Hyun-Kook Kahng
Shigeki Goto (Eds.)

Information Networking

Networking Technologies for Broadband and Mobile Networks

LNCS 3090

International Conference, ICOIN 2004
Busan, Korea, February 2004
Revised Selected Papers

Springer

Lecture Notes in Computer Science 3090

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Hyun-Kook Kahng Shigeki Goto (Eds.)

Information Networking

Networking Technologies for Broadband and Mobile Networks

International Conference ICOIN 2004
Busan, Korea, February 18-20, 2004
Revised Selected Papers

Springer

Volume Editors

Hyun-Kook Kahng
Korea University, Department of Electronics and Information Engineering
208 Suchang-dong Chochiwon Chungnam, Korea 339-700
E-mail: kahng@korea.ac.kr

Shigeki Goto
Waseda University, Department of Computer Science, Goto Laboratory
3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan
E-mail: goto@goto.info.waseda.ac.jp

Library of Congress Control Number: Applied for

CR Subject Classification (1998): C.2, H.4, H.3, D.2.12, D.4, H.5

ISSN 0302-9743
ISBN 3-540-23034-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by DA-TeX Gerd Blumenstein
Printed on acid-free paper SPIN: 11307457 06/3142 5 4 3 2 1 0

Preface

The papers in this book were prepared for and presented at the International Conference on Information Networking 2004 (ICOIN 2004), which was held from February 18 to 20, 2004 at Busan, Korea. It was organized by the KISS (Korean Information Science Society) SIG-IN in Korea and the IPSJ (Information Processing Society of Japan) SIG-IN. The papers were selected through two steps: (1) referral by TPC members and reviewers; and (2) on-site presentation reviews by session chairs and conference participants.

The theme of ICOIN 2004 was "Convergence in Broadband and Mobile Networking", considering high-speed mobile networks. Even though it seems that a kind of "slow-speed" mobile Internet service is to be provided in the market sooner or later, it would not be an ultimate solution for the users, who require more "QoS-rich" services. In the near future, mobile services will adopt high-speed networking technology. At the same time, mobile service users will anticipate that a reasonable level of quality of service and security is provided. Based on this, ICOIN 2004 requested papers on technologies for mobility, quality of service, security, broadband access, and applications, focusing on enhanced Internet protocols and algorithms, their implementation, and the convergence technology required to support wired and wireless Internet.

This book contains articles on the following subjects related to information networking from the low-layer transmission technologies to the higher-layer protocols and services.

- Mobility Internet and Ubiquitous Computing: Concerned with mobile ad hoc networking, mechanisms for ubiquitous environments, routing in mobile networks, and mobile IP protocols.
- QoS, Measurement and Performance Analysis: Concerned with QoS-related routing algorithms and protocols, QoS provisioning Internet protocols, modelling and analysis, and performance measurement.
- High-Speed Network Technologies: Concerned with multicast in high-speed networks, advanced algorithms and protocols, and switching and routing.
- Next-Generation Internet Architecture: Concerned with new Internet protocols such as IPv6, active networks, and optical networks.
- Security: Concerned with security in mobile networks and active networks, security infrastructures, authentication mechanisms, and detection mechanisms.
- Internet Applications: Concerned with services using Internet protocols, and application-level protocols for VoIP, multicast, and reliability.

All papers in this book, we believe, will prove rewarding for all computer scientists working in the area of information networking.

May 2004

Shigeki Goto
Hyun-Kook Kahng

Organizing Committee

General Chair	Haruhisa Ishida (Tama Art Univ., Japan)
Organizing Committee	
Chair	Ilyoung Chong (Hankuk Univ. of Foreign Studies, Korea)
Vice-Chairs	Hideki Sunahara (Nara Institute of Science and Technology, Japan) Kijoon Chae (Ewha Womans Univ., Korea)
Local Arrangement Co-chairs	Sung-Woon Kim (Bukyung National Univ., Korea) Kyungshik Lim (Kyungpook National Univ., Korea)
Publicity Co-chairs	Hwa Sung Kim (Kwangwoon Univ., Korea) Sumit Roy (Univ. of Washington, USA) Richard Lai (La Trobe Univ., Australia) Krzysztof Pawlikowski (Univ. of Canterbury, New Zealand) Osamu Nakamura (Keio Univ., Japan) Suress Ramadass (Universiti Sains Malaysia, Malaysia) Jae Kim (Boeing Co., USA)
Publication Co-chairs	Sungchang Lee (Hankuk Aviation Univ., Korea) Choong Seon Hong (Kyung Hee Univ., Korea)
Registration chair	Meejeong Lee (Ewha Womans University, Korea)
Financial Chair	Jongwon Choe (Sookmyung Women's Univ., Korea)
Patron Co-chairs	Hyun Park (LG Electronics, Inc., Korea) Yongtae Shin (Soongsil Univ., Korea)

Program Committee

Co-chairs	Shigeki Goto (Waseda Univ., Japan) Hyun-Kook Kahng (Korea Univ., Korea)
Vice-Chairs	Khun Kanchana (Asian Institute of Technology., Thailand) Francis Lee (Nanyang Technological Univ., Singapore) Krzysztof Pawlikowski (Univ. of Canterbury, New Zealand) Chai K. Toh (Northrop Grumman Mission Systems, USA)
Members	Sanghyun Ahn (University of Seoul, Korea) Chung-Ming Huang (National Cheng Kung Univ., Taiwan) Joe Hui (Arizona State Univ., USA) Seong-Ho Jeong (Hankuk Univ. of Foreign Studies, Korea) Shin-gak Kang (ETRI, Korea) Richard Lai (La Trobe Univ., Australia) Victor O.K. Li (University of Hong Kong, Hong Kong) Kyungshik Lim (Kyungpook National Univ., Korea) Koji Okamura (Kyushu Univ., Japan) Jae-chul Rhoo (Chungnam National Univ., Korea) Shinji Shimojo (Osaka Univ., Japan) Harsha Sirisena (University of Canterbury, New Zealand) Tatsuya Suda (University of California, Irvine, USA) Kevin Tsai (University of California, Irvine, USA) Masato Tsuru (TAO, Japan) Kenichi Yoshida (University of Tsukuba, Japan)

Table of Contents

Mobile Internet and Ubiquitous Computing

QoS, Measurement and Performance Analysis

High-Speed Network Technologies

Next Generation Internet Architecture

Security

Internet Application

Part I

Mobile Internet and Ubiquitous Computing

Node Configuration Protocol Based on Hierarchical Network Architecture for Mobile Ad-Hoc Networks*

Hyewon K. Lee and Youngsong Mun

School of Computing, Soongsil University, 156743 Seoul, Korea
kerenlee@sunny.ssu.ac.kr, mun@computing.ssu.ac.kr
http://sunny.ssu.ac.kr

Abstract. Mobile Ad-Hoc Network (MANET) is a multi-hop wireless network without any prepared base station. It is capable of building a mobile network automatically with no help from DHCP servers and routers to forward or route messages. In this paper, we present a new configuration protocol based on two-tiered hierarchical network architecture for MANET. To guarantee the uniqueness of address and to support a fast and reliable node configuration, a new model, MUnit (Mobile Unit) is proposed. In a peer to peer environment, this two-tiered model effectively carries out well. This architecture ensures facility for address allocation and duplication check.

1 Introduction

MANET is a multi-hop wireless network without any prepared base station. It is capable of building a mobile network automatically with no help from DHCP servers and routers to forward or route messages. Significant difference from wired and wireless networks is continuous excessive changes of network topology without base stations. Routing protocols such as DSR [1], AODV [2], TBRPF [3], etc. to find shortest or optimistic routes have been proposed, but these protocols assume that nodes have been pre-configured before building a network.

In this paper, we present a new node configuration protocol based on two-tierd hierarchical network architecture for MANET. To guarantee the uniqueness of address and to support fast and reliable node configuration, a new model, MUnit is proposed. This architecture ensures facility for address allocation and duplication check. Finally, we evaluate the performance of the proposed protocol and conclude with discussions of further study.

2 Related Works

MANETconf [4] suggests a node configuration protocol based on a flat-hierarchical architecture, which is using a two-phase address allocation mechanism. To

* This work was done as a part of Information and Communication fundamental Technology Research Program supported by Ministry of Information and Communication in republic of Korea.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 3–12, 2004.

avoid dead-locks and thrashing, it employs the concept of prioritization among concurrent initiations. The proposed scheme is not scalable in a large network, viewed in a number of exchanged broadcasting messages because all nodes respond to soliciting or check process. In addition, it applies a proactive approach in an entire network that every node is required to know all other nodes. [4] does not guarantee fast restoration and stability when some nodes acting as address administrator are crashed abruptly.

Prophet Address Allocation for Large Scale MANETs [6] suggests a new IP address allocation algorithm, namely prophet allocation, which is expectable by an initiating node. The fragility of this protocol is that a node in superior position should not leave until a network terminates.

Zeroconfigration is very useful when no configuration information is available. Zeroconfigration protocol is suitable for wire-lined based networks because the topology that this protocol serves is a physically or logically single network segment where all nodes are connected. Zeroconfigration is not pertained for MANET, which employs multi-hops based communications.

3 Proposed Node Configuration Protocol

In this paper, MANET is grouped into two components; *agent* and the other nodes, called *consignors*. An agent allocates address for other nodes and guarantees the uniqueness of address allocation. A consignor selects one of the nearest agents. One agent and consignors organize a *MUnit*, as seen in Fig. 1. The maximal number of consignors in MUnit, c_{max} and the maximal distance between agent and non-agent, d_{max} are necessary to be predefined for optimally fast restoration and communication between nodes. The variables in (1) are dependent on implementations and devices participating on communications.

$$c_{max} \leq m, d_{max} \leq n. \tag{1}$$

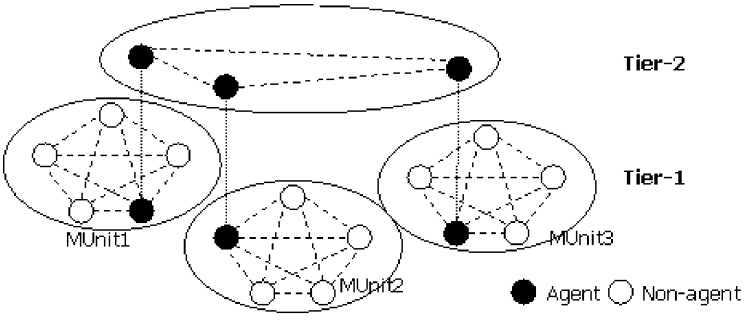


Fig. 1. Hierarchical configuration of MANET

Table 1. Address type

Address Type	Description
Allocated address	(agent address, allocated address, allocated time, the number of hop count)
On-pending address	(agent address, on-pending address, on-pending time)
Free address	(address, de-allocated time)

All members in the same MUnit know each other, but only an agent knows others in other MUnit. An agent located on tier-2 logically learns member information belonging to other agents, so agents hold database about all nodes in a network, and further, an agent knows which nodes belongs to a specific agent by periodic update message exchanges between agents. Besides, an agent is required to send periodic inspection message to its MUnit nodes to check whether they are alive. For stability, non-agents are required to know the nearest agents to recover from an agent absence, so non-agents holds database about neighbored agents. Thus, non-agents know all allocated addresses in same MUnit, neighbored agents' addresses, and further on-pending and free address learned from their agent.

Three types of address are newly defined as shown in Table 1. The allocated address is already allocated to a node in some MUnit and should not be allocated to other nodes. The agent address field specifies the administrator of a node. If the agent address and allocated address fields in a record have same value, the record corresponds to an agent. The number of hop count information is necessary for all nodes to know a distance between nodes. The on-pending address is under address duplication check process and should not be allocated to other nodes, either, before originally invoking agent sends any update message. The status of on-pending address becomes allocated if address uniqueness is verified. The free address is allocated once and free, so it is an available address resource in a MUnit, but still this address is known as unavailable to other MUnits. The free address can be allocated to a new node in same MUnit, so the agent can allocate free address to new node directly without any duplication address check. Once an agent notifies that some address now available, then other agents should delete notified address from their allocated address list or marked as unallocated to use this address later, but unallocated address should pass a duplication address check. The change of address state is not propagated to neighbors immediately, but transmitted via periodic update message, which reduces message exchange overheads.

New message types are defined for node configuration in Table 2. The reply messages for solicit or request message are not specified. Each reply message corresponding to request or solicit will be simply named after solicit or request message. When messages are broadcasted, they should be set with hop limit not to be propagated to an entire network. In neighbor_reply (NR) message, several flags are defined: M, R, and P flags. At first, M flag indicates whether it

Table 2. Message type

Message Type (Abbreviation)	Description
neighbor_search (NS)	Used when a new node wants to find neighbors.
address_request (AREQ)	Used when a node wants to configure new IP address.
address_duplication_check (ADC)	Used when an agent checks address duplication.
be_agent_request (BAREQ)	Used when a node asks other node to be a new agent.
consignor_move_notify (CMN)	Used when a consignor moves other MUnit.
be_agent_notify (BAN)	Used when an agent wants to leave or move a network.
to_join_request (TJREQ)	Used when a node moves and needs to join other agent.
agent_solicit (AS)	Used when a node needs the list of agent addresses.
address_release_request (ARREQ)	Used when a node wants to leave a network.
neighbor_update (NU)	Used when there is a change in a network.

is an agent, or consignor. R flag informs the availability of agent whether it can handle more new nodes. M and R flags are only used by an agent. When a reply message set one of these flags from non-agent is received, it should be ignored. P flag is used to notify whether the response node can be a preliminary agent. If the soliciting node receives reply message set P flag at first, it should remember the response node.

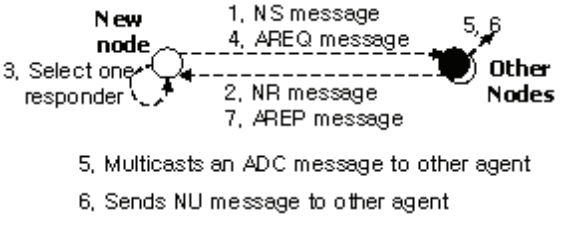


Fig. 2. Joining step

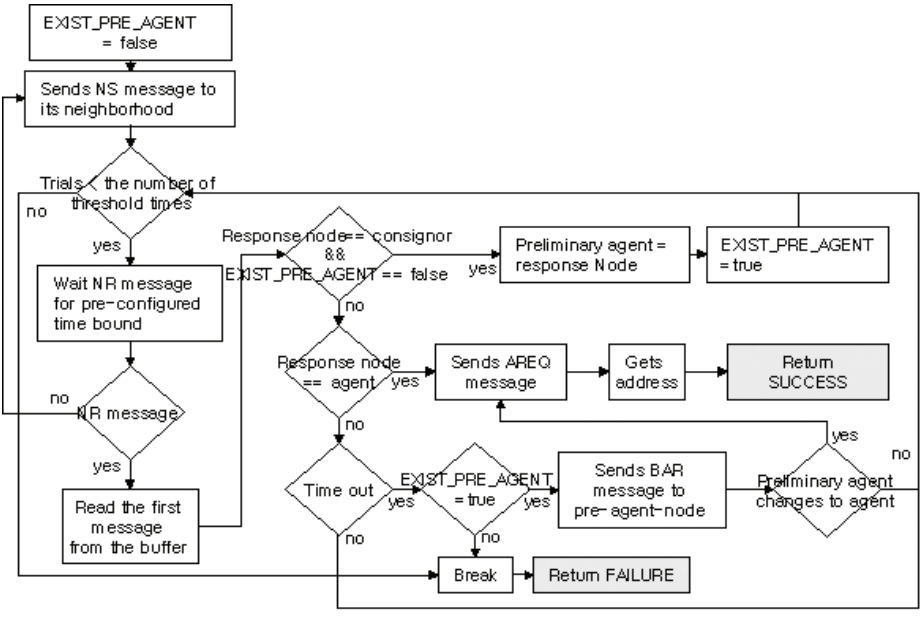


Fig. 3. Flowchart for address allocation

Joining and Address Allocation When a new node enters network, it broadcasts NS message to find neighbors, and it starts a timer. This node will wait until it gets any response, or this timer is expired. If soliciting node gets no response after, it retries this process threshold times which are dependent on implementations. When every trial is ascertained as fail, this node becomes an agent and builds one MUnit.

Once the soliciting node gets reply message, it starts buffering for prefixed time. The order of buffering is response latency between nodes. The node remembers the first response set P flag to 1. If the node finds an agent satisfying (1), it request address, and then the agent will find new address and allocate it to the node. Unless there is any response from agents, the new node requests the first responding node marked P flag to be an agent. Once a node receives with BAREQ message, it requests its original agent to which it belongs once, gets agent information, builds new database, and notifies other agents of a new agent. Now, this new agent can allocate address for other nodes. The joining process is shown in Fig. 2. The address allocation algorithm is given in Fig. 3, whose time complexity is $O(n)$ even in the worst case.

Address Duplication Check To guarantee the uniqueness of address is very important to identify hosts in MANET. In this paper, avoiding duplication policy is employed. When a node requests an address, the requested agent lookups its database, selects temporary IP address, updates on-pending address list and

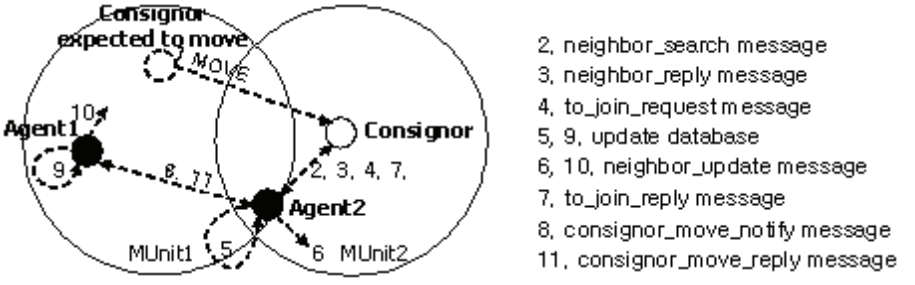


Fig. 4. Handling the moving nodes

sends ADC multicast message to other agents. If agents receive ADC message, they check their database at first, make negative or positive response, and update on-pending address list in case of available address.

Once the agent gets every response from all other agents inner (2), it allocates address to the requesting node. Else, the agent will send check message to non-responding agents. When agents causing trouble do not response even after several trials, new agent is under forming with some reason, such as crashed agent or agent leaving. The requesting agent goes through suspension state until new agents notify of their presence. The waiting time with an assumption that every agent answers to the source agent with no collision can be formulized as (2), where $dist_{max}$ is a distance between an agent and the furthest agent, and t_{propa} is a normalized propagation time.

$$w_{time} = 2 \times dist_{max} \times t_{propa}. \tag{2}$$

If there is any reply indicating duplication, the source agent gives up the on-pending address, selects other one and restarts the address duplication check until there is no address duplication. Once the agent verifies address, address can be allocated. Finally, the agent multicasts new allocated address to its MUnit immediately. Notifying other agents of change in address status is delayed to avoid communication overhead. If two agents solicit same address concurrently, what is called collision, a priority from an arbitrary contention algorithm can be used to solve the problem.

Handling Moving Nodes When a consignor moves one MUnit to another, it does not need to be allocated new address but need to know to which MUnit it belongs. For example, in Fig. 4, if a consignor in MUnit1 moves to MUnit2, and this node is not able to get inspection message from Agent1, then it floods NS message to locate Agent1. If the consignor gets reply messages from any other agents except the previous Agent1, then it selects the most adequate agent, Agent2 and sends TJREQ message. When Agent2 receives TJREQ message, it updates its database and notifies Agent1 of the moving node.

Handling Agent Release When an agent wants to be released from its role, it should select the next agent before leaving. Some contention-based selection algorithm, for instance, node residence time in the network, or power level of device may be adopted to designate or select a new applicant. Once one consignor is appointed, it learns information from the releasing one and configures new database. This newly elected agent informs the other agents of new agent immediately. In case of the leaving agent, address release should be followed by agent release.

Handling Crashed Agents An agent is possibly disappeared with unexpected situations, due to battery life, electrical problem, etc., so crashed agent should be replaced with new one, and one of consignors is elected by some contentions. Once a new agent selected, it should learn all agents' addresses, so new agent sends AS message to its one of neighbored agents and learns the list of agent addresses. The newly elected agent informs the other agents of new agent immediately. The address belonging to crashed agent should be released.

Handling Address Release Nodes which do not want to be a member of network any longer are required to release their addresses. Agents and non-agents are equally in the limit of this application, so ARREQ message should be delivered to an agent before leaving. In case a node unexpectedly leaves a network, agents are required to send inspection packets periodically to check if consignor is alive. Unless there is an answer after several attempts, the agent frees the address. An address duplication problem can be occurred, and it is advised to allocate new address to the deprecated node.

4 Performance Evaluation

Three types of messages are exchanged between nodes in the network: broadcast, multicast and unicast. A broadcast message is flooded to all nodes in a network, so this message expends lots of network resource and causes very high communication overhead. Proposed scheme broadcasts only NS message to all nodes to attain maximal availability of network resources. A multicast message is sent to only the member of targeted group. This message shortens communication overhead from a broadcast message but still has higher overhead than a unicast message does. A unicast message is forwarded only to a destination node so causes the lowest overhead.

The proposed protocol employs hierarchical two-tiered model, and it lessens decently the number of broadcast, unicast and multicast messages. Only agents participate in address allocation process except the formation new agent, so joining new node requires a minimal number of packets compared the model in [4]. For example, if we consider the workload, W of one NS broadcast messages from one new node to the other nodes to join a network with fault rate r, which identifies a wrong delivery due to collision or crashed node, etc., W becomes $\frac{n}{1-r}$, where n is the number of nodes in a network. The proposed scheme limits the range of broadcast message, so this message uses less than W. Further,

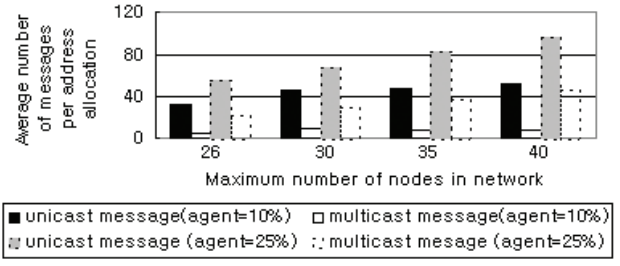


Fig. 5. Communication overhead to allocate address. The agent rate is 10% and 25% respectively when 25 nodes exists in the network

if there is m $(1 \leq m \ll n)$ agents in a network, and l is duplication rate, proposed protocol requires only $\frac{(m+1)}{(1-r)(1-l)}$ for address duplication check. In [4], every node is required to respond to address duplication check, so workload becomes $\frac{2n}{(1-r)(1-l)}$. The second comparison ignores multi-hop routing in both cases, and message is directly forwarded from source to destination. If multi-hop routing is employed, difference between them will be bigger.

The simulation experiments below are primary focused on gathering the number of communication packets to join a new node. It is assumed that there is no collision between packets, and agent rate in the network can be regularly defined. This simulation does not consider the beginning of network formation, so nodes are supposed to be present in the incipient stage. Address adopted in this simulation is assumed as EUI-64 based link-local address defined in [9]. Lifetime of nodes in the network is infinite, and no nodes exit, because this simulation checks the required packets when new nodes come in and are allocated with new address. Each new node comes in every 20 seconds, and a update message is exchanged between agents per every 40 seconds. The expected number of exchanged messages whenever newcomer comes in can be computed by a numerical formula (3), where α is the probability of agent presence in three times of NS message trials, p_i is the probability of agent presence in the buffer, and m_i is the number of exchanged messages between nodes. m_i is equal to the sum of the number of responding nodes to NS messages, number of updates messages between agents and the number of agents in the network.

$$P = \alpha p_1 (1 - p_1)^{n-1} m_i + (1 - \alpha)(2 + m_i). \tag{3}$$

The first simulation shows communication overheads that agent rate is 10% and 25% respectively when 25 nodes exist in the network at first, and 15 newcomers arrive one by one. The second simulation displays communication overheads that 25 nodes and 30 nodes exist respectively in the initial step when agent rate is 10%, and 20 newcomers arrive one by one.

Fig. 5 remarks that high agent rate gives bad effect upon network availability. The main reason of rapid increase of unicast message is that the more agents,

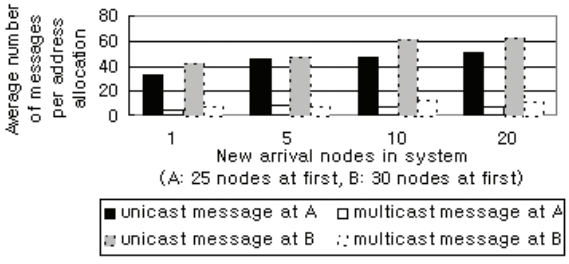


Fig. 6. Communication overhead to allocate address. There are 25 and 30 nodes in the network respectively when the agent rate is 10%

the more number of exchanged unicast messages between agents for address duplication check. This curve rate may be diminished by longer update period. The number of exchanged broadcast message is the infinite series of mean value of neighbor discovery failure possibility. A broadcast message is not specified in this picture because it has very low frequency compared to other message types. When the agent rate is reasonably low, scalability of the protocol is quietly good independent with the number of nodes in a network. The slope rates of unicast and multicast message in Fig. 6 are not fluctuating severely as the number of node increases in both cases, and further, there is a little difference in the number of messages between two cases. From Fig. 6, we can infer that initial size of networks has not so big effect over network scalability.

5 Conclusions

We have presented a distributed, dynamic host configuration protocol for nodes in MANET. This proposed protocol enables nodes in the MANET to configure their address and network formation automatically by two-tiered hierarchal managed network architecture. We have proposed message types and several address types, such as allocated, on-pending, and free address. Especially, this protocol guarantees unique address allocation with reduced overhead by dynamic configuration of MUnit compared with [4]. Besides, fast and reliable node configuration is strongly supported. This protocol may be able to compatible with cluster switch gateway routing for routing protocol on a network. How to build an identifier, i.e., IP address, for each host is necessary to consider as a further study.

References

[1] Johnson, D., Maltz, D., Hu, Y.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Work in progress, IETF (2003) 3

[2] Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, IETF (2003) 3

[3] Ogier, R., Templin, F., Lewis, M.: Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Work in progress, IETF (2003) 3
[4] Nesargi, S., Prakash, S.: MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 2. INFOCOM, IEEE (2002) 23–27 3, 4, 9, 10, 11
[5] Haas, Z.: A new routing protocol for the reconfigurable wireless networks. Universal Personal Communications Record, Vol.2. 6th International Conference, IEEE (1997) 562–566
[6] Zhou, H., Ni, L., Mutka, M.: Prophet Address Allocation for Large Scale MANET. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 2. INFOCOM, IEEE (2003) 1304–1311 4
[7] Boleng, J.: Efficient Network Layer Addressing for Mobile Ad Hoc Networks. (2000)
[8] Corson, S., Macker, J.: Mobile Ad hoc Networking Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, IETF (1999)
[9] Hinden, R., Deering, S.: IP Version 6 Addressing Architecture. RFC 2373, IETF (1998) 10

Vote-Based Clustering Algorithm in Mobile Ad Hoc Networks

Fei Li[1], Shile Zhang[1], Xin Wang[1], Xiangyang Xue[1], and Hong Shen[2]

[1] Department of Computer Science, Fudan University, 200433 Shanghai, China
{021021107,0024131,xinw,xyxue}@fudan.edu.cn
[2] Department of Information Systems, JAIST, Japan
shen@jaist.ac.jp

Abstract. Unlike current clustering methods, the presented vote-based clustering (VC) algorithm uses not only node location and ID information, but also battery time information. In VC, each mobile host (MH) counts Hello messages from its neighbors. At the same time it calculates its own vote that is the weighted sum of the normalized number of valid neighbors and its normalized remaining battery time. The one with higher vote than its neighbors will be selected preferentially as a cluster head (CH). When the number of dominated MHs of a CH is more than a balance threshold, neither of new coming MHs will be permitted to participate in the current cluster. Analysis and simulation results show that VC method can improve cluster structure than Lowest ID (LID) algorithm and Highest Degree (HD) algorithm.[1]

1 Introduction

A MANET is a multi-hop wireless network in which mobile hosts (MHs) communicate without the support of a wired backbone [1]. In a MANET, the network topology changes frequently, the control overhead is very large and a wireless link is easy to break down. So how to reduce the number of control packets and repair a wireless route becomes very important. However, people can only get a tradeoff between the above two ambivalent objects.

Clustering is such an effective method, which is a common method in a communication network topology description, and used to group network nodes into clusters. It provides a convenient framework for the development of important features such as code separation (among clusters), channel access, routing, power control, virtual circuit support and bandwidth allocation. With an underlying cluster platform, non-ordinary MH can be the dominant forwarding nodes. In comparison with fixed communication networks, clustering in MANET turns difficult. Because of node mobility and wireless link weakness, more control information must be paid to clustering a MANET. A representative of each cluster

[1] This work was supported in part by NSFC-60003017, NSFC-60373020, 863-2001AA114120, 863-2002AA103065, SRF for ROCS. SEM, Shanghai Municipal RD Foundation under contracts 035107008, 03DZ15019 and 03DZ14015, and Youth Foundation of Fudan University under No.EXH6286301.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 13–23, 2004.

is named as a cluster head (CH) and a MH belonging to more than 2 clusters at the same time is called a gateway. Other members are called ordinary MHs. Generally a cluster is defined by its CH's transmission range.

Cluster architecture in MANET may be with or without CHs in every cluster [2]. CH-based clustering can reduce storage and exchange information of ordinary MHs. In clusters without CHs, every MH has to store and exchange more topology information, thus the bottleneck of CHs can be eliminated. CH Y. Yi and M. Gera partitioned 2 approaches to construct a MANET cluster platform, i.e. active clustering and passive clustering [3]. In active clustering, MHs cooperate to elect CHs by periodically exchanging information, regardless of data transmission. On the other hand, passive clustering suspends clustering algorithm until the data traffic commences [4]. It exploits on-going traffic to propagate "cluster-related information" (e.g., the state of a node in a cluster, the IP address of the node) and collects neighbor information through promiscuous packet receptions. Thus, it eliminates setup latency and major control overhead of active clustering required collecting neighbor information.

Recently multipoint relays (MPRs) are often used in clustering to reduce the number of gateways in active clustering. MPR Hosts are selected to forward broadcast messages during the flooding process [5]. This technique substantially reduces the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. Using MPRs, the Optimized Link State Routing (OLSR) protocol can provide optimal routes, and at the same time minimize the number of control messages flooded in the network [6]. A good clustering method should be able to partition a MANET quickly with little control overhead. Because of node mobility, it is difficult to construct the best clustering structure in a MANET. To this end, two distributed clustering algorithms are considered. They are Lowest ID algorithm (LID) [7]. and Highest Degree algorithm (HD) [8]. Both of them belong to passive clustering.

In LID algorithm, each node is assigned a distinct ID. Periodically, the node broadcasts the list of nodes that it can hear (including itself). The lowest-ID node in a neighborhood is elected as the CH. LID method has the following 4 rules:

(1) A node which only hears nodes with ID higher than itself is a CH.
(2) The lowest-ID node that a node hears is its CH, unless the lowest-ID specially gives up its role as a CH (deferring to a yet lower ID node).
(3) A node which can hear two or more CHs is a gateway.
(4) Otherwise, a node is an ordinary node.

In HD algorithm, the highest degree node in a neighborhood becomes the CH. The algorithm is described below:

(1) Each node broadcasts the list of nodes that it can hear (including itself).
(2) A node is elected as a CH if it is the most highly connected node of all its "uncovered" neighbor nodes (in case of a tie, lowest ID prevails).

(3) A node which has not elected its CH yet is an "uncovered" node, otherwise it is a "covered" node.
(4) A node which has already elected another node as its CH gives up its role as a CH.

An optimized cluster protocol about LID was proposed in [2]. It ensures MHs who have not received Hello messages during a certain period can issue a new cluster or participate in an existing cluster after a while.

LID method is a quick clustering method, which only uses 2 Hello message periods to get the cluster structure. Also it provides a more stable cluster formation than HD method. In HD style even if one link drops due to node movement, the current CH may fail to be re-elected again. HD method can get fewer clusters than LID. It is very helpful in a large-scale network.

In current clustering schemes, stability, quantity and convergence are of very importance. However, fewer clusters don't always mean better. A CH dominates so many mobile nodes that its resources (e.g. computing, bandwidth, and etc.) will be exhausted soon. So the control of cluster scale is very important. On the other hand, under mobile computing environment, apart from position and ID, power is another important factor for one MH. The fore-mentioned clustering methods didn't mention cluster scale and power factor, we do it in this paper.

The rest of the paper is organized as follows: in Section 2, the vote-based clustering algorithm is presented, which includes 3 parts: vote-based partition algorithm, mobile management method and cluster load balance method. Performance simulation and analysis are shown in Section 3. Finally, conclusions and future work are given in Section 4.

2 Vote-Based Clustering Algorithm

In MANET clustering, we should consider not only position and ID but also other factors. LID is a quick clustering method, which only uses 2 Hello message periods to get the cluster structure. In LID, only ID information was used to distinguish every MH. Obviously, it did not make use of MHs' position information and cannot get few clusters. HD method needs 3 Hello message periods, for that each MH must know its neighbor host's neighbors' number. It uses position information and ID information of one MH.

Fig. 1 shows a simple comparison of HD clustering and LID clustering. In a MANET including 15 mobile hosts, LID method gets 6 clusters but HD method gets only 4 clusters. The virtue of VC is using every MH's mutual location information. On the other hand, Both of LID and HD method cannot trade off among different clusters. Maybe one CH dominates fewer MHs, but another CH holds more MHs. We want to eliminate this possibility to avoid one CH be exhausted. ID is used to distinguish every MH anywhere, anytime. By position information, we know a MH's one-hop neighbors, two-hop neighbors, etc.

In our case, considering the relation between power consumption of one CH and the number of its dominated MHs, we select the lasting time of MH's battery as one performance parameter. Our algorithm is based on two important

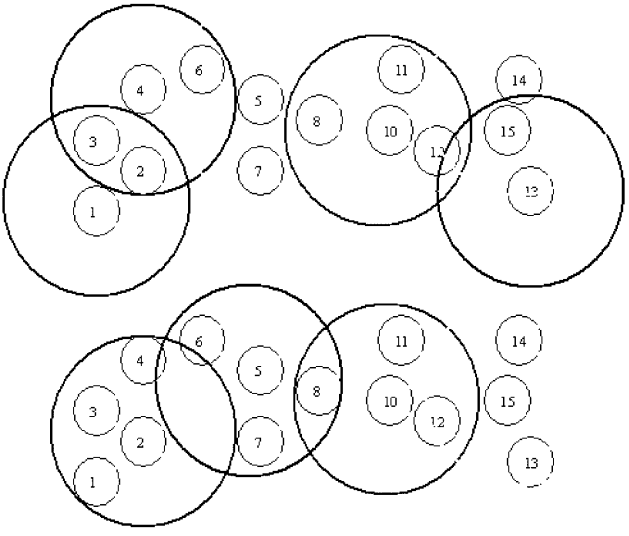


Fig. 1. A simple comparison of HD clustering and LID clustering

performance factors, neighbors' number and remaining battery time of every MH. Then we use voting method to select cluster head and determine members of a cluster.

2.1 Network Model

A MANET can be divided into several overlapped clusters. A cluster is composed of a subset of nodes, which can communicate directly with a cluster head and with each other. Hereby the scenario is modelled as an undirected graph $G(V, E)$ where V is the set of all MHs in the network and E is the set of all links (i, j) where $i, j \in E$. Each link signifies that two MHs are within transmission range of each other. Let S_i be the set of all nodes that can be reached by node i. We assume every link is bi-directional so that link (i, j) exists if and only if $j \in S_i$. The topology of G is the set of nodes and edges.

Each MH has a unique identifier (ID) number, which is a positive integer. The basic information inside the network is Hello message, which is transmitted in the common channel. Every MH acquires information from neighbor hosts' periodic Hello message. We assume that only when the 2 MHs lie inside the mutual transmission range, they can communicate directly with each other, i.e. a bi-directional link exists. One another important information for every MH is its battery lasting time, which is a positive integer.

2.2 Vote-Based Clustering Scheme

In our proposal, we consider the clustering architecture with cluster heads. It is also assumed that one MH can only participate in a unique cluster at the

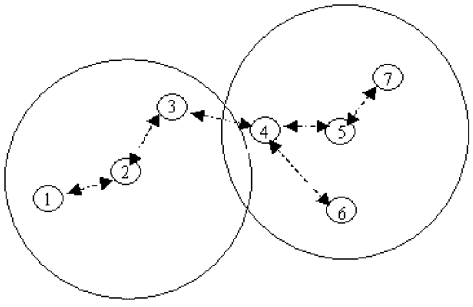


Fig. 2. Communication Procedure between Cluster2 and Cluster5

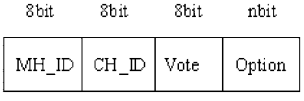


Fig. 3. Hello message format

same time, so Fig. 2 shows this case and the communication procedure between 2 clusters. A cluster is tagged with its CH's ID, e.g. Cluster2 and Cluster5 in Fig. 2. The proposed vote-based clustering protocol includes 3 parts: vote-based partition algorithm, mobile management method and cluster load balance method.

2.2.1 Vote-Based Partition Method

Making use of node location information and power information, we introduce the concept of "vote" and proposed vote-based partition algorithm shown later. Fig. 3. shows the Hello message format in Vote-based clustering algorithm (VC). MH_ID item is MH's own ID and CH_ID item is MH's CH ID Vote item means MH's vote value, i.e. weighted sum of number of valid neighbors and remaining battery time. Option item is used to realize cluster load balance in part 3.

$$Vote = w1 \times (n/N) + w2 \times (m/M). \tag{1}$$

$w1, w2$: Weighted coefficient of location factors and battery time, respectively,
n: Number of neighbors,
N: Network size or the Maximum of members in a cluster,
m: Remaining battery time,
M: The maximum of remaining battery time.

m is a characteristic parameter for every MH. Each MH consumes the battery energy anytime. Each CH spends more than a common MH, obviously.

The algorithm includes the following steps:

(1) Each MH sends a Hello message randomly during a Hello cycle. If a MH is a new user to the MANET, it reset "CH_ID" item. That means the MH does not belong to any cluster and does not know whether it has neighbor hosts.
(2) Each MH counts how many Hello messages it can receive during a Hello period, and considers the number of received Hello messages as its own n.
(3) Each MH sends another Hello message, in which "vote" item is set to its own vote value and got from Equation 1.
(4) Recording Hello message during 2 Hello cycles, each MH knows the sender with highest vote and not belongs to any existing cluster is its cluster head. It set its next sending Hello message item "CH_ID" to the cluster head's ID value. One noticeable issue is when two or more mobile nodes receive the same number of hello packets, the one who owns the lower ID will be prior to others.

Following the above-mentioned approach, every MH knows its cluster head ID after 2 Hello message periods. That is to say, we can finish clustering scheme during 2 Hello cycles. We also know that the cluster head sends a Hello message, in which "MH_ID" is the same as "CH_ID".

2.2.2 Mobile Management Method

All moving MHs can be classified into 2 kinds by their current status. For a moving cluster member, if it receives a Hello message with bigger vote from another CH or non-CH host, the latter will become its new cluster head. For a moving CH host, it uses the same method to participate in a new cluster. However in this case, all its dominated mobile hosts must start a new cluster discovery process. Once a member host finds its CH turns a member host by analyzing the received Hello message, it will reset the "CH_ID" item to 0.

Using this kind of mobile management method, real-time modification of the cluster structure can be realized.

2.2.3 Adaptive Cluster Load Balance Method

In LID or HD clustering scheme, one cluster head can be exhausted when it serves too many MHs. It is not good and the CH becomes a bottleneck. So we proposed an adaptive cluster load balance method. In Fig. 3 , there is an "Option" item. If a sender MH is a cluster head, it will set the number of its dominated MHs as "Option" value. When a sender MH is not a cluster head or it is undecided (CH or non-CH), "Option" item will be reset to 0. When a CH's Hello message shows its dominated MHs' number exceeds a threshold (the maximum number one CH can manage), there will not be any new MH participate in this cluster. As a result, we can eliminate the CH bottleneck phenomenon and optimize the cluster structure.

As stated in the above description, VC can get load balance between various clusters. Thus, resource consumption and information transmission will be distributed to all clusters, not only to some certain clusters. On the other hand, the

consideration of battery lasting time can help us to get a steady cluster structure. Because in VC method, through inducting weighted battery lasting time, the probability of a MH without enough battery energy will be reduced.

3 Performance Simulation and Analysis

In this part, we simulate the proposed clustering protocol under C++ programming environment. The simulation network is a square plane area with $50m*50m$. There are totally N mobile hosts in the square space and N can be 10, 20, 30, and up to 150. Each mobile host stays in the space randomly at the start. Every MH will move at a proportional rate between $0 \sim 5m/s$, and at a random direction. If they move to the boundary, they will be bounced back. The Hello message is sent at $5ms$ period. When using VC with load balance method, the threshold for a dominating MH is defined as 15. Each simulation lasts out 1 minute. The initial battery time of each MH is a random value between 0 and 1, 3, or 5 minutes.

We tested some parameters using LID method and VC method, respectively. From the fore description, it is easy to see that VC without load balance and without battery time limit is HD method indeed. The parameters include number of cluster heads, average change of cluster heads and variance of cluster size.

3.1 Number of Cluster Heads

We count the number of cluster heads every 1s and compute the arithmetic average every 5 simulations. In our simulation, an isolated MH will not become a cluster head, because it cannot communicate with any other mobile host.

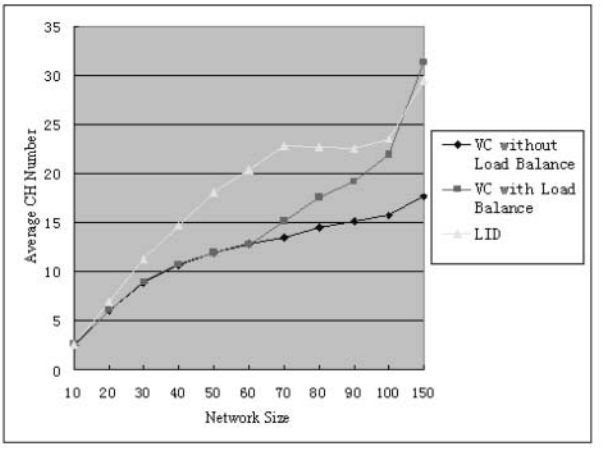


Fig. 4. Average cluster head number with 3 methods

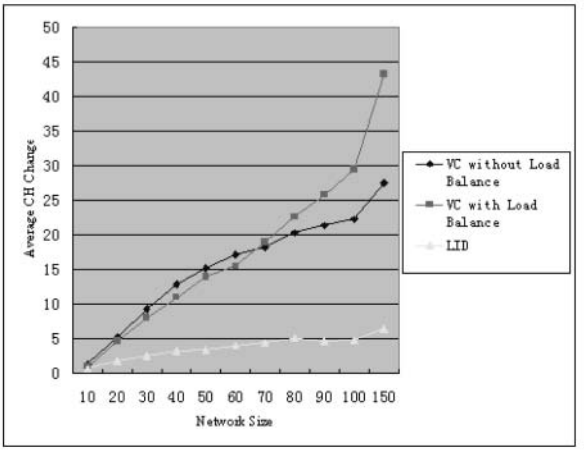


Fig. 5. Average cluster head change with 3 methods

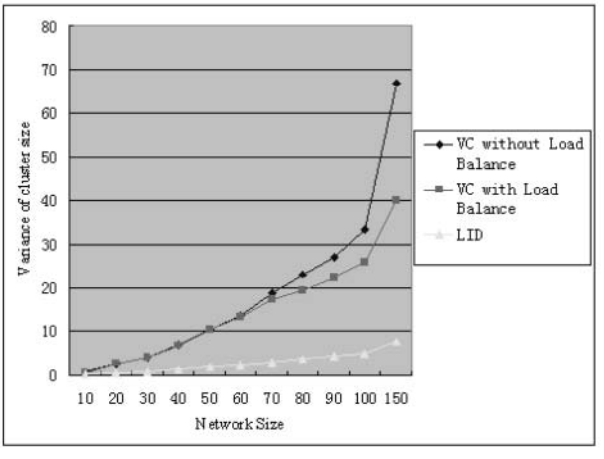


Fig. 6. Variance of cluster size with 3 methods

Fig. 4 illustrated the average cluster head number in the MANET. For a medium-scale network, VC method can reduce clusters obviously. When the network is very sparse, VC method cannot play very well. However, we know that in that case clustering will be not a good mechanism at all. In addition, with the network scale increased, VC with load balance will result in more clusters even than LID. It is because it can save a cluster heads resource to avoid its premature exhaustion.

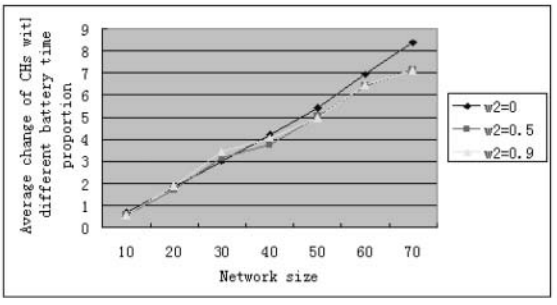


Fig. 7. Average change of CHs with different battery time proportion

3.2 Average Change of Cluster Heads

If we define A and B as aggregate of cluster heads in previous test moment and current test moment, respectively, the change of cluster heads equation holds:

$$\delta = | (A \cup B - | A \cap B). \tag{2}$$

We computed the arithmetic average of every 5 simulations.

Fig. 5 illustrated average change of cluster heads. Since LID only uses not node location information but ID information, so its cluster status is steadier than VC. Obviously, VC with load balance is worse than VC without load balance about this parameter when network size exceeds a certain scale. When N is bigger than 70, in VC with load balance method cluster heads change very frequently.

3.3 Variance of Cluster Size

We recorded current cluster size every second, $C_i, i = 1, 2, \ldots, M$ (M is number of cluster heads). We define C and D as below:

$$C = \sum_{i=1}^{M} C_i / M. \tag{3}$$

$$D = \sum_{i=1}^{M} (C_i - C)^2 / M. \tag{4}$$

Fig. 6 illustrated variance of cluster size, D. Simulation results show that LID methods variance of cluster size is less than VC method.

From the above 3 figures, we know that VC method can optimize cluster structure by reducing cluster head number. The cost is more cluster head change and higher variance of cluster size.

3.4 Average Change of CHs with Different Battery Time Proportion

Like in 3.2, we can get the average of CHs when $w2$ is equal to 1, 0.5 and 0.9, respectively. It is noticeable in the simulation, at every Hello period, a common MHs battery time drops down at a constant space. For a CH, its battery time decreases in proportion to the number of current dominated MHs.

Fig. 7 illustrated average change of cluster heads with different battery time proportion. When $w2$ is equal to zero, it means battery time is not considered in clustering. As $w2$ is increased step by step, battery time takes a more important role in clustering. The curve shows that the more proportion battery time owns, the steadier the cluster structure turns. If $w2$ is equal to 1, the VC method only uses battery time information, not position information.

4 Conclusions and Future Work

In this paper we present a novel vote-based clustering algorithm for MANET. Unlike current clustering method, VC not only uses node location information and ID information, but also battery time information.

In VC method, each MH counts Hello messages from its neighbors. At the same time it calculates its own vote that is the weighted sum of the normalized number of valid neighbors and its normalized remaining batter time. The one with higher vote than its neighbors will be selected preferentially as a CH. When the number of dominated MHs of a CH is more than a balance threshold, neither of new coming MHs will be permitted to participate in the current cluster.

Analysis and simulation results show that VC method can improve cluster structure than LID and HD algorithm. At first, VC can get less cluster number than LID. Secondly, VC can support adaptive cluster size balance to avoid one CH of being exhausted, better than LID and HD. Thirdly, VC can get steadier cluster structure than HD, since it uses battery information.

As to LID and VC, on the one hand, LID uses more clusters since it only uses ID information, which is constant for every MH. On the other hand, VC improves stability of cluster structure by inducting battery information. Any MH with little battery time may be not selected as a CH.

We will study VC-based routing further. In current method, cluster structure maybe is changed even if only one MH comes or leaves, since many MHs' vote is changed. Apparently, it is not very good. In next-step work, we will focus on boosting up VC's robustness.

We will also study VC-based routing and multicast algorithms. In clustering, we reduce the effect of node mobility and link state on cluster structure and make cluster structure repaired with little spending. However in routing, we hope discovery in time and little cost to get a route quickly. We ever used spine structure in multicast [9] and will use VC structure in multicast later.

References

[1] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, Ad hoc on-demand distance vector (AODV) routing, IETF RFC 3561, July 2003. 13

[2] Liu Kai, Li Jiandong, Mobile cluster protocol in wireless ad hoc network, in Proceeding FIP/WCC 2000 (ICCT 2000), Beijing, China, Aug. 2000, pp. 568-573. 14, 15

[3] Y. Yi, M. Gerla, T. Jin Kwon, "Efficient Flooding in Ad hoc Networks: a Comparative Performance Study", 2003082210. 14

[4] Y. Yi, T. J. Kwon and M. Gerla, "Passive Clustering (PC) in Ad Hoc Networks", Internet Draft, draft-ietf-yi-manet-pac-00.txt, Nov.2001. 14

[5] A. Qayyum, L. Viennot and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks", (HICSS'2001). 14

[6] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol", Internet Draft, draft-ietf-manet-olsr-11.txt, Jul. 2003. 14

[7] Jack Tsai and Mario Gerla, "Multicluster, mobile, multimedia radio network", ACM-Baltzer Journal of Wireless Networks, Vol.1, No.3, pp.255-65, 1995. 14

[8] Abhay K. Parekh, "Selecting routers in ad-hoc wireless networks", in ITS, 1994. 14

[9] Xin Wang, Fei Li, Susumu Ishihara and Tadanori Mizuno, A Multicast Routing Algorithm Based on Mobile Multicast Agents in Ad-hoc Networks, IEICE Transactions on Communications, Vol.E84-B No.8, August 2001, pp.2087-2094 22

Load Balanced Onion Relay for Prevention of Traffic Analysis in Ad Hoc Networks

Sungchang Lee[1], Ha Young Yun[1], and Mi Lu[2]

[1] Hankuk Aviation University, Goyang, Korea
{sclee,hyyun}@mail.hangkong.ac.kr
[2] Texas A&M University, College Station, U.S.A.
mlu@ee.tamu.edu

Abstract. In this paper, an ad hoc network anonymous data forwarding method and an associated routing protocol to prevent traffic analysis is presented. This method assumes trusted closed group nodes and every node can play the role of onion relay for the anonymous data forwarding. The route discovery operation of the protocol adopts the load balancing concept. The load balancing not only improves the throughput but also helps the prevention of the traffic analysis since the load cost for camouflage traffic can be reduce and the routes change dynamically. The performance of the proposed protocol is evaluated by simulation, and compared in several major aspects with the fixed mix method for anonymous data forwarding that works over existing ad hoc routing protocol.

1 Introduction

Security problem in mobile ad hoc network is essential and important in many applications. However, the peculiar attributes and limitations of the ad hoc network even make it more difficult. There are numerous types of attacks on both routing and data forwarding in the network, some of them are specific to ad hoc network. The attacks on MANET (mobile ad hoc network) routing include black holes, denial of service, routing table overflow, impersonation, energy consummation, information disclosure [1]. Also, possible attacks against anonymity network are message coding attack, message length attack, replay attack, collusion attack, flooding attack, message volume attack, timing attack, profiling attack [2, 3]. There have been much work against these attacks, but it seems to be impossible to have a single protocol that can perfectly protect the ad hoc network from all kinds of attacks. In order to cover wide range of attacks, it may be inevitable to have a security framework in which separate security capabilities that fight for different attacks cooperate together. Thus, most of the literatures deal with only limited scope of the security problems.

This paper focuses on the prevention of traffic analysis attack. The scope of the paper covers the anonymous data forwarding and the cooperating routing protocol, but the security of the routing itself is excluded. For the security routing, the methods for anonymous route discovery and maintenance using symmetric or asymmetric keys proposed in other literatures [4, 5, 6] may be applied on the routing protocol proposed in this paper.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 24–33, 2004.

Section 2 describes the basic assumptions and the rationales of the proposed method. The proposed load balancing routing and the anonymous data forwarding scheme are presented in section 3 and 4, respectively. The performance simulation results are shown and discussed in section 5, and section 6 concludes this paper.

2 Model of the Proposed Method

The proposed method assumes that the target ad hoc network consists of a finite number of closed member group, thus there is no severe scalability requirement. Also, a trusted environment is assumed, that is, all member nodes can be trusted, and the trust is continuously probed and maintained using some other methods (such as intrusion detection, key management) that is out of the scope in this paper. Since, the main focus is on the prevention of traffic analysis, it is assumed that camouflage traffic is generated by other module or, maybe, by the higher layer function.

As the mix methods [7, 8], fixed length packets with padding will be used for data packets to hide the correlation between the incoming and outgoing packets. However, routing control packet may be used in traffic analysis by the adversary in ad hoc network, especially in the on-demand routing protocols. Therefore, it will be desirable to use two different fixed lengths in ad hoc network, one for data forwarding and the other for the routing control packets since the routing control packets are usually short and the data packets may have wide range of lengths. In this case, two different length camouflage packets will be needed.

The proposed LBOR (load-balanced onion relay) method assumes that every node in the ad hoc network can be relay node for the anonymous data forwarding. The routing scheme adopts the load balancing concept. The rationales behind this are

- An ad hoc network consists of peer nodes, thus the assumption of the existence of mix (or relay) nodes may not be relevant in situations. Also, the collecting and shuffling of packets as in the wire network mixes [7, 8] may not be appropriate for ad hoc network applications due to delay, processing overhead and buffer requirement. Rather, simpler scheme with camouflage traffic may be needed.
- Unlike the wired network, there is no high capacity back bone network in ad hoc network, where mixes are located. Instead, all ad hoc network links have the same, usually, limited bandwidth, even around the mix nodes. As a result, links around the mix nodes becomes bottleneck, causing the degradation of network throughput, delay and packet loss that may be unacceptable for the real-time applications.
- Data forwarding through few mix nodes may cause low reachability (or delivery ratio) due to the bottleneck as mentioned above, or topological problems like hop distance, which is very important for military, emergency applications.

- Load balancing not only reduces the possible bottleneck due to the limited bandwidth, but also lowers the required amount of camouflage traffic to achieve the neural traffic matrix [9, 10] that is pursued to prevent traffic analysis.
- Relay nodes for anonymity data forwarding can be selected randomly and dynamically. This helps the reduction of hops by avoiding detours, and makes it untraceable.

A number of network mix methods [11, 12, 13] provides data forwarding anonymity assuming separate routing protocols that operate below them. Load balancing concept of the proposed LBOR requires the cooperation of routing and data forwarding protocol, thus it covers both routing and data forwarding.

The route discovery and maintenance of LBOR is based on DSR protocol, but with modifications as described in next section. The modification mainly includes the adoption of load balancing routing. The anonymous data forwarding part of LBOR utilizes the well-known onion routing method, but any node in the network can be relay node for the onion routing. Thus, the route from the source node to the destination node is selected based on the load balancing routing, and the nodes that will works as onion routing relays are chosen by the source among the nodes that are en route.

3 Load Balancing Routing

In this section, the routing operation of LBOR is described. As mentioned in the previous section, the main idea of LBOR is load balancing concept. The effect of load balancing on the prevention of traffic analysis is discussed, and the load balancing routing operation is described.

3.1 Load Balancing and Prevention of Traffic Analysis

Mix methods applied in ad hoc networks have been proposed in [13]. In their methods, there are a finite number of mix nodes in the ad hoc network. Every source node has its own designated mix node (fixed mix method) or selects a mix node dynamically (dynamic mix method). In these methods, the mix nodes are well-known even to the adversary, which may be irrelevant in such as military applications. In addition, the link bandwidth of ad hoc networks is limited, and the links around the mix nodes becomes bottleneck. In this paper, it is assumed that any node can play the role of onion relay node. This model makes it possible to eliminate the vulnerable mix nodes and balance the traffic across the network.

The ad hoc network applications are more likely to be time-critical real-time applications. Also, the amount of network traffic is much smaller than the wired network. Thus, the function of collecting and shuffling of packets as in the wired network mix is not appropriate in ad hoc network. Instead, camouflage traffic will be inevitable to prevent traffic analysis. However, ad hoc network has limited bandwidth, thus the load balancing is important.

The proposed method focuses on the prevention of the traffic analysis. The method assumes that the higher layer generates dummy (or camouflage) packets to prevent the eavesdropper from gaining useful information from the traffic pattern. For this end, the camouflage traffic is generated by so that the load on all links may be equal, achieving so called neural traffic matrix [9]. In the paper, the minimum load cost (MLC) to achieve the neural traffic matrix is defined as [9]

$$MLC = n\,(n-1)\,\mu - S \tag{1}$$

where,

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} M\,[i,j] \tag{2}$$

and

$$\mu = \frac{\max\{\sum_{i=1}^{n} M\,[i,j]\,, \sum_{i=1}^{n} M\,[j,i]\,|j = 1,2,...,n\}}{n-1} \tag{3}$$

In the equation, n is the number of nodes in the network, and $M[i,j]$ is the traffic matrix.

According to this proposition, the maximum of link traffic into or out of the nodes in the network should be as low as possible to minimize MLC. The routing protocol of the proposed method tries to find routes so that the traffic on the links may be balanced to minimize load cost of using dummy packets.

3.2 Routing Operation

The proposed routing protocol is based on DSR (Dynamic Source Routing) protocol but with following modifications.

- Every node continuously updates LL (link load) of each link that is alive. LL is defined as

  LL = number of all packets on the link / measurement window (packets/sec)

- If no packet arrives for LEDI (link entry delete Interval), the item is deleted from the LL list. Measurement window (MW) is a sliding time window, and LL is measured for each direction of a link separately.
- Upon receiving RREQ, each node on the route appends the LL of the corresponding link to RREQ before it broadcasts RREQ. RREQ contains the list of the nodes and LLs of the links it travels.
- Upon receiving RREP, each node on the route updates the route cash with the following criteria.
  1. If the node does not have the same route to the destination, then add the new route entry.
  2. If the cash has route(s) to the destination, add new route and sort.
  3. The priorities for the sorting of the route entries are, in order:

(a) bottleneck load (BL) = max{LL_i} of the route,
(b) total route load (TRL) = the sum of LL_i's,
(c) hop count,

4. Each node keeps only top n_entries entries for each destination.

– If any destination is not referred for routing for CEDI (cash entry delete interval), the entries are deleted from the cash.
– Cash entry is updated every CEUI (cash entry update interval) for each destination alive in the cash.
– Every time an entry is referenced, the BL, TRL, are updated as follows, and entries are sorted again.

$$BL^{+} = BL + \lfloor \frac{n\_hop}{2} \rfloor, \quad TRL^{+} = TRL + \lfloor \frac{n\_hop}{2} \rfloor$$

Other operations are just like DSR. This routing scheme focuses on the load balance of the network links to minimize the load cost for the camouflage traffic. Specifically, the route is chosen so that the bottlenecks of the network can be prevented in advance. This scheme not only improves the throughput of the network, but also reduces the required camouflage traffic.

4 Anonymous Data Forwarding

In this section, the proposed method to anonymize the network traffic is described. For fixed networks, many solutions for untraceable communication have been proposed to keep confidential who communicates with whom. Most of them are based on Chaum's[8] method. In such solutions, special network node(s) (called mix) is(are) used to provide unlinkability between the source and the destination. However, mix method has restricted application since it need to collect sufficient number of (a batch of) packets to shuffle the order randomly before it send out a batch of packets. It may cause intolerable delay or camouflage traffic. Also, the mix needs high processing power, and if the link capacity around the mix(s) is limited, the traffic bottleneck degrades the performance of the network significantly. These aspects make the mix method infeasible for Ad Hoc network.

The dummy traffic to make network traffic even to prevent the adversary to deduce useful information from the traffic pattern may congest the limited bandwidth of the ad hoc network. In order to circumvent some of the drawback of mix method, NMD was proposed for mobile IP network, but ad hoc network has additional constraints of the limited bandwidth, the morphosis of the network topology itself, and the limited processing power of the nodes. In this section, how to provide the unlinkability of the traffic that is adopted in the proposed method is described.

A number of protocols for anonymity have been proposed based on the Chaum's untraceable electronic email solution [8]. The basic data forwarding of LBOR is also based on onion routing [12, 14]. The public keys of all nodes are known to the network members. When a node is to send a packet, the source selects the route according to the routing information. With the selected route, the source determines the number of ad hoc onion relay nodes (ORNs) as following.

Step 1: Select the route to the destination according to the routing cash. (n_hop : the number of hops to destination)

Step 2: Decide the number of ORNs (n_ORN) along the route between the source and destination as follows.

$$n\_ORN = \max(1, \lceil n\_hop/\alpha \rceil) \tag{4}$$

where, α is sparseness of ORNs along the route. $\alpha = 1$ means that all node en route to the destination are onion relay nodes.

Step 3: Select randomly n_ORN nodes along the route.

$$S_R = \{R_i |\ i = random(1,\ n\_hop)\}, \quad where \quad |S_r| = n\_ORN \tag{5}$$

Once the ORNs along the route to the destination are determined, the packet goes through a sequence of $ORNs\{ORN_1,\ ORN_2, ...,\ ORN_{n\_ORN}\}$. The onion sent by the source is

$$K_1(ORN_2, R_2, K_2(ORN_3, R_3, K_3(\cdots K_n(D, R_n, K_D(M, R_0))\cdots)))$$

where, K_i 's are public keys, and M is the message sent from the source to the destination. If the processing overhead of public key cryptosystem is not acceptable, the symmetric key cryptosystem can be used instead.

5 Performance Evaluations

To evaluate the performance of the proposed routing protocol in several aspects, the simulation of the proposed protocol and fixed mix method is implemented using *ns*-2 [15]. The ad hoc network consists of 40 wireless nodes moving in 675m x 675m space with mobility model of random waypoint [16]. The moving speed has a uniform distribution between 0m/s and 20m/s. The nodes are initially paced randomly in the space, and the radio bandwidth is 2Mb/s. The range of the transmission power is 250m. The simulation is done for several different pause times as indicated in the figures. The simulation time is 900s excluding 100s initial warm up time. 15 pairs of source and destination are randomly chosen and the connections are set up. The sources are constant bit rate (CBR) sources, and send 3 data packets of 512 bytes every second. The source nodes begin the transmission starting at random time between 0 to 8 second. All simulation results are obtained by averaging 5 different simulations with different seeds. The parameters for routing operation are MW=LEDI=2 seconds, n_entries=3, CEDI=7.5 seconds, CEUI=45 seconds.

To compare the performance, fixed mix (FM) method is also simulated. In the fixed mix method, every node has its designated mix node among a finite number of ad hoc network mix nodes. We assume that the fixed mix method operates over DSR (Dynamic Source Routing) protocol, and there are 6 mix nodes among 40 nodes including the mix nodes. Other environments are the same as the above.

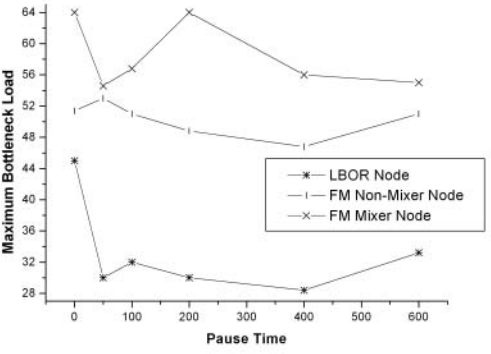


Fig. 1. The bottleneck load is the load of the link of which the load is the maximum in the network in a given measurement window. The maximum bottleneck load is the highest one throughout the simulation

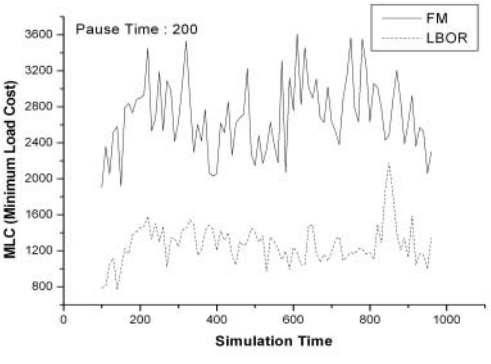


Fig. 2. The comparison of the minimum load cost of fixed mix method and LBOR is shown

In Fig. 1, the maximum bottleneck loads (MBLs) of two methods are compared. The bottleneck load (BL) is defined as the load of the link of which the load is the maximum in the network in a measurement interval. MBL is the maximum of BLs throughout the simulation. As expected the links of mix node and links around mix nodes are crowded by heavy load, and this phenomenon becomes severe as the traffic increases.

In Fig. 2, MLC (minimum load cost defined in section 3.1) to achieve neutral traffic matrix is shown. Due to the load balancing routing, it is shown that required MLC for LBOR is remarkably lower than that of fixed mix method. Similar MLC comparison results were obtained for different pause times by simulation.

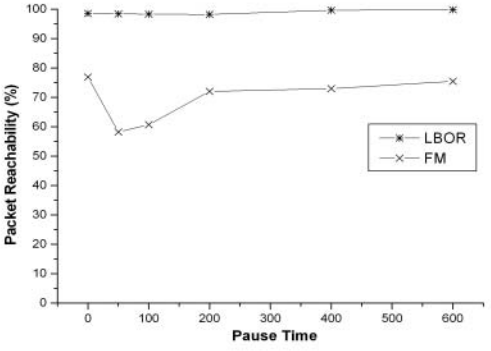


Fig. 3. The successful packet delivery ratio

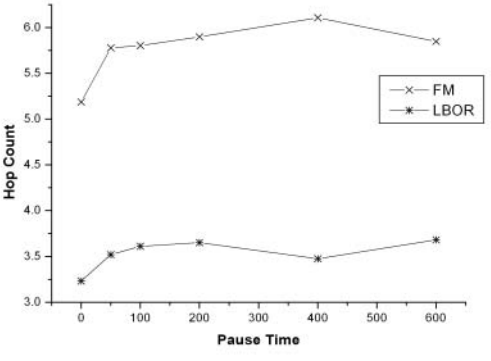


Fig. 4. Average hop count from the source to the destination

Military and emergency applications are the examples of the important applications of ad hoc network. In such circumstances, the successful packet delivery ratio is very important. In Fig. 3, the successful packet delivery ratios of the two methods are compared. The successful packet delivery ratio is defined as the ratio of the number of successfully delivered packets to the number of the total generated packets by the sources.

In the case of fixed mix method, the detour to go through a mix node causes the increase of the hop count between the source and the destination. On the other hand, LBOR does not choose the optimal route in terms of hop count, but it tries to choose the best route by avoiding bottleneck link. Even so, Fig. 4 shows that the average hop count of LBOR is much smaller than that of fixed mix method.

In Fig. 5, the ratio of the number of the routing control packets sent to the

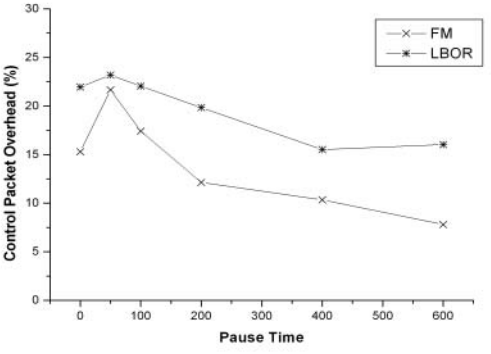


Fig. 5. Control packet overhead

number of total packets sent is shown. LBOR shows better successful packet delivery ratio, but the routing control packet overhead is slightly higher than the fixed mix method over DSR. This overhead implies the efficiency of the combined routing and data forwarding protocol.

6 Conclusion

The prevention of traffic analysis is an important part of network security. In this paper, we presented an anonymous data forwarding method accompanied by an associated routing protocol. The method assumes the nodes to be a trusted group and all nodes can take the role of onion relay node. The routing protocol is based on well-known DSR, but the load balancing concept is adopted and some operations are modified to help data anonymity and to improve other performances. The simulation results show the improved performance of the proposed method compared to the fixed mix method. The load balancing routing efficiently avoids bottleneck links in the network, which not only improves the throughput and delay but also reduces the amount of required camouflage traffic to prevent traffic analysis. Also, the packet delivery ratio is shown to be improved remarkably, which is important in ad hoc network applications.

References

[1] H. Deng, Wei Li, and D. P. Agrawal, "Routing security in wireless ad hoc networks," IEEE Communications Magazine, pp. 70-75, October 2002. 24

[2] M. Rennhard, S. Rafaeli, L. Marthy, B. Plattner, D. Hutchinson, "An architecture for an anonymity network," Proc. of 10th IEEE Intl. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2001), pp, 165-170 , Boston, USA, June 20-22, 2001. 24

[3] J. Raymond, "Traffic analysis: protocols, attacks, design issues and open problems," In H.Federrath, editor, Anonymity 2000, Volume 2009 of Lecture Notes in Computer Science, pages 10-29, Springer-Verlag, 2000. 24

[4] Y. Hu, A. Perrig, D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in The 8th ACM International Conference on Mobile Computing and Networking, September 2002. 24

[5] J. Kong, X. Hong, M. Gerla, "An anonymous on demand routing protolcol with untraceable routes for mobile ad hoc networks," Technical Report TR-030020, April, 2003, UCLA Computer Science Department, Los Angeles, California, 90025. 24

[6] A. Fasbender, D. Kesdogan, O. Kubitz, "Variable and Scalable Security: Protection of Location Information in Mobile IP," in 46th IEEE Vehicular Technology Society Conference, Atlanta, Mar. 1996. 24

[7] A. Fasbender, D. Kesdogan and O. Kubitz, "Analysis of Security and Privacy in Mobile IP," in 4th International Conference Telecommunication Systems, Modeling and Analysis, Nashvill, Mar. 21-24, 1996. 25

[8] D. L. Chaum, "Untraceable electronic mail, return address, and digital pseudonyms," Communications of the ACM 24, 2 (February 1981). 25, 28

[9] R. Newman-Wolfe, B. Venkatraman, "High level prevention of traffic analysis," Proceedings of the Seventh Annual Computer Security Applications Conference, pp. 102-109, San Antonio, December 2-6, 1991. 26, 27

[10] Y. Guan, C. Li, D. Xuan, R. Bettati, W. Zhao, "Preventing traffic analysis for real-time communication networks," in Proceedings of Milcom '99 (November 1999). 26

[11] O. Berthold, et. al., "Project anonymity and unobservability in the Internet," Computer Freedom and Privacy Conference 2000 (CFP 2000), Workshop on Freedom and Privacy by Design, 2000. 26

[12] M. G. Reed, P. Syverson, D. Goldschlag, "Anonymous connections and onion routing," in Proceedings of the IEEE Symposium on Security and Privacy (Oakland, California, May 1997), pp. 44–54. 26, 28

[13] S. Jiang, N. Vaidya and W. Zhao, "A dynamic mix method for wireless ad hoc networks," in Proceedings of IEEE Military Communication Conference (Milcom), Oct 2001. 26

[14] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding Routing Information," Workshop on Information Hiding, Cambridge, UK, May, 1996. 28

[15] K. Fall and K. Varadhan, Eds., The *ns* Manual, 2003; available from http://www-mash.cs.berkely.edu/ns/ . 29

[16] J. Broch, D. A. Maltz, D. B. Johnson, Y-C Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom '98), 1998. 29

Performance of New Broadcast Forwarding Criteria in MANET

Lijuan Zhu[1], Bu-Sung Lee[1], Boon-Chong Seet[2], Kai-Juan Wong[3], Genping Liu[1], Shell-Ying Huang[1], and Keok-Kee Lee[1]

[1] Centre for Multimedia and Network Technology
School of Computer Engineering
[2] Network Technology Research Centre Research TechnoPlaza, 4th Storey
Nanyang Technological University, Nanyang Avenue, Singapore 639798
[3] Institute for Computing Systems Architecture Informatics
University of Edinburgh JCMB, Mayfield Road U.K, Edinburgh, EH9 3JZ

Abstract. In a mobile ad hoc network (MANET), packet broadcast is common and frequently used to disseminate information. Broadcast consume large amount of bandwidth resource, which is scarce in MANET environment. The problem is to find the ideal forward node set so as to minimize the bandwidth consumed, which is a NP-Complete problem. In this paper we will investigate three dominant node approximation algorithms: dominant pruning, total dominant pruning and partial dominant pruning algorithm. An extension to the original algorithms, modified termination criteria, is proposed. Simulation results show that the new termination criteria ensure the same coverage using a reduced number of forward nodes.

1 Introduction

A mobile ad hoc network (MANET) is a self-constructing network that consists of mobile hosts roaming around and communicating with each other freely. Due to the radio transmission range limit, packets transmitted from the source may need intermediate hosts to help relaying before they can reach the destination. Such network finds applicability in military environment, wherein a platoon of soldiers may establish an ad hoc network in the region of their deployment. It has been used as well in non-military environment, e.g. Inter-vehicular communication.

There are several types of message communication services in ad hoc network. When one packet needs to be sent from one node to all other nodes in the network, a broadcast service is needed. The straightforward approach for broadcast is flooding, in which each node retransmits the received packet once. Many MANET routing protocols such as Ad Hoc On Demand Distance Vector (AODV) [1], Dynamic Source Routing (DSR) [2], Location Aided Routing (LAR) [3] and Zone Routing protocol (ZRP) [4] use flooding or its derivation to establish routes. This traditional flooding causes excessive redundant retransmissions, contention and congestion, which is referred as broadcast storm problem.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 34–43, 2004.

Some related works have been done to reduce redundant broadcast. Multipoint Relaying [6] restricts the number of neighbor nodes, which can reduce the retransmission by efficiently selecting a small set of neighbor nodes, which can cover the same region as the whole. This is a distributed mechanism; each node independently selects its own multipoint relaying set without the information of others' selection. In [7] Jie Wu and Wei Lou used a greedy algorithm to select the forward node set at each node, and then broadcast in clustered networks based on forward node set. H. Lim and C. Kim proposed a dominant pruning algorithm [8], and later, in [9] Wei Lou and Jie Wu proposed other two algorithms: total dominant pruning algorithm and partial dominant pruning algorithm, which reduced some of the redundancy in the dominant pruning algorithm. They also discussed two termination criteria: *marked* and *relayed*.

In this paper, we modify the termination criteria, and proposed a hybrid termination criterion that shows marked improvement in the reduction of number of forwarding nodes. The rest of this paper is organized as follows: section 2 introduces some graph definitions and forward node selection algorithms. Section 3 proposes the modification of the *marked*/*relayed* mechanism and the hybrid termination criteria, and examples are given at section 4. Section 5 shows the simulation results. Finally section 6 concludes the paper.

2 Problem Definition

An ad hoc network can be represented by a graph, $G = (V, E)$, where V is the set of nodes in the network, and E is the set of edges between every two nodes. An edge exists between two nodes only when they are within the transmission range of each other, and then each one is called one-hop neighbor by the other node.

For a node v, we use $N(v)$ to represent its one-hop neighbor set (including v), and $N(N(v))$ to represent its two-hop neighbor set (the union of the $N(v)$'s one-hop neighbor). The hosts can get the one-hop or two-hop neighborhood information by periodically exchanging their "Hello" messages.

The ideal way for broadcast is to select the minimum connected dominating set (MCDS) [5] nodes to do the rebroadcast. Finding MCDS is a NP-Complete problem, and extensive work has been done in the theoretical community on finding a good approximation of MCDS. The author in [5] proposed an approximation algorithm (AMCDS), which assumes that it has full network connectivity information.

AMCDS Process:

1. Color all the nodes in V white.
2. Select the node with the maximum node degree, and put it into the set C. color this node black and all its neighbors white.
3. Select the grey node with the maximum white node degree, and put it into the set C. Color this selected grey node black and all its white neighbors grey.
4. If there are still some nodes white, go to 3; else go to 5.

5. The set C is an approximation for the MCDS.

However, in the real MANET environment the nodes will only have limited information, eg. 1-hop or 2-hop neighbor information. Distributed algorithms [8][9] that make use of 1-hop and/or 2-hop information, has been proposed to find the forwarding node set in the MANET environment. These algorithms have two major tasks: selection of forwarding nodes and termination of broadcast.

2.1 Forward Node Selection Algorithms

Three distributed algorithms for forward node selection were investigated. In the discussion that follows, the following parameters are defined:

- $N(v)$ is defined as the 1-hop neighbors of node v.
- $N(N(v))$ is defined as the 2-hop neighbors of node v.
- $B(u, v) = N(v) - N(u)$. This is the set of nodes in the neighborhood of node v that are not the 1-hop neighbors of node u.
- $U_x(u, v)$: the set of forwarding nodes that would relay the packet based on the algorithm X.
- $F(u, v)$: the forwarding node set, initialized to zero members.
- Z: initialized to zero members (empty set).

The general algorithm for forward node selection when node v receives a broadcast packet from node u is as follows:

1. For every node $w_i \in B(u, v), S_i = N(w_i) \cap U(u, v)$.
2. Find w_j with the maximum size of its corresponding set S_i.
3. Add w_j to $F(u, v), Z = Z \cup S_j$, and for all the $S_i, S_i = S_i - S_j$.
4. If no new node is added to Z, exit; otherwise return to step 2.

Dominant Pruning (DP) Algorithm When node v receives a broadcast packet from node u, it will do the following steps to select the minimum forwarding nodes based on the process described above. For DP [7] the set of nodes that should be considered for forwarding the packet are as follows:

$$U_{DP}(u, v) = N(N(v)) - N(v) - N(u). \tag{1}$$

Thus, they are the nodes that are 2-hops away from node v, which are not member of the nodes that are 1-hop away from node v and node u. $U_{DP}(u, v)$ is shown as shaded areas in Figure 1. The two nodes u and v are marked as black dots.

Total Dominant Pruning (TDP) Algorithm In TDP [8] the forwarding nodes set is given by:

$$U_{TDP}(u, v) = N(N(v)) - N(N(u)). \tag{2}$$

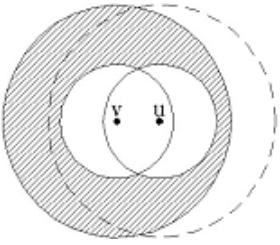


Fig. 1. Elimination of neighbor in DP

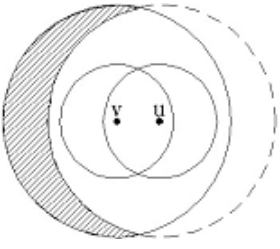


Fig. 2. Elimination of neighbor in TDP

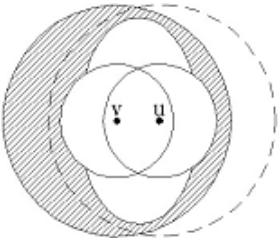


Fig. 3. Elimination of neighbor in PDP

TDP makes full use of the 2-hop neighbor information in the selection of the forwarding nodes.

Figure 2 shows the coverage of the neighbor nodes for TDP. The shaded region represents $U_{TDP}(u, v)$, the neighbor nodes area that would act as the broadcast forwarding nodes. The area is less than that of DP, thus less number of forwarding nodes. However, it suffers from the need for node u to piggyback its 2-hop neighbor set along with the broadcast packet.

Partial Dominant Pruning (PDP) Algorithm In PDP [8], they remove the nodes that are 1-hop neighbors of node u and node v as well as the 2-hop neighbors of the nodes that are 1-hop neighbors of both node u and v.

Let

$$P = N(N(v)N(u)). \tag{3}$$

then

$$U_{PDP}(u,v) = N(N(v)) - N(v) - N(u) - P. \tag{4}$$

The shaded areas in Figure 3 represent the area where the forwarding nodes can reside as defined by equation 4, i.e. $U_{TDP}(u,v)$. The area covered is slightly larger than TDP. The advantage of PDP over TDP is that it does not need to piggyback the 2-hop neighbor set information of the sender with the broadcast packet. Thus, reducing the overhead.

2.2 Termination Criteria

When a broadcast packet is sent, every intermediate node will use the DP/PDP/ TDP algorithm to select the forward nodes, and every selected forward node will use termination criteria to determine whether to broadcast the packet. Wei Lou and Jie Wu [9] proposed two termination criteria:

- *Marked:* When node v receives a broadcast packet it will not rebroadcast if all its one-hop neighbors' status are marked.
- *Relayed*: When node v receives a broadcast packet it will not rebroadcast if it has previously relayed the packet.

3 Modified Marked/Relayed

3.1 Modified Marked

We modify the *marked* termination criterion described above, and propose a hybrid one based on the two termination criteria, which is as follows:

> When node v is performing its forward node selection process, it will drop its marked neighbors out of consideration. Node v will stop rebroadcast only if all its one-hop neighbors' status are marked.

Theorem 1. *If node v is marked, then its one-hop neighbor $N(v)$ has been marked before or will be marked later.*

Proof. Node v is marked, assume that it receives the packet from its neighbor u, there are then two conditions: (1) u will select v as the forward node, if v really broadcasts the packet using the termination criteria, then its one-hop neighbor $N(v)$ will receive that packet and be marked later, otherwise its $N(v)$ has been marked before. The theorem is thus correct; (2) u will not select v as the forward node, then u will select other forward nodes to mark $N(N(u))$, $N(v) \subseteq N(N(u))$, so $N(v)$ will be marked later thus the theorem is correct.

Theorem 2. *All the nodes in the network will receive the packet through the termination criteria, when initializing one source node as the forward node.*

Proof. When one source node is a forward node and is marked, then from theorem 1, the one-hop neighbor of source will be marked later. By iteratively putting theorem1 into application, all the nodes will be marked later.

3.2 Modified Relayed

In the *modified relay*, node v will not rebroadcast the packet if it checks its status as having received the packet previously, irrespective of whether it was selected as a forward node or not.

Theorem 3. *If a node v receives the packet from node u and is not selected by u as its forward node, then there is no need for another node w to select v as its forward node.*

Proof. If node u does not select v as its forward node, then it will select other nodes to cover $N(N(u))$ including $N(v)$, so $N(v)$ will later be set as relayed, so does not need w to select v as the forward node to cover $N(v) \cup N(N(w))$.

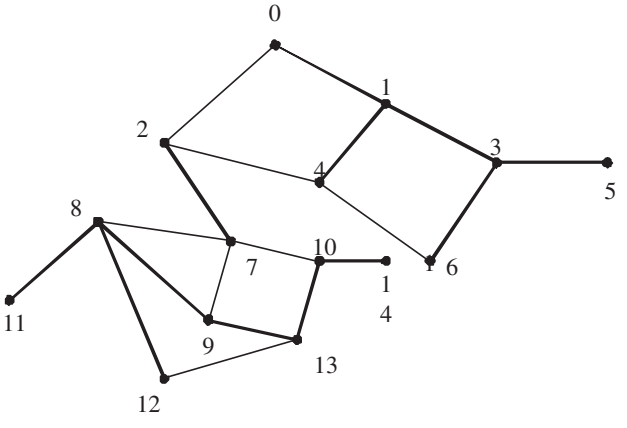


Fig. 4. Example MANET topology with source node 0

Table 1. Forward nodes list for Figure 4 with different algorithm and termination criteria

	Original Marked	Modified Marked	Original Relayed	Modified relayed
DP	0,1,2,3,4,7,8,10,12	0,1,2,3,7,8,10,12	0,1,2,3,4,7,8, 9,10,12,13	0,1,2,3,7,8,10,12,13
PDP	0,1,2,3,4,7,8,10	0,1,2,3,7,8,10	0,1,2,3,4,7,8,9,10,13	0,1,2,3,7,8,10,13
TDP	0,1,2,3,4,7,8,10	0,1,2,3,7,8,10	0,1,2,3,4,7,8,10	0,1,2,3,7,8,10

4 Example

Figure 4 shows the connectivity map of the MANET environment used in our example to illustrate the difference between the forwarding algorithms and termi-

nation criteria. Table 3.2 depicts the differences between *"modified"* and *"original"* in the DP algorithm. Using the original marked termination algorithm, node 2 will select node 4 as its forward node, and for the termination criteria, node 4 will do the rebroadcast, since node 6 is unmarked. When the new modified marked termination algorithm is used, node 2 will not choose node 4 as its forward node. This is because after node 1 has selected node 3 as its forward node, node 4 will be marked, then node 2 will drop node 4 out of consideration as a forward node.

In the relay termination algorithm, node 13 will select node 9 as its forward node. Since node 9 has not relayed the packet before, it will act as the forward node. In the case of modified relay, node 13 will select node 9 as its forward node. When node 9 receives the packet, based on the new but termination criteria, node 9 will not rebroadcast the packet since its status is relayed. In this example, we have illustrated how the different dominant nodes selection algorithms have benefited from using the *"modified"* termination mechanism. The number of forward nodes is reduced in this example.

5 Simulation Results

Simulations are done using the unit disk graph [10]. The set-up for the simulation is as follows:

- Graph area = 100x100
- Transmission range = 40
- Number of nodes = 20 to 100

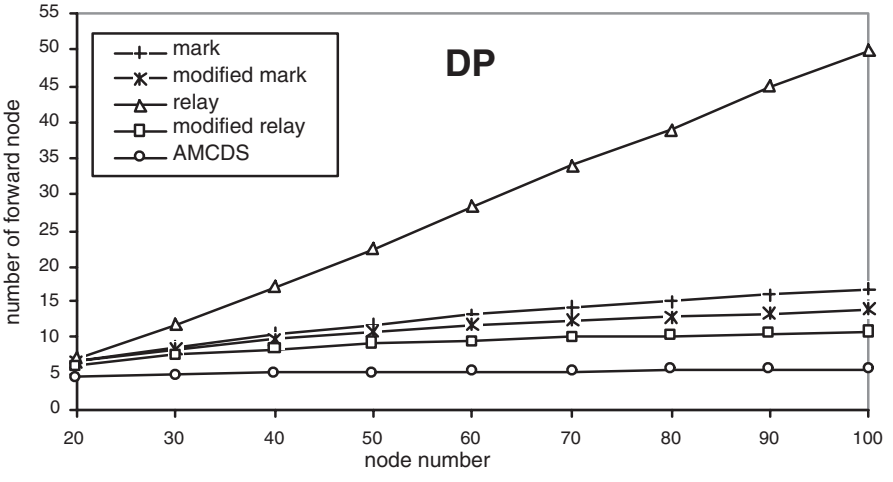


Fig. 5. Average number of forward nodes for DP algorithm

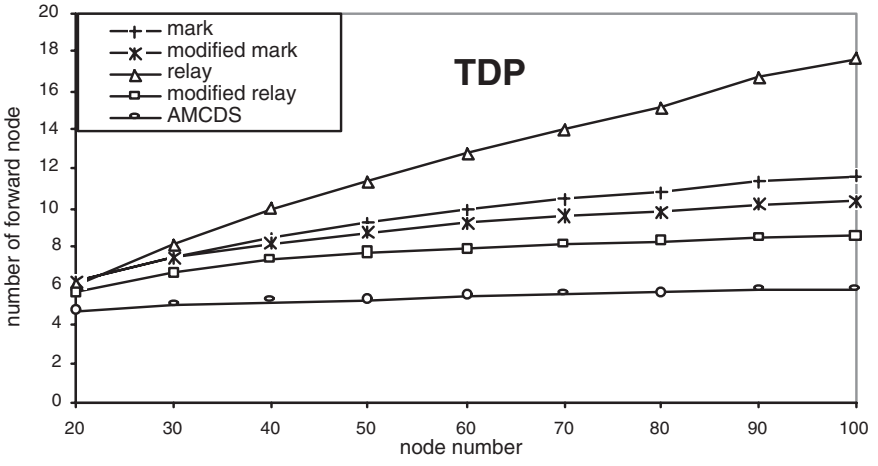


Fig. 6. Average number of forward nodes for TDP algorithm

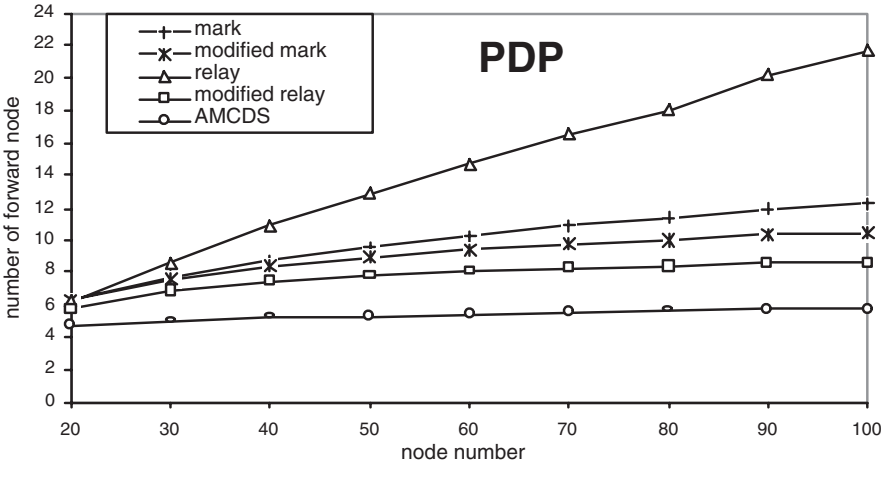


Fig. 7. Average number of forward nodes for PDP algorithm

A total of 400 sample graphs are generated and for each graph 20 nodes are selected as the broadcast source. The results on number of forward nodes are then averaged across all the experiment and plotted. The experiments were carried out for the three dominant node selection algorithms: DP, TDP, and PDP.

Figure 5, Figure 6 and Figure 7 show the average number of forward nodes with different termination criteria and different dominant node algorithm. In all the figures, AMCDS is the lower bound for the performance as it's a centralized

system, while the others use a distributed algorithm. All the algorithms perform poorly when they use the relay termination criteria. This is to be expected as it only makes use of its own information, i.e. whether it had previously forwarded the packet. The other termination criteria makes use of its 1-hop neighbors information.

The modified termination criteria reduces the number of forwarding nodes. The performance of the algorithm using different termination criteria increases as the number of nodes increase. Example for node number of 100, the percentage reduction in the number of forward nodes based on modified marked compared with original marked termination criteria is 16.8% for DP, 14.6% for PDP, and 11.7% for TDP. The percentage of the reduced forward nodes based on modified relay compared with the original relay is 78.3% for DP, 60.0% for PDP and 51.5% for TDP.

6 Conclusions

In this paper, we have summarized some previously promising algorithms to select the forward nodes, and related termination criteria to determine whether node should broadcast or not. Since the number of forward nodes depends largely on the termination criteria, we have modified one termination criteria and proposed a hybrid one. From the simulation results, we can see that the new termination mechanisms show marked improvement when incorporated with existing forwarding node selection algorithms, especially when the number of nodes in the network increases.

References

[1] C. Perkins, E. Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing", Internet Draft: draft-ietf-manet-aodv-09.txt, Nov. 2001. 34

[2] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks". In T. Imelinsky and H. Korth, editors, Mobile Computing, Kluwer Academic publishers, pp.153-181, 1996. 34

[3] Y.Ko and N. H. Vaidya, "location-aided routing (LAR) in mobile ad hoc networks", In proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), pp.66-75, 1998 34

[4] Zygmunt J. Haas, Marc R. Pearlm, and prince Samar, "The Zone Routing Protocol (ZRP) for ad hoc networks", IETF MANET Internet Draft, July 2002. 34

[5] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in Ad hoc Networks Using a Virtual Backbone", Proc. Int'l Conf. Computer Comm. And Networks '97, pp. 1-20, Sept. 1997. 35

[6] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Message in Moble Wireless Networks", Proc. 35th Annual Hawaii International Conference on System Science-2002. 35

[7] J. Wu and W. Lou, "Forward-Node-Set-Based Broadcast in Clustered Mobile Ad hoc Networks", Technical Report CSE-02-15, June 2002. 35, 36

[8] H. Lim and C. Kim, "Flooding in Wireless Ad hoc Networks", computer Comm. J.,vol. 24, no. 3-4, pp. 353-363, 2001. 35, 36, 37

[9] W. Lou and J. Wu, " On Reducing Broadcast Redundancy in Ad hoc Wireless Ad hoc Networks", IEEE Transactions on Mobile Computing, vol. 1, no. 2 APRIL-JUNE 2002. 35, 36, 38

[10] B. N. Clark, C. J.Colbourn, and D. S.Johnson, "Unit Disk Graphs", Discrete Math., vol. 86, pp. 165-177, 1990. 40

[11] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a mobile Ad hoc Networks", Proc.MOBICOM '99, pp. 151-162, Aug. 1999.

A Simple Load-Balancing Approach in Secure Ad Hoc Networks*

Younghwan Yoo[1] and Sanghyun Ahn[2,**]

[1] School of Computer Sci. & Eng., Seoul Nat'l University, Seoul 151-742, Korea
yhyoo@archi.snu.ac.kr
[2] School of Computer Science, University of Seoul, Seoul 130-743, Korea
ahn@venus.uos.ac.kr

Abstract. Most ad-hoc routing protocols such as AODV and DSR do not try to search for new routes if the network topology does not change. Hence, with low node mobility, traffic may be concentrated on several nodes, which results in long end-to-end delay due to congestion at the nodes. Furthermore, since some specific nodes are continuously used for long duration, their battery power may be rapidly exhausted. Expiration of nodes causes connections traversing the nodes to be disrupted and makes many routing requests be generated at the same time. Therefore, we propose a load balancing approach called Simple Load-balancing Approach (SLA), which resolves the traffic concentration problem by allowing each node to drop RREQ or to give up packet forwarding depending on its own traffic load. Meanwhile, mobile nodes may deliberately give up forwarding packets to save their own energy. To make nodes volunteer in packet forwarding we also suggest a payment scheme called Protocol-Independent Fairness Algorithm (PIFA) for packet forwarding. To evaluate the performance of SLA we compare two cases where AODV employs SLA or not. Simulation results show that SLA can distribute traffic load well and improve performance of entire ad-hoc networks.

1 Introduction

The mobile ad-hoc network (MANET) is a wireless network that has neither fixed communication infrastructure nor fixed base stations. Nodes in an ad-hoc network can freely move, hence the network topology may continuously change. In addition, the characteristics of wireless channels such as the limited data transmission range, low bandwidth, high error rate, and limited battery power make routing on an ad-hoc network a difficult problem to deal with. The most prominent ad-hoc routing protocols are AODV [1] and DSR [2]. AODV tries to find a new route by broadcasting route request (RREQ) messages and maintains only one route for a destination. On the other hand, DSR performs source routing and maintains more than one route for a destination. Therefore, if a current route

* This work was supported by Korea Research Foundation Grant. (KRF-2003-041-D00501)

** Corresponding author

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 44–53, 2004.

is not available, DSR selects one of the alternate routes as a new route without triggering the route discovery mechanism.

According to the study [3] about both AODV and DSR, as the node mobility decreases, the packet delivery ratio increases and the routing overhead decreases. An interesting result in [3] is that the packet delivery delay increases as the node mobility decreases. If nodes do not move much, current routes become used for long duration because AODV and DSR reinitiate route discovery mechanisms only when current routes are stale. This may result in traffic concentration on several specific nodes, and this traffic concentration not only causes a high transmission delay but also forces a few specific nodes to consume their all power to forward others' packets. In turn, expiration of some nodes makes other nodes increase their transmission power to make up packet relaying roles of expired nodes. As a result, the lifetime of entire ad-hoc networks becomes far shorter than the cases where traffic is distributed well.

Thus, we propose a new routing approach called Simple Load-balancing Approach (SLA) which considers load balancing in the ad-hoc network. The concept of SLA differs from load balancing in the wired network. In the wired network, the main objective of load balancing is to reduce congestion to enhance the overall network performance. On the other hand, SLA tries to extend the expiration of mobile node power by preventing traffic concentration on a few nodes, which may frequently occur under the low mobility situation. AODV and DSR do not search for new routes as long as current routes are available. In the case with low mobility, this feature may cause nodes on the current routes to be congested. Hence, SLA allows each node to determine whether it is under heavy load condition or not and to let some other nodes take its place by explicitly giving up packet forwarding or implicitly dropping RREQ from other nodes. Consequently, this spreads traffic uniformly on a whole network and extends the lifetime of an entire ad-hoc network by making all MANET nodes fairly consume their energy.

SLA is not an independent routing protocol but a supplementary part to any existing ad-hoc routing protocols like AODV and DSR. SLA is a simple method based on the autonomy of each node, assuming that it is operating in a secure ad-hoc networks where all MANET nodes honestly forward other nodes' packets. Actually, however, some selfish nodes may deliberately give up packet forwarding to save their own energy, if they do not receive an appropriate compensation. We propose a credit-based scheme called Protocol-Independent Fairness Algorithm (PIFA) for urging nodes to voluntarily participate in forwarding packets. Although PPM and PTM in [4] and Sprite [5] were designed for the same purpose, they require full path information from source to destination to apply their schemes. Thus they can be used only with DSR, not with AODV. On the other hand, PIFA is compatible to all types of routing protocol.

The paper is organized as follows: Section 2 introduces previous load balancing approaches and fairness schemes for power consumption. In Sections 3 and 4, we describe the operations of SLA and PIFA respectively. Section 5 shows the results of performance evaluation for SLA and Section 6 concludes this paper.

2 Related Work

Both Load-Balanced Ad hoc Routing (LBAR) [6] and Dynamic Load-Aware Routing (DLAR) [7] consider load balancing in the MANET environment. They try to find the least loaded route by checking the network traffic condition. As the cost of each route LBAR uses the node activity and the traffic interference, which mean the number of paths through each node and the sum of activity of all its neighboring nodes respectively. For DLAR, the number of packets buffered in the interface of intermediate nodes is used as the primary route selection criteria. In LBAR, a setup message broadcast by a source node carries the cost, which is updated by every intermediate node based on its node activity value. Similar to LBAR, DLAR ROUTE REQUEST messages collect load information, or the number of packets in buffers, of every intermediate node while they proceed to destination.

The above algorithms assume that all the nodes in MANETs cooperate for a common object. However, some research have raised a question that some selfish nodes may attempt not to forward others' packets for saving their own energy. To remove this possibility, Packet Purse Model (PPM) and Packet Trade Model (PTM) in [4] pay for packet forwarding in virtual currency named *nugget*. Sprite (a simple, cheat-proof, credit-based system) [5] also utilizes credit to give incentive to the nodes that forward packet. Unlike PPM and PTM depending on any tamper-proof hardware, Sprite suggests a security algorithm to prohibit cheating by the help of a credit management server called Credit Clearance Service (CCS). However, since Sprite should report a *receipt* for every message to CCS, it can be too much overhead to MANET. All of PPM, PTM, and Sprite should know full path information from source to destination to apply its algorithm, so they are compatible only to DSR not to AODV.

3 Simple Load-Balancing Approach (SLA)

When MANET nodes move fast, AODV and DSR can automatically achieve the load balancing effect because they search for new paths whenever the network topology changes. On the other hand, in low mobility situation, AODV and DSR do not need to discover new routes, which forces the same routes to be used for long duration. This may cause severe congestion on the routes, which eventually degrades the network performance.

To overcome this problem we propose Simple Load-balancing Approach (SLA) that can be implemented as an additional module to any existing routing protocols. In SLA each node independently determines whether it is suffering from traffic concentration or not. Calculating the ratio of its own traffic load to the whole average, each node periodically checks if the ratio is over two threshold values τ_l and τ_h ($\tau_l \leq \tau_h$). According to the result, it selects one of the three states: the normal state, the passive load-balancing state, and the active load-balancing state. In the normal state, a node operates as a common MANET node, broadcasting RREQs and forwarding data packets. On the other hand, in

Table 1. The traffic load ratios presented by M1, M2, and M3

	P_i	E_i	R_i	L_i(M1)	L_i(M2)	L_i(M3)
node A	100	1000	800	1.33	0.8	0.56
node B	50	200	100	0.67	2	4.5
average	75	600	450			

(Required energy to forward one packet, $f = 2$)

the passive and active load-balancing state each node tries to balance its own traffic load with the average in the MANET.

Meanwhile, the traffic load ratio L_i of node i can be presented by the following three different ways M1, M2, and M3. E_i and P_i are node i's current energy and the number of buffered packets, and E and P are their averages in the whole network. The way to calculate the averages is described in Appendix.

– M1: considering only the number of packets buffered in queues

$$L_i = \frac{P_i}{P}$$

– M2: considering the ratio of the number of packets to the residual power

$$L_i = \frac{P_i/E_i}{P/E}$$

– M3: considering the ratio of the expected power R_i left over after all buffered packets are forwarded

$$L_i = \frac{R}{R_i} = \frac{E - f \times P}{E_i - f \times P_i}, \text{ where } f \text{ is the energy required to forward one packet}$$

M1 is very simple but it does not consider node energy information at all. On the other hand, both M2 and M3 take into account it. M2 is simpler than M3, but M3 is more effective in load-balancing as shown from the example in Table 1. Although node A has two times as many packets as node B, its energy also is five times more. Thus it is desirable for node A to deal with more packets for balancing the speed of node expiration. Comparing between L_i(M2) and L_i(M3), we can see that M3 reflects this desirability more clearly. To the contrary, M1 misjudges that it is desirable that node B treats more packets than node A from now on, since M1 does not consider node energy at all.

3.1 Ignore Route Requests (IGN_REQ)

When the ratio of its own traffic load to the average is between τ_l and τ_h, node i moves to the passive load-balancing state called IGN_REQ. In this state RREQs from other nodes are ignored until L_i goes below τ_l. As a result, new routes passing through node i are prevented from being established, in turn the traffic passing through node i is reduced eventually.

3.2 Give Up Packet Forwarding (GIVE_UP)

When a node is overloaded over τ_h times as much as the average, it changes its state to the active load-balancing state called GIVE_UP. In this state the node tries to reduce traffic load by giving up forwarding some packets, not to mention that it ignores new RREQs. A node i in the GIVE_UP state sends a GIVE_UP message to the source node of the first packet passing through node i after it enters the GIVE_UP state. The GIVE_UP message contains the source identification and the destination list of the source initiated routes traversing node i. When the source node receives the message, it initiates a route discovery mechanism to the destinations to find new routes detouring the node i.

One thing to notice is that node i in the GIVE_UP state does not discard the corresponding routing table entry as long as data packets come from the source, since the timer for the entry will not be expired due to those data packets. As a result, even though a new route cannot be found, node i can continue to forward data packets along the current route passing through node i. On the other hand, if a new route is found, the routing table entry will be discarded in the timer expiration since no more data packets arrive. As in the IGN_REQ state, node i in the GIVE_UP state does not participate in any route discovery procedure. Node i returns to the IGN_REQ or the normal state if its traffic load ratio L_i goes below the specified thresholds τ_h or τ_l respectively.

SLA allows each node to resolve its own congestion situation so that the limited resource of mobile nodes can be fairly used. Irrespective of the degree of node mobility, SLA works well in terms of traffic distribution as we will see in Section 5. Also SLA can be easily applied to any existing ad-hoc routing protocols as an additional SLA-specific module. SLA is independent of the types of routing protocol.

4 Protocol-Independent Fairness Algorithm (PIFA)

Nodes in the preceding section are assumed to volunteer in packet forwarding, in other words they honestly decide whether they should enter the IGN_REQ or the GIVE_UP state or not. Some selfish nodes, however, may not cooperate with one another and may attempt not to forward others' packets for saving their own energy. To isolate selfish nodes or to make them voluntarily participate in forwarding packets, we suggest a credit-based payment scheme for packet forwarding. Although PPM/PTM [4] and Sprite [5] are also credit-based methods, they can be used only with a source routing protocol like DSR, but our method called Protocol-Independent Fairness Algorithm (PIFA) can be adopted irrespective of the types of routing protocol.

In MANETs using PIFA, nodes can originate packets only when they have enough credits, and they earn the credits by forwarding others' packets. PIFA can detect and isolate a single malicious node which tries to cheat others on the number of forwarding packets to acquire more credits than it should actually receive. PIFA assumes that there is no collusion between two or more nodes and

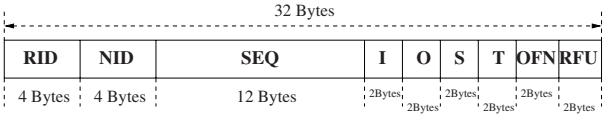
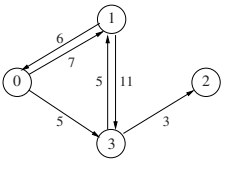


Fig. 1. A report message to CM

RID	NID	SEQ	I	O	S	T	OFN
0	1	128	6	7	7	2	2
0	3	128	0	5	1	0	0
1	0	128	7	6	2	3	7
1	3	128	5	11	7	1	3
2	3	128	3	0	0	3	2
3	0	128	5	0	0	4	1
3	1	128	11	5	3	9	7
3	2	128	0	3	2	0	0

(a) Report messages to CM



(b) Topology calculated with the reports

Fig. 2. Process to compute the current topology from reports

that nodes do not know full path information from source to destination just like the case of AODV. One hop packet transmission between two nodes always succeeds.

PIFA needs a server node called Credit Manager (CM), which manages nodes' Credit Database (CDB). Other MANET nodes periodically report to CM on the number of packets they forwarded in each time interval; and CM verifies the credibility of the reports and infers the current MANET topology from them.

Fig. 1 shows the fields of a report message that a MANET node sends to CM for each of its neighbors. The meaning of each field is as follows:

- **RID**: the ID of a reporter
- **NID**: the ID of a neighbor node
- **SEQ**: the sequence number of the current node's reports for synchronizing report messages
- **I**: the number of input packets from the neighbor
- **O**: the number of output packets to the neighbor
- **S**: the number of packets starting at the current node among ones going to the neighbor
- **T**: the number of packets terminated at the current node among ones coming from the neighbor
- **OFN**: the number of packets originated from the neighbor itself among ones coming from the neighbor
- **RFU**: reserved for future use

Fig. 2 (a) gives an example of report messages to CM. Having collected report messages with the same sequence number, CM verifies the credibility of

the reports by three checkpoints. First, for every link the number of output packets from a node should be the same as the number of input packets to the opposite side node. For instance, if nodes n and m are neighbors, and $\mathbf{A}_{n,m}$ is the **A** field of a message whose **RID** and **NID** are n and m, then $\mathbf{O}_{n,m} = \mathbf{I}_{m,n}$.

Second, the difference between the total number of input packets ($\sum \mathbf{I}$) and the total number of terminated packets ($\sum \mathbf{T}$) at a node is the number of forwarding packets. Also, the difference between the total number of output packets ($\sum \mathbf{O}$) and the total number of starting packets ($\sum \mathbf{S}$) is the same as the number of forwarding packets. Therefore, if F_n is the number of packets forwarded by node n and A_n is the set of adjacent nodes of node n, then

$$F_n = \sum_{m \in A_n} \mathbf{I}_{n,m} - \sum_{m \in A_n} \mathbf{T}_{n,m} = \sum_{m \in A_n} \mathbf{O}_{n,m} - \sum_{m \in A_n} \mathbf{S}_{n,m} \ . \tag{1}$$

Each node receives credits in proportion to this F_n from CM; and these credits are spent as much as $\sum \mathbf{S} \times H_{avg}$, where H_{avg} is the average hop count between two nodes in the network. In principle the credits of a node should decrease in proportion to the number of nodes which forward its packets until they arrive at destination, but because actual full path information is not known in AODV, we utilize the average hop count.

Finally, **S** at a node's report has to be identical with **OFN** of the next hop node. The purpose of **OFN** is to prevent a malicious node from manipulating the number of forwarding packets by changing both $\sum \mathbf{T}$ and $\sum \mathbf{S}$ in Equ. (1). This **OFN** is recorded by counting the number of packets originated at a neighbor out of ones coming from the neighbor. If **S** at a node's report does not accord with **OFN** of the next hop node, CM adopts the **OFN** since the next hop node has no motive to cheat on the number of forwarding packets of the preceding node. After passing these all checkpoints, CM infers the topology from the reports as shown in Fig. 2 (b). This topology may not be the same as the actual current topology due to node mobility, but it does not matter because it does not have an effect on the calculation of forwarding packet numbers.

In MANETs using PIFA, nodes would not deliberately enter the IGN_REQ or GIVE_UP state because they need to earn credits to send their own packets. At the initial time, CM assigns a fixed amount of credits to all nodes so that they can send some packets before earning credits for themselves.

5 Performance Evaluation

To evaluate the performance of SLA, we used the $ns-2$ simulator. The simulation environment consists of 50 mobile nodes in the range of 1000m × 1000m, and the transmission range of each node and the channel capacity are set to 250m and 2Mbps respectively. The moving direction of each mobile node is randomly chosen and we apply the free space propagation model in which the power of the signal decreases $1/d^2$ for the distance d. The IEEE 802.11 is adopted as the medium access control protocol and AODV is used as the underlying unicast routing protocols. Source and destination node pairs are randomly selected and

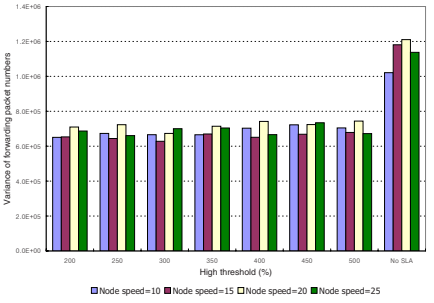


Fig. 3. Variance of forwarding packet numbers with node mobility

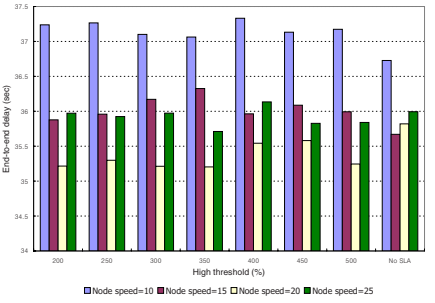


Fig. 4. End-to-end delay with node mobility

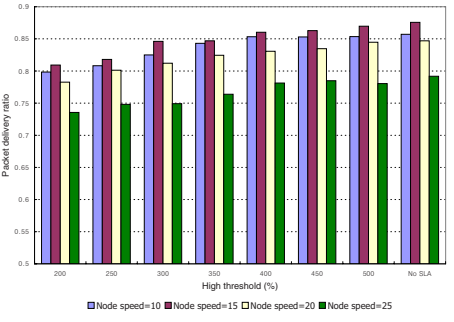


Fig. 5. Packet delivery ratio with node mobility

each source generates two 512 byte packets per second in constant bit rates (CBR). The total simulation time is set to 1000 seconds.

Figs. 3, 4, and 5 show the performance with changing the average speed of node mobility (10, 15, 20, and 25 m/s). Nodes stay for 60 seconds on every new location. The horizontal axes of all the figures represent the high threshold τ_h. If the ratio of a single node traffic to the whole network average, say L_i of M1, is higher than τ_h, the node changes its state to GIVE_UP. The low threshold τ_l is fixed at 150. The results of the basic AODV without SLA are shown at the rightmost part of all the figures. Fig. 3 shows the variance of the number of packets that each node forwards for 10 seconds. As shown in the figure, SLA balances load remarkably better than the basic AODV, but its effect gradually diminishes as the node mobility increases. This is because AODV naturally initiates the route discovery procedure more frequently with higher mobility, resulting in a better traffic distribution. Also, we can see that the lower threshold, the better traffic distribution. Figs. 4 and 5 show the end-to-end delay and the packet delivery ratio. The delay of SLA is slightly longer than the basic AODV, since new routes found after GIVE_UP messages arrive are generally longer than the shortest path which the basic AODV uses. The

lower threshold induces the more frequent route discovery procedures, in turn the longer end-to-end delay. Meanwhile, the packet delivery ratio goes down as the delay increases because nodes move more times and fartherer during the longer delay. However, the difference in the delay and packet delivery ratio can be ignored, considering SLA's outstanding achievement in load-balancing. Therefore, we can conclude that SLA improves the performance of AODV since it works pretty well in terms of traffic distribution without compensating other performance criteria.

6 Conclusion

In AODV and DSR, some nodes may be overburdened with forwarding packets especially when the network mobility is low. This traffic concentration problem is undesirable since MANET nodes undergoing this concentration will be shortly expired due to its limited power. To overcome this problem, we proposed a simple method for load-balancing, SLA, in which each node autonomously checks its traffic condition and asks a source node to find an alternate route detouring it. SLA can be added to any ad-hoc routing protocols as an independent module. Simulation results showed that SLA can distribute traffic load well without sacrificing the overall packet delivery performance and can achieve the fairness of energy consumption among mobile devices. In addition, we suggested a credit-based payment scheme PIFA to make some selfish nodes volunteer to forward packets, which can be used regardless of the types of routing protocol.

Appendix

In this Appendix, we propose a simple method to calculate the whole average of energy or buffered packets in the entire MANET. Our method utilizes Hello Message [1], which is a RREP message with TTL = 1. The Hello Messages are broadcast to confirm link connectivity when a RREQ or another appropriate layer 2 message is not broadcast for HELLO_INTERVAL whose default value is 1,000 milliseconds. The operation of our method is similar to a distance vector protocol. A RREP that is used as the Hello Message has a reserved field of nine bits in its header. Putting its own energy and buffered packet information into this field, a node exchanges Hello with its neighbors. Five bits of the reserved field are used for the number of buffered packets and the remaining four bits contain residual power information divided into 16 levels. The detailed operations are described below. For convenience' sake only energy information is mentioned and each step is depicted in Fig. 6.

1. Node i initially exchanges its own energy E_i with its neighbor nodes.
2. Computing the average E_i' of E_i and all E_j (node j is node i's neighbor), node i informs neighbor nodes of E_i'.
3. Node i periodically updates E_i' with the average of E_i' and all E_j' (node j is node i's neighbor).

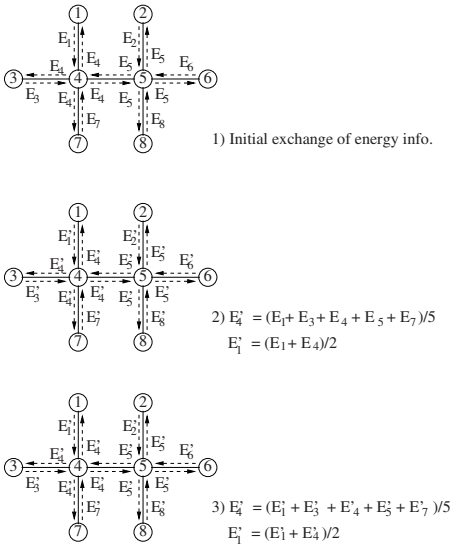


Fig. 6. Steps to calculate the whole average of node energy

In our simulation, E_i' becomes stable and almost the same as the whole average after Step 3 is repeated three times or so, when a MANET consists of 30 nodes.

References

[1] Perkins, C. E., Royer, E. M., Das, S. R.: Ad hoc On-demand Distance Vector (AODV) Routing. IETF MANET Internet-draft (2003) 44, 52

[2] Johnson, D. B., Maltz, D. A., Hu, Y.-C.: The Dynamic Source Routing (DSR) Protocol for Mobile Ad hoc Networks. IETF MANET Internet-draft (2003) 44

[3] Das, S. R., Perkins, C. E., Royer, E. M.: Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks. Proc. IEEE INFOCOM (2000) 3–12 45

[4] Buttyán, L., Hubaux, J.-P.: Enforcing Service Availability in Mobile Ad-Hoc Networks. Proc. ACM MobiHoc (2000) 87–96 45, 46, 48

[5] Zhong, S., Chen, J., Yang, Y. R.: Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks. Proc. IEEE INFOCOM (2003) 1987–1997 45, 46, 48

[6] Hassanein, H., Zhou, A.: Routing with Load Balancing in Wireless Ad hoc Networks. Proc. ACM MSWiM (2001) 89–96 46

[7] Lee, S.-J., Gerla, M.: Dynamic Load-Aware Routing in Ad hoc Networks. Proc. IEEE ICC (2001) 3206–3210 46

An Energy-Efficient Reliable Transport for Wireless Sensor Networks

Keun Soo Yim, Jihong Kim*, and Kern Koh

School of Computer Science and Engineering
Seoul National University, Seoul 151-742, Korea
{ksyim,kernkoh}@oslab.snu.ac.kr
jihong@davinci.snu.ac.kr

Abstract. In a wireless sensor network, sensor devices are connected by unreliable radio channels. Thus, the reliable packet delivery is an important design challenge. The existing sensor-to-base reliable transport mechanism, however, depends on a centralized manager node, incurring large control overheads of synchronizing reporting frequencies. In this paper, we present a *decentralized* reliable transport (DRT) with two novel decentralized reliability control schemes. First, we propose an independent reporting scheme where each sensor node stochastically makes reporting decisions. Second, we describe a cooperative reporting scheme where every sensor node implicitly cooperates with its neighbors for the uniform reporting. In the reporting step, DRT uses a reliable MAC channel, which is specifically optimized for reducing the energy dissipation. Experimental results show that DRT satisfies the desired delivery rate reliably in a decentralized manner while it significantly reduces the energy consumption of the radio device and the communication time.

1 Introduction

As a wireless sensor network reflects the network environment of next-generation ubiquitous computing, researches on a wireless sensor network are becoming increasingly active [1]. A wireless sensor network is organized with numerous tiny sensor devices, which collect various physical data such as temperature, light, sound, and movement in a cost-efficient manner. The sensor nodes, i.e. Mote [2] and Smart-Its [3], are interconnected by a harsh radio channel so as to build an ad-hoc communication path. This routing path is mainly used to transfer the sample data from sensor nodes to base nodes, which report the data to users. Due to the relative ease of construction, various practitioners are trying to use this sensor network for monitoring and collecting tasks.

In fact, there exist mainly two challenges that should be addressed to practically use the wireless sensor networks in various social fields. One is the reliability problem. In this paper, we define the *reliability* as the ratio of the number of collected sample data to the number of interested sensor nodes. Since a radio

* This research was supported by University IT Research Center Project in Korea.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 54–64, 2004.

device used in the sensor nodes has a high packet error rate of about 50% [6], the reliable data delivery is an important challenge, especially when the network dimension is large. The other is the energy consumption problem. As the sensor nodes are powered by their small batteries that usually cannot be recharged, the power saving is a paramount design objective in the wireless sensor networks. Hill and Culler *et al.* [4, 5] reported that a radio device forms about 20-60% of the total power consumption of a sensor node. Therefore, the radio device should be managed efficiently to lessen the power consumption.

One of the effective ways of improving the reliability is adopting a reliable transport protocol [1]. Recently two different reliable transports of RMST [7] and ESRT [8] were developed for the sensor networks. RMST employs ARQ [13] protocol in link layer and a selective NACK protocol in transport layer. Thus, it guarantees the complete end-to-end reliability from sensor nodes to base nodes. However, the sensor networks are often interested in reliable detection of the collective information provided by the numerous sensor nodes not in their individual reliable reports. Thus, the complete end-to-end reliable transports include RMST are not generally applicable in the sensor network regime.

In order to provide a reliable detection of events occupied in the sensor network, a centralized reliable transport, namely ESRT [8], was recently presented. In ESRT, a base node directly controls the reporting frequency of all sensor nodes, so that it ensures the desired partial reliability. Specifically, when the current measured reliability is lower than the desired, the base node aggressively adjusts the reporting frequency so as to reach the desired one as soon as possible. If the measured one is higher than the required, the base node conservatively reduces the reporting frequency so as to conserve the energy.

However, since ESRT frequently changes the reporting frequency of all sensor nodes using a multicast protocol, it incurs serious control overheads of the energy consumption and the network congestion. Even though when the base node uses a powerful radio device for controlling the reporting frequency, the powerful yet expensive base node can not efficiently deliver the control signal to all sensor nodes in the wireless sensor networks. For example, as the network diameter increases, the radio device becomes gradually expensive so as to provide the high radio signal power. Also when the physical network topology is complex, the base node cannot directly transfer the control signal to all sensor nodes even though the distance to the sensor nodes is shorter than the communication range of its radio device.

To overcome these technical obstacles, in this paper, we present an energy-efficient decentralized reliable transport (DRT) that does not require a powerful radio device. In DRT, the querying packet embeds the desired reliability value and is propagated to all sensor nodes placed in the routing path. Then, all sensor nodes use one of two decentralized reliability control schemes of an independent reporting and a cooperative reporting. In the independent reporting, each sensor node stochastically makes the reporting decision whether it will report the sample data or not. Next, in the cooperative reporting, each sensor node implicitly cooperates with its neighbor nodes for the uniform reporting. These two decen-

tralized reliability control schemes assure the measured reliability as similar to the desired one because the sensor nodes use reliable channel in the delivery of their sample data. The reliable channel is specifically optimized for improving the delivery speed and reducing the energy dissipation.

To study the performance of DRT over the existing transports, we use a byte-level sensor network simulator, which has been specifically developed for this purpose. The simulation results show that DRT with the cooperative scheme always guarantees the desired reliability. The results also show that DRT reduces both the energy consumption of the radio device and the communication time significantly. Finally, we show that the cooperative reporting scheme distributes the reporting density more uniformly than the independent reporting scheme.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the overall organization and specific working mechanisms of DRT. We describe the simulation methodology and the simulation results in Section 4 and 5, respectively. Section 6 concludes with a summary.

2 Related Work

In this section, we provide the performance of an unreliable protocol that uses a retransmission technique. Then, we summarize the existing reliable MAC- and transport-layer protocols in comparison with DRT.

The main characteristic of the wireless sensor networks is that they have a high packet error rate, and the error rate is liable to change depending on the physical state. Rubin [4] reported that a harsh radio channel used in the sensor networks has a high bit error rate (e) of about 0.5%. If a packet length (L) is twenty bytes, the average packet error rate (E_L) exceeds 55%. With an unreliable protocol, the successful packet delivery rate can be described as $(1-E_L)H$ where variable H means a number of hops. When a hop count is larger than or equal to four, the delivery rate is less than 10%.

A packet retransmission technique is a simple way of improving the delivery rate. As a retransmission count (R) increases, the effective packet error rate ($E_{L;R} = E_L^{R+1}$) is gradually decreased. For example, when a packet error rate is 50% and a retransmission count is three, the effective packet error rate is lower than 6.5%. However, this technique results in the heavy network traffic in direct proportional to the retransmission count. This heavy traffic consequently incurs a large amount of power consumption. Thus, the retransmission technique does not efficiently improve the delivery rate in terms of power consumption.

The automatic repeat request (ARQ) [14] is a reliable MAC protocol that guarantees the reliable hop-by-hop delivery. Basically, ARQ is classified into three major types depending on the spresence of sender and receiver buffers.

First, in the stop-and-wait ARQ, the sender transfers a data packet and waits until it receives the acknowledgement (ACK) packet. If the sender does not receive the ACK packet before its retransmission timer is expired, it retransfers the data packet and repeats this procedure until it receives the ACK packet. Since this technique delivers the packets one by one, both the sender and the

receiver require a small size buffer. However, this technique at the same time results in a slow packet delivery speed.

Second, in the go-back-N ARQ, the sender transfers a group of data packets without having to wait an ACK packet. The sender stores the sent packets in its buffer. Then, the receiver replies by using a cumulative ACK packet. For example, when the sender transfers packets (P_1, P_2, P_3), and the packet (P_1) is lost due to an error, the receiver requests to the sender for resending the data packet from (P_1). Then, the sender retransfers the data packets (P_1, P_2) even though the packet (P_2) is correctly delivered to the receiver in the first transmission because the receiver node does not have a buffer. The selective-repeat ARQ addresses this drawback by using a receiver buffer. Thus, the selective-repeat ARQ generally provides the best performance among these three techniques.

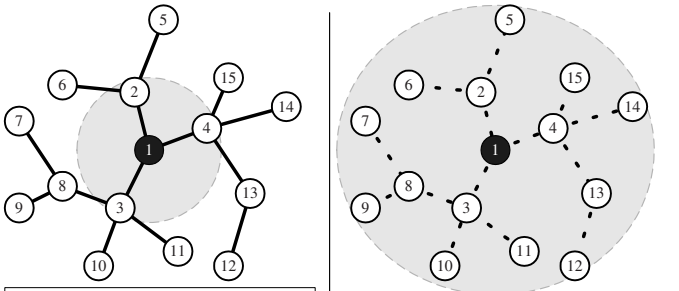
Next, as partly mentioned before, recently several reliable transports were developed for the sensor networks. These include PSFQ [9] and RMST [7]. PSFQ is mainly used as a reliable transport for multicasting the control data such as a new software image from a base node to sensor nodes [10]. Thus, it is not appropriate for the forward direction reliable transport that uni-casts the data from a sensor node to a base node. One the other hand, RMST is used as a forward direction reliable transport. RMST employs the stop-and-wait ARQ in link layer and a selective NACK protocol in transport layer. Thus, it ensures the complete end-to-end reliable data delivery. However, the sensor networks are often interested in reliable detection of the collective information provided by the numerous sensor nodes not in their individual reliable reports. Therefore, the complete end-to-end reliable protocols include RMST are not generally applicable for the forward direction reliable transport in the sensor network regime.

In order to provide a reliable detection of events occupied in the sensor networks, a centralized reliability control technique, namely ESRT [8], was presented. In ESRT, a base node directly controls the reporting frequency of all sensor nodes by using either a multicast protocol or a powerful radio device. However, this centralized transport is not generally applicable to the wireless sensor networks because as it frequently changes the reporting frequency of all sensor nodes, it incurs serious control overheads of the energy consumption and the network congestion. Therefore, we present a decentralized reliable transport for the wireless sensor networks in Section 3.

3 Energy-Efficient Decentralized Reliable Transport

In this section, we briefly compare DRT with the centralized transport of ESRT. Then, we explain the two novel decentralized reliability control schemes and describe the reliable MAC channel that is specifically optimized for DRT.

Figure 1 illustrates the key characteristics of DRT and ESRT. In ESRT, the powerful base node directly controls the reporting frequency of all sensor nodes in a centralized manner. Since the sensor nodes use unreliable channel, the number of successful packet delivery from sensor nodes to the base node is vary depending on the network state, such as number of hops and average packet



(a) Decentralized Transport (DRT) (b) Centralized Transport (ESRT)

Fig. 1. The decentralized reliable transport vs. the centralized reliable transport

error rate. Thus, the base node adaptively changes the reporting frequency, but this generally incurs lot of control overhead in terms of time and network traffic. Moreover, it has a strong design constraint that the base node has to directly deliver its control signal to all sensor nodes if it does not use a multicast protocol.

Fortunately, in DRT, the base node does not have to frequently control the all sensor nodes because each sensor node controls its reporting frequency in a decentralized manner as demonstrated in Figure 1(a). We specifically construct DRT based on three key steps.

First, in querying step, the base node distributes a query packet that embeds the desired partial reliability degree, which is configured by network users, to all sensor nodes. The query packet also includes the target sensor node conditions, the sample types, the reporting period, the number of reporting counts, and the quantization degree. In order to efficiently diffuse the querying packet, we use a reliable multicasting protocol, such as PSFQ, in this step. Since we assume that the routing topology is fixed before this step in this paper, DRT is orthogonal to any routing protocols, whose goal includes the uniform use of sensor nodes.

Second, in reliability control step, each sensor node makes the reporting decision, whether it will reports its sample data or not, so as to ensure the desired reliability with the minimum energy consumption. We use two novel decentralized reliability control schemes as described in Figure 2.

In the independent reporting scheme, each sensor node stochastically makes the reporting decision. Particularly, the sensor node generates a random rate using its system time and its node ID. If the random rate is lower than or equal to the desired reliability, it reports the sample data as shown in Figure 2(a). Otherwise, it does not report.

Contrastively, in the cooperative reporting scheme, each sensor node cooperates with its neighbor nodes so as to uniformly report their sample data. As shown in Figure 2(b), the sensor node compares the desired reliability with the

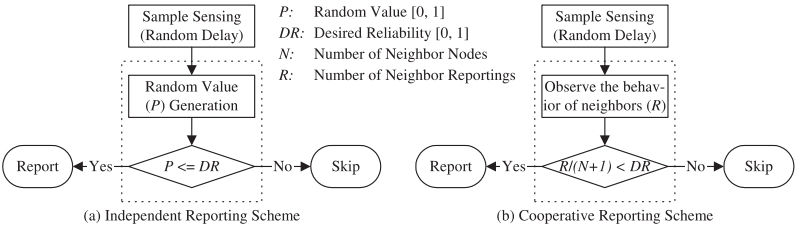


Fig. 2. The decentralized reliability control schemes of DRT

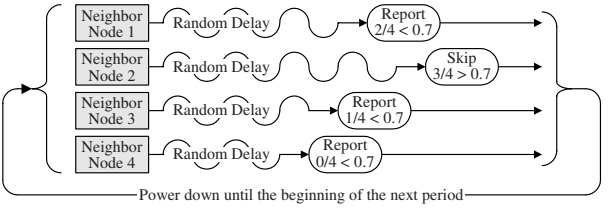
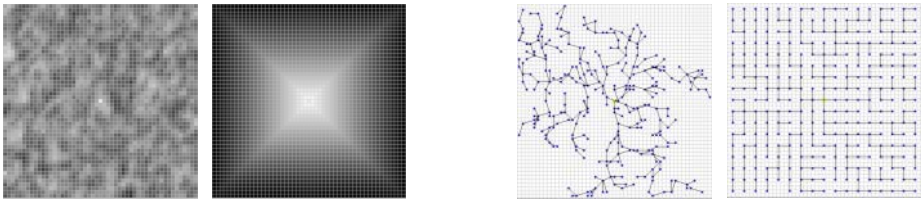


Fig. 3. An example of the cooperative reporting scheme

ratio of the reporting count of its neighbors to the number of its neighbors plus one. It means that when its neighbor nodes locally satisfy the desired reliability, the node does not report its sample data to lessen the energy dissipation. Optionally, the sensor node can omit to report its sample data, if its sample data is quiet similar to the reported sample data of its neighbors.

This cooperative scheme is exemplified in Figure 3 where the desired reliability is 70% and the neighbor node count is three. The sensor nodes can implicitly observe the reporting behavior of their neighbors because it is a wireless network where all packets are broadcasted, and the radio device consumes similar power in both listening and receiving modes. Moreover, since we consider that the neighbor node count is calculated in either routing or querying step, the cooperative scheme does not incurs any extra overhead.

Finally, in reporting step, the sensor nodes use the reliable MAC channel in order to guarantee the desired reliability with these decentralized reliability control schemes. We specifically optimize the reliable MAC channel, which is based on the selective-repeat ARQ for reducing the energy consumption. We present the parameter optimization procedure of the reliable channel such as retransmission timer and the buffer size in Section 4. Furthermore, in order to reduce the overhead of packet header, we propose the packet unification technique that unifies the several sample data packets to one packet. Actually, it is possible because DRT uses the direct diffusion technique [12] that makes the sample data as name and value pairs. We also present the optimization of the packet unification parameter in Section 4.2.



(a) Outdoor (b) Indoor (c) Random (d) Grid

Fig. 4. Medium error condition maps and network routing topologies

In this paper, we presume that each sensor node can turn off its radio devices as quick as it delivers its all data packets by using the partial state reporting scheme of TAG [11].

4 Experimental Methodologies

To study the performance of the various transports, we have developed a byte-level sensor network simulator using Java. The simulator includes various types of sensor nodes that are specifically modeled as finite state machines with software timers and job queues. The simulator core divides the simulation time into units of sending one byte, and it schedules all sensor nodes in every time unit.

The simulator provides two error condition maps as illustrated in Figure 4(a) and 4(b). The maps consist of a fifty-one by fifty-one matrix of small cells, whose extent is $10m^2$ and whose color means the medium error rate. The darker color means the higher error rate. We assume the average bit error rate to be 0.5% and the error range to be between 0% and 1% according to [4]. We calculate the bit error rate between two nodes by using the average value of their error rates. In fact, Figure 4(a) represents an outdoor error condition, and Figure 4(b) represents an indoor error condition, where cells near to the boundary walls have a high error rate due to the interference of the walls.

The simulator also supports four routing topologies as partly shown in Figure 4(c) and 4(d). Figure 4(c) means random topologies, while Figure 4(d) means grid topologies. In these topologies, a base node is placed in the center of each map. We used the four routing topologies in combination with the aforementioned two error maps in every experiment, and then we calculated the mean values as summarized in Section 4.2.

We selected the hardware parameters of a radio device based on Mote [2]. For example, we set the radio bandwidth to be 40Kbps, the radio communication radius to be 30 meters, and the radio power consumption to be 5μA in idle state, 4.5mA in either listen or receive state, and 12mA in transfer state.

We preformed a pre-simulation to choose the appropriate software parameters. As a result, we set a retransmission timer of the reliable MAC protocol to be 30-60ms. The retransmission time is dynamically changed depending on the medium error rate. We also set the retransmission timer gap between consequent

Table 1. Performance summary

Scheme	Spec.	Reliability (%)		Time (ms)	TX (KB)	RX (KB)	Power (μA)
		MIN	AVG				
Unreliable - Retransmission - R: Retransmission count	R=0	2	4	223	5	134	731
	R=1	3	5	293	12	137	854
	R=3	4	7	948	38	177	1,385
	R=5	5	8	3,277	97	277	2,681
	R=7	5	10	22,260	329	665	7,730
	R=9	5	10	93,780	897	1,594	19,985
ARQ - Stop and wait - Proposed packet unification tech. - U: Packet unification parameter	U=1	100	100	12,730	117	1,648	9,823
	U=2	100	100	10,352	120	1,554	9,389
	U=3	100	100	11,340	146	1,778	10,852
	U=4	100	100	15,772	178	2,130	13,052
	U=5	100	100	20,859	224	2,557	15,806
ARQ - Selective repeat - U=2 - B: Buffer size in packets	B=2	100	100	10,760	122	1,562	9,450
	B=4	100	100	8,199	133	1,344	8,504
	B=6	100	100	8,188	140	1,294	8,356
	B=8	100	100	8,984	124	1,375	8,539
	B=10	100	100	8,742	155	1,355	8,859
DRT - Independent Rep. - U=2, B=6 - DR: Desired reliability (%)	DR=100	100	100	8,479	143	1,342	8,638
	DR=90	84	91	7,751	129	1,252	7,985
	DR=70	68	71	6,185	91	1,046	6,453
	DR=50	46	51	5,268	60	828	4,947
	DR=30	24	31	3,408	34	589	3,400
DRT - Cooperative Rep. - U=2, B=6 - DR: Desired reliability (%)	DR=100	100	100	8,201	143	1,354	8,692
	DR=90	99	100	7,457	142	1,300	8,408
	DR=70	78	82	6,932	108	1,138	7,141
	DR=50	61	64	5,240	79	935	5,744
	DR=30	44	50	5,060	61	802	4,825

packets as more than the packet roundtrip time. Because in the simulator all sensor nodes handle the ACK packets as a highest priority job, this configuration removes the useless retransmissions in most cases. We assumed the processing time of a data packet and an ACK packet to be 10ms and 2ms, respectively. We considered the size of packet header and sample message as eight bytes and six bytes, respectively. In every experiment, each sensor node performs one sampling after a random delay of at most 200ms.

5 Performance Evaluations

In this section, we describe the simulation methodology and evaluate the performance of DRT over the existing unreliable and reliable transports. We use the partial reliability, the communication time, and the power consumption of a radio device per each sensor node as performance metrics. The communication time is the elapsed simulation time to deliver all reporting messages to

a base node. We calculated the power consumption per each node by using the aforementioned radio power model.

Table 1 summarizes the simulation results of the existing unreliable and reliable transports. First, the results show that the centralized transport with unreliable channel, i.e. ESRT, accomplishes extremely low reliability of 4% in an average case. Fortunately, this technique works very fast and consumes quite low power. When we use the bulk retransmission technique with unreliable channel, it improves the reliability up to 10% in average cases. However, it at the same time incurs a long finishing time and large power dissipation. Therefore, the centralized transport with unreliable channel is not efficient to provide the high reliability in terms of power consumption and communication time.

We optimized the performance of the reliable MAC channel for improving the delivery speed and reducing the energy dissipation. First, we evaluated the stop-and-wait ARQ with the proposed packet unification technique. The results show that when the unification parameter is two packets, this technique provides the optimal performance. Second, we evaluated the selective-repeat ARQ where the unification parameter is two packets. The simulation results show that the selective-repeat ARQ with the unification technique performs best when the buffer size of both sender and receiver is set to six packets. When the buffer size is larger than six packets, the performance is degraded due to the network congestion and the packet collisions.

In DRT, each sensor node determines the delivery of its sample data. The data packets, which is chose to report, are always delivered to the base node. Contrasted to this, in an unreliable scheme, each sensor node attempts to deliver all sample data. However, some of them are lost in the delivery to the base node because of communication errors. The lost packets uselessly waste the energy of radio devices. This useless energy consumption is avoided in DRT because it uses the reliable channel and makes the reporting decision as early as possible. Therefore, DRT accomplishes the desired reliability with lower energy consumption as compared with the unreliable scheme.

Based on these optimal parameters, we analyzed the performance of DRT with two different decentralized reliability control schemes. The results show that DRT with the independent scheme provides the measured reliability more similar to the desired one than DRT with the cooperative scheme in average cases. One the other side, the cooperative DRT always guarantees the desired reliability. Therefore, these two schemes can be alternatively used in various application domains depending on their objectives.

In the sensor networks, we can induce some semantic information by using a part of sample data. The required ratio of sample data is defined as the partial reliability in this paper. DRT provides an infrastructure where users can easily control the partial reliability. We analyzed the performance gains obtained with this partial reliability control mechanism of DRT. The selective-repeat ARQ is used as a complete reliable transport. The result is that DRT notably reduces both the communication time and the power consumption in comparison with the complete reliable transport. For example, when the desired reliability is 70%,

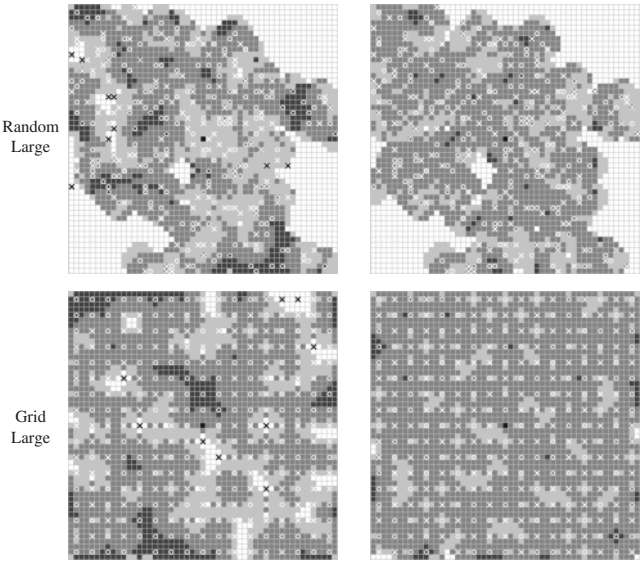


Fig. 5. Reporting density. (white: none, light gray: low, dark gray: fit, black: over)

the independent DRT and the cooperative DRT reduce the communication time by 24% and 15%, respectively, and the power consumption by 23% and 14%, respectively.

In order to enhance the reliability in the sensor networks, the numerous sensor nodes should uniformly report their sample data to the base node. For example, when we monitor the temperature, we generally require at least one temperature sample per every physical room. Figure 5 visualizes the reporting density of DRT with different reliability control schemes. Here, the white cells mean no reporting, the light gray cells mean lower reporting than the desired reliability, the dark gray cells mean appropriate reporting, and the black cells mean excessive reporting. The figure implies that the cooperative DRT distributes the reporting density more uniformly than the independent DRT due to its local cooperation mechanism.

6 Conclusions

We have presented a decentralized reliable transport mechanism for wireless sensor network. With its unreliable radio channels, a wireless sensor network cannot accurately predict the successful packet delivery rate to the base node. Unlike the previous work, which depends on the powerful centralized base node, our approach makes reporting decisions locally. Using custom-optimized reliable MAC channel, sampled sensing data are sent to the base node. Experimental

results show that DRT accurately guarantees the requested reliability while it reduces the power consumption of the radio device and the communication time significantly.

References

[1] I. F. Akyildiz, S. Weilian, *et al.*, "A Survey on Sensor Networks," *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, 2002.
[2] The TinyOS and Mote Project, http://webs.cs.berkeley.edu/tos/.
[3] The Smart-Its Project, http://www.smart-its.org/.
[4] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Network Sensors," *Proc. ACM ASPLOS*, pp. 93-104, 2000.
[5] J. Hill and D. Culler, "A Wireless-Embedded Architecture for System Level Optimization," *UCB Technical Report*, 2002. (http://webs.cs.berkeley.edu/tos/.)
[6] R. Rubin, "Analysis of Wireless Data Communication," *UCB TR*, 2000.
[7] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102-112, 2003.
[8] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," *Proc. of the ACM International Symposium on Mobile Ad-Hoc Networking and Computing*, pp. 177-188, 2003.
[9] C.-Y. Wan, A. T. Campbell, L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," *Proc. IEEE WSNA*, 2002.
[10] P. V. Krishnan, L. Sha, and K. Mechitov, "Reliable Upgrade of Group Communication Software in Sensor Networks," *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 82-92, 2003.
[11] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks," *Proc. USENIX OSDI*, 2002.
[12] J. Heidemann, F. Silva, C. Intanagonwiwat, *et al.*, "Building Efficient Wireless Sensor Networks with Low-Level Naming," *Proc. ACM SOSP*, pp. 146-159, 2001.
[13] G. Fairhurst and L. Wood, "Advice to Link Designers on Link Automatic Repeat reQuest (ARQ)," *Request for Comments (RFC)*, No. 3366, 2002.

A Ubiquitous Streaming Framework for Multimedia Broadcasting Services with QoS Based Mobility Support

In-Soo Park[1], Won-Tae Kim[2], and Yong-Jin Park[1]

[1] Division of Electrical and Computer Engineering, Hanyang University, 17 Haengdang-dong, Seongdong-Gu, Seoul, 133-791, Korea
{ispark,park}@hyuee.hanyang.ac.kr
[2] Rostic Technologies, Inc., #B207, H.I.T. Bld., Hanyang University, 17 Haengdang-dong, Seongdong-Gu, Seoul, 133-791, Korea
wtkim@rostic.com

Abstract. A ubiquitous streaming framework architecture is proposed to support mobile multimedia broadcasting services with QoS-guaranteeing. U-Stream framework consists of a system part and a network part. The system part includes a new mobile streaming format, a media encoder, a media server and a client system. The network part has a mobile multicast mechanism and mobile QoS supporting technologies. A testbed including WLAN and CDMA2000 1X EV-DO is built in order to perform effective experiments. Finally, the experimental results of handoff delay time and packet loss are measured and analyzed.

1 Introduction

Ubiquitous computing comes to be accepted as one of the most well-known terms in world wide. Users can communicate and compute with each other whenever, wherever and whatever. The evolution of mobile technologies initiates the ubiquitous computing era and mobile computing devices come to be enough small to be integrated or embedded in our clothes and even in glasses. Various sensors will be scattered around us and as a result invisible computing or pervasive computing will come true. It is strongly believed that there will be still IP (Internet Protocol) in the eye of that digital storm. Today the mobile technologies have made so dramatic evolution from the first generation mobile network to third generation mobile networks such as UMTS (Universal Mobile Telecommunications System) or IMT-2000 (International Mobile Telecommunications-2000). The needs for mobile Internet services drive the radio technologies and network technologies to support broadband multimedia services. Additionally mobile user terminals are evolving to take various designs and functionalities integrating the legacy digital devices such as a mobile game station, a MP3 player and a digital camera/camcoder, etc.

In this paper, a multimedia multicast framework for a preliminary ubiquitous computing environment, named U-Stream (Ubiquitous- Streaming), is suggested.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 65–74, 2004.

The related works are reviewed on the next section. The U-Stream architecture design and implementation will be described in section 3. Finally the testbed and the results of experiments will be given in section 4.

2 Related Works

2.1 IP Multicasting Technologies

There are various approaches to support IP multicasting on each protocol layer. In this subsection, multicast mechanisms on network layer are reviewed respectively. For the purpose intended by this paper, IP layer multicast routing will be the format for our discussion. For intra-domain multicast services, several multicast routing protocols have been developed which can be divided into two categories: some making shortest path trees (SPT) and others making shared trees [1]. The former makes a SPT expanding to all receivers from each sender. In other words, it has to make routing table entries for all senders in each on- tree router. Furthermore, SPT type multicast routing protocols require too many network resources and control information in order to establish and continuously maintain the multicast tree states. There are distance vector multicast routing protocol (DVMRP), multicast OSPF (MOSPF) and protocol independent multicast- dense mode (PIM-DM) in SPT types [2, 3, 4]. The shared trees type of multicast routing protocols such as CBT (Core Based Tree) and protocol independent multicast-sparse mode (PIM-SM) on the other hand, establish shared trees connecting all receivers to the core routers or rendezvous points [5, 6].

2.2 Mobile RSVP

There have been many proposals to support QoS (Quality of Service) in Mobile IP networks. The legacy mobile RSVP (ReSource reservation Protocol) proposals are summarized in Table 1.

3 U-Stream Framework Architecture

3.1 U-Stream Architecture Overview

On the stage of designing U-Stream architecture, the consideration of a system part and a network part should be made respectively. The first requirement of the system part is the capability of multi-channel streaming over heterogeneous networks and devices. Since there are various network capacity and device types, U-Stream system can send single source data to different networks and devices at the same time. Secondly, the system must support international standard protocols in order to have interoperability with the existing multimedia systems such as RTP (Realtime Transport Protocol), RTSP (Real Time Streaming Protocol) and SIP (Session Initiation Protocol). Thirdly, U-Stream media encoder should have a powerful transcoding function to reuse stored contents. For example,

Table 1. The comparison of legacy mobile RSVP proposals

Name	Features
Talukdar [7]	A RSVP proxy on a visiting cell performs active reservation. Other RSVP proxies around neighbor cells do passive reservation process and switch to active mode.
Mahadevan [8]	Much similar to Talukdar [**?**], there are active/passive reservation modes. It is assumed that not a MN (Mobile Node) but a BS (Base Station) knows neighbor BSs. This reduces load of a MN.
Terzis [9]	It is the most straightforward mechanism integrating RSVP and MIPv4 protocols. If a MN handoffs, a new tunnel is established between HA (Home Agent) and new FA (Foreign Agent) by RSVP signaling on the new tunnel.
Chen [10]	Conventional reservation is performed by the RSVP proxy that a MN visits, and predictive reservation is done by around RSVP proxies. The proxies join a multicast tree rooted on a sender and perform join and leave the multicast tree with movements of the corresponding MN. RSVP messages and IP packets are delivered to the MN through the multicast tree.
Shen [11]	Routers on the overlapped path between a new path and the old path are excluded on handoff updating process after handoffs is finished. Only new routers are under the process of path updating.

it can convert AVI media compressed by MPEG-4 to RTP stream with H.264 codec. At least the transcoding operation should support conversion functions of streaming format, compression codec, AV resolution and frame rate. Finally, it may be optional to support file system plug-in by which U-Stream systems can be easily installed on different file systems or OS (Operating System) platforms.

The key issues of U-Stream network part are the mechanisms of mobile QoS and mobile multicast or integration of them. The most straightforward mechanism for mobile multicast uses IETF Mobile IP operation. The first mechanism is to make a HA a proxy agent for multicast receivers which are mobile nodes registered to the HA, that is to say, the HA forwards multicast packets to each mobile node by means of a tunneling mechanism. In this mechanism network congestion on gateway may be made by replicated enormous traffic from a home agent as well. Moreover the propagation delay caused by Mobile IP triangular transmission should not be ignored because mobile service roaming should be considered in national wide or continental wide. The second choice is to make a FA a DR (Designated Router) which performs as an edge multicast router managing multicast members in its subnet. Although it does not make a triangular problem or an avalanche trouble on networks, it is so difficult to deploy DRs on every ubiquitous network. The deployment problem is the most serious reason why multicast networks can not be an infrastructure of the current In-

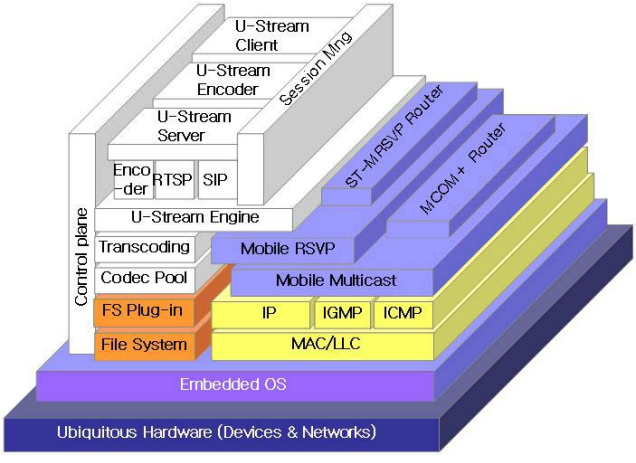


Fig. 1. U-Stream framework architecture

ternet. In this paper the first choice of mobile multicast approaches is adopted. The second choice is described very well in the paper titled MCOM [12].

Secondly, the interoperation between Mobile IP agent and multicast management module in applications is not defined. It may be a part of developing issue. It, however, must be solved so as to support seamless mobile multicast services on mobile terminals and network nodes as our mobile experience. Thirdly, it is one of critical problems that a mobile terminal maintains a logical communication pipe for guaranteeing QoS during a multimedia session. The solution may be summarized as Mobile RSVP in this paper. Although many excellent researchers have suggested some solutions as shown in table 1, an effective interworking solution with multicast should be suggested. As the results of the above considerations, the entire architecture of U-Stream is given in Fig. 1.

3.2 U-Stream Engine

Mobile Streaming Format RSF (Real-time Streaming Format) was invented by the authors as a new streaming format to especially support mobile environments. One of RSF design concept is open architecture to support any kind of multimedia codecs. Although RSF basically encodes media compressed by MPEG-4 and MP3, RSF does not depend on any specific codecs. Of course, the reason to introduce RSF here never means that U-Stream systems should support only RSF media. U-Stream is designed to be able to support any kinds of streaming formats including RSF as mentioned above. RSF is optimized to send mobile multimedia stream in UDP datagram not in RTP because RSF already has the features of RTP encoding. RSF reduces overhead of a multimedia streaming format in order to minimize packet size. RSF has very simple structure compared with ASF of Microsoft.

Table 2. The comparison between ASF and RSF

Functionality	ASF	RSF	Unit
chunk structure	Yes	Yes	
chunk header size	16	2	Byte
chunk data field size	8	2, 4	Byte
Main header size	Variable (¡Kbyte)	44	Byte
Header repetition	No	Yes	
Text data support	No	Yes	
AV codec information	Yes	Yes	
Join on going session	No	Yes	
AV packet header size	¡24	8 12	
AV synchronization Info.	Yes	Yes	Byte
AV sync. Mechanism	Time stamp	Ref. audio	

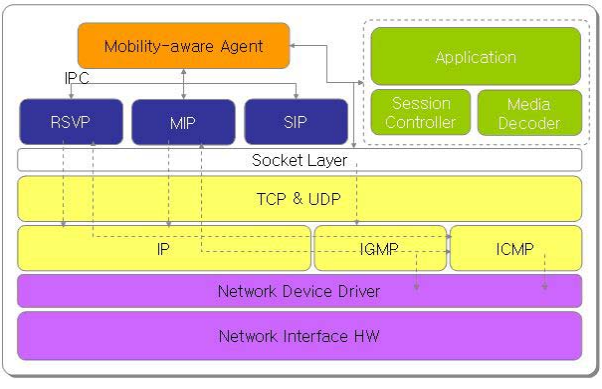


Fig. 2. Structure of a U-Stream client system

It makes synchronization between an audio stream and its corresponding video stream based on audio timeslot. The legacy streaming formats uses timestamp on each media stream because they are originally developed for the purpose of playing on high speed networks or local systems. The timestamp information of each media makes so much overhead and wastes network bandwidth as a result. An advantage of RSF is to allow a late user to join a broadcasting session at any time.

U-Stream System Part The structure of a U-Stream client is suggested in Fig. 2. The clients are designed and developed for Microsoft windows including MS Pocket PC and Linux because the platforms are the most well-known OSs. Since the structure design is so simple and light-weight, it can be adapted to any kind of embedded operating systems such as PALM or symbian OS. Application modules perform user interfacing, multicast session control and multimedia

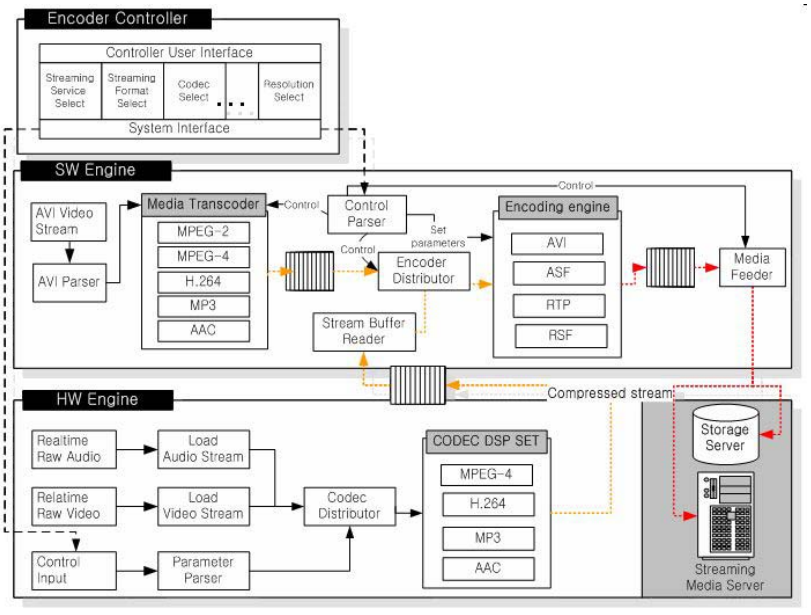


Fig. 3. The structure of a U-Stream encoding system

stream rendering with inter-media synchronization. It is too difficult to sophisticatedly control inter-media synchronization in Linux because Linux does not support exact time information to applications and, moreover, clock period of Linux varies around maximum 20 msec as our experiments. If the time error accumulates, the inter-media synchronization can be never achieved. Time correction algorithm for the missed accuracy is devised by means of audio rendering delay mechanism. Media decoder decodes incoming streams which is pumped up to the application module. Session controller does functions of starting, maintaining and eliminating a session. Signaling modules include RSVP, Mobile IP and SIP daemons. The daemons keep the standard specifications. Communication among them is controlled and coordinated by mobility-aware agent module (MAM) which connects the application module as well. MAM notifies the application module of terminal handoff and makes it rejoin multicast session. Broken arrows mean the control directions and target protocol modules of each component.

Another essential component of ubiquitous multimedia services is a powerful encoding system as mentioned Sect. 3.1. The logical system design is given in Fig. 3. In actual implementation, each part of the U-Stream encoding system is separately built on multiple systems. Since it is an implementation problem, the mechanism will be no more described in this paper. In a SW engine, an AVI stream from a HW engine or a storage system is injected into AVI parser which pushes the stream into a media transcoder to convert input stream to other type of the AV data by using selected AV codecs. After AV compression, the stream

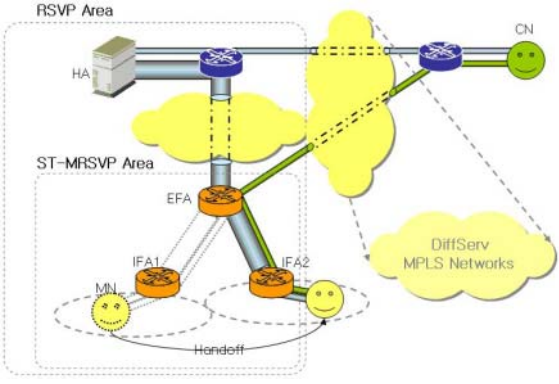


Fig. 4. Logical Mechanism of ST-MRSVP

is encoded into a selected stream format. The sequential process is controlled by an encoder controller system which receives user commands and manages the entire encoding system. Users can select multiple options to simultaneously make different types of AV streams originated from one source. The output streams are transmitted to U- Stream media servers which make direct connections with multiple clients. U- Stream media servers are not described further in this paper. As a result, any kinds of client systems can be served by the U-Stream encoder.

3.3 Split Tunnel Based MRSVP

The mobile Internet should support enough bandwidth and packet loss ratio to reliably transport multimedia streams. Our previous work on mobile RSVP, named Split Tunnel based MRSVP (ST-MRSVP), has been introduced [13]. A tunnel between a HA and a FA is divided into two split tunnels with HA-EFA (External Foreign Agent) and EFA-IFA (Internal Foreign Agent) as shown in Fig. 4. In order to enhance handoff delay time, handoff process is localized instead of reserving tunnel resources in end-to-end as standard RSVP does.

An EFA is a FA located on a gateway which is a kind of a border router between an intranet and a core network. An IFA has direct connections with MNs. An EFA receives registration requests relayed from an IFA which manages MNs, and relays the requests to the MNś HA. While the EFA behaves as a HA of the IFA, the EFA pretends to be MNs FA for the MNś HA. Of course, the IFA works as a normal foreign agent for the MN. A MN finds a FA in the same way as standard Mobile IP does and sends registration requests to the FA, that is, the IFA. After the IFA changes the Care-of Address (CoA) in the registration request message to an EFAś address, it forwards the modified message to a designated EFA. During this process a dynamic tunnel interface is established and the EFA sends the message toward MNś HA. If address binding about the MN is finished, the HA start relaying packets for the MN through an established tunnel between the HA and the EFA. When EFA receives the tunneled packets, it transforms

the current tunnel header to a new tunnel header with the destination address toward the IFA which decapsulates the tunneled packets and delivers them to the MN. When a MN handoffs, it registers a new CoA assigned by a new IFA through the IFA. The new IFA forwards the registration request message to the EFA. Since it is assumed that the EFA keeps the information about the MN, the EFA does not need to forward the message to the HA. It is very straightforward mechanism that the EFA changes the old tunnel to a new tunnel. The EFA just updates the MNś entry of a binding table maintained in the EFA from the address of the previous IFA to the address of the current IFA. RSVP PATH messages travel through the established tunnel via the HA. RSVP reservation messages are sent back along with the tunnel path. On tunnel sections such as HA-EFA and EFA-IFA, each MRSVP router reserves tunnel resources by standard RSVP operation. Protocol modification is mainly done in Mobile IP, but RSVP hardly needs to be changed.

4 Experiments and the Analysis

The U-Stream testbed is built as shown in Fig. 5. 7 PC routers including mobility agents have Intel Petium3 800MHz CPU, 128Mbyte RAM and Linux kernel version 2.4.17 respectively. 5 client systems have Intel Petium3 800MHz CPU, 512Mbyte RAM and MS window2000/Linux kernel version 2.4.17. 3 clients with Linux are based on Mobile IP and 2 clients with MS windows use simple IP. Mobile IP clients uses ST-MRSVP networks and simple IP clients receive data over the legacy multicast networks, that is, DVMRP networks. Wireless LAN APs (Access Points) are made by Orinoco IEEE 802.11b. The resource reservation status can be monitored in debugging mode of RSVP router and traffic flow over the testbed can be measured by Ethereal, a well-known traffic monitoring tool. The U-Stream server is set up to transmit the saved RSF source file and its transmission rate is about 512byte/30msec. It tells that the bandwidth of the media is tuned about 140kbps because current CDMA2000 1X EV-DO can guarantee minimum bandwidth of 144Kbps. Media is fixed to transmit as fast as possible since the bandwidth is enough to handle in CDMA2000 1X EV-DO networks. Buffering of the received multicast packets makes the transmission rate higher than rendering speed. RSF stream is transmitted seamlessly for single channels, and the packet loss is hardly occurred in WLAN networks. Some packets, however, are lost in cellular networks. The packet loss ratio is about 1% in well tuned CDMA networks. Although the lost packets make somewhat noise on a screen, the noisy screen, however, does not an obstacle to our visual recognition system. The handoff delay and packet loss are shown in Fig. 6.

The handoff test with ST-MRSVP is performed between a home network and two foreign networks as shown in figure 5. The experiment is about an impact from dynamic handoff. Handoff delay time takes about 1.0 seconds to register its current location because there is no handoff notification from hardware driver. If hardware drivers can inform the instance of handoff to the Mobile IP module, the registration time may be reduced dramatically below 0.5 sec. Since resource

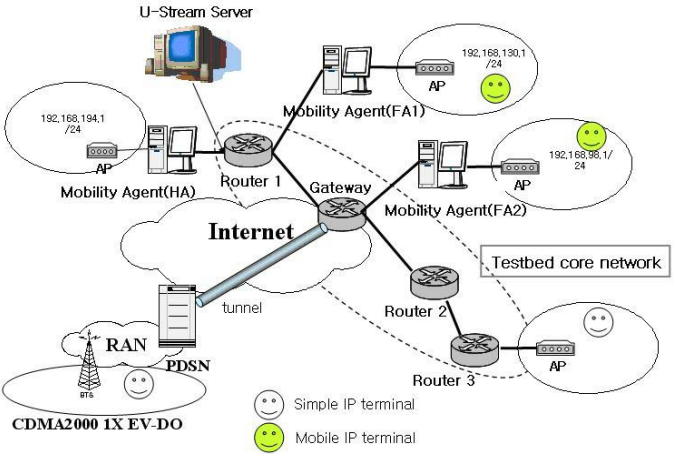


Fig. 5. The configuration of U-Stream testbed

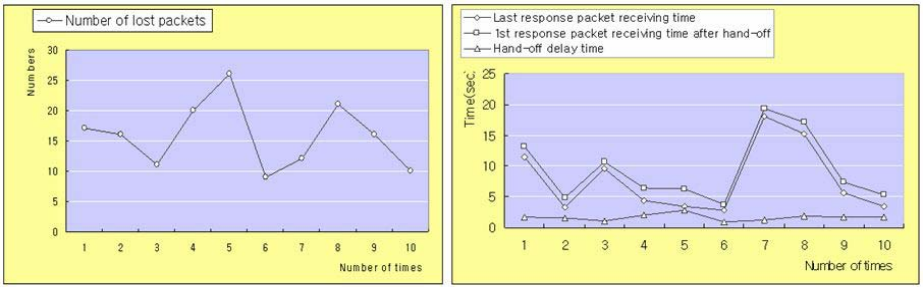


Fig. 6. The results of experiments (a)the number of lost packets (b)handoff delay time

reservation time is added on the handoff time, the experiment results show that the average time to receive first packets from U-Stream media server is about 1.5 seconds.

5 Conclusion

U-Stream is a ubiquitous streaming framework architecture supporting mobile multimedia broadcasting services with QoS-guaranteeing. Overall architecture is suggested and the functionalities of each component are described in detail. An advanced mobile streaming format and U-Stream client/server/encoder systems are introduced as well. On network part, RT-MRSVP is reviewed in relation with mobile multicasting services. Experiments of streaming multimedia multicast were performed over U-Stream testbed. As the results, it was proved that the level of handoff delay time and packet loss by U-Stream framework is acceptable for the human visual systems.

References

[1] D. Kosiur : IP Multicasting: The complete guide to Interactive Corporate Networks. Wiley computer publishing(1998). 66
[2] T. Pusateri : Distance Vector Multicast Routing Protocol. Internet RFC 1075 (1997). 66
[3] J. Moy : Multicast Routing Extensions for OSPF. Communications of the ACM (1994). 66
[4] S. Deering : Protocol Independent Multicast Version 2, Dense Mode Specification. Work in Progress (Internet draft) (1997). 66
[5] A. Ballardie : Core Based Trees (CBT) Multicast Routing Architecture. IETF RFC2201 (1997). 66
[6] S. Deering and D. L. Estrin : The PIM Architecture for Wide-Area Multicast Routing. IEEE/ACM Transactions on Networking (1996). 66
[7] A. K.Talukdar, B. R.Badrinath, and A.Acharya : MRSVP: A Reservation Protocol for an Integrated Services packet Network with Mobile Hosts. Tech. Rep. Dcs-tr-337, Dept. of CS, Rutgers Univ. (1997). 67
[8] I.Mahadevan and K. M.Sivalingam : An Experimental Architecture for providing QoS guarantees in Mobile Networks using RSVP. IEEE PIMRC, Boston (1998). 67
[9] A.Terzis and M.Srivastava and L.Zhang : A Simple QoS Signaling Protocol for Mobile Hosts in the Integrated Services Internet. INFOCOM 1999. 67
[10] W.Chen and L.Huang : RSVP Mobility Support: A Signaling Protocol for Integrated Services internet with Mobile Hosts. INFOCOM 2000. 67
[11] Qi.Shen and W.Seah and A.Lo : Flow Transparent Mobility and QoS Support for IPv6-based Wireless Real-time Services. IETF Internet-Draft (2001). 67
[12] Won-Tae Kim and Yong-Jin Park : Scalable QoS-based IP multicast over label-switching wireless ATM networks. IEEE Network Magazine, Sept./Oct. 2000. 68
[13] In-Soo park, Yong-Jin Park and Haeng-Bong Kang : The Mobile RSVP to support the local mobility in the wireless LAN. ICOIN-14 (2000). 71

A Reflective Approach to Dynamic Adaptation in Ubiquitous Computing Environment

Soo-Joong Ghim, Yong-Ik Yoon, and Jong-Won Choe

Department of Computer Science, Sookmyung Women's University
Chungpa-Dong 2-Ga, Yongsan-Gu, 140-742, Seoul, Korea
{sjghim,yiyoon,choejn}@sookmyung.ac.kr

Abstract. To provide users with persistent services in distributed ubiquitous environments, it is required for applications and middleware to be aware of the frequent and unpredictable changes in users requirements as well as environmental conditions, also to be able to adapt their behaviour as such changes. One of the main limitations of current approaches for supporting adaptability is that applications themeselves are responsible for triggering and adaptive mechanism when the underling infrastructure notifies them about any changes. Hence, we design an adaptive middleware framework using reflection and propose the meta-meta-level to support a policy-based adaptation. We implement mobile agents (adaptation, context and meta agents) to adapt user-level and application-level changes dynamically for mobile users and applications.

1 Introduction

Ubiquitous Computing advocates the construction of massively distributed systems that help transform physical spaces into computationally active and intelligent environments [1]. Within such environment, not only can devices or software services be added to or removed from the system at anytime, but also are contexts or preferences of users changing seamlessly.

Adaptability is one of the most important requirements for ubiquitous computing systems, since such environments are highly dynamic, characterized by frequent and unpredictable changes in different contexts. Hence, applications need to be capable of adapting their behavior to ensure they continue to offer the best possible level of service to the user. Adaptation should be driven by awareness of a wide range of issues including communication performance, resource usage, location, cost, and application preference [2].

The current approach to providing adaptable services or applications is based upon the classic layered architectural model where adaptation is provided at the various layers (data link, network, transport or application layers) in isolation [3]. One of the main limitations of current approaches is that applications themeselves are responsible for triggering and adaptive mechanism when the underling infrastructure notifies them about any changes [4].

Hence, we need a sophisticated approach, which can manipulate mixed or customized adaptation in contextual changes. It is more desirable and effective

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 75–82, 2004.

to manage adaptation at the middleware for providing different adaptive solutions in various situations. For ubiquitous computing environments, middleware architecture itself should be context-aware to manage the communication among objects in a transparent fashion.

In this paper, we propose the design of middleware framework in reflective architecture, which can support adaptability for mobile and context-aware applications. Section 2 depicts related research and section 3 describes our reflective approach to separate concerns on adaptation using mobile agents and gives an overview of the design of reflective middleware framework. Section 4 describes how adaptations are triggered by a policy and represents implementations on agents. Section 5 presents our conclusions and future works.

2 Related Work

Odyssey Odyssey [5] supports a type of adaptation called *application-aware adaptation*. Odyssey's approach to adaptation is to adjust the quality of accessed data to match available resources. The agility of Odyssey is determined by the adaptive decision loop; select fidelity, place request, detect change, and notify application. The first step of this process, fidelity selection, is the province of the application and server. However, Odyssey's notification approach can be shown to lead to inefficient solutions because it is lack of support for enabling coordination between the adaptation policies and enable to increase the burden of the application developer.

K-Components K-Components [6] uses asynchronous architectural reflection to build context-adaptive applications. Adaptation logic specifying adaptive behaviour, encapsulated in the Adaptation Contract Description Language (ACDL), can be written by programmers to build self-adaptive systems, but can also be modified and updated at runtime by users, allowing them overall control of the application's adaptive behaviour. Adaptation occurs in response to adaptation events raised by the application components or from the evaluation of adaptation rules themselves. The main issue with K-Component in relation to this system is its inability to accept new types in the configuration graph since the configuration graph is a static representation of the architecture of the system.

Chisel Chisel [7] project is to investigate the use of reflective techniques as a vehicle for the development of a framework for dynamic adaptation, using middleware as a case study. The approach will be to allow different application-aware, user-aware, and context-aware policies to control the dynamic adaptation of component behaviours defined as new Iguana metatypes. A policy-based approach was chosen to drive the adaptation mechanism by incorporating user and application specific semantic knowledge and intelligence, combined with low-level monitoring of execution environment.

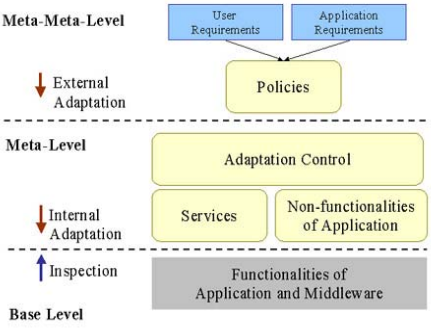


Fig. 1. Conceptual model for adaptation

3 An Architecture for Reflective Middleware

This section describes a reflective approach to separate concerns on adaptation and design an adaptive middleware framework. Also, we present our middleware architecture and mobile agents.

3.1 Reflective Approach

For the dynamic management and adaptation of changes, we use reflection to separate aspects of adaptation at the application level and the middleware system level. Reflection refers to the capability of a system to reason about and act upon itself [8]. The base level of reflective middleware addresses the application program's functionality, while the meta-level designates collections of components that form the internal architecture of the middleware platform [9]. Reflective middleware techniques enable autonomous changes in application behaviour by adapting core software and hardware mechanisms dynamically without the need for explicit control by applications or end-users [10]. By separating aspects, applications and middleware service components will be dynamically adapted at the meta-level, in contextual changes of users and applications. We propose the meta-meta-level for decoupling policies from adaptation control and supporting the policy-based adaptation at the meta-level.

Our aim is to build a middleware framework supporting user-specific and application-specific adaptation in context-aware fashion, hence, providing users with right services persistently in ubiquitous environments. Fig.1 represents our conceptual model for adaptation. External adaptation is required for making different adaptive decisions by collaborating user-specific and application-specific policies, while internal adaptation is required for monitoring changes of the system context, dynamic reconfiguration and transparent adaptation. At current stage of our work, we focus on external adaptation, including a policy-based mechanism.

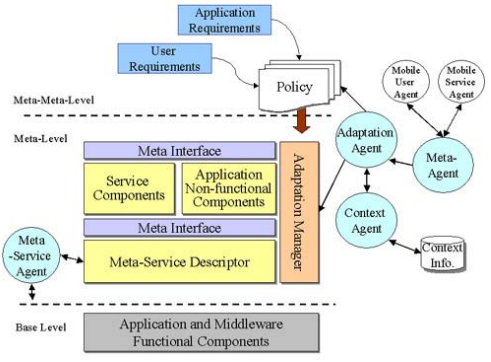


Fig. 2. Architecture of reflective and adaptive middleware framework

3.2 Design of Reflective and Adaptive Middleware Framework

We give an overview the design of reflective middleware framework that can support the meta-level adaptation by a policy-based manner and we describe our components in a meta-architecture (see Fig.2).

Service Component Service components involve non-functionalities of middleware system and mobile agents. These components implement not only middleware services but also monitoring, analysis and communication methods for mobile agents. A set of service components is reconfigurable and extensible by adding new service components in plug-in manner. From a remote host, mobile agents can download service components and application's non-functional components, to deploy new services or upgrade existing services.

Meta-Service Descriptor The meta-service descriptor can provide abstraction of the base level components. At run-time, meta-service agents will perform inspection, and then configure meta-service descriptor.

Adaptation Manager The adaptation manager will be responsible for the management and adaptation of the application's non-functional components and service components. It can interpret the adaptation policy, which is a human-readable document, and associate the selected decision with the real behaviour. When adaptation is requested, that is an adaptation event occurs, this adaptation manager will perform adaptation operation triggered by changes in the policy and context. Users and applications can trigger adaptation explicitly using policy documents.

3.3 Mobile Agents for Adaptation and Mobility

We propose meta-agent, adaptation agent, context agent, and meta-service agent to collaborate for dynamic adaptation at the meta-level.

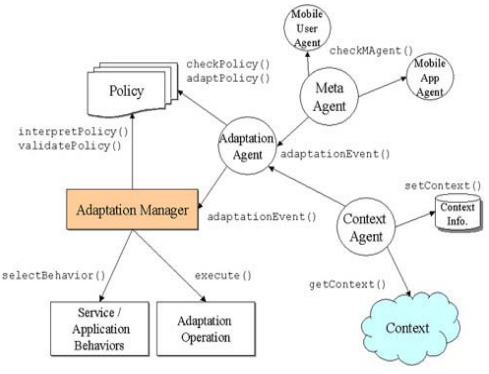


Fig. 3. Adaptation using agents

Meta-Agent At the meta-level, meta-agents can be considered as a different kind of mobile agents that run on their own execution environment. Main purpose of meta-agent is to manage and control of adaptation processing. Thus meta-agents are responsible for monitoring the execution of mobile agents and for transferring them to remote host. The association between meta-agents and mobile agents may be performed on the basis of one-to-one or one-to-many. In the former case, one meta-agent can be created per mobile agent, therefore it allows a meta-agent to migrate itself containing one mobile agent and to keep track of the behaviour of that agent on every node it visits. In the latter case, one meta-agent can manage a group of mobile agents. According to meta-agent's own policy for management, the number of mobile agents that one meta-agent can encompass is regulated.

Adaptation Agent and Context Agent An adaptation agent will be responsible for monitoring changes in the policy and coordinating system responses to changes in the environment and acquiring a certain type of meta-information available to applications. It can analyze types of events caused y other agents and also throw adaptation events to adaptation manager. A context agent is a mobile agent, which is responsible for gathering and analyzing context information. It can also monitor contextual changes.

Meta-Service Agent A meta-service agent is a stationary agent, which is responsible for monitoring configuration changes of base level components and supporting reconfiguration. Another important task of this agent is inspection on the base level to configure meta-service descriptor.

4 Policy-Based Adaptation

This section describes behaviours of agents on adaptation and how adaptations are triggered by a policy. Also, we represent implementations on agents.

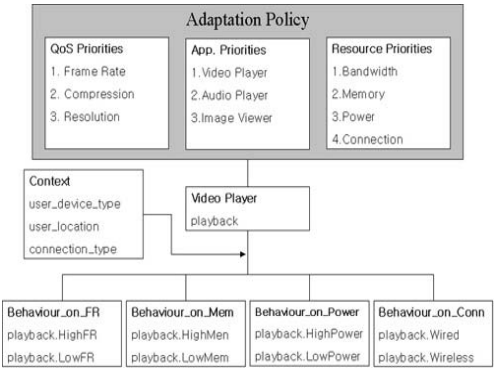


Fig. 4. Adaptation policy and application's behaviours

4.1 Adaptation Control

The meta-agent establishes point-to-point channel by remote communication with the destination meta-agent, which will receive that mobile agent and deserialize it. When a mobile agent arrives at its destination, a meta-agent examine configuration and make it restart its execution. The meta-agent continues monitoring the execution of mobile agents and throws an adaptation event when any change occurs by mobile agents.

Adaptation agent tracks changes in the policy. On the adaptation event, an adaptation agent analyses the subtype of that event and also throw an adaptation event. Adaptation manager interprets and validates policy rules, and triggers adaptation by selection of appropriate service bahaviour (see Fig.3).

The adaptation policies can be specified by the user-specific and application-specific priorities on user's preference to applications and quality of service, and application's resource requirements (see Fig.4). When any of the contextual changes occurs, the context agent perceives that change and then the adaptation manager has to decide which adaptation method should be invoked. In the example, application's non-functional behaviours are separated into several groups according to the application's resource priorities. In order to decide which bahaviour to invoke, the adaptation agent checks the prioritization of the applications.

4.2 Implementation of Agents

The meta-agent is responsible for monitoring execution of mobile agents, while the adaptation agent and the context agent are responsible for monitoring changes; the former monitors changes of policy and the latter monitors contextual changes. For the dynamic adaptation, we present MetaAgent, AdaptationAgent, and ContextAgent classes to implement our policy-based mechanism, as follows:

Meta-Agent Class

```
class MetaAgent extends Agent implements AgentInterface{

    ...
    public void initAgent(Agent agent) //initializes the agent
    public void checkMAgent(){...} //monitors mobile agents
    public void migrationEvent(){...} //acts on migration event by mobile agents
    public void arrivalEvent(){...} //acts on arrival event by mobile agents
    public void adaptationEvent() {...} //acts on adaptation event
    }
```

Adaptation Agent Class

```
class AdaptationAgent extends Agent implements AgentInterface {

    ...
    public void initAgent(Agent agent) {...}
    public void adaptContext(Context context) {...} //adapts context
    public void checkPolicy(Policy policy) {...} //monitors adaptation policy
    public void adaptPolicy(Policy policy) {...} //adapts adaptation policy
    public void adaptationEvent() {...} //acts on adaptation event
    }
```

Context Agent Class

```
class ContextAgent extends Agent implements AgentInterface {

    ...
    public void initAgent(Agent agent) {...}
    public void getContext() {...}  //gets current context
    public void setContext(Context context) {...} //sets changed context
    }
```

5 Conclusions and Future Work

In this paper, we described the design of reflective middleware framework that can support dynamic adaptation for mobile users and applications in ubiquitous environments, and we introduced the meta-meta-level for supporting a policy-based adaptation at the meta-level.

Fundamental to our reflective approach is the idea of separating concerns on adaptation and making different decisions on adaptation policies. The adaptation policies can be specified by the user-specific and application-specific priorities on user's preference to applications and quality of service, and application's resource requirements. Our model can provide meta-level adaptation method using mobile agents. For this, we implement adaptation, context and meta-agents to adapt user-level and application-level changes dynamically.

The implementation of our adaptive middleware framework is currently ongoing, focusing on supporting mobile applications. In the future work, we intend to develop adaptive middleware services and management mechanism for context information.

References

[1] A. Ranganathan and R. H. Campbell.: A Middleware for Context-Aware Agents in Ubiquitous Computing Environments, In ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, Brazil (June 16-20, 2003)
[2] G. S. Blair, G. Coulson, A. Anderson, et al.: A Principles Approach to Supporting Adaptation in Distributed Mobile Emvironments. Proceedings of the 5th International Symposium on Software Engineering for Parallel and Distributed Systems (PDSE'2000), Nixon P. & Ritchie I. (eds), Limerick, Ireland (June 10-11, 2000)
[3] Z. J. Haas: Designing Methodologies for Adaptive and Multimedia Networks, IEEE Communications Magazine, Vol. 39, N.11, (November 2001) 106-107
[4] C. Efstratiou, K. Cheverst, N Davices and A. Friday: An Architecture for the Effective Support of Adaptive Context-Aware Applications, Proceedings of the Second International Conference on Mobile Data Management (MDM '2001) (January 8 - 10, 2001) 15-26
[5] B. Noble: System Support for Mobile, Adaptive Applications, IEEE Personal Communications, Vol. 7, No. 1 (February 2000)
[6] J. Dowling and V. Cahill: The K-Component Architecture Meta-Model for Self-Adaptive Software, Proceedings of Reflection 2001, LNCS 2192 (2001)
[7] J. Kenney and V. Cahill: Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework, In Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy (June 4-6, 2003) 3 – 14
[8] G. S. Blair, G. Coulson, et al.: What is Reflective Middleware?, IEEE Distributed Systems Online Journal 2(6) (2001)
[9] M. Román, F. Kon, and R. H. Campbell: Reflective Middleware: From Your Desk to Your Hand, IEEE Distributed Systems Online Journal, Special Issue on Reflective Middleware (2001)
[10] N. Wang, M. Kircher, and D. C. Schmidt: Applying Reflective Middleware Techniques to Optimize a QoS-enabled CORBA Component Model Implementation. COMPSAC 2000, Taipei, Taiwan (October 2000)

Personal Service on Application Level Active Network for Ubiquitous Computing Environments*

Sungjune Hong[1], Sunyoung Han[2], Keecheon Kim[2], Jinpyo Hong[3], and Kwanho Song[4]

[1] Department of Information and Communication, Yeojoo Institute of Technology
454-5 Yeojoo-goon, Kyungki-do 469-800, Korea
sjhong@mail.yeojoo.ac.kr
[2] Department of Computer Science and Engineering
Konkuk University 1 Hwayangdong, Kwangin-gu, Seoul, 143-701, Korea
{kckim,syhan}@kkucc.konkuk.ac.kr
[3] Department of Information and Communication
Hankuk University of Foreign Studies 89
Wangsan-ri, Mohyun-myon, Yongin-si, Kyongki-do 449-791, Korea
jphong@hufs.ac.kr
[4] Korea Network Information Center
khsong@nic.or.kr

Abstract. This paper describes the Self-adaptive Personal Service (SPS) on the extended Application Level Active Network (ALAN) for ubiquitous computing environments. It is expected that a customized service personalization in a ubiquitous computing environment can be deployed. However, the existing service personalization does not support location information, Quality of Service (QoS) policy, device type, etc., since a user's preference depends on the origin of the web servers. To address this issue, the Internet Engineering Task Force (IETF) standard called Framework for Service Personalization (FSP) on Open Pluggable Edge network Service(OPES) is underway. Nevertheless, FSP on OPES does not accommodate the ubiquitous computing environments because OPES does not consider self-adaptation. As a result, existing networks have the inability to support additional services. Therefore, this paper suggests the use of Generic Modeling Environment (GME) tool as Service Creation Environment (SCE). By using a GME tool, the SPS on the extended ALAN can support self-adaptation with functions such as a user's changing constraints, a decision-making, and a service composition.

1 Introduction

The interest in ubiquitous computing environments has recently increased. The objective of ubiquitous computing environments is to provide users with seamless, ubiquitous access to services, irrespective of users' end device or location. It

* This research was supported by University Research Center Project.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 83–92, 2004.

is expected that the customized service personalization in ubiquitous computing environments can be deployed. Although the existing service personalization includes features such as virus scanning, content translation, packet-filtering, and content adaptation, the major drawback of the current service personalization is its dependency to content origin of the web server to perform personalizing tasks. This is because the personalization task is performed using incomplete information about the user. The content provider may not be aware of the various types of information about the user, including geographic location, Quality of Service (QoS) policy, device type, and access rate. A potential solution to this problem is to shift responsibility for personalizing content to an intermediary device. To address this issue, the Internet Engineering Task Force (IETF) standard called Framework for Service Personalization (FSP) [1] on Open Pluggable Edge network Service (OPES) [2][3] is underway. OPES is an intermediary such as an ISP's web caching proxy server that performs valued-added services on content. OPES can support the service personalization including location information, QoS policy and device types. But FSP on OPES does not accommodate the ubiquitous computing environments because OPES does not consider self-adaptation. As a result, existing networks have the inability to support additional services.

Therefore, this paper suggests the Self-adaptive Personal Service (SPS) on the extended Application Level Active Network (ALAN) [4][5] for ubiquitous computing environments by using a Generic Modeling Environment (GME) [6] tool. This paper considers the GME as Service Creation Environment (SCE) invented by Intelligent Network (IN), albeit the GME does not currently support SCE. The objective of this paper is as follows:

– To support self-adaptation for service personalization including functions such as user's changing constraints, decision-making and service composition on Active Network (AN) by using SCE tool.

The SPS means an active application on the extended ALAN meets the user's requests according to the user's changing constraints. We designed a network architecture that includes a connectivity layer, a control layer, and a modeling layer for the SPS. The existing TCP/IP acts as the connectivity layer, the extended ALAN acts as the control layer, and the GME acts as the modeling layer.

ALAN is an agent-based active network, which supports self-configuration and expedited deployment. When the active capabilities are provided at the lower layers, this is regarded as the pure form of active networking. Nevertheless, when the active capabilities are provided at the higher layers, this is regarded as the provision of active services. The advantage of the active service approach is that operations at the lower levels are not affected and deployment can be incremental and thus faster. But the existing ALAN is supported by a java applet, which cannot maintain its state, unlike the mobile/intelligent agents. Therefore, we extend a proxylet of ALAN implemented by a java applet to a mobile/intelligent agent called Jade [7].

A GME is a configurable toolkit for creating domain-specific modeling and program synthesis environments. A GME is used mainly as a real-time and embedded domain. We use a GME as SCE and service composition mechanisms. The QoS-supported GME tool is recently being developed by a GME-related project named the Component Synthesis using Model Integrated Computing (CosMIC) [8]. But the CosMIC project is mainly based on supporting QoS on Common Object Request Broker Architecture (CORBA) middleware and does not support a user's changing constraint on AN for ubiquitous computing environments.

Therefore, we describe the design and implementation of the SPS using a user's constraint-supported GME tool on the extended ALAN for ubiquitous computing environments. This paper is organized as follows: Section 2 illustrates the design of the SPS; section 3 describes the implementation of the SPS; Section 4 compares the features of the SPS. Finally, section 5 presents concluding remarks

2 The Design of the SPS

2.1 The Architecture of the SPS

The comparison between the existing service personalization and the SPS is summarized in Table 1. The existing service personalization is considered as a service personalization on a web server. Service Personalization on OPES moves one step further and makes computation (processing and transcoding) an infrastructure

Table 1. Comparison of the existing service personalization and the SPS

	The existing Service Personalization	*Service Personalization on OPES*	*SPS on the extended ALAN*
System overview	The existing web server	An overlay network of application proxies	an overlay network of applications proxies on Active Network
Services	User's preference-based personal service	User's preference and QoS policy-based personal service	Self-adaptive personal service
Protocol	Hyper Text Transfer Protocol (HTTP)	Simple Objet Access Protocol (SOAP)	Agent Communication Language (ACL), Remote Method Invocation (RMI) and HTTP
Distribution channel	Web contents	Value-added services and applications	Value-added services and applications

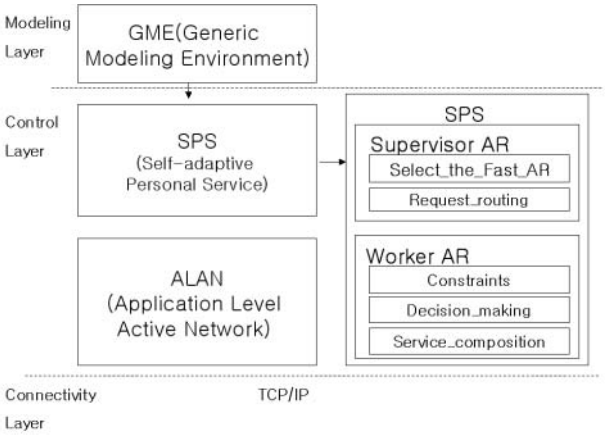


Fig. 1. The architecture of the SPS

service through the development of another overlay network. The SPS is on AN, while the architecture of the service personalization on OPES has the overlay network on the existing network. The processing of a service can support a user's changing constraints, a decision-making, and a service composition. The protocol uses an Agent Communication Language (ACL) of the mobile/intelligent agent, a Remote Method Invocation (RMI) of ALAN, and Hyper Text Transfer Protocol (HTTP). Distribution channel is used for services and applications for a service personalization.

A scenario using the SPS is as follows. A user decides to go outside while playing an online game on a wired PC. The device of the online game needs to be changed from a wired PC to wireless devices. The user can process the button to change the game on the PC to the game on the wireless device. Simply, the user can consecutively enjoy the game on the wireless device.

Fig. 1 shows the architecture of the SPS. The network architecture of the SPS consists of a connectivity layer, a control layer, and a modeling layer. AN-based service personalization mechanism on a control layer is to realize the result of a user's changing constraints, decision-making, and a service composition on a modeling layer. The control layer includes the extended ALAN integrated with the mobile/intelligent agent. The extended ALAN can support service personalization on a control layer. A modeling layer includes a GME considered as SCE, which can specify the user's changing constraints and ISP's service logic and can re-compose each service component on a modeling layer. The function of the SPS is as follows: *Supervisor AR* consists of of *Select_the_fastAR* and *Request_routing*. *Select_the_fastAR* is to select the suitable worker AR for the user's changing constraint. *Request_routing* redirects the user's request to the worker AR. A *Worker AR* consists of *Constraints*, *Decision_Making* and *Service_Composition*. *Constraints* is to specify user's changing constraints, *De-*

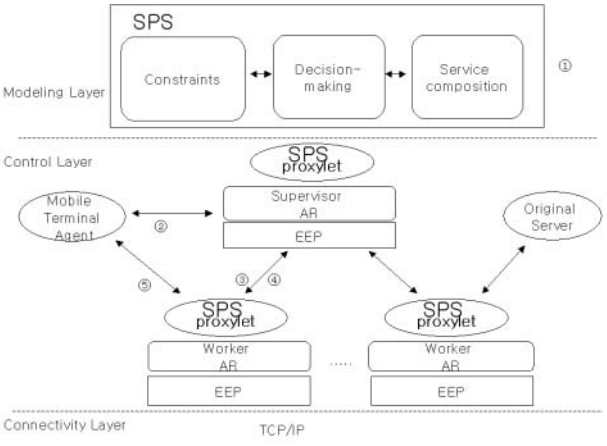


Fig. 2. The operation of the SPS on AN

cision_Making is to make a decision to re-compose each service by the defined rule. *Services_Composition* re-composes each service.

2.2 The Operation of the SPS

The operation of the SPS including a mobile terminal agent, a supervisor Active Router (AR), a worker AR, and a GME is depicted in Fig.2. The role of a GME is to specify the user's changing constraint, to make a decision by the rule, and to re-composite each component on a modeling layer. The role of a supervisor AR is to select the suitable worker AR, and to redirect the user's request to the chosen worker AR. The role of the worker ARs is to get the changing constraints of a user and the service logic of the ISP and to make a decision for service composition on the Execution Environment for Proxylet (EEP).

The order of the operation is as follows.

1. The ISP's service logic and user's changing constraints is defined by a GME.
2. A mobile terminal agent sends the constraints such as gold class, silver class, and economy class to the supervisor AR.
3. The supervisor AR gets the constraints of a user and the logic of the ISP and then selects the suitable worker AR.
4. The supervisor AR redirects the user's request to the chosen worker AR.
5. The worker AR does its own task on EEP and returns the result of the re-composite service to a mobile terminal agent.

2.3 The Mechanism for a Service Personalization

Fig. 3 shows domain variables, values and constraints to specify the changing constraints of a user. *Domain* consists of a *Subscriber-Profile*, *Service-Profile*, and

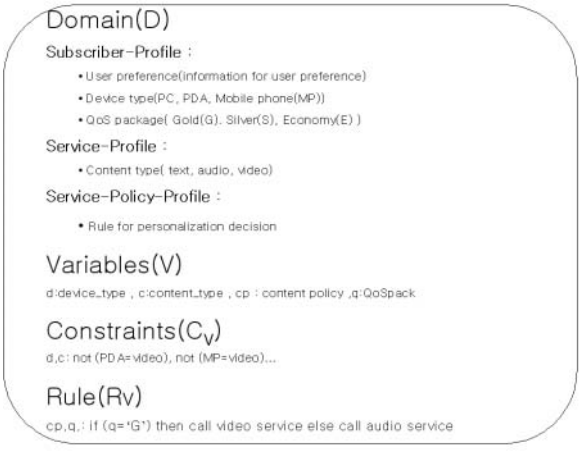


Fig. 3. Domain variables, Value and Constraints for service personalization

a *Service-Policy-Profile* for service personalization. *A Subscriber-Profile* includes properties such as a *User-preference*, *Device-type*, and *QoS-package*. A *Service-Profile* includes a property such as *Content-type*that means text, audio, video. A *Service-Policy-Profile* includes a property such as rule to decide a service personalization. The defined *Variables* of *Domain* are as follow: *d* is device-type, *c* is content-type, *q* is QoSPackage. Whether the device-type is the PC or a wireless device is defined in the *Device-type* of a *Subscriber-Profile*. Whether data is audio or video is defined in the *Content-type* of a *Service-Profile*. *QoS-Package* of a *Subscriber-Profile* is categorized into gold, silver and economy. A rule depends on *Content-Policy-Profile*. The defined variables specify the *Constraints*. *d,c : not (PDA = video) and (MP = video)* means a video is not supported in PDA and Mobile Phone (MP). *if (q=' G') then call video service else call audio service* means the rule that calls a video service in case that *QoSPackage* is gold class and calls an audio service in case that *QoSPackage* is silver class.

Fig.4 shows the mechanism named the SPS proxylet for a service personalization. The SPS proxylet consists of *Constraints*, *Decision_Making* and *Serivce_Composition*. *Constraint* is to specify a user's changing constraint. *Decision_Making* is to determine how to re-composite proxylets for service by the constraint and the rule. *Service_Composition* is to re-composite the services by the rule for a service personalization.

3 Implementation of the SPS

The implementation of the SPS is based on Windows 2000 server, the Java language-based public software of ALAN that was developed by UTS and BT and a GME2000 tool that was developed by Vanderbilt University. A GME tool can automatically generate the C programming language according to the

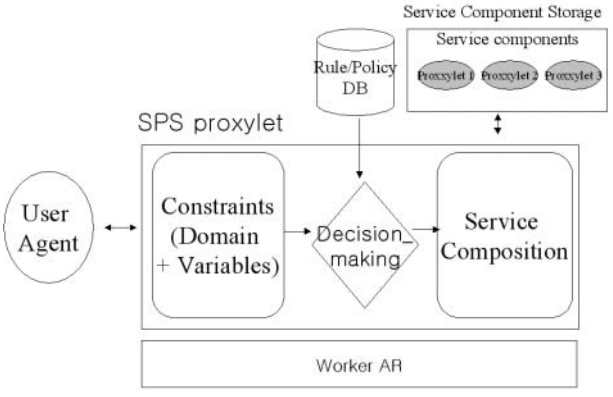


Fig. 4. The mechanism for a service personalization

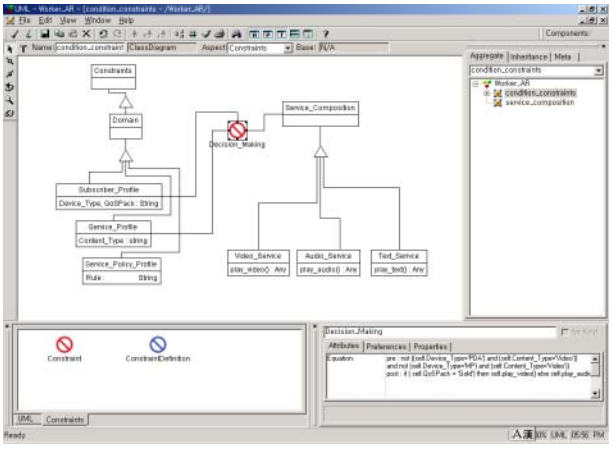


Fig. 5. The mechanism for the SPS proxylet using a GME

specific modeling but does not currently support the java programming language. So we currently map manually the modeling to a Java-based mobile/intelligent agent called Jade.

3.1 The SPS Using a GME Tool

The meta-modeling of a GME supports meta-model composition. The SPS is presented by a GME. A GME is specified by Object Constraint Language (OCL) [9]. We limit the constraints of a user to a gold class and a silver class and limit a service composition to a video service and an audio service. The constraints of a user on a PC are defined as a gold class while the constraints of a user on a wireless device are defined as a silver class. The example of the service logic is as follows: When the worker AR gets the gold class as the constraints of a user,

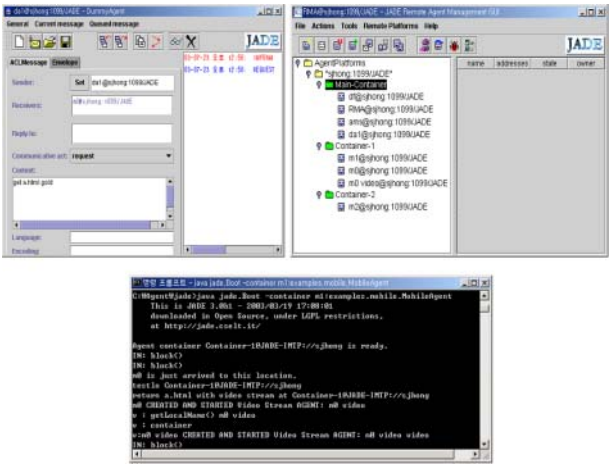


Fig. 6. Test of a mobile terminal agent, a supervisor AR, and a worker AR

the worker AR can choose a video service. When the worker AR gets the silver class as the constraints of a user, the worker AR can choose an audio service.

Fig. 5 shows the mechanism for the SPS proxylet using a GME tool. The mechanism for the SPS proxylet consists of *Constraints*, *Decision_Making* and *Service_Composition*. *Constraints* has the user's constraints in a *Subscriber_Profile*, a *Service_Profile* and a *Service_Policy_Profile*. *Decision_Making* is used by OCL. The user's constraint in a *Subscriber_Profile* and *Service_Profile* is expressed by a pre-condition in OCL. A rule depends on a *Service_Policy_Profile*. The rule in *Service_Policy_Profile* is expressed by post-condition in OCL. For instance, *pre: not((self.devicetype='PDA') and (self.contenttype='video'))* means the video service is not supported in PDA. *pre* means a pre-condition of OCL. *post : if(self.QoSPack ='G') then self.videoplay() else self.audioplay() endif* means a video service is called in case that *QoSPackage* is a gold class and an audio service is called in case that *QoSPackage* is a silver class. *post* means a post-condition of OCL.

3.2 The Execution of the SPS

Fig. 6 (top left chart) shows the test of a mobile terminal agent, a supervisor AR and a worker AR. We use a dummy agent of jade as the mobile terminal agent. We are developing the mobile terminal agent on the mobile device platform called Lightweight Extensible Agent Platform (LEAP) on J2ME devices such as PALM IIIc. The mobile terminal agent requires the object called a.html with gold class. When the mobile terminal agent requires an object called an a.html on the PC, the message stated as *get a.html gold* is sent to the supervisor AR. When the mobile terminal agent requires an object called *a.html* on a wireless device, the message stated as *get a.html silver* is sent to the supervisor AR. Fig. 6 (top right

chart) shows the test of the supervisor AR. When the supervisor AR gets the user constraint called gold class, the supervisor AR redirects to the worker AR close to the users. the worker AR chosen by the supervisor AR provides the video service at the EEP called *Container-1*. Fig. 6 (bottom chart) shows the result of the worker AR. As the worker AR composes the video service at the EEP, the message stated *m0 video CREATED and STARTED Video Stream Agent* is shown at the EEP. As the worker AR composes the audio service at the EEP, the message stated *m0 audio CREATED and STARTED Audio Stream Agent* is shown at the EEP.

4 Comparison of the Features of the Existing Service Personalization and the SPS

Table 2 shows the comparison of main features of the existing service personalization, service personalization on OPES and the SPS on AN. The SPS has more features, such as a self-configuration, a fast deployment, and a self-adaptation, than the existing service personalization. A self-adaptation includes functions such as a user's changing constraints, a decision-making, and a service composition.

There is no difference of performance between the existing SP and the SPS when the constraints of a user are fixed. However, in the case of frequently changing constraints of a user, the SPS provides more an enhanced performance than the existing SP. This is attributed to the fact that the SPS supports self-adaptation, whereas, the existing SP does not have the self-adaptation functionality.

Table 2. Comparison of main features

	The existing Service Personalization	*Service Personalization on OPES*	*SPS on AN*
Self-configuration	-	X	X
Fast deployment	-	-	X
User's changing constraints	-	-	X
Decision-making	-	-	X
Service-composition	-	-	X

5 Conclusion

This paper has presented our service on AN for ubiquitous computing environments. We believe that this service addresses a new mechanism to support the self-adaptation on AN. We expect our mechanism to comply with the industry standard such as FSP. Our future work will involve more studies on self-adaptation for the situation-aware mechanism, which extends the context-aware mechanism, using AN and the advanced Modeling Integrated Computing (MIC) for ubiquitous computing environments

References

[1] Barbir et al.,"A Framework for Service Personalization", draft-barbir-opes-fsp-03.txt, work in progress, March 2003.
[2] Tomlinson, G., Chen, R. and M. Hofmann, "A Model for Open Pluggable Edge Services", draft-tomlinson-opes-model-00.txt, work in progress, June 2001.
[3] Govindan Ravindran, Muhammad Jaseemudin, Abdallah Rayhan, "A Management Framework for Service Personalization," 5th IFIP/IEEE International Conference on Management of Multimedia Networks and Services, MMNS 2002, Santa Barbara, CA, USA, October 6-9, 2002.
[4] I. W.Marshall, et, al, "Application-level Programmable Network Environment," BT Technology Journal, Vol. 17, No2 April 1999.
[5] K. T. Krishnakumar, M. Sloman, "Constraint- based Network Adaptation for Ubiquitous Applications," Proceedings of the 6th Interna- tional EDOC Conference, Sep. 2002, pp 258-271, Laussane, Switzerland.
[6] Ledeczi A., Maroti M., Bakay A., Karsai G., Garrett J., Thomason IV C., Nordstrom G., Sprinkle J., Volgyesi P, "The Generic Modeling Environment," Workshop on Intelligent Signal Processing, accepted, Budapest, Hungary, May 17, 2001.
[7] Bellifemine,F.,Poggi,A., and Rimassa,G.,"Developing mulit-agent system with a FIPA-compliant agent framework," Software - Practice and Experience 31(2), p.103-128,2001.
[8] Aniruddha Gokhale et al. "CosMIC: An MDA Generative Tool for Distributed Real-time and Embedded Component Middleware and Applications," Proceedings of the OOPSLA 2002 Workshop on Generative Techniques in the Content of Model Driven Architecture, Seattle, WA, November 2002.
[9] J. Wing,, and Kleppe, A.,"OCL : The Constraint Language of the UML,"JOOP, May, 1999.

A New Directional Flooding Protocol for Wireless Sensor Networks*

Young-Bae Ko[1], Jong-Mu Choi[2], and Jai-Hoon Kim[2]

[1] Division of Information and Computer Engineering
[2] Graduate School of Information and Communication, Ajou University, South Korea
Tel. +82-31-219-2443, Fax. +82-31-219-1614
{youngko,jaikim}@ajou.ac.kr
jmc@dmc.ajou.ac.kr

Abstract. Flooding is commonly used both for conventional ad hoc networks and wireless sensor networks. Most of previous works on developing efficient flooding protocols are focused on making an optimal broadcast tree, with an implicit assumption that all nodes should be reachable from the source. However, this aim as well as the implied assumption may no longer be true because flooding protocols in wireless sensor networks are used to deliver the data packets towards a single or only subset of destination node(s). In this paper, we propose a new flooding protocol for utilizing directional information to achieve the efficiency in data delivery. The proposed directional flooding can lead flooded packets to flow in the "right direction" towards their destinations, hence eliminating unnecessary packet forwarding and reducing the total energy consumption. Our simulation results show the average number of transmitted packets can be significantly reduced over the existing flooding algorithms, in which possibly all nodes are participated in a flooding process.

1 Introduction

Wireless sensor network [1] has received great amount of research attention in recent years, due to its several uniqueness distinguished from a wireless ad hoc networks. Those unique characteristics include much higher number of densely (or spatially) deployed nodes that are extremely limited in energy and computational resources. Another uniqueness is that wireless sensor nodes often have no global identification (ID) like IP address.

The wireless sensor network is still considered as one form of an ad hoc wireless network, however, because both networks are quite similar in many aspects; for instance, they are capable of self-configuration and self-maintenance. Transferring packets via multi-hop wireless links (i.e., capable of multi-hop routing) is another important similarity between the two networks. Actually, the study of routing protocols in mobile ad hoc networks has been a very active area of

* This work was supported by the Korea Research Foundation (KRF-2003-003-D00375).

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 93–102, 2004.

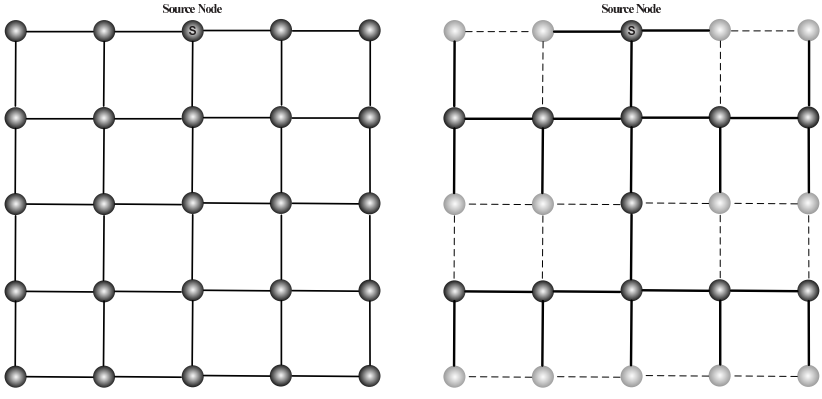
research, producing a number of protocols [2,3,4]. For wireless sensor networks, there has been recent attention on routing (also, called a data dissemination) from sensor nodes to a sink node.

Most of these routing/data dissemination protocols in wireless ad hoc or sensor networks are based on variations of "flooding," even though they use some optimizations. Flooding is clearly a straightforward and simple solution, but it is very costly in general. Furthermore, most MAC protocols for mobile ad hoc networks and wireless sensor networks use the carrier sensing multiple access with collision avoidance (CSMA/CA) based MAC approaches. These CSMA/CA based MAC protocols could cause serious contention when many adjacent nodes decide to broadcast simultaneously. These contention, collision and redundant broadcasts can be referred to as the *broadcast storm problem* [5]. Thus, without careful designs, it will result in serious performance degradation. To solve such a problem, a few solutions have been presented in ad hoc networks environments [5,6,7,8]. Their basic idea is to build some optimal broadcast tree, through which packets can be delivered to every node reachable from a source (eventually, all nodes in the networks).

In this paper, we argue that even such an optimized tree approach is not well suited for wireless sensor networks, causing unnecessary overhead. This is because, in sensor networks, packets are not necessary to reach all nodes. Especially, for the case when sensing data packets are towards a single destination (i.e., a centralized sink) from other sensors, the packets are required to reach the sink only. We demonstrate how even more optimal routing can be possibly made by means of the proposed "Directional Flooding" protocols for wireless sensor networks. The proposed flooding schemes use directionality information (which shows whether the packets are traveling toward a sink or not) to reduce the number of intermediate nodes unnecessarily forwarding packets. Limiting the number of nodes results in less energy consumption.

2 Related Works

In this section, we examine the current approaches to develop efficient flooding protocols in mobile ad hoc networks. We then motivate our work by pointing out that those protocols may not be the best in wireless sensor network environments. As mentioned earlier, several modified methods for flooding have been proposed to minimize the defect of broadcast storms in mobile ad hoc networks [5,6,7,8]. Most of these methods have been developed, based on the idea of producing an optimal broadcast tree and thus minimizing the number of transmitted packets in the whole flooding process. Fig. 1 illustrates a comparison between a pure flooding tree and an optimal flooding tree. Thus, in Fig. 1(a) with a pure flooding, data packets are being transmitted via all nodes that are all required to forward the packets exactly once. This pure flooding is quite reliable but clearly too expensive. For instance in Fig. 1(a), 25 times of transmission would occur. More efficient way of flooding is to find some optimal tree, through which all nodes can still be reachable from the source with the minimal times of packet transmission.



(a) Pure flooding tree (b) Optimal flooding tree

Fig. 1. Optimal broadcast flooding

For instance, in Fig. 1(b), only 12 times of transmission would be needed to reach all nodes.

However, [8] verified that the problem for finding such an optimal broadcast tree is NP-complete. Therefore some alternatives for efficient flooding protocols have been proposed to construct an approximate form of optimal broadcast tree. In [9], the authors have categorized and compared these methods into three groups – probability based, area based, and neighbor knowledge methods. The probability-based methods, such as [5], is similar to a pure flooding, except that intermediate nodes only rebroadcast with a predefined probability Area based [6] and neighbor knowledge based [7] methods also decide whether to rebroadcast or not by predefined calculating its additional coverage area and maintaining neighbor nodes information, respectively. All of these approaches aimed at making efficient broadcast tree using adaptive or heuristic based algorithm.

However, in case of wireless sensor networks, one question is arisen – "Are these optimal-tree-based flooding protocols still necessary for wireless sensor networks?" We believe the answer is "probably not." Because making an efficient broadcast tree assumes successful delivery in all nodes in the networks. To the contrary, in wireless sensor networks, goal of flooding is not reach all nodes but to reach the only one or several destination(s), just like a unicast or geocast. This means that packet flow of flooding protocol with destination(s) can have the directionality toward destination. Fig. 2 shows the tree constructed by directional flooding protocol. As you see in the figure, only 4 times of transmission is enough to reach the destination.

Using this motivation, we suggest a protocol to create packet flow that is gradually approached to the destination. Thus, by utilizing directionality information, we attempt to reduce the number of nodes unnecessarily involved in the flooding process.

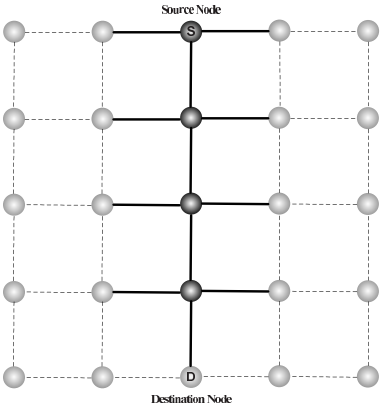


Fig. 2. Tree constructed by directional flood

There have also been a number of recent works on efficient data delivery in wireless sensor networks. [10, 11] These schemes also use routing cost to determine whether to forward or not. However, such schemes typically require full neighbor information, thus necessitating initial set-up time when the nodes get to know their neighbors. In case of dynamic environment, these protocols require periodic monitoring of neighbors' information and it causes many overhead in the aspect of number of packets, energy consumption, and delay.

3 The Proposed Directional-Flooding Scheme

In this paper, we propose a new directional flooding protocol, where the flooding decision is made with considering the directionality information towards a destination. To realize our approach, we have made certain assumptions in utilizing the directional information to flood protocol. These assumptions include: all sensor nodes know *their own location information* and *sink nodes' location information*. It is not unreasonable to expect such localization features in a sensor node using some of the techniques described in the recent literatures [12, 13]. Of course, it can also be assumed for a sensor node to possess a fully functional GPS receiver.

We now describe how our directional flooding protocol works in wireless sensor networks:[1]

- Initially, each sink node announces its own location information to all sensor nodes in a network field. Because it is often assumed that a sink is quasi-stationary, this global announcement by the sink will be done only once at a deployment time. After receiving an announcement, a sensor node calculates an estimated minimum hop count between itself and the sink. This

[1] The procedures here one for data flows from sensor nodes to the sink. The opposite direction of data flows will be discussed in a later section.

estimation is based on the knowledge about the two location information and a sensor nodes' transmission range.
- When a sensor node initiates a flooding of packet, it puts its minimum hop-count toward a destination sink node in the packet.
- When a node receives a packet, it compares its own value of minimum hop-count toward destination sink with that in the receiving packet. Of all neighboring nodes receiving the broadcast, only those that have a smaller value than the stated minimum hop value in the packet will forward it. All other sensor nodes (having a higher or an equal value) will simply ignore the packet. In this way, the packet slides down to the sink in a "right" direction.

The above procedures continue until the destination sink is reached. Fig. 3 provides an illustration for more detailed operations of our directional flooding protocol.

Fig. 3(a) shows an initial topology of our example with one source(S) and one sink. For all the figures, a distance between node S and a sink is assumed to be 100 units and each sensor nodes' maximum transmission range is 30 units. Therefore, node S become aware of the sink's location initially, it knows about the minimum hop count value for the sink as 4.

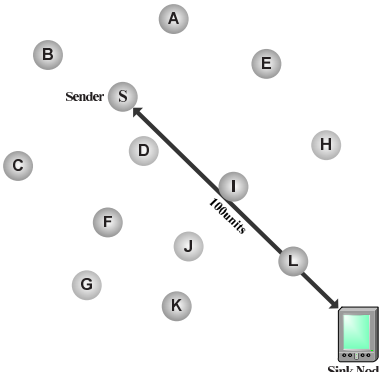
In Fig. 3(b), among the node S's one-hop neighbors (i.e., node A, B, and D) receiving the packet, only node D forwards it because D's minimum hop count (=3) is less than that of the packet (=4). Note that, nodes A and B do not rebroadcast the receiving packet as their minimum hop counts for the sink are not smaller than 4 in the figure. Thus, these two nodes conclude that they are not in the right direction for this particular packet. The dark black line in the Fig. 3(c) shows an edge of the tree constructed by the first step of our directional flooding protocol. Fig. 3(d) and (e) also show the further steps based on the same rules. After the final step where the sink node finally receives the flooded packet, the tree constructed by our directional flooding is shows in Fig. 3(f). Observe that only four intermediate node (D, I, J, and L) out of twelve are involved in a flooding process.

4 Performance Evaluation

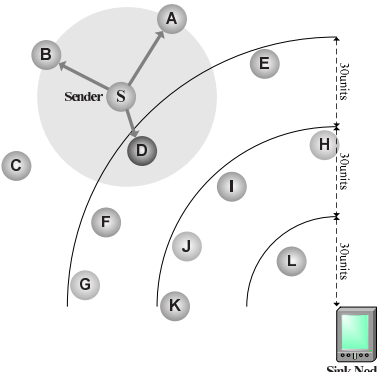
In this section we evaluate the performance of our directional flooding, compared to the blind flooding protocol. For our simulation we have modified the CMU wireless extended version of ns-2 [14].

4.1 Simulation Model

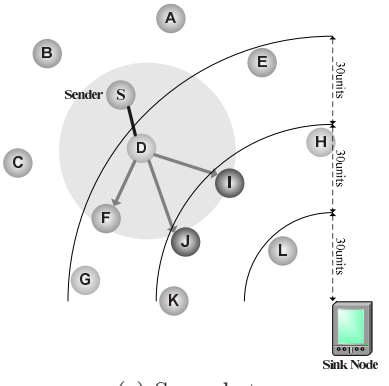
To evaluate the performance of proposed protocol, we use the following two metrics for a measurement: *total number of transmitted packets* during the flooding process and *average energy consumption* on each node. By reducing these two metrics, nodes can conserve its remained energy to expand the network lifetime of wireless sensor network. The assumptions for our simulations are as follows.



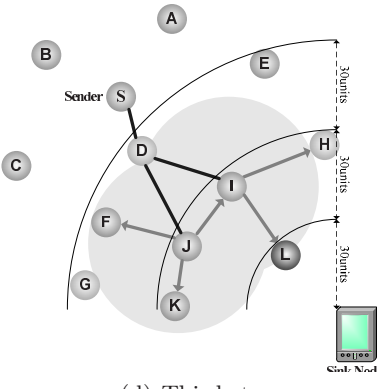
(a) Initial Topology



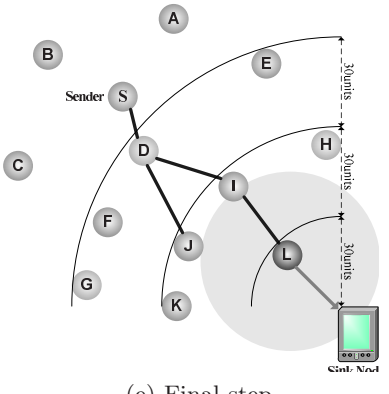
(b) First step



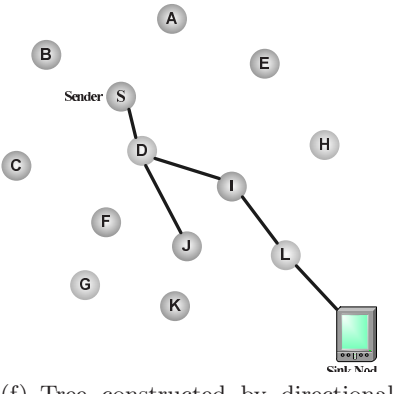
(c) Second step



(d) Third step

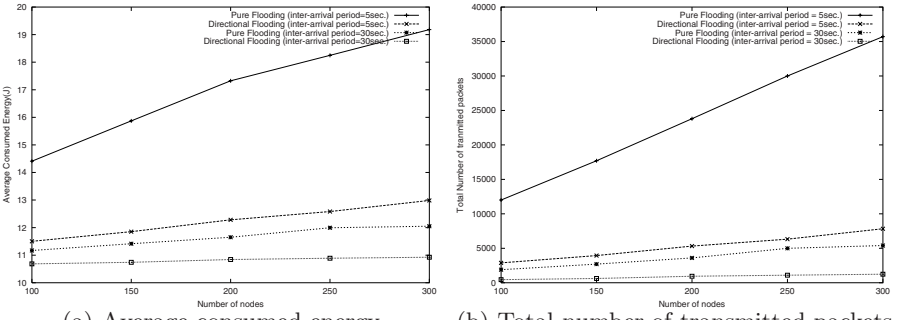


(e) Final step



(f) Tree constructed by directional flooding

Fig. 3. Operation example of directional flooding



(a) Average consumed energy (b) Total number of transmitted packets

Fig. 4. Performance of the directional flooding by varying the number of nodes

– Initially nodes are randomly distributed in a confined space of $200 \times 200m^2$
– Transmission range of each node is 30.5 meters.
– Each event packet size is fixed to 64bytes.

We also refer to the reference of the Berkeley Motes [15] to model the energy consumption behavior and wireless physical layer of wireless sensor node. In our simulation, we only performed one way of packet delivery from sensor nodes to sink [2].

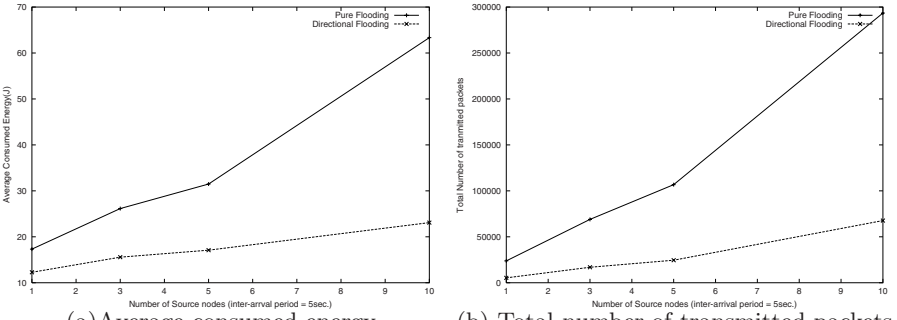
We varied the number of nodes, the number of source nodes, and an event generation inter-arrival period. Varying the number of nodes (from 100 to 300 by 50) intends to observe the effect of nodes' density on each metrics. By varying the number of source nodes (from 1, 3, 5, and 10), we can see the effect of multiple source nodes. And we also vary the event generation inter-arrival period to simulate various traffic environments. It varies from 1 to 30 seconds in our simulations. Each simulation has duration of 300 seconds.

4.2 Simulation Results

The effect of varying the number of nodes is shown in Fig. 4. As the number of sensor nodes increases, both the average consumed energy and total number of transmitted packets begin to increase for both protocols. However, our directional flooding requires much less cost than a pure flooding protocol for both cases of inter-arrival periods 5 (The two upper lines in the figure) and 30 (The below two lines). The reason is clear that directional flooding reduces the unnecessary packet forwarding according to the nodes' hop count towards a destination.

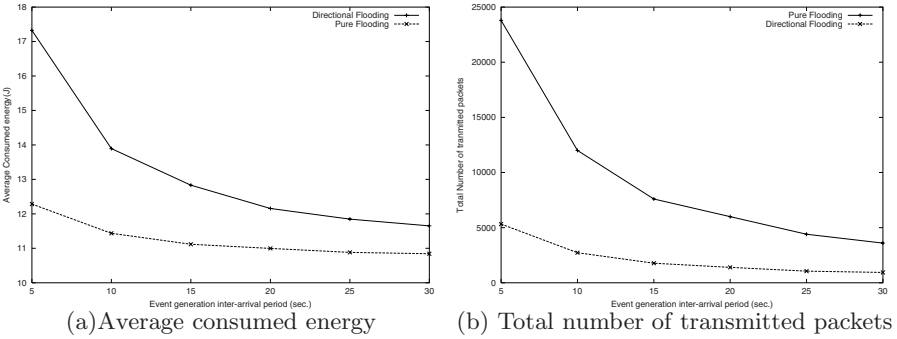
Fig. 5 displays the effect of varying the number of source nodes while the other two simulation parameters are fixed (i.e., 200 nodes and 5 seconds inter-arrival period). This graph also shows that the directional flooding has better

[2] In case of packet delivery from sink to sensor nodes, we remain this example for future works.



(a)Average consumed energy (b) Total number of transmitted packets

Fig. 5. Performance of the directional flooding by varying the number of source nodes



(a)Average consumed energy (b) Total number of transmitted packets

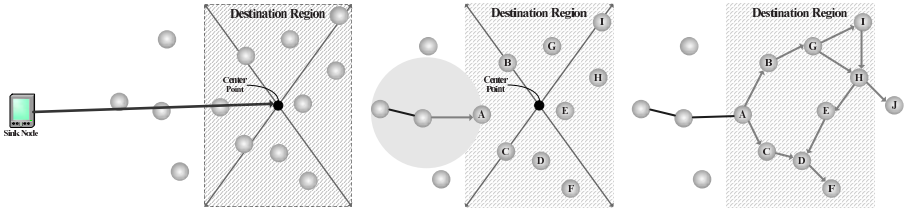
Fig. 6. Performance of the directional flooding by varying the event generation inter-arrival periods

performance than the pure flooding for the same reason. Especially, with an increase of the number of source nodes, a pure flooding has more degradation in their performance. This is because, as the network traffic goes up, the pure flooding suffers from more defect of broadcast storm problem.

Finally, Fig. 6 shows the comparison results by varying the event generation inter-arrival rate, with the fixed number of nodes (=200) and a source node (=1). As we expected, our directional flooding shows the better performance than the pure flooding as the traffic is increased (i.e., smaller event generation inter-arrival period).

5 Discussion

In this section, we consider the packet delivery from sink node to sensor nodes. Example of such packets is some initial announcements or queries. In this case we need an assumption that the sink node knows location of destination region



(a) Direction toward center coordination of destined region (b) Reaching destination region (c) Local flooding

Fig. 7. Operation example of directional flooding in delivery from sink to sensor

for queries (e.g., "Retrieve the average temperature in the southern quadrant.") As seen in Fig. 7(a), the same algorithm described in the previous section can be applied for this case, except for the fact that a center of destination region is used to get a distance between the sink and the target sensors. In this case, when the sink node initiates query packet, following information to be added on packets by the sink node.

- Destination region's coordination of center point[3]
- Minimum hop count from the sink (or rebroadcasting sensor node) to the destined region
- Coordination of destination region
- A region flag which represents whether packet arrives the destination region or not

If any node in the destination region receives a query packet, it sets the "Region flag" on the packet (In the Fig. 7(b), node A sets this value). When the Region flag is set, the local flooding is used inside the destination region to enhance reliability. Namely, by default, all sensors within the region are required to forward any receiving packets to other node. However, any sensors located outside the destination region (e.g., node J in the Fig. 7(c)) simply drop receiving packets without forwarding them further.

6 Conclusions

Flooding is commonly used both for conventional ad hoc networks and wireless sensor networks. Many approaches have been proposed to develop efficient flooding protocols for mobile ad hoc networks. These previous works assume that all node be reachable from the source. In this paper, we argue that this assumption may not suitable for wireless sensor networks. This is because, flooding protocol in wireless sensor networks are used to deliver data packets towards only one

[3] For ease of exposition, we choose the destined region to be a rectangle defined on some coordinate system; in practice, this might be based on GPS coordinates.

or subset of destination node(s). Based on this assumption, we propose a directional flooding protocol for wireless sensor networks. To verify the excellence of proposed protocol, we have done simulation study. The performance evaluation clearly shows that our directional flooding performs much better than the pure flooding protocol in terms of both average energy consumption and total number of transmitted packets.

References

[1] I. F. Akylidiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," *in Elsevier Computer Networks*, Vol. 38, pp. 393-422, 2002.

[2] D. Johnson, D. Maltz, and J. Broch, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *IETF MANET Working Group, draft-ietfmanet-dsr-03.txt*, Nov. 1999.

[3] C. E. Perkins and E. M. Royer, "Ad-hoc On demand Distance Vector Routing," *IETF MANET Working Group, draft-ietfmanet-aodv-05.txt*, Mar. 200.

[4] Y.-B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *in proc. of ACM Mobicom '98*, pp. 66-75, Oct. 1998.

[5] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *in proc. of ACM Mobicom '99*, Aug. 1999.

[6] V. K. Paruchuri, A. Durresi, D. S. Dash, and R. Jain, "Optimal Flooding Protocol for Routing in Ad Hoc Networks," *in proc. of IEEE WCNC '03*, Mar. 2003.

[7] H.-J. Lim and C.-K. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," *in proc. of ACM MSWiM*, 2000.

[8] W. Liang "Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks," *in proc. of ACM Mobihoc '02,* pp. 112-123, Jun. 2002.

[9] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," *in proc. of ACM Mobihoc '02*, Jun. 2002.

[10] C. Intanagonwiwat, R. Govinda, D. Estrin, and J. Heidemann, "Directed Diffusion for Wireless Sensor Networking," *in IEEE/ACM Transaction on Networking*, Vol. 11, No. 1, pp. 2-16, Feb. 2003.

[11] F. Ye, G. Zhong, S. Lu, L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Prtocol for Large Scale Sensor Networks," *to appear at the ACM WINET Journal*.

[12] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," *in proc. of IEEE INFOCOM '01*, Apr. 2001.

[13] A. Savvides et al. "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *in proc. of ACM MOBICOM '01*, pp. 166-179, 2001.

[14] The CMU Monarch Project, "The CMU Monarch Project's Wireless and Mobility Extensions to NS"

[15] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Mobile Networking for Smart Dust," *in proc. of ACM/IEEE MOBICOM '99*, Aug. 1999.

An Efficient Scheduling Scheme for Bluetooth Scatternets Using the Sniff Mode*

Woosin Lee[1], Hyukjoon Lee[1], Seung Hyong Rhee[2], and Hyungkeun Lee[1]

[1] School of Computer Engineering, Kwangwoon University
447-1 Wolgye-Dong, Nowon-Gu, Seoul 139-701, Korea
wlee@kw.ac.kr
{hlee,hklee}@daisy.kw.ac.kr
[2] School of Electronics Engineering, Kwangwoon University
447-1 Wolgye-Dong, Nowon-Gu, Seoul 139-701, Korea
rhee@ieee.org

Abstract. Several Bluetooth piconets can be interconnected using an inter-piconet unit called a bridge node to form a Bluetooth scatternet. The bridge node can make its presence in each piconet on a time division basis by switching its frequency hopping sequence. Without a careful coordination among nodes, piconet switching can result in conflicts and idle time in communication, which impact the throughput of the entire scatternet. In this paper we present an efficient inter-piconet scheduling scheme called LSRR (Limited Sniff Round Robin). LSRR is a locally coordinated scheduling scheme that assigns presence points in round robin fashion among different piconets. LSRR uses the sniff mode as the inter-piconet switching mechanism and dynamically adjusts the duration of communication events based on the past history of transmissions. Simulation results show LSRR outperforms the pure round robin scheme in terms of throughput and delay.

1 Introduction

The emergence of Bluetooth [1] as a low power, low cost, short-range radio interface for small handheld devices, such as PDAs, mobile handsets, MP3 players, DVD players and notebook computers, allows high speed data communication between them. Although the major application of Bluetooth technology is the cable replacement of point-to-point links, its networking capability enables a new networking paradigm, i.e., a personal area networking (PAN).

A Bluetooth piconet consists of two or more Bluetooth devices sharing the same frequency-hopping channel by time division multiplexing in a star topology with one master node controlling up to seven other nodes (slaves). Several piconets can be interconnected in an ad hoc fashion via one or more inter-piconet nodes called bridge nodes participating in more than one piconet to form a Bluetooth scatternet. Bluetooth scatternets offer a wide range of applications.For example, people in a conference room can have their notebook computers and/or

* This work was supported by Grant No. R01-2001-00349 from the Korea Science & Engineering Foundation and the Research Grant of Kwangwoon University in 2003

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 103–113, 2004.

PDA's interconnected to exchange various kinds of information such as business cards for a collaborative work.

Currently, the Bluetooth specification does not define a mechanism for the formation of topology or coordination of communication in scatternets. A number of research works related to the formation of scatternet topology have been published [2, 3, 4] . These works however do not address how the Bluetooth nodes should be coordinated to let data traffic flow efficiently over the scatternet.

Data communication between two Bluetooth devices participating in neighboring piconets is achieved by the so-called bridge node, which participates in both piconets forwarding data packets between the two devices. Since any Bluetooth node including the bridge node can make its presence in one piconet at a time, it must switch between different piconets on a time division basis. This inter-piconet switching may result in a loss of bandwidth if the bridge node switches to a piconet where the peer node has no data to transmit or is not participating in the piconet at all. Therefore it is important to carefully coordinate the presence of the bridge node in the different piconets (i.e., inter-piconet scheduling) such that the loss of bandwidth is minimized.

The polling schedules of two directly connected nodes in a Bluetooth scatternet are inter-dependent. That is, they have to coordinate their own schedulers to ensure that the communication between them starts and ends at the same instances. Therefore, the inter-piconet scheduling problem becomes a global coordination problem that expands to the entire scatternet. As pointed out in [5], constructing the optimal link schedule that maximizes total throughput in a Bluetooth scatternet is an NP hard problem even if scheduling is performed by a central entity. A scatternet-wide scheduling approach would in general not only require a large amount of signaling overhead but also would react rather slowly to varying network traffic conditions. On the other hand, a locally coordinated scatternet scheduling approach does not require the scatternet-wide coordination overhead and is able to adapt rapidly to varying traffic conditions. A local scheduling approach may however result in a loss of bandwidth since it cannot guarantee conflict-free inter-piconet scheduling.

Our inter-piconet scheduling scheme, LSRR, is based on the locally coordinated scatternet scheduling approach. The presence points are assigned in a round robin fashion among different piconets. Communication durations are dynamically adjusted based on the information about the past link utilization.

The rest of this paper is organized as follows. In section 2 a brief overview of the Bluetooth technology is presented. In section 3 we give an overview of related works in Bluetooth scatternet scheduling. In section 4 our proposed scheme is described and simulation results are given in section 5. Finally, in section 6, we conclude our discussion.

2 Bluetooth Overview

Bluetooth technology uses frequency hopping scheme in the unlicensed 2.4 GHz ISM (Industrial Scientific-Medical) band. The frequency spectrum is divided

into 79 or 23 channels of 1 MHz bandwidth depending on the country where it is operated. The hop frequency is up to 1600 hops per second. Its radio transmission uses slotted protocol with TDD (Time Division Duplex) and the time length of each slot is 0.625ms. Communication between Bluetooth devices is based on a mater-slave scheme. That is, any Bluetooth device operates as a master or slave. The master unit controls the communication. Each master unit can connect to a maximum of 7 active slaves to form a piconet. All Bluetooth units belonging to the same piconet share the same frequency hopping sequence determined uniquely by the master and remain synchronized.

The Bluetooth specification defines two types of links between a master-slave pair, i.e., a Synchronous Connection Oriented (SCO) link and an Asynchronous Connectionless link (ACL). The former is typically used for voice transmission and the latter for data. The transmission of a packet in the ACL link from the master should begin in an even numbered slot. The receiving slave should send a packet in response to the master in the immediately following odd numbered slot. Thus, if the master has no data for the slave, it may send a poll packet to receive data from the slave. If the slave does not have data to send, it may send a null packet (a poll-null sequence).

The frequency hopping sequence of a piconet is uniquely determined from the clock and the Bluetooth address of its master. Since a Bluetooth device with a single transceiver can follow a single frequency hopping sequence at a time, it can switch between different piconets by changing its frequency hopping sequence.

Bluetooth offers three power saving modes to allow its devices to minimize their power consumption: i.e., sniff, hold and park mode. In the sniff mode two devices can begin their communication only in specific slots (sniff slots) that they have agreed on to reduce the duty cycle at which a slave needs listen to the master. The time interval between two sniff slots is called a sniff period and denoted by T_{sniff}. At each sniff slot the slave listens to the master for at least $N_{sniff\_attempt}$ slots. If the master does not send packets, the slave aborts listening to the master in $N_{sniff\_timeout}$ slots. A sniff event is the duration of an actual communication session that may continue until the next sniff slot arrives and may be extended dynamically until one of the devices decides to end the communication. The sniff mode can be used to implement the timing mechanism of inter-piconet switching by the bridge node. We describe this in more detail in section 4.

3 Related Works

Currently, there exists only a handful of research works on inter-piconet scheduling published in the literature. S. Baatz *et. al.* in [6] present and analyze an adaptive scheme that determines the presence points and communication durations locally. Their scheme is based the credit scheme, which is inspired by leaky bucket traffic shaping and the deficit round robin fair scheduling mechanism, and tries to serve the links in a fair manner. In [7], P. Johansson *et. al.* propose

an inter-piconet scheduling algorithm called the Maximum Distance Rendezvous Point (MDRP). The main idea behind the MDRP algorithm is that "Rendezvous Points," i.e., a slot at which a master and the bridge unit have decided to meet, should be placed as far from each other as possible. However MDRP only works for the case that the bridge unit does not assume its role as a master unit. In [8] Racz *et. al.* present a pseudo random coordinated scheduling algorithm, in which every node randomly chooses a communication checkpoint. The checking periods are adjusted by a fixed multiple based on the link utilization in order to adapt to various traffic conditions, while the duration of communication events are not changed. Although this scheme achieves some degree of coordination among the schedules of individual units with little overhead, it may not scale up well since conflicts are expected to occur more often as the number of nodes increases. In [9] Tan *et. al.* presents a locally coordinated dynamic and distributed scatternet scheduling algorithm (LCS). What separates LCS from others is that it achieves a scatternet wide schedule by aligning meetings in a hierarchical fashion. Nodes coordinate communication among neighboring nodes, while allocating bandwidth based on current local traffic conditions. LCS however not only has a limited applicability to a loop-free (i.e., tree topology) scatternet, but it also suffers from amount of signaling overhead, which may become serious as the size of the network becomes large.

4 Limited Sniff Round Robin (LSRR) Scheduling Scheme

LSRR is a locally coordinated scatternet scheduling approach where each bridge node is scheduled independently. The presence points are assigned in a round robin fashion among different piconets, and communication durations are dynamically adjusted based on the information about the past link utilization. Coordinating scatternet scheduling locally removes the signaling overhead and reacts rather quickly to changing traffic conditions. On the other hand, it has the disadvantage of conflicts among the schedules of individual bridge nodes in a scatternet. The scatternet wide coordination of piconets is not our main concern in this paper. Instead we focus on the efficiency of the inter-piconet scheduling scheme for a bridge node, especially when the sniff mode is used as the inter-piconet switching mechanism.

The sniff mode is one of the recommended timing mechanisms for implementing inter-piconet switching by the bridge node (cf. [1]). When the sniff mode is used to implement the bridge node's switching between different piconets, some inefficient use of slots may occur, which impacts the throughput of the entire scatternet. In what follows we first explain how the inter-piconet switching is implemented using the sniff mode.

A sniff slot can be used as a presence point at which the bridge node starts to communicate with one of its peering devices. In each sniff slot the corresponding master must attempt to communicate with the bridge node. When the sniff mode is used as the switching mechanism, some slots can be wasted (Fig. 1). When the bridge node stops communicating with master A and switches to master B in the

next sniff slot, A may continue to send packets to the bridge node without success for $N_{sniff\_timeout}$ before it finally decides to stop communicating with the bridge node. As a result, A wastes $N_{sniff\_timeout}$ slots that could have been used for the communication with one of its slaves (case 1). If the bridge node ignores the sniff slot and continues to communicate with A, B may try to communicate with the bridge node in the next sniff slot. With no response from the bridge node, B aborts immediately, but it has already wasted two slots (case 2) .

Since the master unit's clock defines the piconet timings independently from each other, the slot boundaries of different piconets may not be aligned. Thus, the bridge node may have to wait until the next even slot begins after the piconet switch. This waiting period may waste up to two slots (so-called guard time). Since wasted slots result in reduced throughput of the scatternet, frequent inter-piconet switching should be avoided. Conversely, long switching intervals may cause forwarding delays at each hop in a connection across the scatternet.

One straightforward approach of the inter-piconet scheduling is to have the bridge node switch between the links to all of its peer nodes (including both the masters and slaves) with an equal amount of service time, i.e., the sniff period T_{sniff}. The maximum length of a sniff events T_{switch} (the duration for which communication lasts) is calculated by $T_{switch} = T_{sniff}/N_{link}$, where N_{link} is the number of peer nodes. All peer nodes are served in a round robin fashion, and T_{switch} is fixed throughout the rounds. We call this scheme Sniff Round Robin (SRR). Figure 2 and 3 show an example scatternet and the sniff activities under SRR where $T_{sniff} = 16$ slots. Note that the bridge node can act as a master only in one piconet.

With SRR, when the system load is low, some actual sniff events may last less than T_{switch}, and hence, some idle slots may occur. A shorter sniff period may reduce the idle slots. But it causes frequent piconet switching, which results in wasted slots as described above. A better approach would be to have the bridge node switch to a new peer node immediately after the communication terminates with the current peer node. However, it is impossible to determine *a priori* how long the communication between the bridge node and each of its peer nodes would last. In what follows we describe how LSRR reduces the number of idle slots to achieve higher throughput.

A bridge node can operate as a master only in one piconet, but as a slave in many other piconets. Most of the idle slots in inter-piconet switching can be removed by excluding the slaves of the bridge node in its own piconet from the scheduling rounds, but instead, polling them in the idle slots. When the bridge node detects the end of a sniff event, it immediately switches its role from a slave to the master and begins to poll its own slaves according to its intra-piconet polling scheme until the next sniff slot arrives. In this scheme, the bridge node gives a higher priority to its peering masters than to its slaves. Hence it is possible that the bridge node's own piconet never gets a chance to be scheduled, if the entire T_{switch} of each peering master is consumed. This leads to the starvation of the links in the bridge node's own piconet. The starvation can be avoided by imposing an upper limit for T_{switch} to guarantee the minimum

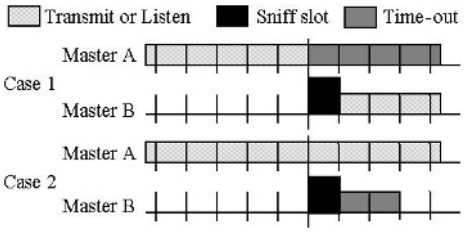


Fig. 1. Wasted slots in piconet switch using the sniff mode

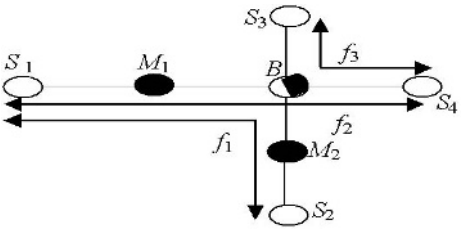


Fig. 2. A Bluetooth scatternet of three piconets

amount of service time for the slaves of the bridge node. Hence, the bridge node is allowed to communicate with each peering master only up to a time limit $T_{limit} < T_{switch}$. The value of T_{limit} is determined adaptively based on the utilization of the slots in the previous sniff event. Based on the assumption that the traffic load is almost evenly distributed among all links, the initial value of T_{limit} is set to $T_{limit} = T_{sniff}/N_{link}$, where N_{link} is the number of the links between the bridge node and its peer nodes. This value decreases by α slots (α>0 is an integer constant) if the previous sniff event terminates by a poll-null sequence, and increases by α slots if the previous sniff event terminates by the limit. Note that when $T_{limit} = T_{switch}$, it is possible that the starvation of the links occurs. Thus, the maximum value that T_{limit} can take is set to T_{switch}– $2N_{link}$. Figure 4 illustrates an example of LSRR applied to a scatternet of three piconets in Fig. 2. LSRR's algorithm is given in Fig. 5.

LSRR is an inter-piconet scheduling scheme that tries to minimize the number of idle slots. LSRR follows the notion of max-min fairness (cf. [10]) in that, the unused slots of the bridge-master nodes are released to the bridge node's intra-piconet communication.

5 Simulation

In order to study the performance of the LSRR scheduling scheme, we modified the ns2-based Blueware simulator [11] to include SRR and LSRR. The simulator has a detailed model of the baseband. The physical layer has a simplified model of radio channels with no errors due to interference. IP is run on top of Bluetooth link layers and fixed routes are used.

We set up the scatternet topology as shown in Fig. 2. The scatternet consists of three piconets inter-connected by a single bridge node B acting as a master in one piconet and a slave in the other two piconets. Bridge node B is connected to two master nodes M_1 and M_2, which have S_1 and S_2 as the other slaves, respectively. B has two slaves S_3 and S_4. This topology was chosen to include three types of traffic flows through the bridge node: (1) traffic flow between two piconets, that are managed by two different master nodes (e.g., f_1), (2) traffic flow

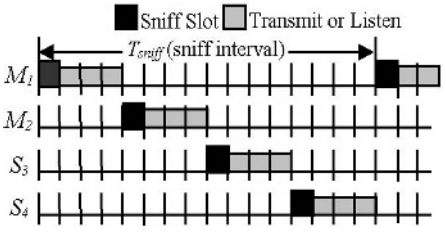


Fig. 3. Sniff Round Robin scheme applied to the scatternet in Fig. 2

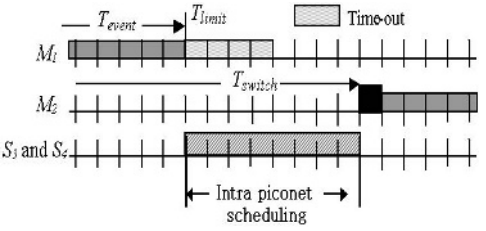


Fig. 4. Limited Sniff Round Robin scheme applied to the scatternet in Fig. 2

For each master-bridge unit, it maintains the following sniff parameter and procedure:
T_{event} : number of comm. slots
N_{inter} : number of connected master and inter-slave
T_{limit_min} : minimum value of T_{limit} (T_{sniff} / N_{link})
T_{limit_max} : maximum value of T_{limit} ($T_{switch} - 2N_{slave}$)

Algorithm 1 ***When bridge unit sends packet to master;***
if ($POLL-NULL$ *sequence occurs*) ***then:***
if ($T_{limit} > T_{limit\_min}$) ***then decrease*** ($T_{limit}$);
Switch to intra-piconet comm.;
else if ($T_{event} = T_{limit}$) ***then:***
if ($T_{limit} < T_{limit\_max}$) ***then increase*** ($T_{limit}$);
Switch to intra-piconet comm.;
else increase (T_{event});

Algorithm 2 ***When bridge unit received packet from inter-slave;***
if ($POLL-NULL$ *sequence occurs*) ***then*** *switch to intra-piconet comm.;*

Algorithm 3 ***When*** N_{inter} ***changes***
$T_{switch} = T_{sniff}/N_{inter}$;
Send LMP_sniff_req to each master and inter-slave;
while:
if (*LMP_sniff_req received*) ***then*** *recalculate* T_{switch};
Resend LMP_sniff_req to each master and inter-slave;
if (*LMP_accepted received from all master*) ***then break;***
Initiate sniff parameter;

Fig. 5. Pseudo-code of LSRR scheme

between the bridge node's own piconet and another piconet (e.g., f_2), and (3) intra-piconet traffic flow within the bridge node's own piconet (e.g., f_3). Given this network set up, we ran our simulation in three different traffic scenarios.

The first scenario involves all three types of flows carrying constant bit rate (CBR) traffic. This scenario was set up to analyze how LSRR adapted to varying

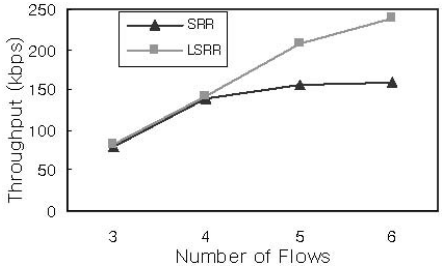


Fig. 6. System Throughput of 64 Kbps H.263 with 10 Kbps CBR flows

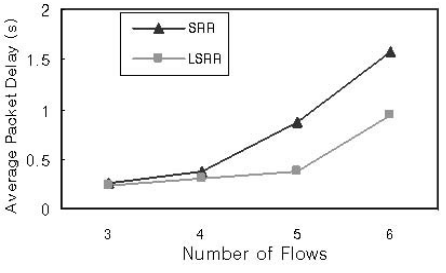


Fig. 7. Average packet delay of 64 Kbps H.263 with 10 Kbps CBR flows

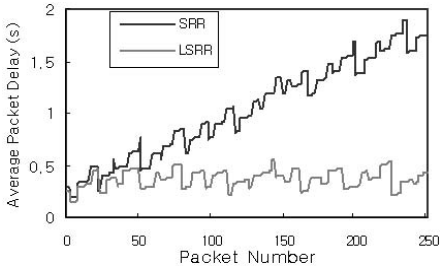


Fig. 8. End-to-end per-packet delays of three 64 Kbps H.263 video streaming flow

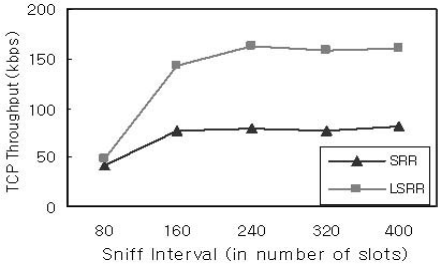


Fig. 9. TCP throughput with a 64 Kbps H.263 flow and a 10 Kbps CBR flow

inter-piconet traffic load (i.e., f_1). The traffic load was increased by the multiples of H.263 video streams of the average bit rate of 64 Kbps. The trace files for the video streams were taken from [12]. The CBR traffic used for f_2 and f_3 were artificially generated at 10 Kbps. The video stream data were delivered in UDP datagrams of 335 bytes, which were segmented into one or more DHx packets by L2CAP based on available slots. The value of T_{sniff} was 240 slots. The simulation was run for 20 seconds.

The next scenario was set up to evaluate the effectiveness of LSRR in reducing idle times in sniff periods. In this set up, a TCP traffic was given to flow f_2. A 64 Kbps H.263 video stream and a 10 Kbps UDP stream were used for f_1 and f_3, respectively. The maximum segment size of the TCP flow was 512 bytes. TCP NewReno version was used for the TCP protocol. TCP throughput was measured as T_{sniff} was varied in the range of 80 to 400 slots. The simulation was run for 100 seconds.

In the last scenario, we analyzed how effectively LSRR adjusted its schedule such that high priority was given to inter-piconet links than intra-piconet links. This is an important characteristic in an application where, for instance, the bridge node (the master) is a laptop computer and the slaves are Blue-

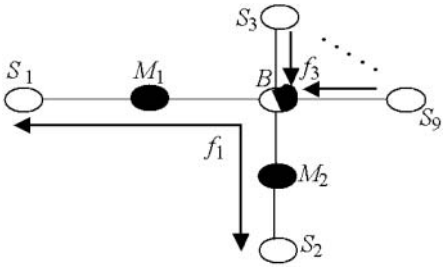


Fig. 10. Scatternet with varying number of intra-piconet links

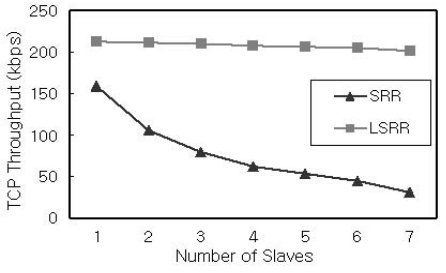


Fig. 11. TCP throughput with 10 Kbps CBR flows

tooth enabled peripheral devices such as a mouse. In order to compare how the throughput of f_1 changed under LSRR and SRR, we increased the number of intra-piconet links from 1 to 7 by adding slave nodes (S_3, ..., S_9) (See Fig. 10). f_1 and f_3 (the flow between the bridge node and its slaves) were given a TCP traffic and 10 Kbps CBR traffic, respectively.

In all three scenarios, LWRR (Limited and Weighted Round Robin) [13] with the values of the limit 6 was used as the intra-piconet scheduling scheme. The value of constant α was 10 slots.

Figure 6 shows the achieved throughput of the CBR traffic in the multiples of H.263 flows. The rate of increase in the throughput of SRR decreased considerably faster than that of LSRR as the number of CBR flows increased to 5 and 6. This implies that the sniff intervals for links (M_1, B) and (B, M_2) were quickly exhausted as its maximum length was fixed. Note that since the bridge node spent about one quarter of its time for each of its links, the maximum throughput achievable was about 180 Kbps. When there were five CBR flows the throughput reached 158 Kbps which included the two 10 Kbps flows f_2 and f_3. On the other hand, the throughput of LSRR continued to increase, since it extended the limit of sniff events for the same links, as the number of CBR flows increases to 5 and 6. Figure 7 compares the average IP packet delays for the two scheduling schemes. As expected, the average delay of SRR increased faster than that of LSRR. This implies that in case of SRR, as the bandwidths allocated to links (M_1, B) and (B, M_2) got almost filled up with the CBR flows, the scheduling delay increased rapidly. Figure 8 shows the variations of end-to-end packet delay of the H.263 streams when there are five CBR flows.

Figure 9 shows simulation results for the second scenario. The TCP throughput of f_2 under SRR and LSRR are plotted against the length of sniff periods. When the length of sniff periods is 80, the difference is about 6.6 Kbps, while it is about 77.9 Kbps when the length of sniff periods is 160. This implies that, a lot of slots were left idle under SRR when the sniff period was long, while the unused slots were allocated to the intra-piconet traffic under LSRR.

Figure 11 shows how the TCP throughput changes as the number of slaves connected to the bridge node increases. One can see that the throughput of SRR decreases faster than LSRR. This is due to the priority scheme of LSRR that allocates communication time to the inter-piconet links first. Only the idle time is shared by the intra-piconet links, while they are given equal amount of time in SRR. The throughput slightly decreases because the initial value T_{limit} becomes smaller as the number of slaves increase.

6 Conclusions

In this paper, we addressed the problem of wasted slots in using sniff mode to switch the bridge unit between different piconets. As we described, having the bridge unit sniff with its peering masters with time limits and communicate with its slaves in the remaining slots can reduce the idle slots. We also described how the sniff time limits are determined adaptively based on the amount of past traffic in each piconet. We presented an efficient scatternet scheduling scheme based on these observations. Simulation results show LSRR scheme outperforms a straightforward scheme, i.e., SRR. We will further improve our scheme such that QoS guarantee is provided and scatternet wide coordination is performed as well.

References

[1] Specification of the Bluetooth System - Core vol.1 v1.1, `www.bluetooth.com` 103, 106
[2] Salonidis, T., Bhagwat, P., Tassiulas, L., LaMaire, R.: Distributed Topology Construction of Bluetooth Personal Area Networks. In Proc. of IEEE INFOCOM'01. USA. (April, 2001) 22–26 104
[3] Basagni, S., Chlamtac, I., Záruba, G. V.: Bluetrees - Scatternet Formation and Routing in Bluetooth-Based Ad Hoc Networks. IEEE ICC'01. (2001) 273–277 104
[4] Miklós, G., Rácz, A., Turányi, Z., Valkó, A., Johansson, A.: Performance Aspects of Bluetooth Scatternet Formation. ACM Mobihoc'00. (Aug, 2000) 147–148 104
[5] Johansson, N., Korner, U., Tassiulas, L.: A distributed scheduling algorithm for a Bluetooth scatternet. In Proc. of ITC' 17. Salvador da Bahia, Brazil. (Sept, 2001) 104
[6] Baatz, S., Frank, M., Kühl, C., Martini, P., Scholz, C.: Adaptive Scatternet Support for Bluetooth using Sniff Mode. In Proc. of the 26th Annual Conference on LCN. USA. (2001) 112–120 105
[7] Johansson, P. G., Kapoor, R., Kazantzidis, M. I., Gerla, M.: Rendezvous Scheduling in Bluetooth Scatternets. In Proc. of ICC'02. (2002) 318–324 105
[8] Rácz, A., Miklos, G., Kubinszky F., Valkó A.: A pseudo random coordinated scheduling algorithm for Bluetooth scatternets. ACM MobiHoc'01. (2001) 193–203 106
[9] Tan, G., Guttag, J.: A Locally Coordinated Scatternet Scheduling Algorithm. In Proc. of 27th Annual Conference on LCN. USA. (2002) 293–303 106
[10] Jaffe, J.: Bottleneck Flow Control. IEEE Transactions on Communications. (July, 1981) Vol. 29(7), 954–962 108

[11] Tan, G.: Blueware: Bluetooth Simulator for ns. MIT Technical Report, MIT-LCS-TR-866. Cambridge, MA. (October, 2002) 108
[12] Fitzek, F., Reisslein, M.: MPEG-4 and H.263 Video Traces for Network Performance Evaluation. TKN Technical Report TKN-00-06, Technical University Berlin, Telecommunications Networks Group. (October, 2000) 110
[13] Capone, A., Kapoor, R., Gerla, M.: Efficient Polling Schemes for Bluetooth Picocells. ICC. (2001) Vol. 7, 1990–1994 111

Efficient Route Discovery for Reactive Routing Protocols with Lazy Topology Exchange and Condition Bearing Route Discovery*

XuanTung Hoang, Soyeon Ahn, Namhoon Kim, and Younghee Lee

Computer Networks Lab. , Information and Communications University
58-4 Hwaam-dong, Yuseong-gu, Daejeon 305-714, Korea
Tel. +82-42-866-6162
{tung_hx,syahn,nhkim,yhlee}@icu.ac.kr

Abstract. We propose LTE and CBRD to reduce route discovery overhead of reactive routing protocols for Mobile Ad hoc Networks (MANETs). LTE proactively distributes topology information of the network by using a lazy update policy. That topology information is used by CBRD to optimize route discoveries. CBRD is a reactive route discovery mechanism which employs topology information provided by LTE to restrict route discovery floods to limited regions containing desired destinations. Our simulation results have shown that LTE and CBRD efficiently reduce route discovery overhead as well as route discovery delay. Also, they improve routing performance of flooding dependent reactive routing protocols like AODV in low- and moderate-traffic networks.

1 Introduction

In Mobile Ad hoc NETwork (MANET), where mobile nodes function as end nodes as well as routers, routing packets to desired destinations is challenging due to topology changes and limited resources such as bandwidth and nodes' battery life. In such an environment, a routing protocol should maintain routing state with a small number of control packets.

Proactive routing protocols such as DSDV [8] exchange routing updates periodically as well as instantly in response to topology changes. Since topology of MANETs can change unpredictably, nodes need to exchange enough routing information within a certain period of time; otherwise data packets will be misrouted. For example, as shown in [9], DSDV can provide acceptable performance in low mobility networks, but it fails to converge when highly increased mobility causes frequent changes in topology. Moreover, proactive routing protocols keep generating many periodic control messages to calculate routes even when there is almost no traffic load. These two disadvantages discourage proactive protocols from being used when mobility is high and the traffic load of networks is low.

* This work was supported by grant No. R01-2003-000-10562-0 from Korea Science and Engineering Foundation.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 114–123, 2004.

Rather than establishing routes proactively, reactive routing protocols [3, 7, 10] set up routes only when nodes need routes to send data. This reduces routing overheads since routes that are not required for sending data will not be set up. Reactive routing protocols establish routes on-demand by using a route discovery scheme in which route request packets (RREQs) are flooded to search desired destinations. Unfortunately, that flood-based route discovery tends to be costly and can severely degrade the performance of routing protocols.

In this paper, we propose Lazy Topology Exchange (LTE) and Condition Bearing Route Discovery (CBRD) to reduce the overhead of the flood-based route discovery in reactive routing protocols. The former, LTE, aims at setting up an approximate topology of the network at each node by proactively and lazily distributing topology information, and the latter, CBRD, is a route discovery heuristic that employs topology information provided by the former to restrict the route discovery floods to a small region around the shortest path to the destination.

The outline of this paper is as follows: In section 2, related works are discussed; section 3 describes our proposed scheme; simulation results are presented in section 4; and finally, we conclude the paper in section 5.

2 Related Work

Since route discoveries base on flooding, one approach for reducing cost of route discoveries is reducing cost of flooding. Works [1, 5, 6], that has been done on improving the effectiveness of the flooding operation in MANETs, can be directly applied to route discovery of reactive routing protocols.

Restricting route discovery floods is also a promising approach to the reduction of cost of route discoveries. Following this approach, a source node floods route request packets just within a limited region that contains its desired destination. This results in a less number of flooded packets. In Query Localization (QL) technique [4] and Location Aided Routing [11], a source node first estimates a requesting zone containing the current location of its destination based on some historical information and then floods route request packets in that location only. All route request packets rebroadcasted outside the requesting zone are suppressed. Although these techniques can effectively reduce overhead of flooding in re-discovering a route that was recently used, they cannot avoid overhead in the first, and most expensive, route discovery.

Different from LAR and QL, FRESH [12], a recent proposal in restricting route discovery floods, can narrow the route discovery area even for the first route discovery attempt. Restricting global floods in FRESH is based on *mobility diffusion* as follows: Due to mobility, nodes may encounter each other, that means they become one-hop neighbors. When encounters happen, nodes record this kind of events in their *encounter tables*. If a source node s needs to find route to a destination d, s looks for intermediate node k that has encountered d more recently than s by flood-based *expanding ring searches*. Node k then looks for another node that has encountered d more recently than itself. This procedure is

repeated until d is found. Although FRESH can reduce flooding overhead even for the first route discovery, routes formed by FRESH tend to run along nodes' trajectories. Hence, routes can be sub-optimal. Also, if mobility is not sufficiently high, reduction in route discovery cost may not be significant.

3 Proposed Scheme

Lazy Topology Exchange (LTE) and Condition Bearing Route Discovery (CBRD) introduced in this paper follow the approach of restricting floods to solve the problematic route discovery of reactive routing protocols. Different from previous works, this scheme bases on distance information that is proactively distributed among nodes by using LTE. The update policy that LTE uses is periodic and lazy in order to set up an approximate distance-based representation of the network topology at each node. CBRD then uses this topology representation to narrow flood areas resulting cheaper route discoveries. The total overhead for building approximate topology representation and for discovering routes with CBRD, according to our simulation with 100 mobile nodes, is about from 20% to 25% lower than the overhead of pure flood-based route discovery.

Using CBRD, a source node who wishes to find a destination locally floods a route request packet within μ hops to find any intermediate node that has better distance information to the destination than itself. This node called anchor node will find the next anchor node toward the destination. This procedure iterates until the destination is found. Details of LTE and CBRD will be discussed in the following subsections.

3.1 Lazy Topology Exchange (LTE)

Only bidirectional ad hoc networks are considered in our scheme. Two nodes in the networks can communicate with each other if they are within their transmission range R_{Tr}. Otherwise, none of them can receive packets from the other.

LTE, in fact, is a variation of DSDV [8] but with a lazy update policy and a multi-path extension. Similar to DSDV, in LTE, distance information between a source-destination pair is a combination of cost information, represented by the number of hops between nodes, and the freshness of that cost information, represented by a destination-generated sequence number. Each node j stores and maintains distance information of multiple paths from node j to all other nodes via different neighboring nodes of j in a distance table DT_j. The structure of the distance table DT_j is as follows:

1. A set N_j of neighboring nodes of node j.
2. Corresponding to each tuple (d, k), where d is a destination node known by node j, and k is a neighbor of j, the following information is maintained:
   - The most recent update of the shortest distance (in hops) from node j to node d via neighbor k, denoted as $D_j^k(d)$

- The corresponding sequence number $S_j^k(d)$ of $D_j^k(d)$, originated from node d and received by node j

By examining the above $D_j^k(d)$ and $S_j^k(d)$, for each destination node d, j selects a neighbor k corresponding to the best distance to d, called distance entry to d. The distance entry to node d in DT_j, denoted by $DE_j(d) = (d, S_{jd}, D_{jd})$, is that which has the latest sequence number S_{jd} and the fewest number of hops D_{jd}. For convenient explanation, $DE_j(d).D_{jd}$ and $DE_j(d).S_{jd}$ are used to refer to D_{jd} and S_{jd} of $DE_j(d)$, respectively.

Topological information is exchanged among nodes in periodical update messages which contain one or more distance entries in nodes' distance tables. Each node j in the network keeps refreshing the distance entry to itself, $DE_j(j) = (j, S_{jj}, D_{jj} = 0)$, by monotonically increasing the sequence number S_{jj} and sending $DE_j(j)$ in every update message to advertise the current position of j to j's neighbors. Other distance entries can also be sent in an update message together with $DE_j(j)$ if they have been changed but have not been advertised. Update messages are sent periodically after an update interval τ. An update message will be fragmented into several smaller update messages if it does not fit in a single Maximum Transmission Unit (MTU). The distance table is updated according to the following rules:

- When node j receives a distance entry $DE_k(d)$ in an update message from node k, node j updates $D_j^k(d)$ and $S_j^k(d)$ in its distance table if $DE_k(d)$ is a "better distance" to d via k than that being maintained at j, i.e. $DE_k(d)$ contains a higher sequence number than $S_j^k(d)$, or $DE_k(d)$ represents a equally fresh but fewer number of hops than $S_j^k(d)$ and $D_j^k(d)$.
- When node j discovers a new neighboring node k, j sends its distance table to k.
- When node j detects that the link to neighbor k is broken, j removes $D_j^k(d)$ and $S_j^k(d)$ from its distance table.

Any change in $D_j^k(d)$ causes an immediate recalculation of $DE_j(d)$. If a $DE_j(d)$ is changed, it will be marked for sending in the next update message. Thus, the update policy of LTE is periodic without triggered updates.

The update operations of LTE described above may not help node route data packets to destinations since global knowledge of network topology at each node is not up-to-date when mobility causes link states between nodes to change. However, that global knowledge is useful for reducing route discovery overhead if CBRD is used to set up routes.

3.2 Condition Bearing Route Discovery

CBRD is proposed to utilize the information that is provided by LTE to optimize the flood-based route discovery. Any route setup mechanism, such as the dynamically modifying routing table in AODV [3] or source routing in DSR [7], can be integrated with CBRD to establish routes. Like flood-based route discovery

procedure, CBRD is a query-response mechanism, i.e., source node broadcasts Route Request packets (RREQ) to search for its destination; intermediate nodes that receive a route request packet will forward this packet further; and when the destination is found, a route reply packet is sent back to the source node. However, a RREQ in CBRD implicitly carries some conditions, and only intermediate nodes that meet the conditions inside a received RREQ will forward the packet further. Thus, the number of nodes that need to forward RREQs is restricted.
CBRD works as follows:

- The route discovery process starts when a source node j broadcasts a route request packet, RREQ, to find a destination d. The RREQ packet contains the distance entry $DE_j(d)$. This RREQ is a broadcast message which is limited within μ hops by some counter value such as the TTL field in the IP packet header [2].
- When an intermediate node k receives an RREQ for a destination d, if the distance entry at k for destination d, $DE_k(d)$, is "better" than that contained in the RREQ, node k drops the RREQ and initiates a new broadcast RREQ within μ hops for destination d with its distance entry $DE_k(d)$ attached into the RREQ. Otherwise, k rebroadcasts or drops the received RREQ by examining the counter value or the TTL field of the RREQ packets. The criterion for determining a "better" distance entry $DE_j(d)$ is based on $DE_j(d).S_{jd}$ and $DE_j(d).D_{jd}$. More specifically, $DE_k(d)$ is better than $DE_j(d)$ if: $(DE_k(d).S_{kd} > DE_j(d).S_{jd})$ **OR** $((DE_k(d).S_{kd} = DE_j(d).S_{jd})$ **and** $(DE_k(d).D_{kd} < DE_j(d).D_{jd}))$

The idea of CBRD is that when node j needs a route to a destination, j "locally floods" within μ-hop to find nodes, called *anchor nodes*, that "know" the destination better than j. A found anchor node k then searches for the next anchor node toward the destination. This search procedure, called *anchor search*, then iterates until the desired destination is reached resulting in a successful route discovery. If nodes choose sufficiently large values for μ, anchor searches will succeed with high probability, and finally, the route discovery process will end with a route set up to the destination. Since an anchor node is a node that has better distance information to the destination than its previous anchor node[1], searching an anchor node can be considered as discovering the next node toward the destination.

μ, which is called *topological estimation radius*, is a local parameter that node j uses to estimate the position of the next anchor node. The parameter μ used by node j should take the probability of success π of the anchor search into consideration. In fact, π is a function of the topological estimation radius μ, the update interval τ in LTE, and node density. On the one hand, if μ is not sufficiently large to compensate the "laziness" of LTE, which is parameterized by τ, the success probability of route discovery will be low. On the other hand, a too large value of μ leads to unnecessary route request messages injected into

[1] The source node is considered as the first anchor node.

the network. Thus, a suitable configuration of τ and μ is a significant factor for an optimal performance of LTE and CBRD.

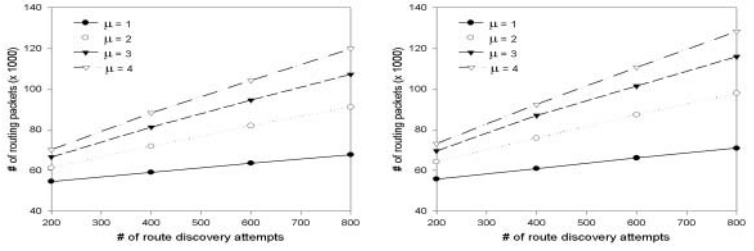
4 Simulation Studies

We have performed several simulations with ns2 (version 2.1b9) to analyze the route discovery operation with LTE and CBRD. We also evaluated the effects of LTE and CBRD on routing performance. Our simulation scenario is arranged as follows: 100 nodes move around in a rectangular region of $1250m \times 1250m$. Random way-point model [13] is used to simulate node mobility. In this mobility model, a node randomly chooses a destination and moves to the destination with a random speed chosen uniformly between 0 and *maxspeed* value. Once a node reaches its destination, it stays there for a pre-defined *pause time* before moving to a new random destination. Following simulations in previous works [9, 14], we set *maxspeed* to 20 meters per second, and vary *pause time* to adjust node mobility. The wireless transmission model is parameterized to be similar to Lucent's WaveLAN interface which has a nominal trans-mission range of 250 meters and a shared radio medium using IEEE 802.11 standard. Since current hardware does not support link layer feedback, we decide to use Hello messages for monitoring link connectivity in both AODV and AODV+&CBRD. The Hello message interval in our simulations is set to one second.

4.1 Route Discovery Analysis

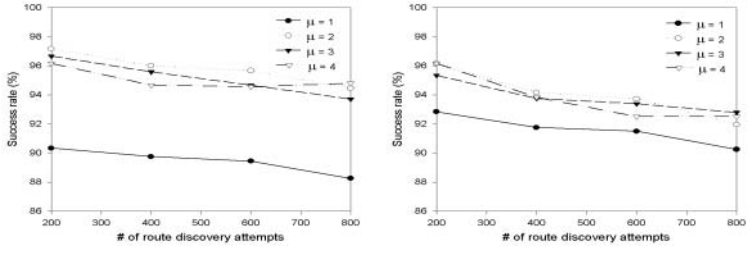
Our first set of simulations is a comparison of route discovery performance between AODV, a flooding dependent reactive routing protocol, and AODV with LTE and CBRD, which is labeled as AODV+LTE&CBRD.

In order to figure out a suitable set of configuration τ and μ parameters for LTE and CBRD, we vary the topological estimation radius μ while the update interval τ of LTE is fixed to 5 seconds. Simulation experiments on route discovery performance of AODV+LTE&CBRD with different values of topological estimation radius μ are conducted with this scenario to figure out a suitable value of topological estimation radius μ. Results shown in Fig. 1 and Fig. 2 help us conclude that $\mu = 2$ is an optimal value in the scenario described above since with $\mu = 2$, LTE&CBRD provides the lowest number of routing packets and a competitive packet delivery fraction. In this comparative simulation with AODV, since we are interested in route discovery of compared protocols, traffic in our simulated network is comprised of hundreds of short conversations occurring in 200 seconds of simulation time. The traffic of conversations is Constant Bit Rate (CBR) between randomly selected source-destination pairs. Each conversation sends a random number of 64-byte packets uniformly distributed between 5 and 10 packets with rate 10 packets per second. The number of conversations is changed to adjust the condition of the traffic load in our simulations. Comparisons in routing overhead and packet delivery fraction under different pause time values in Fig. 3 show that LTE&CBRD enhancement for AODV roughly reduces



(a) Low mobility (pause time = 100s) (b) High mobility (pause time = 0s)

Fig. 1. Total number of route request packets and update packets



(a) Low mobility (pause time = 100s) (b) High mobility (pause time = 0s)

Fig. 2. Success rate of route discoveries

20% to 25% the number of routing packets while it does not hurt the packet delivery fraction. The reduction in routing overhead also indirectly improves route discovery latency as shown in Fig. 4. Since conversations in this experiment are very short, the dominant reason of packet delay is route setup delay. In all observations, AODV+LTE&CBRD results in a smaller mean value of packet delay, usually a half of that of AODV. Also, packet delay in AODV+LTE&CBRD is rather stable. It implies that AODV+LTE&CBRD provides faster and more stable route setups. To explain this simulation result, consider two source nodes simultaneously starting route discovery in the network. In the case of AODV, all nodes in the MANETs will receive RREQs from both the two source nodes. In other words, the two route discovery areas completely overlap and cover the whole network. In the case of AODV+LTE&CBRD, a limited overlapped region exists between the two route discovery areas as the result of route discovery localization. It mitigates mutual interference between concurrent route discoveries. Hence, the network becomes more stable, and route discovery latency is improved.

4.2 Routing Performance

To evaluate effects of LTE and CBRD on routing performance, we run a set of simulations with multiple long-lived traffic streams: the number of streams is 20;

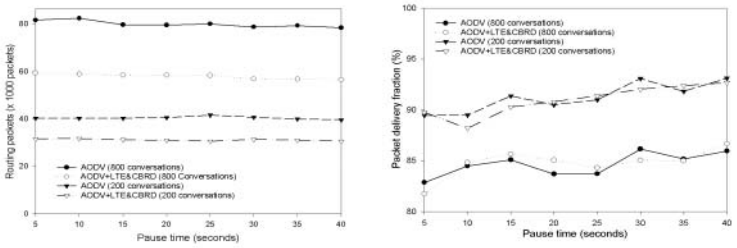


Fig. 3. Route discovery analysis

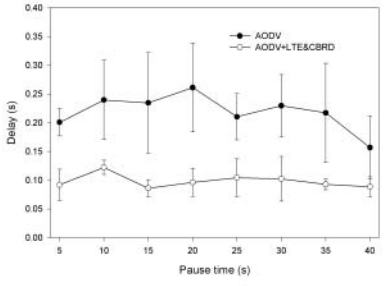
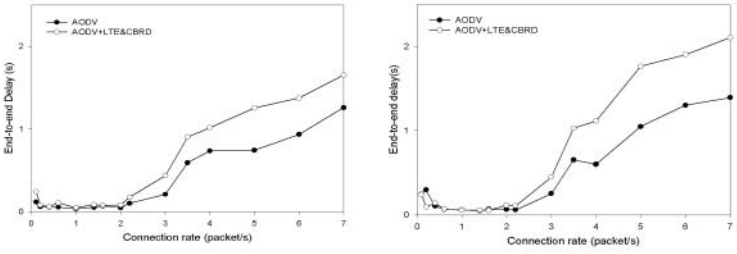


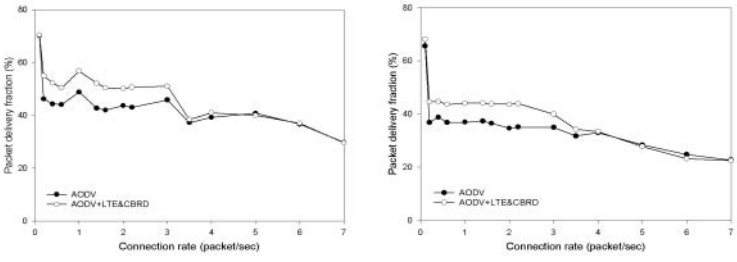
Fig. 4. Route discovery delay

traffic type is Constant Bit Rate (CBR) with 512-byte packet size; each stream is between a randomly chosen pair of source and destination; simulation time is set to 500 seconds; simulation area, node population and moving pattern are the same as those in the route analysis experiment presented above. Connection rates are varied to adjust the traffic load. Fig. 5 shows comparisons in packet delay between AODV+LTE&CBRD and AODV. When the connection rate is greater than 3 packets per second, equivalent to about 150 Kbps total offered load, packet delay in both the two protocols dramatically increases implying that network congestion happens. We also notice that AODV+LTE&CBRD produces larger packet delay than AODV does. Since LTE and CBRD localize route discovery area around the shortest path instead of carrying a global flooding like AODV, if congestion happens to be in a route discovery area, it will prevent route request messages from reaching the desired destination. Thus, the source, in case of AODV+LTE&CBRD, has to re-discover a route while the source, in case of AODV, is able to find a route around the congested area in the first trial. This explains why AODV+LTE&CBRD suffers from higher delay when the network is congested. However, comparison on packet delivery fraction in Fig. 6 shows that, in terms of the packet delivery fraction, AODV+LTE&CBRD is still similar to AODV under network congestion while AODV+LTE&CBRD outperforms AODV when network is not congested. Thus, we can conclude that



(a) pause time = 100s (b) pause time = 0s

Fig. 5. Routing analysis - End-to-end delay



pause time = 100s pause time = 0s

Fig. 6. Routing analysis - Packet delivery fraction

LTE and CBRD can improve routing performance in the low and moderate traffic networks.

5 Conclusion and Future Works

We have presented LTE and CBRD for reactive routing protocols to restrict route request floods. Our scheme is beneficial even to the first route discovery attempt. Based on topology information that is formed by lazily exchanging updates between nodes, our scheme replaces a global flood by successive anchor searches to narrow route request area. Our simulation has shown that with LTE and CBRD, AODV can reduce overall routing overhead and mitigate the effects of the flood-based route discovery. Tradeoffs for these advantages are periodic control packets and degradation in reliability of route discovery. However, the proactive overhead is low and spreads out over time instead of occurring in a short time like flooding. Also unreliability in route discovery can be alleviated by an appropriate configuration of parameters and a retry strategy. Our simulations show that LTE and CBRD have positive effects on route discovery performance, and they can improve reactive routing protocols on low- and moderate-loaded networks. Our future direction will focus on an extended version of LTE and CBRD for multicast routing, and it may result in some considerable results.

References

[1] M. Gerla, T. J. Kwon and G. Pei "On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering," In Proceedings of IEEE WCNC 2000, Chicago, IL, 2000. 115

[2] DAPRA Internet program, "Internet Protocol Specification," IETF RFC 791, Sep. 1981. 118

[3] Charles E.Perkins and Elizabeth M. Royer, "Ad hoc On-demand Distance Vector Routing," In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, Feb. 1999. 115, 117

[4] R. Castaneda and S. Das, "Query Localization Techniques for On-demand Routing Protocols in Ad hoc Networks," In Proceedings of the ACM International Symposium on Mobile Ad hoc Networking and Computing (MOBIHOC), 1999. 115

[5] L.Li, J. Halpern, and Z. Haas, "Gossip-based Ad hoc routing," In Proceedings of the IEEE conference on Computer Communications (INFOCOM), New York, NY, June 2002. 115

[6] B. Williams and T. Camp, "Comparison of Broadcasting techniques for Mobile Ad hoc Networks," In Proceedings of the ACM International Symposium on Mobile Ad hoc Networking and Computing (MOBIHOC), 2002. 115

[7] David B Johnson and David A Maltz, "Dynamic source routing in ad hoc wireless networks," In Imielinski and Korth, editors, Mobile Computing, volume 353. Kluwer Academic Publishers, 1996. 115, 117

[8] Charles E. Perkins, Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," SIGCOMM'94 Computer Communication Review 24(4):234-244, Oct. 1994. 114, 116

[9] D. A. Maltz, J. Broch, and D.Johnson, "A performance comparison of Multi-hop Wireless Ad hoc Network routing protocols," In Proceedings of ACM MOBICOM, October 1998. 114, 119

[10] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," IETF, Internet draft, 1997. 115

[11] Young-Bae Ko and Nitin H. Vaidya, "Location Aided Routing (LAR) in mobile ad hoc networks," ACM/Baltzer Wireless Networks (WINET), Vol.6-4, 2000. 115

[12] Henri Dubois-Ferriere, Matthias Grossglauser and Martin Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks" In Proceedings of MobiHoc '03, 2003. 115

[13] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002. 119

[14] P. Johansson, T.Larsson, N. Hedman, B. Mielczarek, and M.Dagermark, ôScenario Based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks,ö in Proceedings of ACM Mobicom'99, Seattle, Washington, August 1999. 119

Chumcast in Two-Tier Networks

Seung-Seok Kang and Matt W. Mutka

Department of Computer Science and Engineering, Michigan State University
East Lansing, MI 48824, USA
{sskang,mutka}@cse.msu.edu

Abstract. This paper provides a scheme to reduce the cost to distribute multimedia content to a set of nearby mobile peers, which we call *chums*. One peer, called the *proxy*, downloads multimedia content via a telecommunication link, and distributes it (which we call *chumcast*) to the *ad hoc* network formed from the set of nearby peers. Each peer in the *ad hoc* network takes turns serving as a proxy. Every peer is associated with a server that resides in the Internet. The server for the proxy, called the active server, manages peer information, schedules the next proxy, selects a set of rebroadcasting peers, and detects partitioning of the *ad hoc* network. With support from the servers in the Internet, peers receive benefits of reduced telecommunication bandwidth, computation power, as well as several security features. Simulation results show that 80% of the telecommunication cost is saved with as few as six peers.

1 Introduction

In crowded areas such as airports, subway trains, stadiums, or bumper-to-bumper traffic, people may download multimedia data from the Internet, such as on-air TV shows, sports, and traffic information. Fig. 1(a) illustrates the case when many mobile devices connect to the Internet via their wireless telecommunication link to their ISPs, and then access their favorite services. Some popular multimedia data may be downloaded at the same time by several different people located within the same vicinity. Since it is expected that the cost to download multimedia data is a function of the amount of data downloaded, the costs of the telecommunication connections to access the Internet may be reduced by sharing the Internet connections among users located in a near vicinity. We propose in this paper a low-cost scheme for delivering multimedia data. Mobile devices form an *ad hoc* network and one of the mobile devices, called the *proxy*, connects to the Internet with a metered 3G connection in order to download data, and broadcasts the data to the nearby devices with an unmetered *ad hoc* connection. This may be possible if each mobile device has a 3G network interface for wireless wide area network (WWAN) access, and a wireless local area network interface (WLAN), such as 802.11 or Bluetooth, to form an *ad hoc* network with users in the nearby region. Global wireless networks promise to provide 3G service for voice and data with the transmission speed of up to 2.05Mbps [1], for a fee based on the amount of data traffic. Wireless LANs such as 802.11 or Bluetooth [2] allow the formation of *ad hoc* networks and to provide unmetered bandwidth.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 124–133, 2004.

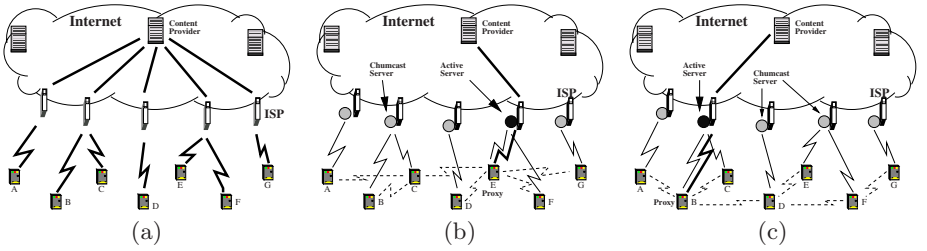


Fig. 1. (a) Mobile devices accessing the Internet through individual ISP connections. (b) A single peer incurs telecommunication charges. The connection is shared by all peers. (c) The proxy migrates to another peer

Our concept for sharing access to the wireless Internet for multimedia data is called CHUM (Cooperating ad Hoc networking to sUpport Messaging). CHUM has previously been described as a means to support instant messaging [3]. It exploits ideas that are often described as *peer-to-peer* computing [4]. The contribution in this paper describes the mechanism for providing the sharing scheme of multimedia streaming among peers. The data delivery mechanism in the CHUM network uses an inherent wireless broadcast scheme, but it is different from known wireless broadcast or multicast schemes in that only the cooperating members of the CHUM network will be allowed to receive the multimedia data. Cooperating members take turns to serve as the proxy for the CHUM network, which is necessary to make 3G network access cost effective.

To illustrate the operation of CHUM, Fig. 1(b) shows that the devices downloading the same content construct a CHUM network and one of them, say E, becomes a *proxy*. As the proxy E receives data from its WWAN connection, it broadcasts the data to its neighbor peers with its WLAN. Some of the participating peers, for example C, may rebroadcast the packets for other peers that are out of transmission range of the proxy. Each peer has its associated *chumcast server* that may be located somewhere on the Internet such as at an Internet Service Provider (ISP), Content Provider (CP), or somewhere else. Each associated server collects neighbor information from its associated peer and reports it to the *active server*, which is the chumcast server of the proxy. The active server maintains membership information of a CHUM network and the global topology of the logical CHUM network, which allows the active server to schedule the next proxy and to compute the minimum rebroadcasting set of peers.

After a proxy serves for a given amount of time, the active server schedules another peer to serve as a proxy. The active server delivers all membership and topology information to the chumcast server whose associated peer is appointed as the next proxy. The chumcast server becomes the new active server and manages the CHUM network. Fig. 1(c) shows that peer B becomes the next proxy. The active server finds peer D and F as the minimum rebroadcasting set, and reports the result to the associated chumcast servers of peer D and F. The associated servers direct their peers to rebroadcast when data packets

are received from the proxy or from one of its neighbor peers. In Fig. 1, the thick connection line indicates that the multimedia data packets are downloaded from a CP. The thin line of the connection implies intermittent control packet exchanges between chumcast servers and their associated peers. The dashed line shows the packet exchanges over the *ad hoc* network.

This paper describes the functionality of CHUM networks and addresses these challenges. Some assumptions are presented in section 2. CHUM network formation is explained in section 3 and the network management is expressed in section 4. Section 5 displays the simulation results of the cost savings with CHUM networks and section 6 draws conclusion.

2 Assumptions

A CHUM network makes a few assumptions of the mobile devices and of the networking services available to the devices. The following list provides a set of assumptions that will be used for the remainder of this paper.

- The peers have networking interfaces that enable the formation of an *ad hoc* network even if there is not a wide-area Internet connection available at a given moment.
- The devices are equipped with an interface that enables a telecommunication connection to an ISP.
- Participating devices within a group are expected to contribute to the group by providing Internet connectivity, and serving as a proxy.
- Users may join and leave the group at any time.
- All mobile devices share the same wireless internal communication channel. They have omni-directional antennas and the same transmission range.

3 CHUM Network Formation

The first step for a mobile device to participate in a CHUM network is to search for an existing CHUM network nearby. If the device fails to find a wanted service beacon from a CHUM network, it connects to a chumcast server via its ISP and downloads its favorite content from the content provider (CP). The chumcast server generates a group ID (GID) for the CHUM network, a network ID (NID) to distinguish between several CHUM networks by the server, a unique peer ID (PID) for the device, and the symmetric group key for secure data packet exchange. The server delivers the group information, the NID, and the PID to its associated device. The device stores the information and periodically transmits a service beacon that contains a description of the CHUM network including the URL of the CP, a summary of the downloading multimedia content, and the beacon sender's PID.

Suppose another mobile device looks for a service beacon and finds a CHUM network that downloads interesting content from the Internet. The device constructs a JOIN packet and broadcasts it to nearby peers in the CHUM network.

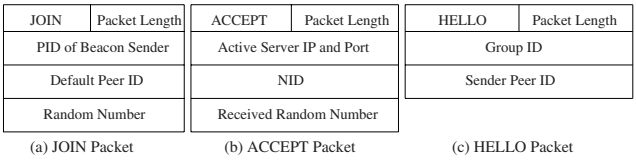


Fig. 2. JOIN, ACCEPT, and HELLO Packet Format

The format of the packet is shown in Fig. 2(a). Since the device does not have a unique PID in the CHUM network, it uses a predefined default PID. In addition, the device generates a random number that distinguishes it from other joining devices. Each peer in a CHUM network sends a service beacon periodically. Any peer receiving the JOIN packet compares the PID of the beacon sender in the packet with its own, and drops it if unmatched. If both PIDs are identical, the peer replies with an ACCEPT packet displayed in Fig. 2(b). The packet contains the transport information of the active chumcast server on the Internet as well as the received random number from the JOIN packet. When the device that transmitted the JOIN packet receives an ACCEPT packet with the same random number, the device becomes a peer of the CHUM network. After examining the packet, the device contacts its ISP in order to communicate with a chumcast server on the Internet.

The device receiving the ACCEPT packet informs its associated chumcast server about the NID and the transport information of the active server. The associated server registers the presence of the new peer to the active server with the NID. As a response, the active server provides the group information of its CHUM network to the associated server including the GID of the network, a symmetric group key for data packet decryption, and the PID of the joining peer, which is unique within the CHUM network. The PID will be used as a network address of the newly joined peer in the *ad hoc* CHUM network. The associated server delivers the CHUM information including the GID, the symmetric key, and the PID to its associated peer. The peer configures with the PID as a network address in the wireless CHUM network, and stores the GID and the group key for data packet decryption. Device mobility does not produce any overhead during the CHUM network formation, because the initial proxy resumes broadcasting downloaded content after the active server collects neighborhood information. The peer movement during the formation does not matter, but it is important to collect the location of peers right before the proxy resumes content transmission.

4 CHUM Network Management

4.1 Membership Management

When a peer joins a CHUM network, described in the previous section, the active server updates the membership information of all peers in the network. The

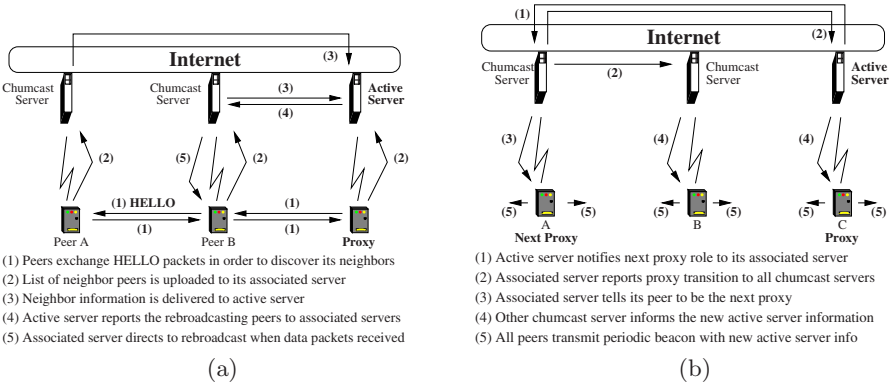


Fig. 3. (a) The sequence of processing neighbor information. (b) The sequence of proxy role transition

peer information is used for scheduling the next proxy, selecting the minimum rebroadcasting set of peers, and detecting a network partition. The membership may change when a peer joins or leaves the CHUM network. The peer that leaves may explicitly inform its associated chumcast server. Nevertheless, some peers may disappear without any notice. The impolite disappearance may be detected by its associated server from the periodic neighborhood information from its peer. When the associated server detects the missing periodic information, it promptly reports the fact to the active server so that it updates the membership information and recomputes the rebroadcasting set as well as any network partition. Moreover, the active server generates a new GID and group key, and passes them to all associated servers, except the one whose associated peer that has left the network. It is because the missing peer may become a freerider using the group key. The associated servers deliver the new group information to their peers. Upon receiving the information, each peer updates the GID and saves the group key. The HELLO packet that follows includes a new GID in order to exclude the left peer from the CHUM network.

In order for the active server to perform its tasks, it requires neighborhood information of peers in the CHUM network. The neighborhood information should be collected from each peer in the network. In order to discover one-hop neighbors, each peer constructs a HELLO packet shown in Fig. 2(c). The neighbors of a peer are the peers from which HELLO packets are received. When a peer receives this packet, it examines the GID in the packet. Only the peers with the same GID will process HELLO packets as well as keep the neighborhood information. Fig. 3(a) presents how the neighbor information is collected and used in a CHUM network. Periodically, each participating peer in a CHUM network broadcasts a HELLO packet with its PID and GID. After a given time, a peer is able to discover its neighbors and to generate a list of its neighbor peers that

sent HELLO packets. The peer, then, uploads the list to its associated server, which delivers the list to the active server.

The active server may create a global topology of its CHUM network from synthesizing the partial neighbor information from each peer. Whenever the active server receives new neighborhood information, it updates the global topology of the network. The global topology is essential to compute the minimum rebroadcasting set of peers and the efficient partition detection of the CHUM network. The active server maintains a boolean connectivity matrix in which an element of (i, j) is set to true if i^{th} peer is a neighbor of j^{th} peer. The matrix represents the global topology of the CHUM network with one-hop connectivity.

4.2 Selection of Rebroadcasting Peers

The main job of the proxy is to contact its favorite content provider on the Internet, to download the multimedia data, and to broadcast them to all peers in its CHUM network. While the proxy downloads the content, it encrypts the content with the symmetric group key and broadcasts them to its neighbors. The peers that are far from the proxy may not receive the content unless some peers in the *ad hoc* network rebroadcast them. Much research have proposed distributed algorithms for selecting the rebroadcasting neighbors while using local neighbor information [5, 6, 7, 8]. The research proposes two-hop neighbor knowledge to minimize the number of rebroadcasting peers. Because wireless broadcasting may cause the duplication of packets and the contention of the wireless medium, it is important to select carefully the set of minimum rebroadcasting peers in a CHUM network.

The problem of selecting the minimum number of rebroadcasting set is very similar to the MCDS (Minimum Connected Dominating Set) problem. The authors in [5] mentioned that the minimum rebroadcasting set problem is harder than the MCDS problem, which has been proved to be NP-complete. Several wireless broadcasting algorithms [5, 6, 7, 8] have been proposed for the approximation of MCDS to compute the backbone wireless peers or the rebroadcasting peers using local neighborhood information. The active server in a CHUM network utilizes the global topology of the CHUM wireless network, and produces the minimum set of rebroadcasting peers. The best known algorithm for the MCDS problem is the Berman's algorithm [9]. The algorithm generates a set of connected nodes and then merges the set into one connected dominating set. In a CHUM network, the partial neighborhood information from each associated server provides a hint to generate a mesh of connected peers. The active server merges the connected peers into one connected dominating set of peers, which includes the proxy in the wireless CHUM network. The active server notifies the associated server of the peers in the set, which is shown in step (4) of Fig. 3(a). Each responsible associated server directs its peer(s) to rebroadcast packets when it receives data packets from its neighbor.

4.3 Proxy Scheduling

One of the attractive advantages of the CHUM network is to save telecommunication cost for downloading multimedia data from the Internet. This benefit comes from the sharing of proxy role with other participating peers in the network. Only one proxy pays for the telecommunication connection at the moment, and other peers share the connection with no charge. It is desirable to evenly share the total cost among all participating peers, but it is difficult to share the cost evenly because of the dynamic membership change of peers. In spite of the dynamicity, the proxy scheduling should be as fair as possible.

The active server performs the proxy scheduling because it maintains the peer membership information. When the proxy serves a CHUM network for a fixed amount of time and there is at least one peer in the network, the proxy role will be moved to another peer. For the CHUM network in this paper, the active server maintains a circular scheduling list of peer members and performs a round robin scheduling. Other methods for fair sharing of the proxy role are possible, but we do not discuss them here, but rather refer to [10]. When a new peer is introduced, the peer will be placed right after the proxy in the circular scheduling list. The reason is because a new member should serve soon as a proxy to ensure that they appropriately participate before benefiting from the service of other peers.

Fig. 3(b) displays the steps for the transition of a proxy role to another peer and the action that follows. The current active server sends a notice packet to the associated server of the next scheduled proxy when the current proxy has finished serving a given amount of time. The packet contains the current peer membership information and the global topology of the wireless CHUM network. Upon receiving the information, the associated server notifies all other chumcast servers by transmitting its transport information such as its IP and port number. In addition, the would-be active server determines the new GID and group key for the new proxy, and distributes the group information to all other chumcast servers. The associated server passes the new group information and directs its peer to be the next proxy. Then, the associated server itself plays a role of the active server. The other chumcast servers inform their associated peers about the transport information of the new active server as well as the new group information. Any peer receiving the new IP and port number updates the active server information and uses the new values for its periodic service beacon. The NID will not be changed. The newly selected proxy includes a new GID in any packet transmitted and encrypts all data packets with a new group key.

4.4 Network Partition Management

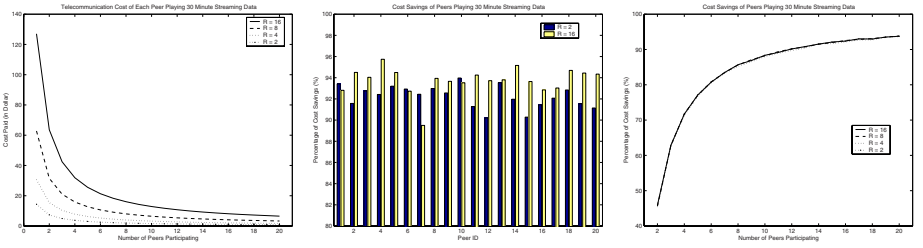
It is possible that a CHUM network is partitioned into two or more subnetworks because some forwarding peers between the proxy and other peers move to another place, and data packets from the proxy cannot be delivered to the peers in the partitioned subnetwork(s). The isolated peers may encounter gaps in their

multimedia data unless they have enough buffered data before the reestablishment of the connection with the proxy. It is not always expected that some peers may fill the broken connection and become a bridge between the partitioned network and the proxy. Rather, it is reasonable for the active server to elect a new proxy among peers in the partitioned network and to resume service within the partitioned network.

The active server has responsibility to detect a partition in a CHUM network because it maintains the global topology of a CHUM network. The server may easily detect a network partition by applying a transitive closure operation on the peer connectivity boolean matrix maintained by the active server. If the proxy may reach all peers in the resultant transitive closure connectivity matrix, there is no partition in the network. If, however, there are some peers that the proxy cannot access, the wireless CHUM network is partitioned. The active server selects the peer(s) in the partitioned network and notifies their associated servers about the isolation from the proxy. In addition, the active server assigns one peer as a proxy in the partitioned network. The associated server of the selected proxy generates new NID, GID and group key, and distributes them to other associated servers whose peers are in the partitioned network. The active server of the original CHUM network deletes the peer information of the partitioned network from both the membership information and the neighbor connectivity information. The partitioned network may resume its operation as the group information is shared by all peers, and the newly elected proxy may start downloading the same content from the Internet.

5 Cost Savings in a CHUM Network

This section shows the amount of cost saving as the network size (the number of peers in the network) increases, and the relationship between the number of peers and the amount of cost saving. The simulation of the CHUM network involves up to 20 peers that download multimedia data from the Internet for 30 minutes. According to [11], each frame size ranges from 1K byte to 30K byte depending on the video contents and their variations. Because CHUM network traffic is targeted to much smaller LCD screens such as PDAs and mobile phones, the simulation generates frames sized from 512 byte to 4K byte. The playback rate varies from 2 frames per second to 16 frames per second. Each peer serves as a proxy for 30 seconds and the scheduled next proxy takes the role. When a peer joins the network, it is assigned as a next scheduled proxy. The round robin scheduling is used throughout the simulation. The simulation runs with the neighborhood timeout (T_N) value of 10 seconds. That is, for every 10 seconds, each peer uploads the neighbor information to its associated chumcast server. All participating peers are honest and their data timers will not fire. This simulation adopts 512 byte per 0.1 cent, which is the rate of the CDMA2000 1xEV-DO 3G service for streaming data in Korea. Both the uploaded neighbor information and the downloaded group information are packed in the packet size of 512 byte. Fig. 4(a) shows the telecommunication cost for 30 minute streaming data



(a) (b) (c)

Fig. 4. (a) Cost of Each Peer Playing 30 min Streaming Data (b) Cost Savings of Each Peer with 20 Participating Peers (c) Average Cost Savings of Peers with Varying Number of Peers

downloaded from a live TV content provider with varying number of peers in a CHUM network. As the playback rate R increases, the quality of the video may increase. However, a single user alone may not afford to enjoy the streaming data because of the high telecommunication cost paid. As the number of peers increases in a CHUM network, the cost paid by each peer drops dramatically.

Fig. 4(b) displays the percentage of cost savings when 20 peers participate in a CHUM network. One peer forms a CHUM network at zero second and the subsequent peer joins after the interval from the previous join event, which is exponentially distributed with an average of 20 seconds. After 25 minutes, peers start leaving at a time exponentially distributed with an average of 15 seconds from the previous leave. Fig. 4(b) illustrates the special case of cost saving with 20 peers. The 20^{th} peer joins at the 344 second and the 19th peer leaves at the 1785 seconds. At least one peer stays throughout the simulation. With 20 peers, the cost saving varies from 90% to 93% when R =2 and from 89% to 96% when R = 16.

Fig. 4(c) averages the percentage of the cost saving with a different number of cooperating peers from 2 to 20. With the network size of 2, the cost saving is not 50% because of the management packet transmissions between peers and their associated chumcast servers. As the number of peers increase, the percentage of the cost saving increases as well. The percentage reaches 80% with 6 peers when R = 16, 8, and 4, and with 7 peers when R = 2. With 20 peers, the cost savings reach at 94% regardless of the playback rate R. It is expected that the telecommunication cost for data traffic will reduce as the technology advances and the market expands. Moreover, the small size of the CHUM network may provide additional benefits to the wireless multimedia users as well as provides catalysts to the related industries.

6 Conclusion

This paper proposes an approach for mobile peers to reduce the telecommunication cost by sharing the connection from one of the peers, called the *proxy*. All

peers take turns to become a proxy, which downloads multimedia data from the Internet and chumcasts the data to its peers. Each peer in a wireless *ad hoc* network is associated with a chumcast server located in wired network. The active server manages peer information, selects rebroadcasting peers, schedules the next proxy, and detects network partitioning. By uploading all neighborhood information periodically to associated servers and then to the active server, all peers are free from performing those tasks. The reduction of the tasks by the peers saves wireless bandwidth. By cooperating with wired and wireless networks, the CHUM network operates in a more secure environment than if it operates only in a pure *ad hoc* network. In addition, the peers save a large amount of the telecommunication cost even if a small CHUM network is formed.

References

[1] Garber, L.: Will 3G Really Be the Next Big Wireless Technology? IEEE Computer (2002) 26–32
[2] Bhagwat, P.: Bluetooth: Technology for Short-Range Wireless Apps. IEEE Internet Computing **5** (2001) 96–103
[3] Zhu, D., Mutka, M.: Sharing Presence Information and Message Notification in an Ad Hoc Network. IEEE Int'l Conference on Pervasive Computing and Communications (2003) 351–358
[4] Oram, A.: Peer-to-Peer : Harnessing the Power of Disruptive Technologies. O'Reilly & Associates (2001)
[5] Lim, H., Kim, C.: Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks. ACM MSWiM 2000 (2000) 61–68
[6] Qayyum, A., Viennot, L., Laouiti, A.: Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. Proceedings of Hawaii International System Sciences (HICSS) 2002 (2002) 3866–3875
[7] Peng, W., Lu, X.: AHBP: An Efficient Broadcast Protocol for Mobile ad hoc Networks. Journal of Science and Technology (JCST) - Beijing, China **16** (2001) 114–125
[8] Calinescu, G., Mandoiu, I., Wan, P., Zelikovsky, A.: Selecting Forwarding Neighbors in Wireless Ad Hoc Networks. ACM Workshop on Discrete Algothrithms and Methods for Mobile Computing and Communications (2001) 34–43
[9] Guha, S., Khuller, S.: Approximation Algorithms for Connected Dominating Sets. Algorithmica **20** (1998) 374–387
[10] Zhu, D., Mutka, M.: Fair Sharing of Proxy Responsibilities in an Ad Hoc Network. IEEE Int'l Conference on Pervasive Computing and Communications (2004)
[11] Solleti, A., Christensen, K.: Efficient Transmission of Stored Video for Improved Management of Network Bandwidth. International Journal of Network Management **10** (2000) 277–288

A Routing Strategy for Metropolis Vehicular Communications

Genping Liu[1], Bu-Sung Lee[1], Boon-Chong Seet[2], Chuan-Heng Foh[1], Kai-Juan Wong[3], and Keok-Kee Lee[1]

[1] Centre for Multimedia and Network Technology, School of Computer Engineering
[2] Network Technology Research, Centre Research TechnoPlaza, 4th Storey Nanyang Technological University, Nanyang Avenue, Singapore 639798
[3] Institute for Computing Systems Architecture Informatics University of Edinburgh JCMB, Mayfield Road U.K, Edinburgh, EH9 3JZ

Abstract. One of the major issues that affect the performance of mobile ad hoc networks (MANET) is routing. Recently, position-based routing for MANET is found to be a very promising routing strategy for inter-vehicular communication systems (IVCS). However, position-based routing for IVCS in a built-up city environment faces greater challenges because of potentially more uneven distribution of vehicular nodes, constrained mobility, and difficult signal reception due to radio obstacles such as high-rise buildings. This paper proposes a new position-based routing scheme called Anchor-based Street and Traffic Aware Routing (A-STAR), designed specifically for IVCS in a city environment. Unique to A-STAR is the usage of information on city bus routes to identify an anchor path with high connectivity for packet delivery. Along with a new recovery strategy for packets routed to a local maximum, the proposed protocol shows significant performance improvement in a comparative simulation study with other similar routing approaches.

1 Introduction

MANET is an autonomous system composed of mobile nodes communicating through wireless links in an environment without any fixed infrastructure support. Nodes in this network are self-organizing and rely on each other to relay messages to their correct destinations. As nodes are free to move randomly, the network topology may change rapidly and unpredictably. Thus, the routing protocol must be able to adapt and maintain routes in the face of changing network connectivity. Such networks are very useful in military and other tactical applications such as emergency rescue or exploration missions where an established (e.g. cellular) infrastructure is unavailable or unusable. Commercial applications are also likely where there is a need for ubiquitous communication services. Particularly in recent years, there is a growing commercial interest on the research and deployment of MANET technology for vehicular communications, e.g. FleetNet [1], VICS [2], CarNet 3 [3], etc. Existing MANET routing protocols work well in scenarios where nodes are uniformly distributed and moving freely in open space. However, these protocols do not work as well for IVCS

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 134–143, 2004.

in a city environment because of some additional inherent challenges. Generally, vehicular nodes are more unevenly distributed due to the fact that vehicles tend to concentrate more on some roads than others. Their constrained mobility by road patterns, along with more difficult signal reception in the presence of radio obstacles such as high-rise buildings, have contributed to greater fragility in the "connectedness" of the IVCS network, and the frequent formation of topology "holes", which could not be dealt with effectively by existing position-based routing protocols.

Recently, a project called BUSNet [4] was initiated to study the performance of MANET routing algorithms in the IVCS, based on a Metropolitan Grid model (M-Grid) [4][5]. It proposes using the regular network of buses to form a stable communication backbone for an otherwise fragile IVCS network. In [5], the performance of existing MANET routing protocols is found to be much lower in the M-Grid model than in the random waypoint model. This is because inter-node connectivity is much harder to establish with constrained mobility and obstacles in the M-Grid model.

For a large, metropolitan-scale IVCS network, the scalability of the routing protocol is very important. Position-based routing is known to be very scalable with respect to the size of the network. Thus, it is a good candidate for metropolitan-scale IVCS. However, applying position-based routing to IVCS may not be without any problems. An example is Greedy Perimeter Stateless Routing (GPSR) [6], one of the most well known position-based protocols in literature. It works best in a free open space scenario with evenly distributed nodes. But when applied to city scenarios [7][8], GPSR is found to suffer from several deficiencies, the details of which we will discuss in the next section.

This paper proposes a new position-based routing scheme called Anchor-based Street and Traffic Aware Routing (A-STAR), designed specifically for IVCS in a city environment. Unique to A-STAR is the usage of information on city bus routes to identify an anchor path with high connectivity for packet delivery. Along with a new recovery strategy for packets routed to a local maximum, the proposed protocol shows significant performance improvement in the M-Grid model. A-STAR is therefore proposed as a potential routing strategy for metropolis vehicular communications.

The remainder of the paper is organized as follows. Section 2 discusses with example the challenges faced by position-based routing in IVCS. Section 3 presents some works in literature related to this area. Section 4 describes the proposed A-STAR protocol. The mobility model and simulation setting are explained in Section 5. Performance results are presented in Section 6. Finally, the paper is concluded in Section 7.

2 Challenges of Position-Based Routing in IVCS

The challenges of position-based routing in a city environment have been discussed thoroughly in [7][8]. An example is given here to illustrate some main

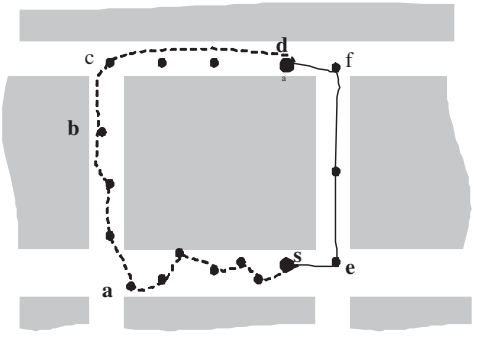


Fig. 1. Challenges of Position-Based Routing in IVCS

problems if typical GPSR is deployed directly to IVCS. Figure 1 shows a partial city environment.

Suppose node s wants to send a packet to node d. Greedy forwarding will fail in this case as there is no neighbor of s, which is nearer to d than s itself. Following the strategy in GPSR, the packet enters into perimeter-mode, using the right hand rule to travel through each node on the dotted route, including nodes a, b and c. At b, it is found that c is nearer to d than a, at which the packet enters into perimeter-mode. Thus, the packet switches back to greedy mode at b, and then reaches its destination d through c. It can be seen that this route is very long in terms of hop count. In fact, s can reach a, and a can reach b, both in one hop. This shows that the perimeter-mode which packet employs to recover from local maximum is very inefficient and time-consuming.

Another observation is that the packet can actually travel from s to d via a route that passes through e and f (shown as solid line), which is much shorter. However, this route is not exploited because the perimeter-mode of GPSR based on right hand rule is biased to a specific direction when selecting for the next hop.

It should be noted that in a city environment, the constrained mobility and frequently encountered obstacles can effectively force GPSR to run into perimeter-mode frequently. As a result, the performance of GPSR could deteriorate dramatically, and therefore may not be suitable for IVCS.

3 Related Work

3.1 Anchor-Based Routing

Anchor-based routing is analogous to the source routing of DSR [10]. In anchor-based routing, the source node includes into each packet a route vector composed of a list of anchors or fixed geographic points, through which packets must pass. Between anchors, the greedy position-based routing is employed. Both Terminode Remote Routing (TRR) [9] and Geographic Source Routing (GSR) [7][8] are

examples of algorithms that employ anchor-based routing to forward packets to remote destinations.

3.2 Spatial Aware Routing

In spatial aware routing, spatial information such as streets map of a city or a description of how several towns are connected by highways, is utilized to assist in making routing decisions. The spatial information reflects the underlying node distribution and topology of the network. Spatial aware routing is usually used in conjunction with anchor-based routing, such as in TRR and GSR where anchored paths are computed using the spatial information.

4 Anchor-Based Street and Traffic Aware Routing (A-STAR)

Considering the challenges faced in a city environment, a new position-based routing scheme called A-STAR is proposed. Similar to GSR, A-STAR adopts the anchor-based routing approach with street awareness. The term "treet awareness" is preferred over "spatial awareness" to describe more precisely the use of street map information in our routing scheme for anchor path computation. That is, using the street map to compute the sequence of junctions (anchors) through which a packet must pass to reach its destination. But unlike GSR, A-STAR computes the anchor paths with traffic awareness. "Traffic" herein refers to vehicular traffic, including cars, buses, and other roadway vehicles.

It is observed that in a metropolitan area, some streets are wider and accommodate more vehicular traffic than others. These are the major streets, served by a regular fleet of city buses. Connectivity on such streets can be higher due to higher density of vehicular nodes and more stable due to regular presence of city buses. With this observation, weight can be assigned to each street based on the number of bus lines by which it is served, i.e. the more bus lines by which a street is served, the less weight it is assigned, and vice-versa. The street map in use by the vehicle is assumed to be loaded with bus route information. An anchor path can thus be computed using Dijkstra's least-weight path algorithm. For such a map with pre-configured information, it is called a ***statistically rated map***.

While bus route information can provide a reasonable estimate of the expected vehicular traffic on each street, the traffic conditions in a city area can be quite dynamic at times. A better weight assignment scheme is therefore one that dynamically monitors and assigns weight to a street based on its latest traffic condition, which can provide higher quality of anchor computation. It could be envisaged that future IVCS would be able to monitor the city traffic condition and distribute such information to every vehicle connected to the IVCS network. This information could then be used to re-compute the weight of each street on the map, e.g. more vehicles, less weight assigned, and vice-versa. Such a map with re-configurable information is called a ***dynamically rated map***.

4.1 Local Recovery

It has been shown that local recovery algorithm of GPSR using perimeter-mode is quite inefficient in a city area. Other recovery algorithms that rely on "right hand rule" such as face-1 or face-2 [11] also face a similar problem. GSR adopts a "switch back to greedy" approach for local recovery: when a packet reaches a local maximum along its anchor path, it switches back to greedy mode. This is not efficient at all as it has been shown that greedy forwarding does perform well in a city environment.

Thus, a more efficient recovery strategy is proposed for A-STAR: a new anchor path is computed from the local maximum to which the packet is routed. The packet is salvaged by traversing the new anchor path. To prevent other packets from traversing through the same void area, the street at which local maximum occurred is marked as "out of service" temporarily, and this information is distributed to the network, or simply carried with the recovered packet. The "out of service" streets are not used for anchor computation or re-computation during the "out of service" duration and they resume "operational" after the time out duration.

5 Mobility Model and Simulation Setting

5.1 M-Grid Mobility Model

Mobility model describes the movement of nodes in a certain environment. In this paper, the M-Grid mobility model [4][5] is used to describe the movement of vehicular nodes in a city area. M-Grid is a variant of the Manhattan model [12], which models the vehicular movement in a typical metropolis where streets are set out on a grid pattern. Key features which distinguish the M-Grid from Manhattan model, include:

- *Node Heterogeneity*: Buses and cars are two types of vehicular nodes modeled in our M-Grid. Buses, which only travel along the bus routes, show higher regularity and lower mobility than cars. For the M-Grid in Figure 2, the bus routes are represented by bold lines in gray. It shows three loop lines (or service numbers), plying the streets in various parts of the city. Each line is bi-directional with buses running clockwise and anti-clockwise.
- *Preferential Movements*: It is observed that in real life, some streets would attract more vehicles than others. More often than not, these are the main streets, which are bustling with people and therefore served by buses. In M-Grid, when a car reaches a junction, it would choose to move into another street with some preference. Given the observation above, the car at the junction shall give greater preference to a street which is on a bus route than one which is not.
- *Radio Obstacles*: The blocking of signal transmissions by objects such as high-rise buildings in the city has been modeled in M-Grid. As Figure 3 shows, the gray areas represent obstacles, which are non-penetrable by the

signals. Thus, for a node pair to communicate directly, they must have a "line-of-sight" to each other, in addition to being in range of one another.

5.2 Simulation Setting

Performance of A-STAR and other related protocols are evaluated using the ns-2 [13] simulator. Four protocols are implemented, namely: i) GPSR, ii) GSR, iii) A-STAR-SR, and iv) A-STAR-DR. Protocol iii and iv refer to the proposed A-STAR with *statistically rated* and *dynamically rated* maps respectively. Presence of a location service is assumed to supply position data to each protocol under study. Table 1 summarizes the parametric settings used in our simulation.

Note that the number of vehicles (nodes) is varied to reflect different vehicle densities under which the performance of each protocol is evaluated. However, throughout the evaluation, the number of buses is a constant, with only the car density varying. For the M-Grid shown in Figure 2, the three bus lines have a total of 37 buses running in the city: two with 12 buses, one with 13 buses. Inter-bus distance is approximately 1 kilometer for each line in the same direction. Moreover, cars at the junction would move into a street which is on a bus route with a probability three times that of which is not. Speed limit of buses and cars are 50 and 70 km/h respectively.

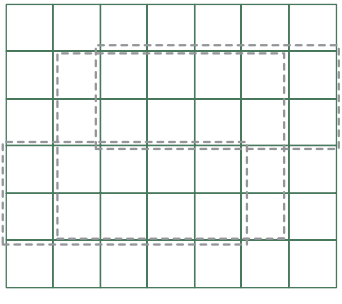


Fig. 2. M-Grid with bus routes

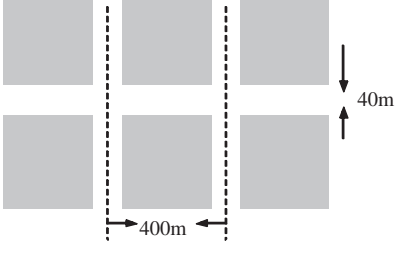


Fig. 3. M-Grid with obstacles

Table 1. Simulation Setting

Parameter	Setting
Mobility model	M-Grid
Traffic model	20 CBR connections
Data packet size	64 bytes
Transmission range	350 meters
Map size	2800x2400m2 (7x6 grid)
Node number	200 to 500, in steps of 50
Simulation time	500 seconds

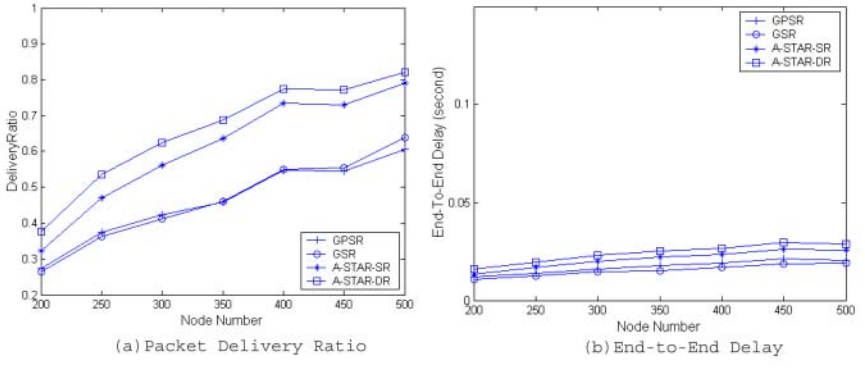


Fig. 4. Performance without local recovery

Performance result for each simulated vehicle density (node number) is the average of five simulation runs. The key metrics of interest are:

- *Packet delivery ratio*: the ratio of packets delivered to the destinations to those generated by the sources.
- *End-to-end delay*: the average time it takes for a packet to traverse the network from its source to destination.

6 Simulation Results and Analysis

Recall that A-STAR differs from GSR and GPSR in two main aspects. Firstly, A-STAR incorporates traffic awareness by using statistically rated and dynamically rated maps. Secondly, A-STAR employs a new local recovery strategy that is more suitable for a city environment than the greedy approach of GSR, or the perimeter-mode of GPSR.

To investigate impacts of each aspect on the routing performance, protocols are evaluated initially without local recovery, and later with local recovery.

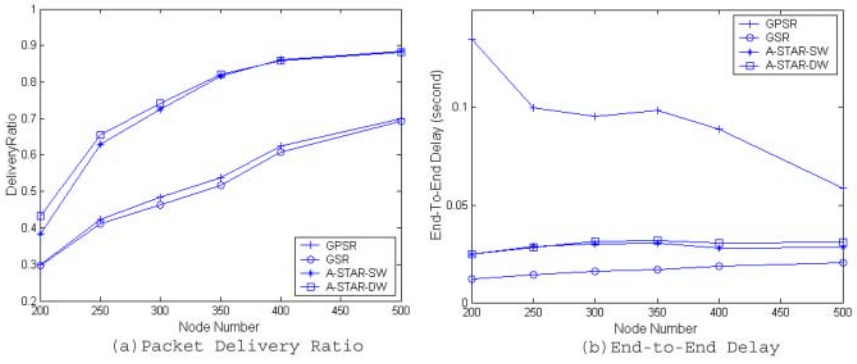


Fig. 5. Performance with local recovery

Without local recovery, a packet is simply dropped when it encounters a local maximum. Figures 4 and 5 show the protocols performance without and with local recovery, respectively.

In Figure 4(a), it is observed that more packets are delivered as node number increases. This is expected since more nodes increases the probability of connectivity, which in turn reduces the number of packets dropped due to local maximum. It is also observed that GSR did not show a better performance than GPSR, possibly because the grid layout of streets did not pose as much problem to GPSR as did one with fork junctions in [8]. With traffic awareness, A-STAR shows the best performance because it can select paths with higher connectivity for packet delivery. As much as 40% more packets are delivered by A-STAR, compared to GSR. Between A-STAR-SR and A-STAR-DR, the latter performs better by using more precise vehicular traffic information. Figure 4(b) shows the result of end-to-end delay. Generally, no significant difference is observed between the protocols. A-STAR, however, shows slightly higher delay that may be attributed to possibly longer, but higher connectivity paths used for packet delivery. With local recovery, packets that encounter local maximum can be rerouted and delivered instead of being dropped. Thus, more packets are delivered by each protocol as shown in Figure 5(a). The increase in packets delivered is more significant at lower node number where local maximum is encountered more frequently. For example, with local recovery, A-STAR-DR delivers 20% more packets at 250 nodes, while only 6% more at 400 nodes. It is also observed that local recovery allows A-STAR-SR to narrow its performance gap with A-STAR-DR. GSR and GPSR show improvement in packet delivery of not more than 15% with local recovery, which suggests that their recovery strategies may not be very effective in a city environment.

Figure 5(b) shows the corresponding result for end-to-end delay. A key observation is that GPSR with local recovery incurs significantly higher end-to-end delay. This is because of frequent attempts by GPSR to salvage packets from local maximum via perimeter-mode, which is generally inefficient and causes con-

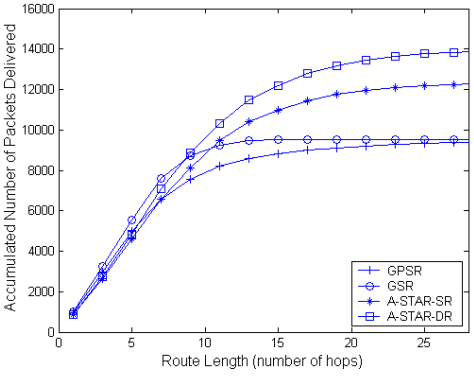


Fig. 6. Route length distribution (for 200 nodes)

gestion especially at lower node number. Delay of A-STAR is lower than GPSR, but seemingly higher than GSR, once again at lower node number. A close analysis of its route length distribution in Figure 6 suggests that the higher delay is likely an artifact due to successful delivery of more long-distance packets that are otherwise dropped without local recovery. These packets inevitably have longer traversal time and thus contribute to a higher average end-to-end delay.

7 Conclusion

In this paper, a new position-based routing protocol A-STAR is proposed for metropolis vehicular communications. A-STAR features the novel use of city bus route information to identify anchor paths of higher connectivity so that more packets can be delivered to their destinations successfully. In our comparative simulation study with other position-based routing schemes, A-STAR demonstrates excellent improvement in packet delivery while maintaining reasonable end-to-end delay. As future work, the traffic awareness in A-STAR shall be extended to include data traffic to provide vehicular nodes with higher performance paths in terms of connectivity as well as delay. Another area that shall be looked into is how information on bus schedules, in addition to bus routes, can be utilized to further optimize the performance of our protocol.

References

[1] W. Franz, R. Eberhardt, and T. Luckenbach, "FleetNet - Internet on the Road", Proc. 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, Oct. 2001. 134

[2] S. Yamada. "The Strategy and Deployment Plan for VICS", IEEE Communications, Vol. 34, No. 10, pp.94-97, 1996 134

[3] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, "CarNet: A Scalable Ad Hoc Wireless Network System", Proc. 9th ACM SIGOPS European Workshop, Sept. 2000. 134

[4] K. J. Wong, B. S. Lee, B. C. Seet, G. Liu, and L. Zhu, "BUSNet: Model and Usage of Regular Traffic Patterns in Mobile Ad Hoc Networks for Inter-Vehicular Communications", Proc. ICT 2003, Thailand, April, 2003 135, 138

[5] B. S. Lee, K. J. Wong, B. C. Seet, L. Zhu, and G. Liu, "Performance of Mobile Ad Hoc Network in Constrained Mobility Pattern", Proc. International Conference on Wireless Networks (ICWN'03), Las Vegas, USA, Jun. 2003. 135, 138

[6] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", Proc. ACM/IEEE MobiCom, Boston, USA, Aug. 2000. 135

[7] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Herrmann, and M. Mauve, "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments", Proc. IEEE Intelligent Vehicles Symposium (IV2003), Ohio, USA, Jun. 2003. 135, 136

[8] J. Tian, I. Stepanov, and K. Rothermel, "Spatial Aware Geographic Forwarding for Mobile Ad Hoc Networks", Proc. MobiHoc, Lausanne, Switzerland, Jun. 2002. 135, 136, 141

[9] L. Blazevic, S. Giordano, and J. Y. Le Boudec. "Self-Organizing Wide-Area Routing", Proc. SCI 2000/ISAS 2000, Orlando, USA, Jul. 2000. 136

[10] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing Protocol for Mobile Ad Hoc Networks", Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer, 1996, pp. 153-81. 136

[11] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks", Proc. 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM'99), Seattle, USA, Aug. 1999. 138

[12] A. Kamat and R. Prakash. "Effects of Link Stability and Directionality of Motion on Routing Algorithms in MANETs". Proc. IEEE International Conference on Computer Communications and Networks (ICCCN), Las Vegas, USA, Oct 2000. 138

[13] Network Simulator (ns-2), `http://www.isi.edu/nsnam/ns/` 139

On Demand Routing Protocol to Support Unidirectional Links in Mobile Ad Hoc Networks

K. Venkataramanan[1], D. Aravindan[1], and K. Ganesh[2]

[1] Department of Information Technology, Crescent Engineering College
Vandalur, Chennai-600048, India
venkat_cec@yahoo.co.in
aravindous@yahoo.com
[2] Department of Information Technology, Velammal Engineering College
Ambattur, Chennai-600066, India
ganesh_krt@yahoo.co.in

Abstract. Mobile ad hoc network (MANET) is a multihop wireless network formed by a collection of mobile nodes communicating through radio links characterized by the absence of any preexisting infrastructure. Many of the proposed routing algorithms for MANET assume that the given network is an undirected graph. For these routing algorithms to work in the presence of unidirectional links, some modifications were proposed. These modifications eliminate unidirectional links from the route computation procedure. As a result, the performance of many popular routing protocols degrades drastically in the presence of many unidirectional links. In our paper we propose a theoretical framework of a novel routing protocol called ORPUL (On demand Routing Protocol with Unidirectional Link support) for mobile ad hoc networks together with the changes needed to be accommodated in the IEEE 802.11 MAC layer protocol.

1 Introduction

Mobile adhoc network consists of a dynamic set of nodes with a network set up by them with no preexisting infrastructure. Majority of the routing protocols proposed for MANET like AODV,TORA,DSDV etc, function with the assumption that all nodes in a given network have the same transmission range or all the wireless links are symmetric or bidirectional . However these assumptions in some cases prove to be wrong. The occurrence of unidirectional links in MANET is attributed to different transmission ranges of the heterogeneous nodes, or discrepancies in the transceiver capabilities or transient channel interference, or due to the power control algorithms. The power control algorithms running at different nodes are aimed at efficient power usage, thus making these nodes function at various transmission ranges at different instances of time.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 144–153, 2004.

2 Previous Work

Several papers have been published answering the problems concerned with unidirectional links [4,5,6,7]. These papers view the unidirectional links as an anomaly and tend to eliminate them from the route computation procedure. For instance, many researchers have stated that the overhead of maintaining unidirectional links in a route computation procedure tend to outweigh the benefits derived by using the unidirectional links [2,8]. A patch work has been proposed for AODV protocol, to deal with unidirectional links [2]. However, this approach is equivalent to the blacklisting scheme in terms of avoidance of unidirectional links in the route computation [1]. These fault tolerant routing scales well only in those networks where the number of unidirectional links is insignificant.

When the number of unidirectional links are large, and the network still remains a connected graph, the routing protocols that don't utilize unidirectional links, perceive the network as an unconnected graph. As a result, all the path or a series of links, which connect the nodes remain unnoticed by these protocols which do not utilize the unidirectional links .The scalability and performance of many routing protocols for MANET decreases drastically with the increase in the number of unidirectional links, for the simple reason that these protocols do not consider the unidirectional links in the route computation procedure.

A clever approach is adopted in [6] through the use of a global positioning system for identifying the position of the nodes and exchanging these node positions in terms of the (x, y) coordinates with the adjacent nodes. Then, the Euclidean distance between the nodes is calculated and the transmission range of a node is varied in such a manner, that the transmission range is greater than or equal to Euclidean distance between these two nodes .

The route computation procedure in RODA protocol is characterized by the probing of independent forward and reverse paths between the source and the destination [5]. This protocol requires the information of the nodes visited to compute the forward and reverse paths. The paper also modifies the existing IEEE 802.11 MAC protocol to support unidirectional links. RODA requires two broadcasts to compute the forward and reverse paths. This is not required in the case when the shortest path from the source to destination is a bidirectional path. The bidirectional path can be used as both forward and reverse paths. Also, the additional overhead of storing the information about all the nodes visited can prove costly, especially in large networks.

3 Proposed ORPUL Routing Scheme

ORPUL (On demand Routing Protocol with Unidirectional Link support) is a source-initiated on-demand approach. It has 2 phases: route discovery and route maintenance. Whenever a source node wants to send packets to the destination it computes the route to the destination in an on-demand manner. The route may be broken due to link failure caused by node movements. This calls for route reconstruction.

3.1 Route Discovery Phase

Each node in the network maintains a routing table which contains { destination node, next hop } entries. It stores only one entry per destination. The routing table is updated periodically to cope up with the network dynamics. Unlike many of the existing protocols, this protocol makes use of unidirectional links to route packets. Here we construct forward and reverse paths only when there are one or more unidirectional links in the shortest path from source to destination. Otherwise if the shortest path has only bi-directional links, the protocol behaves similar to the AODV protocol. In this section we describe how the route discovery process takes place.

HELLO Messages. Every node periodically broadcasts HELLO messages [1] to its neighbours. The HELLO message contains the list of all nodes from which the node can hear. A unidirectional link is detected by a node when it receives an HELLO message from another node and does not find its id listed in it.It should be noted that only a downlink node can detect the presence of an unidirectional link shared with its corresponding uplink node.The uplink node would be notified of an unidirectional link on an on-demand basis during route discovery phase, by the broadcast of REV_PATH_PROBE packet which contains the ids of nodes which share an unidirectional link.

Route Discovery Procedure. Whenever a source has data to send, it floods the ROUTE_PROBE packet. This packet will include the information about the nodes that share an unidirectional link. A node processes only the first copy of a ROUTE_PROBE packet and rejects the subsequent copies. Whenever any node receives a ROUTE_PROBE packet it checks whether it received the packet on an unidirectional link. If yes then the node sets the UNI_PATH flag incase it is not set. It also appends its node id along with the node id of the node from which it received the packet. If a node that knows a path to the destination node receives a ROUTE_PROBE packet with the UNI_PATH flag set, it sets the REM_PATH_KNOWN flag (which means that the packet is going to be sent through a predetermined path to the destination) and appends its own node id to the ROUTE_PROBE packet and unicasts it to the destination through the path that it knows(no more node ids will be added in the forward path).

Unidirectional Link Notification to the Uplink Node. If the destination node receives a ROUTE_PROBE packet with the UNI_PATH flag set, it broadcasts the REV_PATH_PROBE packet containing information about all the nodes that share an unidirectional link in the forward path (from source to destination). In case the destination receives the ROUTE_PROBE packet with UNI_PATH flag not set, it sends the BI_REPLY packet to notify the sender and intermediate nodes about the bi-directional path (similar to AODV protocol functioning).

Table 1. Packets and Functions

Packet Type	**Function**
ROUTE_PROBE	This is used in route discovery process.This packet will include the information of nodes that form a unidirectional link.
BI_REPLY	This is used to notify the sender and intermediate nodes about the bi-directional path by the destination.
REV_PATH_PROBE	The destination, when the UNI_PATH flag is set, floods this packet.It is used to compute an efficient reverse path from destination to the source.
INTER_REPLY	This packet is sent by nodes, that are tail of unidirectional links in forward path, towards preceeding nodes in the list of node ids in ROUTE_PROBE packet when they receive the REV_PATH_PROBE packet.
REV_PATH_NOTIFY	The sender transmits this packet via the forward path set up, to notify the destination about the reverse path from the destination to the source.
REV_PATH	The receiver transmits this packet to notify the nodes along the acquired reverse path forward data packets over the reverse path.

The intended recipients of the REV_PATH_PROBE packet are the source and the nodes that fall on the shortest path from the source to the destination and also have an outgoing unidirectional link. The information about these nodes is extracted from the ROUTE_PROBE packet. Any node that receives a REV_PATH_PROBE packet checks if the node-id of the uplink node of any unidirectional link in the forward path, as mentioned in the packet, matches its own id. If yes then it is one of the intended recipients. The REV_PATH_PROBE packet also includes the ids of the nodes visited, as the packet propagates towards the destination (this information is used to compute the backward path from the destination to the source). A node will process only the first copy of a REV_PATH_PROBE packet.

The recipient nodes (excluding the source) in turn send an INTER_REPLY packet via the reverse path. The INTER_REPLY packet will terminate either at the source or at the node that has an incoming unidirectional link, and informs the intermediate nodes about the next hop to be taken to reach the destination, which is same as that to reach the node that sent the INTER_REPLY. The recipient nodes, sometimes even the source (when it has an outgoing unidirectional), sets its next hop to the node which falls on the shortest path from the source to destination and having an incoming unidirectional link from the recip-

Table 2. Flags and Function

Flag	Function
UNI_PATH	This flag in the ROUTE_PROBE packet when set indicates that the packet has come via a path that has one or more unidirectional links.The destination on receiving such packet broadcasts the REV_PATH_PROBE packet to compute the reverse path.
REM_PATH_KNOWN	This flag when set indicates that an intermediate node in the forward path knows a route to the destination and so no more node ids need to be added in the ROUTE_PROBE packet.

ient node. Thus a forward path is established from the source to the destination. The source sends the REV_PATH_NOTIFY packet to the destination, through the forward path. The REV_PATH_NOTIFY packet contains information about the backward path, which is extracted from the REV_PATH_PROBE packet. On receiving this packet the destination generates the REV_PATH packet, which notifies the nodes along the acquired reverse path forward data packets over the reverse path. Now data is transferred using the path set up between the source and destination.

4 Route Maintenance

In ORPUL, route maintenance is done by the source. After the route discovery phase is completed the sender sends data to the destination via the path (forward path in case there are two paths) being set up. The destination receives the data and sends its acknowledgement via the reverse path (if there are two paths). If the source does not receive any reply from the destination within, say t seconds, it assumes that there is a fault in the route, either in the forward or backward or both. It again initiates a route discovery process as explained above, to compute the new route to the destination. This is extremely advantageous given the mobile nature of the network. Instead of a costly route maintenance process, which incurs additional storage and computation costs, this method is simple and efficient.

5 Link Level Acknowledgement

The IEEE 802.11 MAC protocol is a popular medium access protocol used in mobile wireless communications.The hidden and exposed terminal problems is tackled in this protocol using the signaling protocol RTS-CTS-DATA-ACK [9]. Whenever a node wants to transmit a data packet it transmits a RTS message to the receiver informing about the upcoming data transmission and hence to

defer other data transmission. The receiver then sends a CTS message in response to the RTS message sent by the source, consenting the delivery of the data packets. The source can then proceed with its data transmission by the transfer of DATA packets. For each DATA packet being transmitted, the link layer protocol requires the transmission of the corresponding acknowledgement by the destination to the source.

The entire procedure described above assumes the presence of bidirectional links between two nodes and hence, if a particular node does not happen to receive the acknowledgement for the transmitted packet, it concludes that the given link is broken. In our proposed protocol, which was devised with the unidirectional links in mind, the direct transmission of acknowledgements isn't possible on an unidirectional link. This is because the theoretical studies done by us reveal that there is a combinatorial explosion in the number of control packets, which is due to the additional route discovery procedure for the set up and maintenance of alternate reverse path between nodes that have unidirectional links.Consequently the link level acknowledgement for DATA packets on an unidirectional link is ignored.

The functionality of the IEEE 802.11 MAC protocol remains unchanged for DATA transmissions over bidirectional links and hence the link level reliability offered by the IEEE 802.11 MAC protocol is being preserved.The DATA transmissions over outgoing unidirectional links is done without any corresponding transmissions of ACK messages by the downlink node as the link level acknowledgement of DATA transmissions isn't viable on an unidirectional link. The knowledge of whether a given link is an outgoing unidirectional or a bidirectional link is maintained by the routing layer and the underlying MAC layer is informed on whether a given link is unidirectional or not and correspondingly whether to expect an acknowledgement from a downlink node

6 Working of the ORPUL Protocol

Figure 1 illustrates the functioning of ORPUL in the presence of unidirectional in the shortest path from source to destination. Source A wants to send some data to destination F [Refer Fig 1]. Node A broadcasts the ROUTE_PROBE packet to its neighbors. The intermediate node B has received the packet on an unidirectional link from A. So it appends its node-id along with the node-id of A in the ROUTE_PROBE packet, sets the UNI_PATH flag to 1 and re-broadcasts the packet to its neighboring nodes. After broadcasts from nodes C, D, the packet reaches E in a unidirectional link from D. Now E appends the node-id of D along with its node-id and broadcasts it to F. The ROUTE_PROBE packet when received by F will have the node-ids of A, B, D and E. Now F broadcasts the REV_PATH_PROBE packet carrying this information. The nodes A and D, infer that they have an outgoing unidirectional link to B and E respectively, and set their next hop entry in the routing table accordingly. These nodes send an INTER_REPLY packet to the nodes previous to them in the REV_PATH_PROBE packet; in this case D sends an INTER_REPLY packet to B. This packet sets

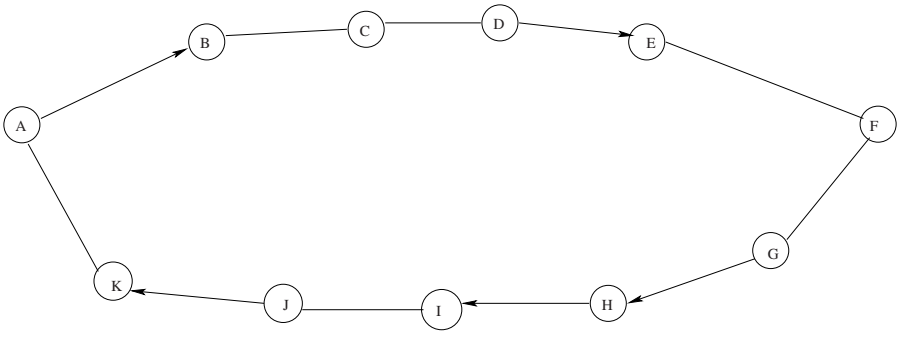


Fig. 1. Scenario 1

Table 3. Routing information stored in nodes A and C

NODE A		NODE C	
Destination	Next Hop	Destination	Next Hop
F	B	F	D

the forward path of the intermediate nodes to F and also to the respective intermediate nodes that send them.

In this case it tells B that C is the next hop to reach F and D (because the INTER_REPLY came from D), and also tells A that, B is the next hop to reach F. The packet also corrects invalid path entries, in this example for nodes C and D the entry for A is removed. Thus the forward path A-B-C-D-E-F has been set up. The REV_PATH_PROBE packet, when received by A, contains the information about the nodes visited from F to A, which is G, H, I, J, K (reverse path). This reverse path information is sent to node F by node A through the forward path being set up using the REV_PATH_NOTIFY packet. Node F after reception of REV_PATH_NOTIFY packet generates a REV_PATH packet, which is sent to make nodes along the acquired reverse path forward data packets over the reverse path. The nodes in the network can examine the REV_PATH_PROBE packet to infer information about other nodes that can be extremely useful in future route computations. This route computation process took two broadcasts, and also the overhead of storing information of intermediate nodes constituting unidirectional links.

Figure 2 illustrates the route maintenance procedure of ORPUL. The route discovery process was explained before using Figure 1. Let us assume that the links D-E and J-K breaks. Neither the receiver receives any data, nor the sender receives any acknowledgement. A time-out occurs at the sender after t seconds, after which the sender assumes disconnection and initiates the route discovery process.

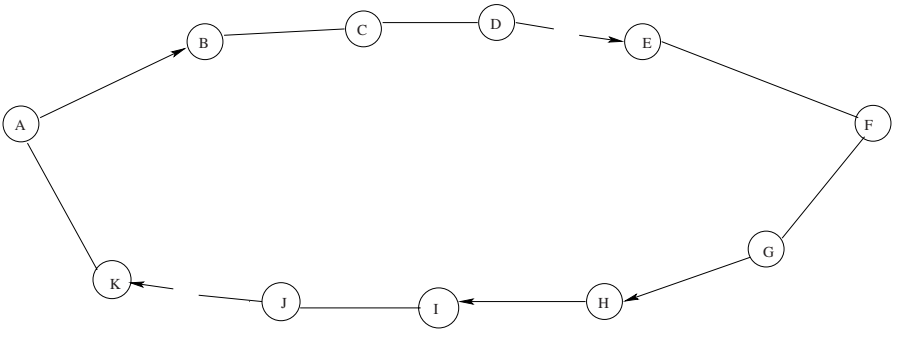


Fig. 2. Scenario 2

Proposed Representation of Node List. In ORPUL the node id's of all nodes that share an unidirectional link in the forward path are included in the ROUTE_PROBE packet, which we shall call as the node list. Additionally an extra bit for each node in the node list (except for the first and last node) is added at the end of the corresponding node id.

If a node in the node list is just a tail of an unidirectional link, in the forward path, but not the head of another such link the special bit is set to 0. This is set when the next node adds its id to the list. If a node in the node list is just the head of an unidirectional link, the special bit is set to 1.

Let n be the total number of nodes in the path and n_u be the number of nodes that share an unidirectional links. Assuming that the node id is 32 bits, then for ORPUL to use lesser packet size than RODA the required condition is

$$32n - 33n_u >= -2 \tag{1}$$

This will be illustrated in figure 3 where X axis is n, the total number of nodes in the path and Y axis is n_u, the number of nodes forming unidirectional links in the path.

7 Conclusion

ORPUL supports unidirectional links and takes them into account during route computation.Unlike DSR and RODA, ORPUL does not store information about all the nodes visited, except for the reverse path. This reduces the packet overhead by a huge margin, especially for large networks. ORPUL functions similar to AODV when the shortest path from source to destination is bi-directional. Backward paths are only calculated when the shortest path to the destination has unidirectional links.

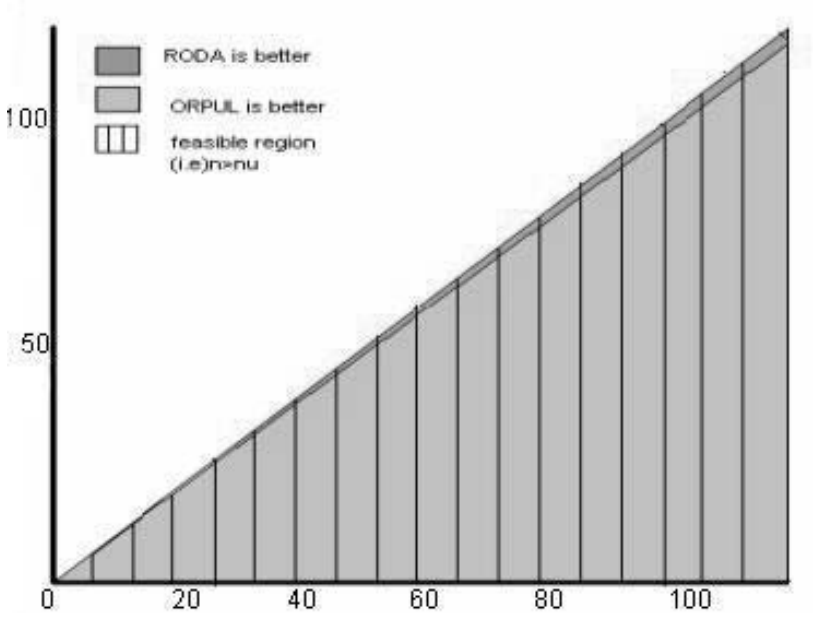


Fig. 3. Comparison of RODA and ORPUL in terms of packet size

8 Future Course of Action

It should be worthwhile to simulate ORPUL using ns-2 and do a performance comparison of ORPUL with other popular routing protocols for MANET such as AODV, DSR in the presence of unidirectional links. This requires amending the IEEE 802.11 protocol to support routing in the presence of unidirectional links. Additionally the setdest function in ns-2 can be used to generate the random way point model in which the nodes move with an arbitrary velocity within the maximum velocity limit,pauses for a random moment of time and then moves with a random velocity. The unidirectional links can be simulated by assigning a transmission range randomly chosen from the set trans comprising of the elements x, y, z where x, y, z correspond to transmission range. We hope to return to this in future.

Acknowledgements

We would like to thank Prof. S.P. Reddy, Head of the Department, Information Technology, Crescent Engineering College and Prof. Mahadevan, Head of the Department, Information Technology Velammal Engineering College for their useful discussions. We would like to thank Dr. K.P. Mohammed, Principal, Crescent Engineering College for his kind encouragement.

References

[1] C. E. Perkins, E. M. Royer, and S. R. Das.: Ad hoc On-Demand Distance Vector (AODV) Routing. `http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-10.txt`, Jan 2002. IETF Internet Draft (work in progress).

[2] Mahesh K. Marina Samir R. Das.:Routing Performance in the Presence of Unidirectional Links in Multihop Wireless Networks, In MOBIHOC'02, June 9-11, 2002, EPFL Lausanne, Switzerland.

[3] Prasun Sinha, Srikanth Krishnamurthy and Son Dao.:Scalable Unidirectional Routing with Zone Routing Protocol (ZRP) Extensions for Mobile Ad-Hoc Networks, In Proceedings of IEEE WCNC 2000, Chicago, September, 2000.

[4] Lichun Bao and J. J. Garcia-Luna-Aceves.: Link-state Routing in Networks with Unidirectional Links, In Eight International Conference on Computer Communications and Networks, pages 358-363, 1999.

[5] D. K.Kim, C.-K.Toh and Y.Choi,: RODA : a new dynamic routing protocol using dual paths to support asymmetric links in mobile ad hoc networks, IEEE ICCCN, LasVegas, USA, 2000.

[6] D. K.Kim, H. S.Jeong, C.-K.Toh and Y.Choi, :GAHA and GAPA : Approaches for Supporting Asymmetric Links in Mobile Ad Hoc Networks, IEEE PIMRC, San Diego, USA, 2001.

[7] Sanket Nesargi and Ravi Prakash. :A Tunneling Approach to Routing with Unidirectional Links in Mobile Ad-Hoc Networks, In Proceedings of IEEE ICCCN 2000, Las Vegas, Nevada, October 16 - 18, 2000.

[8] R. Prakash. :Unidirectional Links Prove Costly in Wireless Ad- Hoc Networks, Proceedings of ACM DIAL M'99 Workshop, Seattle, WA, August 1999, pp. 15-22.

[9] IEEE, P802.11, IEEE Draft Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, D2.0 (July 1995).

On Reducing Paging Cost in IP-Based Wireless/Mobile Networks

Kyoungae Kim[1], Sangheon Pack[2], and Yanghee Choi[2]

[1] LG Electronics Inc., Seoul, Korea
Multimedia and Communication Lab., School of CSE
k2heart@lge.com
[2] Seoul National University, Seoul, Korea
shpack@mmlab.snu.ac.kr
yhchoi@snu.ac.kr

Abstract. For more scalable mobile services, it is important to reduce the signaling cost for location management. In this paper, we propose a cost effective IP paging scheme utilizing explicit multicast (xcast). Xcast is a new kind of multicast scheme for small sized groups which uses unicast with low maintenance overhead. In terms of the paging algorithm, we use a selective paging scheme to minimize the paging cost, by dividing a paging area into several sub-paging areas. For the performance analysis, we develop an analytical model based on the random walk model in the mesh and hexagonal cell configurations. Using this model, we compared the paging cost between IP paging scheme using xcast and that using unicast or multicast. The results indicate that the proposed paging scheme reduces the paging cost by 44-84% compared with traditional paging schemes using multicast.

1 Introduction

In wireless/mobile networks, since mobile users are free to move within the coverage area, the network can only keep track of the approximate location of each mobile user. When a request is made to establish a session with a particular user, the network needs to determine the user's exact location within the cell granularity. The operation of the mobile host (MH) informing the network about its current location is known as *location update*, and the operation of the network determining the exact location of the mobile user is called *terminal paging*.

Recently, with the advent of IP technologies, IP-based location management has become the focus of research in this area. In terms of location update, the Mobile IP Working Group in Internet Engineering Task Force (IETF) proposed various protocols based on Mobile IP. On the other hand, in terms of terminal paging, several protocols were proposed in [1], [2], and [3]. Unlike the paging protocols in cellular networks, these protocols are based on IP-layer messages so that they are called *IP paging protocols*. With time-constraint multimedia applications (e.g. VoIP and message applications) gaining in popularity, IP paging is being considered as one of the essential functions in next-generation mobile

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 154–163, 2004.

networks. However, previous studies didn't focus so much on the issue of cost optimization schemes, which provide system scalability, but only on the basic paging architecture, paging procedure, paging area design, and so on.

In this paper, we propose a cost-effective IP paging scheme. Among the various cost optimization factors we focus on rendering the delivery mechanism of paging request messages more efficient. Since paging in cellular networks is dependent on the specific link technologies, seeking a more efficient mechanism is somewhat redundant. However, a number of different delivery mechanisms (i.e. unicast, multicast, and so on) are available in IP networks. Therefore, it is necessary to determine which mechanism is the best to deliver the paging messages. In previous works, both simple unicast ([1]) and multicast ([2] [3]) were used. Unicast is easy to implement, but it is not an efficient method to page multiple access routers (AR) simultaneously. On the other hand, multicast is an efficient way to page a paging area (PA) consisting of a large number of ARs, however it requires a certain amount of overhead for multicast group management. Therefore, we utilized explicit multicast (xcast) [4] as delivery mechanism. Xcast is a variance of multicast, which is designed for small sized groups. Unlike multicast, xcast does not require group maintenance, and can therefore support simultaneous paging to multiple ARs with minimal overhead. Besides, xcast is an appropriate choice for the selective paging scheme [5], as it dynamically adjusts the size of the PA and thus can minimize the paging cost while meeting the paging delay bound.

The rest of this article is organized as follows. Section 2 proposes the selective IP paging scheme utilizing xcast. In Section 3, we develop an analytic model to evaluate the proposed scheme. Section 4 shows the numerical results and Section 5 concludes this paper.

2 Selective IP Paging Scheme Using Explicit Multicast

2.1 Protocol Overview

In this section, we present an overview of the selective IP paging scheme utilizing explicit multicast. When a paging request is destined for an MH, a paging agent receives the request and sends the paging request to the ARs in order to find the MH. In the selective paging scheme, the number of paging steps to be performed needs to be determined first, before sending a paging request to the ARs, by considering paging delay bound. Let M be the number of paging steps. When determining the value of M, we should consider the paging delay bound and N^*, the feasible group size, which is the number of group members that will enable xcast perform to better than typical multicast in terms of the paging cost.

Once the above paging step has been performed, the PA is divided into M sub-PAs and the paging agent sends the first paging request packet to the sub-group denoted by A_0. If the destination MH is in A_0, the paging procedure is terminated. Otherwise, a second paging procedure should be performed for the second sub-PA, denoted by A_1. These procedures are repeated until the destination MH is found out.

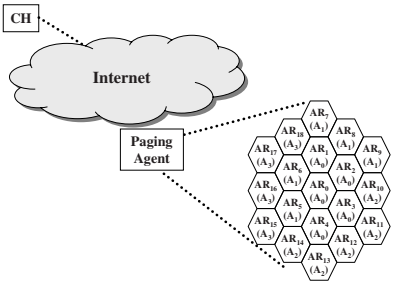


Fig. 1. Protocol overview: Selective IP paging scheme using xcast

Fig. 1 provides an illustrated example of the selective paging scheme in a PA consisting of 19 ARs. It is assumed that the number of paging steps (M) is 4. In the selective paging scheme, the PA is divided into sub-PAs based on the location where the MH recently registered. The detailed method used to determine the size of the sub-PAs and the members of these sub-PAs will be presented in the next subsection.

Let's assume that a correspondent host (CH) would like to initiate a session with an MH, which registered at AR_0 last but has moved to AR_{11} without any location registration because the MH remained in the idle state. First, the paging agent receiving the paging request forwards the request to all of the ARs in the first sub-PA (A_0) using xcast. If there is no response, the paging agent performs the second paging procedure to the sub-PA (A_1). Since the MH is currently located in AR_{11}, the MH responds to the third paging request sent to the third sub-PA (A_2), and at this point the paging procedure is finished.

2.2 Determination of Paging Group Size and Paging Step

Before partitioning a PA into multiple sub-PAs for the selective paging scheme, we have to determine the value of M, the number of paging steps. To determine M, the paging delay bound and the maximum feasible xcast group size (N^*) need to be taken into consideration. As mentioned above, xcast is a more light-weight delivery scheme than multicast, because there is no management cost involved. However, if the group size exceeds a certain value, the packet size becomes too large and this affects the packet processing cost traffic load in wireless access networks. Hence, the maximum feasible group size in xcast should be set to a reasonable value by the network administrator. Once N^* has been determined, we can decide the value of M using the following relationship where D is the paging delay bound and M is an integer value.

$$N_{AR}/N^* \leq M \leq D$$

For example, if there are 61 ARs in a paging area, N^* is 20 and paging delay bound is 5, then M can be either 4 or 5. If the paging delay is more sensitive

factor, we choose the value of 4 for M. On the other hand, if the paging cost is more important factor, M is set to 5. In addition, there are two types of grouping algorithms. One ($G1$) is an algorithm in which the group size is set to $\lfloor N_{AR}/M \rfloor$ and the other ($G2$) is an algorithm in which the group size is set to N^*. For example, if M is 4, the 61 ARs in the example paging area are divided into sub-PAs consisting of 15 ARs, 15 ARs, 15 ARs, and 16 ARs in the case where $G1$ used. In contrast, in $G2$, the 61 ARs are divided into sub-PAs consisting of 20 ARs, 20 ARs, 20 ARs, and 1 AR. $G1$ is more beneficial if the goal is to reduce the paging cost, whereas $G2$ is more advantageous if the objective is to reduce the paging delay. Therefore, in this paper, we utilized both $G1$ and $G2$ as grouping algorithms and compared their performance.

2.3 Paging Group Construction by Partitioning Paging Area

In the previous section, we discussed how to decide the size of the sub-PA groups scheme. Once this has been done, we need to construct the sub-PA groups In this section, we propose a PA partitioning algorithm, which allows the division of a PA into various sub-PAs, which are numbered from A_0 to A_{M-1}.

In the selective paging scheme, A PA is divided into sub-PA groups based on the geographical cell topology [7]. The innermost cell (In this paper, the term *"cell"* refers to the coverage of an AR.) is labeled "0". Cells labeled "1" form the first ring around cell "0" and so forth. In general, r_k ($k \geq 0$) refers to the kth ring away from the cell "0". Let $n(r_i)$ be the total number of cells from ring r_0 to ring r_i.

In [6], the shortest-distance-first (SDF) was proposed for selective paging in dynamic location management. In SDF, since there is no such concept as the feasible group size, sub-PAs are divided into based on the number of rings. In contrast, in the selective paging scheme using xcast, the sub-PAs are constructed by considering the maximum feasible number of sub-PA group members, which can be calculated as explained before.

Algorithm 1 shows how to partition a PA into M sub-PAs. When $n(A_0)$ is greater than $n(r_0)$, all of the cells in r_0 and several of the cells in r_1 become the members of A_0. In such a case, various criteria can be used to select the cells in r_1 which are become members of A_0. For example, one of the choices might be to select those ARs with smaller paging delay. In this paper, the ARs are selected in a random manner for the simplicity of analysis. This grouping procedure is repeated until all of sub-PAs have been constructed.

2.4 Paging Operation

The last step is to perform terminal paging to the sub-PAs constructed using the Algorithm 1. In order to find an MH in idle state, the paging agent sends a paging request to the ARs of each sub-PA. The paging agent keeps on sending paging requests until it receives the paging response from the MH being sought as shown Algorithm 2. When an MH receives a paging request from the paging agent, the MH checks whether or not it is still located in the same AR in which it

Algorithm 1 Paging Group Construction

```
1: i ← 0; j ← 0;
2: initiate n(A_j);
3: while j < M do
4:   if n(r_i) ≥ n(A_j) then
5:     j++;
6:     transfer n(A_j) of the cells in r_i to A_j;
7:     n(r_i) = n(r_i) − n(A_j);
8:   else
9:     i++;
10:    transfer all of the cells in r_i to A_j ;
11:    n(A_j) = n(A_j) − n(r_i);
12:  end if
13: end while
```

Algorithm 2 Paging Operation

```
1: i ← 0;
2: while i < M do
3:   Paging agent sends requests to all ARs in A_i;
4:   if MH in A_i then
5:     MH sends response; break;
6:   end if
7:   i++;
8: end while
```

registered last. If the MH is still located in the same AR, the MH sends a paging response back to the paging agent without any registration. Also, the MH sets its state to the active state and restarts its active timer. Otherwise, the MH starts the registration procedure. Following registration, the MH responds to the paging request by sending a paging response message.

3 Performance Evaluation

3.1 Mobility Model

To evaluate the performance of the selective IP paging scheme using xcast, we developed an analytic model. In terms of the user mobility model, we used the random walk mobility model on the mesh and hexagonal cell configurations [7]. Let $p^K_{(x,y),(x'y')}$ be the transition probability that an MH in cell (x, y) moves to cell (x', y') after K movements. $p^K_{(x,y),(x'y')}$ can be obtained from $\alpha((x, y), (x', y'))$, which denotes the probability that an MH moves from cell (x, y) to cell (x', y').

We use two cell classification schemes proposed in [7] (mesh) and [8] (hexagonal). Based on these classifications, it is possible to draw a state diagram and to find the state transition probability. From the transition probability matrix,

Table 1. Cost comparison

Cost	Unicast	Multicast	Xcast
C_L	$N \cdot L_u \cdot S_u \cdot \alpha$	$L_m \cdot S_m \cdot \alpha$	$L_x \cdot S_x \cdot \alpha$
C_N	$(L_u + 1) \cdot \beta$	$(L_m + 1) \cdot \beta$	$N \times \beta + \sum G \times \beta$
C_M	0	$P_V \cdot \left(\frac{1}{\lambda_s} \cdot r_m \cdot \alpha\right) + (1 - P_V) \cdot (\theta_{Tree} + T \cdot r_m \cdot \alpha)$	0

Table 2. Definition of Terms

α	unit cost when a unicast packet is transmitted over a wired link
β	unit processing cost incurred at an intermediate node
L_u	the average length of a unicast routing path
$L_m(L_x)$	the total length of a multicast(xcast) distribution tree
$S_u(S_m,S_x)$	the relative paging request packet sizes in relation to the size of a unicast packet in case of unicast(multicast,xcast)
P_v	probability that a valid distribution tree exists when a paging request arrives
r_m	the message delivery rate required to maintain the group membership
θ_{Tree}	tree construction cost
λ_s	parameter of Poisson process
T	valid time of a distribution tree is alive

the probability that an MH resides in the cell (x, y) after K movements can be calculated. Detailed equations and derivations can be found in [11].

3.2 Unit Paging Cost

To determine the different paging costs when the various delivery schemes are utilized, the unit paging cost for each scheme should be determined in advance. To do this, we formulate the unit paging cost ($C_T(N)$) when the paging group size is N using the comparative results listed in Table 1 [11]. Also all terms can be defined in Table 2.

Since the transmission cost is generally much larger than the processing cost [9], α and β are set to 10 and 1, respectively. L_m and L_x can be calculated from the relationship, $L_m(or\ L_x) = N^\kappa \cdot L_u$ [9]. The message delivery rate for group management (r_m), is dependent on the type of multicast protocol. We assumed that PIM-SM is used in order to reduce the management overhead. Since there is no periodical message exchange in PIM-SM, r_m is set to 0. θ_{Tree} can be approximated to $N \cdot L_m$ because all of the ARs belonging to the paging group send a join message to the paging agent, which serves as a core node in the multicast tree. In addition, λ_s and T are set to 0.01 and 120, respectively. Table 3 shows the parameter values for the unit paging cost.

3.3 Paging Cost

In this paper, we focus on the design of an efficient delivery scheme to reduce the paging cost. Therefore, we didn't consider the location update cost and we assume that this cost is identical in the three different delivery schemes (i.e. unicast, multicast and xcast).

Table 3. System parameter values

α	β	L_u	r_m	κ	λ_s	T	$S_u(S_m)$
10	1	5	0	0.8	0.01	120	1.0

On the other hand, the paging cost is proportional to the unit paging cost and the number of ARs to be paged. The unit paging cost in each delivery scheme is a function of the number of ARs to be paged as mentioned in the previous section. Let $C_P((x,y))$ be the paging cost when the AR where the MH most recently updated its location is (x,y). The paging cost for the first sub-PA, A_0, is as follows:

$$C_T(n(A_0)) \cdot \sum_{(x',y') \in A_0} \alpha((x,y),(x',y'))$$

On the other hand, the paging cost for the second sub-PA, A_1, is the sum of the paging costs of the first and second sub-PAs. This is because the second paging step is performed only after the first paging step is finished and when the destination MH was not found in the first paging step. Based on this relationship, the total average paging cost $(C_P(x,y))$ when the last registered AR is (x,y) can be expressed as Eq. 1.

$$C_P((x,y)) = \sum_{i=0}^{M-1} \sum_{j=0}^{i} C_T(n(A_j)) \cdot \sum_{(x',y') \in A_j} \alpha((x,y),(x',y')) \tag{1}$$

where M is the number of paging steps and A_i is the i-th sub-PA. $(n(A_i))$ denotes the number of ARs belonging to the i-th sub-PA. $C_T(n(A_i))$ is a delivery cost function when the group size is $(n(A_i))$.

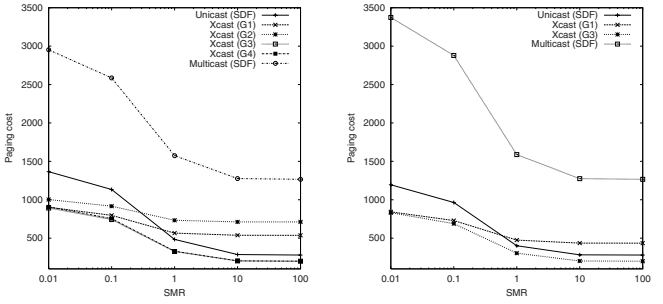
Let's assume that the probability that an MH updated its location in cell (x,y) follows a uniform distribution in $[1, N(n)]$. $N(n)$ denotes the total number of ARs within the areas by n-th ring. In addition, let $N(x,y)$ be the number of ARs of cell type (x,y). Hence, the mean paging cost can be calculated as Eq. 2.

$$C_P = \sum_{all \quad (x,y)} C_P((x,y)) \cdot \frac{N(x,y)}{N(n)} \tag{2}$$

4 Numerical Result

The proposed IP paging scheme can be used not only with the static location management scheme but also with the dynamic location management scheme [10]. However, in this analysis, it is assumed that the dynamic location management scheme (e.g., movement-based, distance-based, or timer-based) is used. Therefore, (x,y), which is the last location updated cell, is simply $(0,0)$. Also we assume that the cell residence time has a Gamma distribution.

To evaluate the performance of the proposed paging scheme, we compare the paging costs when xcast, unicast and multicast are used. The paging schemes



(a) $N^* = 20$ and $M = 4$ (b) $N^* = 20$ and $M = 5$

Fig. 2. Paging cost - Mesh configuration and $N^* = 20$

using unicast and multicast use SDF for the paging group construction. In contrast, xcast is incorporated with G1 and G2 as mentioned above.

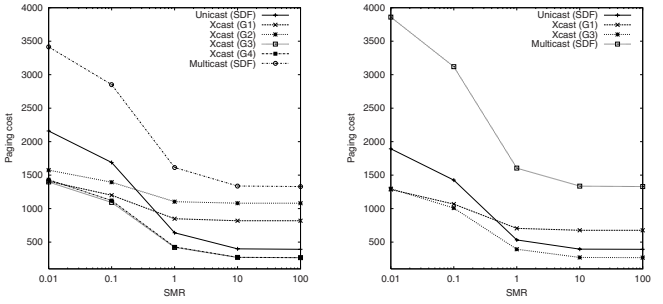
Fig. 2 shows the paging cost as the the session-to-mobility ratio (SMR) is changed under the mesh cell configuration. As the SMR increases, the paging cost decreases. This is because a large SMR implies that an MH's mobility is low (i.e., static MH). The static MH is likely to be connected to an AR in the vicinity of the one to which it was last registered when a paging procedure is invoked. Therefore, the paging cost decreases as SMR increases.

For the purpose of comparison, we defined the paging cost gain (G_{pc}) of scheme A as follows:

$$G_{pc} = \frac{Paging\ \ cost\ \ of\ \ scheme\ \ A}{Paging\ \ cost\ \ of\ \ multicast}$$

The G_{pc} of G1 and G2 are 0.30 and 0.34, respectively, when the SMR is 0.01. However, the gain decreases to 0.42 and 0.56 when the SMR is 100. On the other hand, the G_{pc} values of unicast based on SDF are 0.46 and 0.22 when the SMR is 0.01 and 100, respectively. In other words, the paging cost of unicast can be lower than that of xcast using G1 and G2 when SMR is high.(Of course, this result doee not mean that the paging cost of unicast is necessarily less than that of xcast because, in our calculation, we assumed that the transmission cost is 10 times the processing cost. However, the transmission cost may be much higher than the processing cost due to advances in processing technology [9].) This is because xcast using G1 and G2 creates a sub PA with a fixed size of (N^* or $\lfloor N_{AR}/M \rfloor$). However, SDF makes a sub-PA based on the ring area. These values are much smaller than N^* or $\lfloor N_{AR}/M \rfloor$, which are typically set to a value between 10 and 30. As mentioned above, the probability that an MH remains connected to an AR in near to the most recently registered AR, increases as the SMR increases. Hence, it is wasteful to make a sub PA with a larger fixed group size, in the case of xcast using G1 or G2. As a result, the paging costs of G1 and G2 can be higher than that of unicast based on SDF.

To overcome these drawbacks, we propose two enhanced grouping algorithms called G3 and G4. Unlike G1 and G2, the first sub-PA in G3 and G4 consist of



(a) $N^* = 30$ and $M = 4$ (b) $N^* = 30$ and $M = 5$

Fig. 3. Paging cost - Hexagonal configuration and $N^* = 30$

ARs located in ring 0 and 1. However, subsequent sub-PAs are constructed using the same grouping algorithms, G1 and G2. Fig. 2 shows the paging cost in the case of G3 and G4. The G_{pc} values of G3 and G4 are 0.30 and 0.30, respectively, when the SMR is 0.01. Besides, the G_{pc}, when SMR is 100, is 0.16 for both G3 and G4. Namely, the performance gain of xcast using G3 and G4 is higher than that of unicast using SDF.

The cost variation in other cases (e.g. $N^* = 20$ and $M = 5$) is almost the same as the result of the case where $N^* = 20$ and $M = 4$. In the case of where $N^* = 20$ and $M = 5$, there are no sub-PAs satisfying the grouping algorithm, G2. Therefore, only G1 and its enhancement, G3, are compared.

In terms of the hexagonal cell configuration, N^* is set to 30 because a 6-area location area is composed of 91 cells. In the case of $N^* = 30$ and $M = 3$, unicast exhibits a smaller paging cost than G1 and G2, when the SMR is larger than 1 and 0.1, respectively. However, G3 and G4 exhibit more smaller paging costs than unicast.

When comparing G1 (G3) with G2 (G4), G1 (G3) exhibits a lower paging cost than G2 (G4). Hence, G1 (G3) is a more suitable choice to minimize the paging cost .

5 Conclusion

In this paper, we introduced the selective IP paging scheme utilizing explicit multicast (xcast). The proposed paging scheme is more cost effective than the existing schemes based on unicast and multicast in terms of the paging cost. In order to support IP paging in IP-based wireless/mobile networks using xcast, we proposed two types of grouping algorithms and their enhancements. The numerical results indicated that selective IP paging scheme based on xcast incurs only 16-56% of the paging cost, in the case of the IP paging scheme using multicast or unicast based on SDF

References

[1] X. Zhang et al., "P-MIP: Paging Extensions for Mobile IP," ACM Mobile Networks and Applications, Vol. 7, No. 2, pp. 127-141, Apr. 2002. 154, 155
[2] R. Ramjee et al.,"IP Paging Service for Mobile Hosts," ACM/Baltzer Wireless Networks, Vol. 8, No. 5, pp. 427-441, Sep. 2002. 154, 155
[3] C. Castelluccia, "Extending Mobile IP with Adaptive Individual Paging," ACM Mobile Computing and Communication Review, Vol. 5, No. 2, pp. 14-26, Apr. 2001. 154, 155
[4] R. Boivie et al., "Explicit Multicast (Xcast) Basic Specification," Internet Draft, Work in Progress, Jan. 2003. 155
[5] W. Wang et al., "Effective Paging Schemes with Delay Bounds as QoS Constraints in Wireless Systems," ACM/Baltzer Wireles Networks, Vol. 7, No. 5, pp. 455-466, Sep. 2001. 155
[6] I. F. Akyildiz et al., "Movement-Based Location Update and Selective Paging for PCS Networks," IEEE/ACM Transaction on Networking, Vol. 4, No. 4, pp. 629-638, Aug. 1996. 157
[7] I. F. Akyildiz et al., "A New Random Walk Model for PCS Networks," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 7, pp. 1254-1260, July 2000. 157, 158
[8] G. Xue, "An Improved Random Walk Model for PCS Networks," IEEE Transaction on Communications, Vol. 50, No. 8, pp. 1224-1226, July 2002. 158
[9] J. Chung et al., "Pricing Multicast Communication: A Cost-Based Approach," Telecommunication Systems, Vol. 17, No. 3, pp. 281-297, July 2001. 159, 161
[10] I. F. Alyildiz et al., "Mobility Management in Next-Generation Wireless Systems," Proceeding of IEEE, Vol. 87, No. 8, pp. 1347-1384, Aug. 1999. 160
[11] Kyoungae Kim,"Cost-effective Paging Scheme in IP-based Mobile Network," MS thesis, School of CSE, Seoul Nat'l University, Korea, 2003 159

An Enhanced Handoff Mechanism for Cellular IP

Kyung-ah Kim[1,2], Jong-deok Kim[3], Chong-kwon Kim[1], and Jae-yoon Park[2]

[1] School of Electrical Engineering and Computer Science
Seoul National University, Seoul, Republic of Korea
{kka,ckim}@popeye.snu.ac.kr
[2] R&D Group, KT, Seoul, Republic of Korea
{kka1,jypark60}@kt.co.kr
[3] Telcoware Co.,Ltd. Seoul, Korea
jdkim@telcoware.com

Abstract. Handoff is one of the most important factors that may degrade the performance of TCP connections in wireless data networks. The authors present a lossless handoff scheme called LPM (Last Packet Marking). LPM signals the safe handoff cue by sending a specially marked packet to mobile hosts. Our performance study shows that LPM achieves lossless packet delivery without duplication and increases TCP throughput significantly.

1 Introduction

Currently, there are many efforts underway to provide Internet services on integrated wireless and wireline networks. Supporting efficient IP mobility is one of the major issues to construct IP-based wireless access networks. Mobile users will expect the same level of service quality as wireline users. Even though the access point of mobile user changes, IP connections should be continued transparently. The Mobile Internet Protocol [1] is current standard for supporting global IP mobility in simple and scalable manner. But, in processing frequent handoffs in cellular based wireless access networks, Mobile IP has some limitations. After each migration, a local address must be obtained and communicated to a possibly distant home agent. This incurs increasing handoff latency and load on the global Internet. And mobile user suffers service degradation in handoff period.

A number of solutions [2, 3, 4] have been discussed these problems. These approaches are extending Mobile IP rather than replacing it. To handle local movement of mobile hosts without interaction with the Mobile-IP-enabled Internet, they adopt a domain-based approach. That is, these intra-domain protocols are used for establishing and exchanging the state information inside the wireless access networks, so as to get fast and efficient intra-domain mobility or *micro-mobility* control.

Among these micro-mobility protocols, Cellular IP [4, 5, 6, 7] attracts special attention for its seamless mobility support in limited geographical areas. Since COA (Care-of-Address) is not changed in local mobility control, it eliminates the

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 164–173, 2004.

load for location update in the global Internet. And by using paging concept, Cellular IP provides simple location tracking scheme called cheap passive connectivity, which imposes neither traffic nor processing load on the global network as long as the host is idle, and preserves the power reserves of mobile nodes.

Previous study [8] showed that the performance of TCP degrades significantly due to frequent handoff in small cell wireless networks. Cellular IP introduces *semisoft* handoff mechanism to reduce the number of lost packets. It establishes a routing path to the new BS (Base Station) before handoff, and temporarily bi-casts data to the both old (current) and new BS. The throughput of TCP using semisoft handoff is much better than that of hard handoff. Nonetheless, packet loss or packet duplication still can occur in Cellular IP semisoft handoff, which results in the degradation of TCP performance.

In this paper, we propose a simple handoff scheme called LPM (Last Packet Marking). LPM gives exact cue to mobile host for safe handoff to remove packet loss or packet duplication. This paper is structured as follows. In section 2, we briefly preview the Cellular IP handoff mechanism. In section 3, we describe last packet marking method for improving Cellular IP semisoft handoff. In section 4, we verify LPM by computer simulations. Finally, we conclude in section 5.

2 Cellular IP

In Cellular IP [4, 5, 6, 7], a group of BSs (Base Stations) forms one access network. Each access network attaches to the Internet via a gateway router. A BS, which provides wireless access service to MHs (Mobile Hosts), is a special-purpose router with mobility-related functions. Cellular IP assumes the tree network topology to simplify routing within an access network. An MH, which visits a certain access network, uses the IP address of the gateway router as its COA (Care Of Address). Within an access network, the MH is identified by its home address. Since COA is not changed in local mobility control, Cellular IP eliminates the load for location update in global Internet.

Let us examine Cellular IP handoff schemes briefly. Cellular IP proposes *hard* and *semisoft* handoff. The *hard* handoff scheme, the basic handoff mechanism of Cellular IP, makes a new route to the new BS after real handoff. Hard handoff suffers from severe packet losses and results in performance degradation. To provide adequate performance to both TCP and UDP traffic while maintaining the lightweight nature of the base Cellular IP protocol, Cellular IP introduced a new handoff method called *semisoft* handoff (Fig. 1).

Prior to make a real handoff, an MH makes a brief connection to a new BS and sends a *semisoft request* packet. The semisoft request packet is forwarded toward the gateway router and eventually it reaches a crossover router which is a branching point of the path to the old BS and the path to the new BS. Receiving the semisoft request packet, the crossover node updates its route cache by adding the path to the new BS and starts to buffer packets destined to the MH in a delay device. The MH makes a real handoff after a pre-determined *semisoft delay*. The MH issues a *routing update* packet to the crossover node after the real handoff.

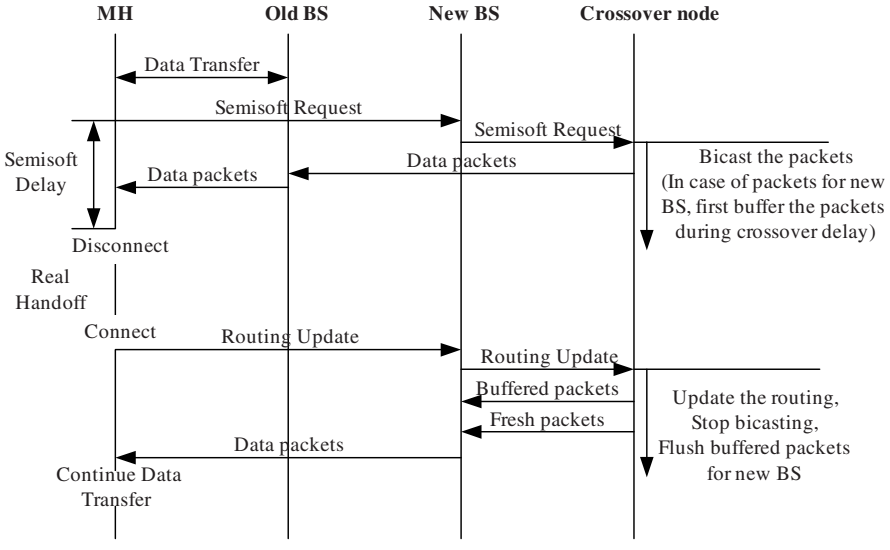


Fig. 1. Cellular IP Semisoft Handoff

The crossover node updates the route cache and transmits packets buffered in a delay device and packets arrived thereafter.

Since optimal semisoft delay cannot be predicted accurately, Cellular IP sets the semisoft delay as the worst-case value that is proportional to the mobile-to-gateway round-trip delay. On various network topology and network traffic, applying constant semisoft delay may cause an MH to receive duplicated packets or to suffer packet losses. Duplicated packets do not disrupt many applications, but packet loss limits TCP's performance severely. To eliminate the packet loss, Cellular IP temporarily introduces a constant delay along the new path between the crossover and the new BSs using a delay device on the crossover node. As a result, just after a handoff, most MH suffers packet delay or duplication, which can cause quality degradation of real time traffic or can trigger TCP congestion control. And depending on the network topology, delay device may not be sufficient to hold data packets during handoff. So, packet loss still can occur in Cellular IP semisoft handoff, which results in the degradation of TCP performance.

3 Last Packet Marking

We develop a new handoff scheme called LPM (Last Packet Marking) which can signals exact handoff initiation time to an MH. Let us explain the proposed mechanism (Fig. 2).

Before make a regular handoff, an MH sends semisoft request packet to the new BS as in Cellular IP. When a crossover node receives the semisoft request,

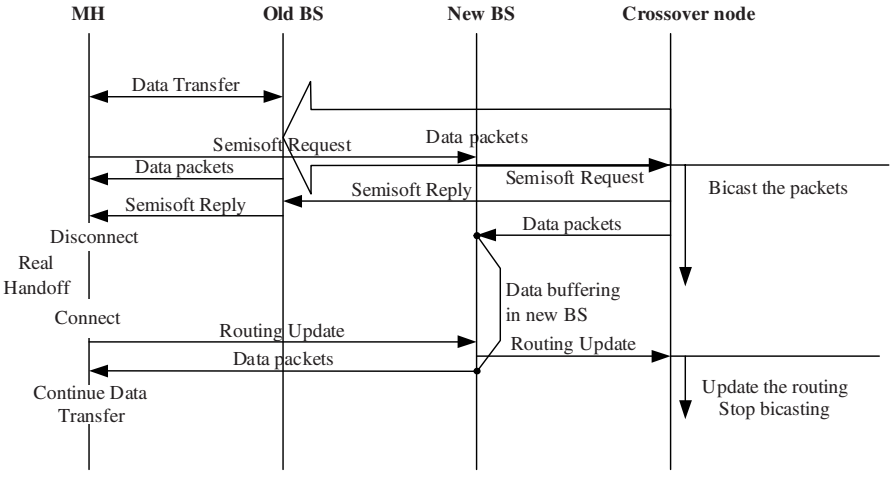


Fig. 2. LPM

it creates a mapping for the new path and transmits *semisoft reply* to the old BS. As the *semisoft reply* assures the MH that it has received all the packets that can be received only through the old BS and it can receive following data packets from the new BS. So the semisoft reply can be used as a trigger for safe handoff by the MH. Therefore, receiving the semisoft reply through the old BS, MH initiates real handoff immediately rather than waiting for the expiration of constant semisoft delay as in Cellular IP. After sending this signal, the crossover node bi-casts data packets to both the new BS and the old BS. The new BS receives packets, which follow the semisoft reply, and it buffers packets for fast delivery.

The purpose of bi-casting is to prevent packet losses when an MH initiates semisoft request but abandons real handoff, which is not shown in Fig. 2. The path through the new BS will be automatically timed out by the Cellular IP soft state routing update strategy and data will be unicasted only to the old BS.

The required buffer size in new BS is proportional to the difference between the delay to the old BS and the delay to the new BS from the crossover node. The BSs are able to know the delay differences in setup time by exchanging control information and set the maximum buffer size that can store data delivered during maximum delay difference plus some margin including link layer handoff time.

Cellular IP introduces the crossover delay to synchronize the delay difference between the new path and the old path from the crossover node in case the new path is shorter than the old path. However this may introduce undesirable additional delay in many situations. LPM makes another distinction in that the crossover node sends data packets to the new BS immediately instead of buffering packets in its delay device for the constant crossover delay. In case packets sent to the new path arrive the new BS before the handoff, the new BS buffer them

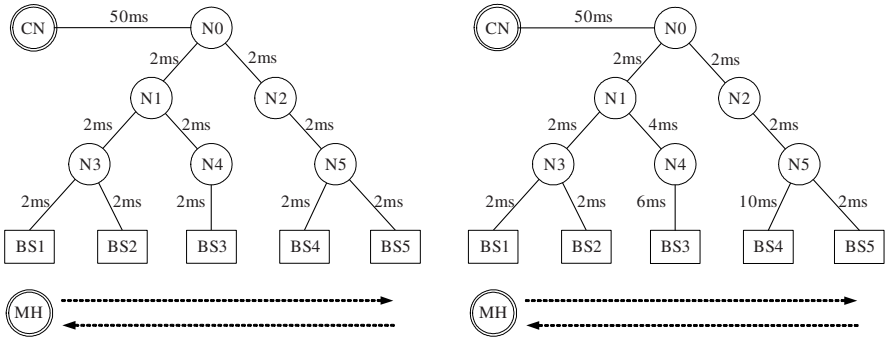


Fig. 3. Simulation topology. left: (a) Topology 1, right: (b) Topology 2

for fast delivery after handoff. This approach eliminates undesirable packet delay and packet duplication in many cases.

Cellular IP does not place any restriction on delaying a real handoff for the constant semisoft delay. However delaying a handoff would be restricted in real situations. Therefore there would be situations that an MH should make a real handoff even if it does not receive the semisoft reply, or it could make a real handoff only after it has received several data packets after the semisoft reply. The former case may result in the packet loss and the latter case may result in the packet duplication. To overcome these, we propose packet forwarding to the new BS by the old BS for the former case and packet duplicate elimination at the new BS for the latter case. The old BS uses the semisoft reply as a delimiter for the packet forwarding to the new BS. That is, packets arrived at the old BS after the last packet mark will not be forwarded, which alleviates the forwarding overhead compared to the general forwarding mechanism proposed. As there is no sequence number in IP Header, the new BS uses a hash function to identify and eliminate the duplicated packets from its buffer.

Compared to Cellular IP, LPM only introduces a simple handoff signal, semisoft reply, that enables MH to adapt to the network topology and dynamics instead of using fixed parameters like semisoft, crossover delays. In case delaying a handoff is constrained, which is not considered in Cellular IP, LPM makes use of efficient packet forwarding and duplicate elimination mechanisms which help to recover from packet loss and duplication.

4 Simulation Results

We used computer simulation for performance analysis. Two network topologies are considered. One is shown in Fig. 3(a) where all possible MH-to-Gateway delays are same, and possible MH-to-Crossover node delays are also same before/after handoff. This topology is used for the performance study of Cellular IP [6, 7]. Another topology is shown in Fig. 3(b) where neither possible MH-to-Gateway delays nor possible MH-to-Crossover node delays before/after handoff

Table 1. UDP results: number of lost, duplicated and disoredered packets on topology 1. (Lost packets / Duplicated packets / Disordered packets)

	BS1 to BS2	BS2 to BS3	BS3 to BS4	BS4 to BS5	BS5 to BS4	BS4 to BS3	BS3 to BS2	BS2 to BS1
Hard	8/0/0/	10/0/0	13/0/0	9/0/0	7/0/0	12/0/0	10/0/0	7/0/0
Semisoft	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4
LPM	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0

Table 2. UDP results: number of lost, duplicated and disoredered packets on topology 2. (Lost packets / Duplicated packets / Disordered packets)

	BS1 to BS2	BS2 to BS3	BS3 to BS4	BS4 to BS5	BS5 to BS4	BS4 to BS3	BS3 to BS2	BS2 to BS1
Hard	7/0/0/	12/0/0	19/0/0	12/0/0	12/0/0	19/0/0	12/0/0	7/0/0
Semisoft	0/0/4	0/3/6	0/1/5	0/0/3	0/0/3	0/0/3	0/0/3	0/0/4
LPM	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0

are same. Considering dynamic network delay due to short burst characteristics of data traffic and competing backgrounds traffics, we assert that this one would be more realistic. N1 through N5 are Cellular IP nodes and N0 is the gateway. Each wired connection between Cellular IP nodes is modeled as 10 Mb/s duplex links with 2 ms delay. CN (Correspondent Node) transmits UDP or TCP traffic to MH from time 3. An MH oscillates between BS1 and BS5 at the constant speed from time 5. The MH stays for about 10 seconds before moving to the next BS. Cellular IP semisoft delay is fixed to 50 ms, and crossover delay to 10 ms. Link Layer handoff delay is 10 ms. Mobile hosts connect to AP (Access Point) using the ns-2 CSMA/CA 2Mb/s wireless link model. We use micormobility extension for the ns-2 network simulator based on version 2.1b6 [9]. UDP traffic is directed from CH to MH and consists of 210 bytes packets transmitted at 2 ms intervals. TCP Reno congestion control is used for TCP connection.

4.1 UDP Results

Table 1 and 2 compares the performance of UDP traffic with three handoff schemes: hard, semisoft and LPM.We used the number of packet losses, duplications and disordering as the performance criteria. Note that LPM has no packet loss, duplication or disordering. In hard handoff, it always suffers from the packet loss. The number of lost packets is proportional to the sum of the transmission delay from the new BS to the crossover node and the transmission delay from the crossover node to the old BS. As shown in Table 1, no loss or duplication

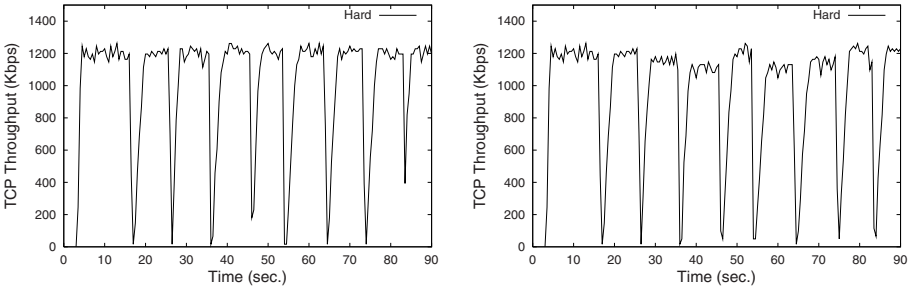


Fig. 4. TCP throughput by hard handoff. left: (a) topology 1, right: (b) topology 2

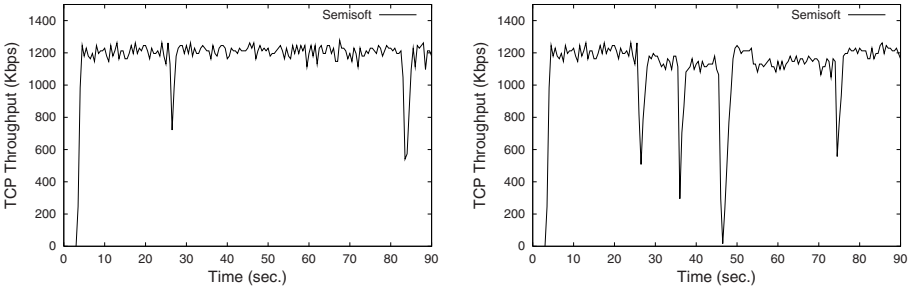


Fig. 5. TCP throughput by semisoft handoff. left: (a) topology 1, right: (b) topo. 2

is occurred in semisoft handoff. It means that the semisoft delay and cross over delay is set properly for topology 1. Disordered packet is occurred by simultaneous packet forwarding in crossover node just after handoff, namely the packets in the delay device and newly arrived packets. In real time UDP traffics, this disordering can be ignored by buffering in application layer. But this characteristic of semisoft handoff degrades the TCP throughput. In topology 2, semisoft handoff suffers either from the packet loss or from the packet duplication. As fixed semisoft and crossover delay do not adapt to the network dynamics, they are too small (packet loss) in some cases and too large (packet duplication) in other cases.

4.2 TCP Results

Figure 4 through 6 show the TCP connection throughput as a function of time. The TCP throughput is measured every 0.5 second. All hard handoff has abrupt glitches caused by lost packets. It is well known that a packet loss decreases the TCP performance significantly due to the TCP congestion control. The most distinguishing difference between the performances of semisoft handoff and LPM is that semisoft has the occasional abrupt glitches while LPM shows no precip-

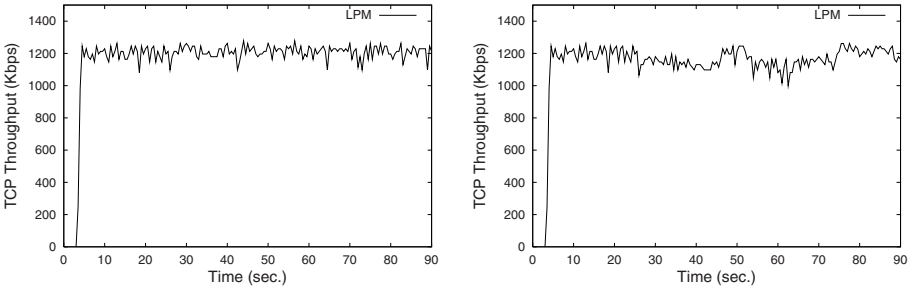


Fig. 6. TCP throughput by LPM. left: (a) topology 1, right: (b) topology 2

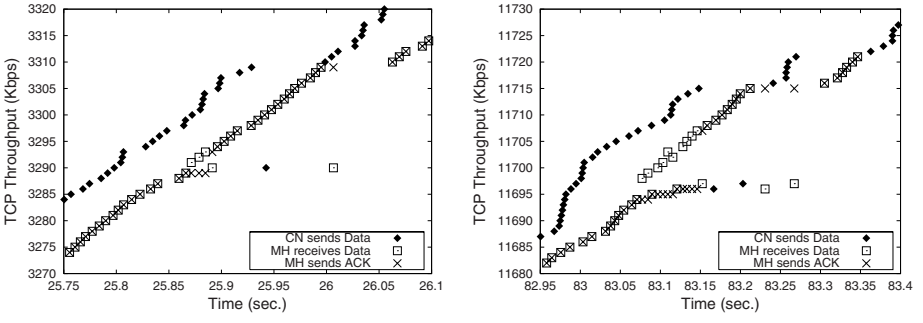


Fig. 7. Sender and receiver traces of TCP connection by semisoft handoff on topology 1. Crossover delay is 10 ms. At 25.89, sequence no. 3290 is disordered and from 83.09 to 83.15, sequence no. 11695 through 11697 are disordered

itous change. Although semisoft handoff improves the number of abrupt drops, semisoft can't get rid of all drops. On the other hand, LPM shows no throughput drops on any handoff. The plateaus of throughput are changed after each handoff in topology 2, since the RTTs to CH is changed for each handoff. TCP throughput is disproportional to the RTT of TCP connections. One example of sender and receiver packet trace for LPM during handoff is shown in Fig. 10. Other packet traces for LPM during handoff are same style.

All throughput drops of semisoft handoff in topology 2 are caused by packet losses (Fig. 8). The crossover delay (10 ms) is not sufficient to delay packets during handoff. Before the MH hands off to the new BS, packets are already arrived to the new BS and these packets are lost. But increasing crossover delay is not advisable too. Because on other conditions like further new path, increasing crossover delay incurs more duplicated ACKs or reordered packets that degrades TCP performance. Figure 9 is semisoft handoff trace on topology 2, except that crossover delay is doubled (20 ms). Although we get rid of the packet loss, but packet disordering is introduced.

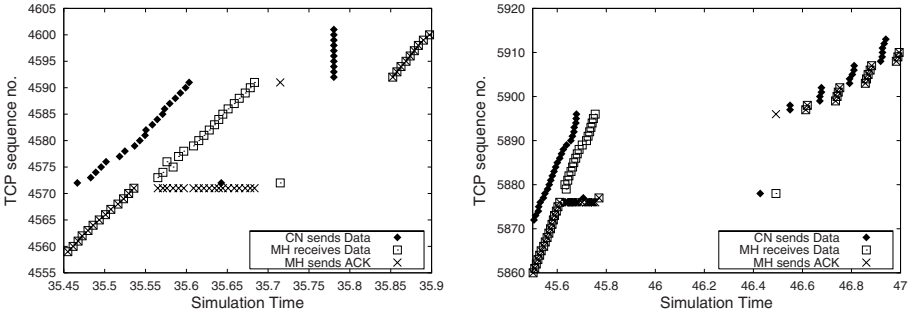


Fig. 8. Sender and receiver traces of TCP connection by semisoft handoff on topology 2. Crossover delay is 10 ms. At 35.55 sequence no. 4572 is lost and at 45.62, sequence no. 5877 and 5878 are lost

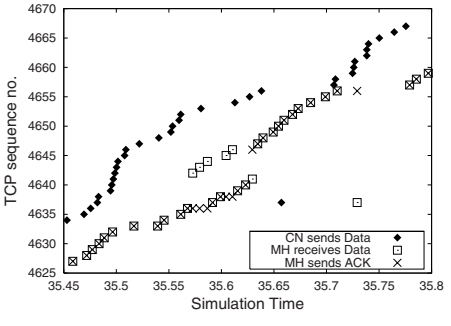


Fig. 9. Sender and receiver traces of TCP connection by semisoft handoff on topology 2. Crossover delay is 20 ms. From 35.59 to 35.63, sequence no. 4637 through 4641 are disordered

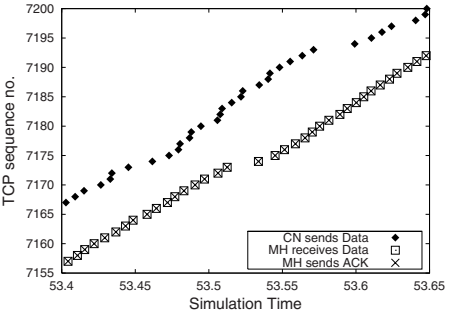


Fig. 10. Sender and receiver traces of TCP connection by LPM on topology 2. Crossover delay is 10 ms. From 53.52 to 53.54 MH hands off from BS5 to BS4, where the difference of transmission time between new and old path is the largest

Disordered packets or duplicated packets may also decrease throughput by triggering the fast recovery mechanism of TCP congestion control. As TCP uses the accumulated ACK mechanism, rather than relying only on the time-out mechanism, it uses sequential duplicate ACKs more than 3 times as an indication of packet loss. If a TCP receiver receives packets which had been already received or which has the sequence number that is above the expecting sequence number, it may generate duplicate ACKs that trigger the congestion control of the sender. All throughput drops of semisoft in topology 1 are caused by disordered packets (Fig. 7). This is because the crossover node forwards packets in the delay device and newly arrived packets simultaneously just after handoff.

5 Conclusion

We have proposed a new handoff scheme called LPM (Last Packet Marking) for micro-mobility in wireless packet networks. LPM uses the simple signal to trigger the handoff timing to MHs. We studied the performance of LPM using computer simulation. Our simulation study showed that LPM received all packets without duplication or loss in case of UDP traffic. Also in case of TCP traffic, LPM is free from packet loss and duplication, and its throughput is only affected by the variance of RTT. In future work, we plan to study LPM on S-MIP [10], which provides a way of combining a location tracking scheme (fast-handoff) and the hierarchical MIP style handoff scheme in the IPv6 network, based on Mobile IPv6.

References

[1] C. Perkins : IP Mobility Support, Internet RFC 2002, Oct. (1996) 164
[2] R. Ramjee, T. La Porta, S. Thuel, K.Varadhan and S. Y. Wang: HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless networks. International Conference on Network Protocols, ICNP (1999) 164
[3] Karim El Malki, Hesham Soliman: Fast Handoffs in Mobile IPv4. draft-elmalki-mobileip-fast-handoffs-03.txt, Internet Draft, IETF, Sep. 27 (2000) 164
[4] Andrew T. Campbell, Javier Gomez, Sanghyo Kim, András G. Valkó, Chieh-Yih Wan: Design, Implementation, and Evaluation of Cellular IP. IEEE Personal Communication, Vol. 7, No. 4, Aug. (2000) 42–49 164, 165
[5] András G. Valkó: Cellular IP: A New Approach to Internet Host Mobility. ACM Computer Communication Review, Vol. 29, No. 1, Jan. (1999) 50–65 164, 165
[6] Andrew T. Campbell, Javier Gomez, Sanghyo Kim and Chieh-Yih Wan: Comparison of IP Micromobility Protocols. IEEE Wireless Communications, Vol. 9, No. 1, Feb. (2002) 2–12 164, 165, 168
[7] Andrew T. Campbell, Javier Gomez, Sanghyo Kim, Zoltán R. Turányi, András G. Valkó and Chieh-Yih Wan: Internet Micromobility. Journal of High Speed Networks, Vol. 11, No. 3–4, Sep. (2002) 177–198 164, 165, 168
[8] S. Seshan, H. Balakrishnan, and R. H. Katz: Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience. Wireless Personal Communications, Vol. 4, No. 2, Mar. (1997) 141-162 165
[9] Columbia IP Micromobility Software (CIMS) home page, `http://comet.columbia.edu/micromobility` 169
[10] Robert Hsieh, Zhe Guang Zhou, Aruna Seneviratne: S-MIP: A Seamless Handoff Architecture for Mobile IP. INFOCOM, IEEE, Mar. (2003) 173

A State-Based Fast Handover Scheme for Hierarchical Mobile IPv6

Kiyoung Kim[1], Myung-Kyu Yi[2], Yongtae Shin[1], and Jaesoo Kim[3]

[1] Dept. of Computer Science, Soongsil University
{ganet89,shin}@cherry.ssu.ac.kr
[2] Dept. of Computer Science & Engineering Korea University
kainos@disys.korea.ac.kr
[3] Locus Corporation
jaekim@locus.com

Abstract. To support seamless handover at realtime applications, IETF fast handover scheme was proposed in wireless IP networks by minimizing handover latency. However, when a correspondent node sends packets to the mobile node during the handover period, inefficient routing occurs in hierarchical Mobile IPv6 networks. In this paper, we propose a novel fast handover scheme to minimize the handover latency. In our proposal, a mobile node sends a fast binding update messages to the mobility anchor point instead of the old access router. Thus, our proposal can reduce the handover latency incurred by inefficient routing from old access router to new access router. In addition, our proposal can reduce the duplicate address detection handling time during the handover by maintaining a list of the confirmed new care-of-address beforehand. Analysis results using the discrete analytic model shows that our proposal can have superior performance than existing handover scheme when packet arrival rate and L2 layer switching time are high.

1 Introduction

Mobile IP provides an efficient and scalable mechanism for host mobility within the Internet[1]. Using Mobile IP, mobile nodes may change their point of attachment to the Internet without changing their IP address. Mobile IP has three functional entities in Mobile IPv6 (MIPv6) : Mobile Node (MN), Home Agent (HA), and Correspondent Node (CN). To obtain a Care-of-Address (CoA), the MN can use either stateful or stateless address autoconfiguration when it changes a point-of-attachments. And then, the MN sends a Binding Update (BU) to its HA or other CNs in its list to notify its current CoA to them. A Binding Acknowledgement (BA) is sent to the MN by its HA or any other CN to indicate that the BU was successfully received. If a CN wants to know the CoA of an MN, it sends a Binding Request (BR) to the MN. CNs are usually expected to deliver packets directly to the MN's CoAs, so that the HA is rarely involved with packet transmission to the MN. To support real-time sensitive traffic, Rajeev Koodli[2] proposed the fast handover protocols in Mobile IPv6 networks (FMIPv6) by

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 174–183, 2004.

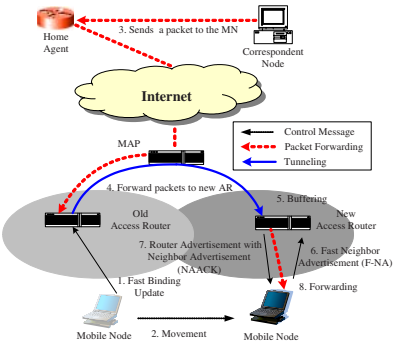


Fig. 1. The IETF Fast Handover Scheme in HMIPv6 networks

minimizing handover latency. The main idea of this approach is that it is to allow an MN to configure a new CoA even before it connects to its new Access Router (nAR). Therefore, the MN can use the new CoA when it connects with the nAR. Although this approach can reduce connectivity and reception latencies, it has a potential disadvantage in terms of the delay and bandwidth efficiency in Hierarchical Mobile IPv6 networks (HMIPv6)[3] as shown in Fig. 1.

Let us consider the case a CN sends a packet to the MN during the MN's handover period. In this case, the CN sends the packet to the Mobility Anchor Point (MAP) first using the RCoA in its binding cache for the MN. And then, the MAP forwards the packet to the old Access Router (oAR) using the LCoA. Upon reception of Fast Binding Update (F-BU) message from the MN, the oAR begins forwarding a packet intended for the MN to the nAR. Finally, the nAR buffers any packets arriving for the MN. After the MN establishes link connectivity with the nAR, the buffered packets are sent to the MN. In this case, a bi-directional tunnel will be establish between oAR and nAR. This could be inefficient in terms of delay and bandwidth efficiency since packets will traverse the MAP-oAR link twice and packets arriving out of order at the MN. To overcome the above problems in HMIPv6, we propose a novel fast handover scheme called FHMIPv6 to reduce the handover delay in HMIPv6. In FHMIPv6, an MN sends a Fast Binding Update (F-BU) messages to the MAP instead of the oAR. Therefore, FHMIPv6 can reduce the handover latency by eliminating inefficient routing. The rest of the paper is organized as follows. In Section 2, we illustrates the system model used in FHMIPv6. Section 3 illustrates the proposed fast handover scheme called FHMIPv6. Section 4 shows the performance analysis and numerical result. Finally, in Section 5, we conclude this paper and discuss some key future research directions.

2 System Model

To minimize packet loss during the handover procedure, we propose to extend the F-BU message with an extra flag 'Simultaneous Bindings ('S') taken from

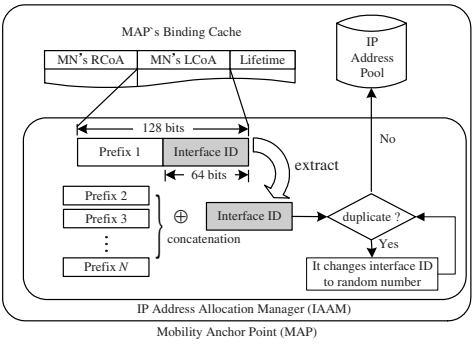


Fig. 2. The IP Address Allocation Manager (IAAM) at the MAP

the reserved field as similar to [4]. If the 'S' flag is set, it indicates a request for bicasting all packets to both CoAs of the MN.

It is well know that the Duplicate Address Detection (DAD) mechanism is used to assure the uniqueness of the address when stateless address autoconfiguration is used[5]. However, the use of DAD adds delays to an HMIPv6 handover. Some works have already investigated so as to deal with the DAD handling issues[6, 7]. However, these approaches did not consider the hierarchical scheme to reduce signaling cost in HMIPv6. Therefore, each AR maintains a generated new CoAs in its address pool or proxy cache for each neighboring ARs. It involves horrendous overhead both in terms of computation and memory space. In FHMIPv6, an address pool for the generated new CoAs is totally maintained at the MAP. To reduce the DAD handling time during the handover, as shown in Fig. 2, each MAP has a new component called *IP Address Allocation Manager* (IAAM) in FHMIPv6. The IAAM is module that generates the confirmed new LCoAs to be used for supporting MIPv6 handover. The IAAM has a list of all nodes and network prefixes on its MAP domains, therefore, it can confirms whether the generated new MN's LCoA is a duplicate or not. As soon as the MN perform registration with the MAP, the IAAM extracts the MN's interface identifier from the MN's LCoA. Based on the MN's interface identifier, the IAAM creates the new LCoAs for the MN beforehand by concatenating it to the another network prefixes on its MAP domains. And then, it confirm whether the generated new MN's LCoAs are a duplicate or not. If the generated new MN's LCoA is duplicated, the IAAM changes the MN's interface identifier to random number. And then, it recreates the new LCoA based on the new MN's interface identifier (i.e., random number). After the creation of the new LCoA, it must be rechecked whether this is a unique address or not. Finally, the IAAM puts the confirmed new LCoAs into its IP address pool. As a result, the IAAM has an IP address pool that maintains a list of the confirmed new LCoAs to be used for supporting HMIPv6 handover. Using the IAAM, DAD handling procedures can be avoided so that the MAP could be immediately available in response to the

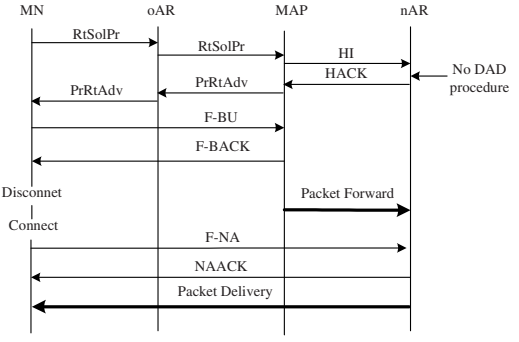


Fig. 3. The Proactive Fast Handover Message Interaction

new LCoA request from the MN. Notice that the above procedure is performed once for all when the MN resides within an MAP domain.

3 Protocol Description

First of all, we assume that the ARs and MAP share necessary security association established. When an MN moves into a new MAP domain, it needs to configure two CoAs: an Resional CoA (RCoA) on the MAP's link and an on-Link CoA (LCoA). The LCoA is the on-link CoA configured on an MN's interface based on the prefix advertised by its default router. An RCoA is an address obtained by the MN from the visited network. An RCoA is an address on the MAP's subnet. It is auto-configured by the MN when receiving the MAP option. After the MN's registration with the MAP, the IAAM creates the confirmed new LCoAs and it puts the confirmed new LCoAs into its IP address pool as mentioned in the previous section.

The FHMIPv6 scheme can be classified into two operational modes based on the state of the MN: proactive and reactive. If the MN does not receive Fast Binding Acknowledgment (F-BACK) message from the MAP prior to its movement, the handover procedure is similar to the FMIPv6[2]. We call this case *reactive* mode. On the contrary, if the MN has received a F-BACK message from the MAP prior to its movement, it performs proposed fast handover scheme to reduce the latency in HMIPv6. We call this case *proactive* mode. Fig. 3 shows the proactive fast handover message interaction. The procedures of the proactive fast handover are as follows:

1) If the fast handover procedure is initiated by using the L2 trigger information, the MN sends a Router Solicitation for Proxy (RtSolPr) message to the oAR. It includes the identifier of its prospective attachment point.
2) After the reception of the RtSolPr message, the oAR forwards it to the MAP. Based on the identifier of the MN's prospective attachment point, the MAP selects a confirmed new LCoA using the IAAM. And then, the IAAM deletes it from its IP address pool.

3) The MAP delivers the confirmed new LCoA to oAR via the responding Proxy Router Advertisement (PrRtAdv) message. The oAR forwards it to the MN.
4) Upon reception of the RtSolPr message, the MAP sends a Handover Initiate (HI) message to the nAR so as to establish a bi-directional tunnel. The HI message contains the link-layer address, current LCoA, and the confirmed new LCoA of the MN. After the reception of the HI message from the MAP, the nAR creates a host route entry for current LCoA and sends a Handover Acknowledgement (HACK) message to the MAP. Finally, the nAR start buffering for any packets arriving for the MN. Notice that the HI/HACK exchange is performed only for tunnel establishment in FHMIPv6 scheme.
5) In response the PrRtAdv message, the MN sends a F-BU message to the MAP before it is disconnected from its link. This F-BU message includes the current LCoA and confirmed new LCoA for the MN to use. To minimize packet loss at the MN, the MN can send a F-BU message with 'S' flag set to the MN. After the reception of the F-BU message with 'S' bit set, in that case, traffic for the MN will be sent from the MAP to both oAR and nAR during the handover procedure.
6) In response the F-BU message, the MAP sends a F-BACK message to the MN. The result of F-BU and F-BACK processing is that MAP begins tunnelling MN's packets to new LCoA.
7) If a CN sends a packet to the MN before MN is able to attach to the nAR, the MAP forwards packet to the nAR and the nAR buffers any packets arriving for the MN.
8) When the MN moves into the nAR's network, it sends the Fast Neighbor Advertisement (F-NA) to initiate the flow of packets at the nAR.
9) As a response, the nAR sends a Router Advertisement with Neighbor Advertisement (NAACK) and forwards the buffered packet to the MN.

The procedures of the reactive fast handover are simple. If the MN does not send F-BU message or receive F-BACK message prior to its movement, the oAR creates a host route entry for current LCoA. And then, the MN must send a F-BU message as soon as it attaches to nAR. During the L2 layer switching time, in this case, packets destined to the MN are routed to its oAR first and are forwarded to the MN via the MAP and nAR as same to FMIPv6[2].

4 Performance Evaluation

In this section, we will compare our proposed scheme with fast handover scheme in MIPv6 (FMIPv6), presented in [1]. Let us call our proposal the FHMIPv6. The performance metrics is the total signaling cost for location update and packet delivery.

4.1 Signaling Cost in FMIPv6

As similar to [8], the following parameters are assumed for cost evaluation of the location update.

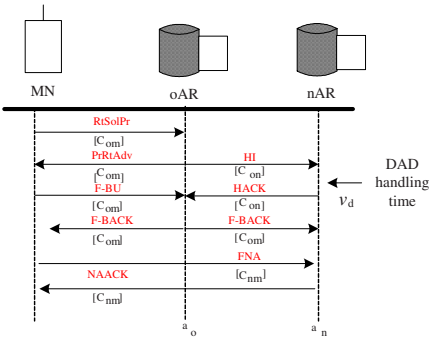


Fig. 4. IETF Fast Handover Protocol Message

- C_{pn}: The transmission cost of control signal between the MAP and the nAR.
- C_{po}: The transmission cost of control signal between the MAP and the oAR.
- C_{nm}: The transmission cost of control signal between the nAR and the MN.
- C_{om}: The transmission cost of control signal between the oAR and the MN.
- C_{on}: The transmission cost of control signal between the oAR and the nAR.
- C_{pm}: The transmission cost of control signal between the MAP and the MN.
- v_d: The processing cost for DAD handling.
- a_o: The processing cost at the oAR.
- a_n: The processing cost at the nAR.
- a_p: The processing cost at the MAP.
- l_{om}: The average distance between the oAR and the MN.
- l_{nm}: The average distance between the nAR and the MN.
- l_{pn}: The average distance between the MAP and the nAR.
- l_{po}: The average distance between the MAP and the oAR.
- δ_U: The proportionality constant for location update.
- T: The total time that the MN performs fast handover procedure.

Fig. 4 shows the control signaling message for fast handover procedure with oAR, nAR, MAP, and MN in FMIPv6. According to signaling message flows for fast handover, the total signaling cost for location update can be calculated as follows:

$$C_{LU} = 2a_o + 2a_n + 4C_{om} + 3C_{on} + 2C_{nm} + v_d \tag{1}$$

For simplicity, we assume that the transmission cost of location update is proportional to the distance between the source and destination mobility agents such as oAR, nAR, MAP, and MN. Thus, C_{on} can be represented as $C_{on} = C_{po} + C_{pn}$. Using the proportional constant δ_U, each cost of location update can be rewritten as follows:

$$C_{LU} = 2a_o + 2a_n + a_p + (4l_{om} + 3(l_{po} + l_{pn}) + 2l_{nm})\delta_U + v_d \tag{2}$$

In term of the packet delivery cost, we consider the costs associated with both forwarded and tunnelled packets from oAR to nAR. For cost evaluation of the packet delivery, the following parameters are assumed.

- T_{cp}: The transmission cost of packet delivery between the CN and MAP.
- T_{po}: The transmission cost of packet delivery between the MAP and oAR.
- T_{pn}: The transmission cost of packet delivery between the MAP and nAR.
- T_{om}: The transmission cost of packet delivery between the oAR and MN.
- T_{nm}: The transmission cost of packet delivery between the nAR and MN.
- l_{cp}: The average distance between the CN and the MAP.
- v_o: The processing cost of tunnelling at the oAR.
- v_n: The processing cost of tunnelling at the nAR.
- δ_D: The proportionality constant for packet delivery.
- δ_T: The proportionality constant for tunneling.
- t_{L2}: The average time taken from the start of the L2 trigger event to the reception of F-BACK message from the MAP at the MN (where $t_{L2} \leq T$).

Before the reception of F-BACK message from the MAP, packets are forwarded to the MN via the MAP and oAR. After the reception of F-BACK message, however, packets will traverse the MAP-oAR link twice and forward to the MN via nAR. Using the proportional constant δ_D, thus, the packet delivery cost can be expressed as follows:

$$\begin{aligned} C_{PD} \;=\; & (T_{cp} + T_{po} + T_{om}) \cdot t_{L2} \\ & +(v_o + v_n + T_{cp} + 2T_{po} + T_{pn} + T_{nm}) \cdot (T - t_{L2}) \end{aligned} \tag{3}$$

We assume that the transmission cost of packet delivery is proportional to the distance between the sending and receiving mobility agents. With the proportionality constant δ_D, T_{cp}, T_{po}, T_{pn}, and T_{nm} can be represented as $T_{cp} = l_{cp}\delta_D$, $T_{po}{=}l_{po}\delta_D$, $T_{pn} = l_{pn}\delta_D$, and $T_{nm}{=}l_{nm}\delta_D$. Also, we define a proportionality constant δ_T which is a tunnelling process constant for packet forwarding at the oAR and nAR. And, we assume that the processing cost of the oAR and nAR are same. Therefore, v_o can be represented as $v_o{=}\lambda_\alpha\delta_T$ and v_n can be represented as $v_n{=}\lambda_\alpha\delta_T$. Finally, we can get the packet delivery cost as follows:

$$\begin{aligned} C_{PD} \;=\; & (l_{cp} + l_{po} + l_{om})\delta_D \cdot t_{L2} \\ & +(2\lambda_\alpha\delta_T + (l_{cp} + 2l_{po} + l_{pn} + l_{nm})\delta_D)(T - t_{L2}) \end{aligned} \tag{4}$$

Based on the above analysis, we induce the total signaling cost function in FMIPv6 from (2) and (4):

$$C_{TOT}(\lambda_\alpha, T, t_{L2}) \;=\; C_{LU} + C_{PD} \tag{5}$$

4.2 Signaling Cost in FHMIPv6

For cost evaluation in FHMIPv6, the following parameters are assumed.

- v_p: The processing cost of tunnelling at the MAP.
- α: The probability that an MN performs proactive fast handover scheme.
- $1-\alpha$: The probability that an MN performs reactive fast handover scheme.

According to signaling message flows for proactive fast handover scheme as shown in Fig. 3, the total signaling cost for location update can be calculated follows:

$$C_{LU}^{'} = a_o + 2a_n + 2a_p + 2C_{om} + 2C_{po} + 2C_{pm} + 2C_{pn} + 2C_{nm} \quad (6)$$

The C_{pm} can be represented as C_{pm}=$C_{po} + C_{om}$. Using the proportional constant δ_U, each cost of location update can be rewritten as follows:

$$C_{LU}^{'} = a_o + 2a_n + 2a_p + (4l_{om} + 4l_{po} + 2l_{pn} + 2l_{nm})\delta_U \quad (7)$$

As similar to Eq. (3) and (4), v_p can be represented as v_p=$\lambda_\alpha \delta_T$. Thus, the packet delivery cost can be expressed as follows:

$$C_{PD}^{'} = (T_{cp} + T_{po} + T_{om}) \cdot t_{L2} + (2\lambda_\alpha \delta_T + T_{cp} + T_{pn} + T_{nm})(T - t_{L2}) \quad (8)$$

As a result, we can get the packet delivery cost as follows:

$$C_{PD}^{'} = (l_{cp} + l_{pn} + l_{nm})\delta_D \cdot t_{L2} + (2\lambda_\alpha \delta_T + (l_{cp} + l_{pn} + l_{nm})\delta_D) \cdot (T - t_{L2}) \quad (9)$$

Based on the above analysis, we induce the total signaling cost function in FHMIPv6 from (7) and (9):

$$C_{TOT}^{'}(\lambda_\alpha, T, t_{L2}) = \alpha \cdot (C_{LU}^{'} + C_{PD}^{'}) + (1-\alpha) \cdot (C_{LU}^{'} + C_{PD}) \quad (10)$$

4.3 Analysis of Results

In this section, we demonstrate some numerical results. Table 1 shows the some of parameters used in our performance analysis that are discussed in [8]. For simplicity, we assume that the distance between mobility agents are fixed and

Table 1. Performance Analysis Parameter

Parameter	a_p	a_o	a_n	v_p	λ_α	T	δ_D	δ_U	δ_T	t_{L2}
Value	10	5	5	5	0.01 - 50	0.01 - 100	0.2	15	15	0.01 - 100

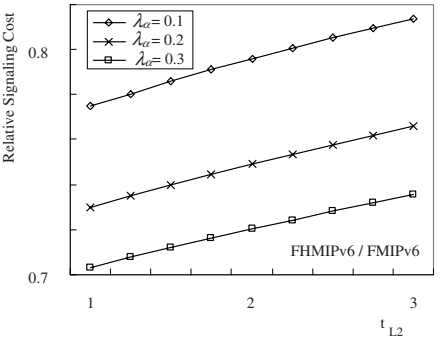


Fig. 5. Impact of Packet Arrival Rate on the Total Signaling Cost

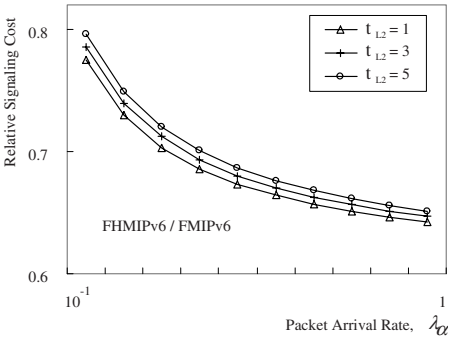


Fig. 6. Impact of the t_{L2} on the Relative Signaling Cost

same number and α and $1-\alpha$ are same (i.e., $\alpha = 0.5$). We define the relative signaling cost of the FMHIPv6 as the ratio of the total signaling cost for the FHMIPv6 to that of the FMIPv6. A relative cost of 1 means that the costs under both schemes are exactly the same. Fig. 5 shows the impact of t_{L2} on the relative signaling cost for $T = 10$. As shown in Fig. 5, the relative signaling cost increases as t_{L2} increases. We can see that the performance of FHMIPv6, on the whole, results in the lowest total signaling cost compared with FMIPv6. For the fixed value of t_{L2}, the performance of FHMIPv6 is better than FMIPv6 for high value of a packet arrival rate. This result is logical. During the fast handover period, the CN sends a packet to the MN directly without the inefficient routing via the oAR under the FHMIPv6. In addition, there is no DAD handling time during the handover. Thus, the FHMIPv6 scheme can reduce the handover latency by reducing the signaling cost. Fig. 6 shows the impact of packet arrival rate λ_α on relative signaling cost when $T = 10$. As shown in Fig. 6, the relative signaling cost decreases as the packet arrival rate λ_α increases. For the fixed value of λ_α, the performance of FHMIPv6 is better than FMIPv6 for low value of t_{L2}. This is because as t_{L2} increases, packet delivery cost for inefficient routing decreases in

FMIPv6. Thus, the performance of FMIPv6 and FHMIPv6 are almost same for the high value of t_{L2}. This implies that the performance of FHMIPv6 is better than FMIPv6 when L2 layer switching time (i.e., $T - t_{L2}$) is high. From the above performance analysis, we come to know that our proposal achieves highly considerable performance improvements when packet arrival rate and L2 layer switching time are high.

5 Conclusion and Future Works

In this paper, we proposed novel fast handover scheme to reduce the handover delay in HMIPv6. In FHMIPv6, an MN sends a F-BU message to the MAP instead of the oAR. Therefore, our proposal can reduce the handover latency by reducing the signaling cost in terms of packet forwarding and DAD handling process. Analysis results using the discrete analytic model shows that FHMIPv6 can have superior performance than FMIPv6 when packet arrival rate is high and L2 layer switching time are high.

References

[1] D. B. Johnson and C. E. Perkins, "Mobility support in IPv6", IETF Internet draft, draft-ietf-mobileip-ipv6-24.txt (work in progress), Jun 30, 2003. 174, 178
[2] Rajeev Koodli, "Fast Handovers for Mobile IPv6", IETF Internet draft, draft-ietf-mobileip-fast-mipv6-05.txt (work in progress), Mar 2003. 174, 177, 178
[3] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, "Hierarchical Mobile IPv6 mobility management (HMIPv6)", IETF Internet draft, draft-ietf-mobileip-hmipv6-08.txt (work in progress), Jun, 2003. 175
[4] K. ElMalki and H. Soliman,"Simultaneous Bindings for Mobile IPv6 Fast Handoffs", draft-elmalki-mobileip-bicasting-v6-03 (Work in progress), Apr 2004. 176
[5] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration", IETF Request for Comments 2462, Dec, 1998. 176
[6] Y. Han, J. Choi, H. Jang, S. Park, "Advance Duplicate Address Detection", draft-han-mobileip-adad-01.txt (Work in progress), Dec 2003. 176
[7] Hee Young Jung, Seok Joo Koh, Dae Young Kim, "Address Pool based Stateful NCoA Configuration for FMIPv6", draft-jung-mipshop-stateful-fmipv6-00.txt (Working in Progress), Aug 2003. 176
[8] Jiang Xie and I. E. Akyildiz, "A distributed dynamic regional location management scheme for Mobile IP", Proc of IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), vol. 2, pp.1069 -1078, 2002. 178, 181

An Efficient Handoff Mechanism with Reduced Latency in Hierarchical Mobile IPv6

Jae-Myung Jang, Dong-Hee Kwon, and Young-Joo Suh

Department of Computer Science and Engineering
Pohang University of Science & Technology (POSTECH)
San 31 Hyoja-dong, Nam-gu, Pohang, 790-784, Korea
{locust,ddal,yjsuh}@postech.ac.kr

Abstract. As handoff latency increases, mobile hosts experience performance and service quality degradations. Typically, it gives a significant impact on the performance of real-time multimedia applications. In this paper, an efficient handoff mechanism that reduces the handoff latency in HMIPv6 is proposed. In the proposed handoff mechanism, each access router learns its neighboring access router information. For this, a neighbor access router discovery scheme is proposed. When a mobile host senses that it will perform a handoff to a new network, it performs an Address Auto-configuration procedure for its LCoA (RCoA and LCoA in case of the handoff to a new MAP domain) in advance using the neighbor access router information received from the current access router. These features give us the reduction of latency both in intra-MAP and inter-MAP handoffs. The simulation results using ns-2 show that the proposed handoff mechanism reduces handoff latency.

1 Introduction

Mobile IPv6 [1] proposed by IETF (Internet Engineering Task Force) provides a basic host mobility management scheme. In Mobile IPv6, when a MH (Mobile Host) moves from one Access Router (AR) to another, it configures a new Care-of-Address (CoA) and requests the Home Agent (HA) to update its binding. A binding maintained by the HA is an association of a MH's home address and its CoA. When the HA has the binding for the MH, the HA intercepts any packets destined to the MH, and tunnels them to the MH's CoA. Thus, it is necessary for the MH to register its current point of attachment to the HA whenever it handoffs. During this process, there is a time interval that the MH can't send or receive any application traffics. This time interval is referred to as handoff latency.

In base Mobile IPv6, handoff latency is divided into two distinct types of latency: Layer 2 (L2) handoff latency and Layer 3 (L3) handoff latency. The L2 handoff latency is the period between the time when the air-link with current AR is disconnected and the time when the MH connects to the air-link of the new

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 184–194, 2004.

AR that it will handoff. The L3 handoff latency is the sum of CoA acquisition latency and binding update (BU) latency. An Address Auto-configuration (AA) in stateless or stateful manner [2] occupies a major part of the CoA acquisition latency. The BU latency is the period from the time when the MH sends a BU message to its HA to the time when it receives a Binding Acknowledgement (BAack) message from the HA. The overall handoff latency including both of the two types of latency is long enough to make some packets to be lost. In addition, when a real-time application such as VoIP is used, this would be intolerable. Thus, reducing the handoff latency is a very critical problem in Mobile IPv6.

Hierarchical Mobile IPv6 (HMIPv6) [5] has been proposed to reduce the signaling overhead and latency. However, HMIPv6 focused on the intra-MAP (Mobility Anchor Point) domain handoff, not on the inter-MAP domain handoff [7-11]. In this paper, we propose a neighbor AR (Access Router) discovery mechanism in HMIPv6, and we propose an efficient handoff procedure using the AR discovery mechanism. The neighbor AR discovery mechanism lets each AR and MAP know its neighbor ARs or MAPs, and thus a MH can perform handoff process in advance by the proposed handoff mechanism. When a MH senses that it will perform a handoff, the MH generates a new CoA from the AA (Address Auto-configuration) process in advance and sends it to the corresponding AR and MAP. The AR and MAP also perform the DAD (Duplicate Address Detection) process on behalf of the MH for the newly configured MH's CoA in advance.

The rest of this paper is organized as follows. Section 2 briefly describes the HMIPv6 mobility management protocol. In Section 3, we present the proposed mechanisms to reduce the handoff latency during the intra-MAP and inter-MAP handoffs in HMIPv6. We evaluate the proposed mechanisms in Section 4. Section 5 concludes this paper.

2 Hierarchical Mobile IPv6 [5]

The Hierarchical Mobile IPv6 (HMIPv6) protocol is proposed to minimize the BU signaling latency and overhead, and it separates mobility management into intra-domain mobility and inter-domain mobility. A Mobility Anchor Point (MAP) in HMIPv6 treats the mobility management inside a domain. Thus, when a MH moves around the sub-networks within a single domain, the MH sends a BU message only to the current MAP. When the MH moves out of the domain or moves into another domain, Mobile IPv6 is invoked to handle the mobility. The basic operation of the HMIPv6 can be summarized as follows. A MH entering a MAP domain will receive from ARs a Router Advertisement (RA) message containing the information on local MAPs. From this message, the MH forms two CoAs: LCoA (Local CoA) and RCoA (Regional CoA). It binds its current location (LCoA) with its current domain (RCoA). LCoA is configured from the network prefix of the AR to which it is currently attached, and RCoA is configured from the network prefix of the MAP. Acting as a local HA, the MAP receives all packets on behalf of the MH it is serving, and encapsulates and for-

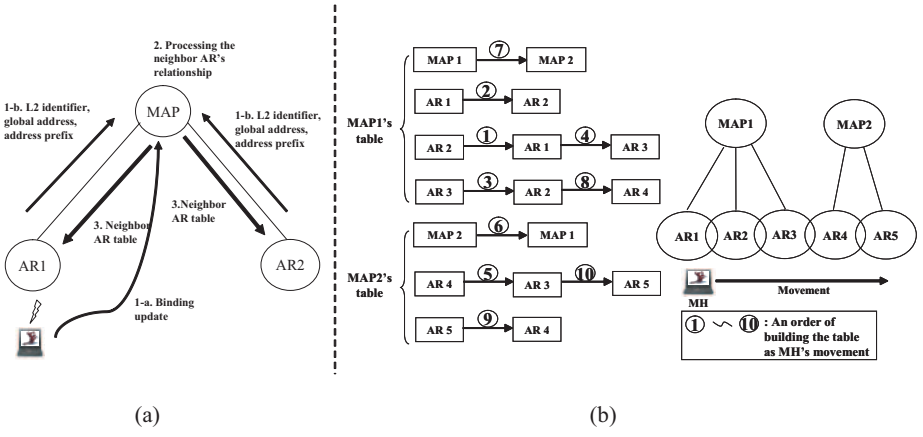


Fig. 1. The neighbor discovery procedure (a) and an example of a neighbor AR table at a MAP (b)

wards them directly to the MH's LCoA. If the MH handoffs to a new AR and changes its current address (LCoA) within a local MAP domain, it only needs to register the new LCoA with the MAP. Therefore, only the RCoA needs to be registered with the HA. The RCoA does not change as long as the MH moves within a MAP domain. This makes the mobile node's mobility to be transparent to the CN and the HA. A MAP domain's boundaries are defined by the ARs advertising the MAP information to the attached MH.

3 Proposed Protocol

In both Mobile IPv6 and HMIPv6, no information is exchanged among ARs. Thus only after completing the L2 handoff, a MH can receive the information about the AR to which the MH will handoff via Agent Advertisement message. On-going communication sessions with other hosts are impossible before the completion of this handoff process.

In the proposed mechanism, each AR can learn the information about its geographically adjacent ARs. Typically, this information includes the global address, L2 identifier, and the network prefix of the AR that is currently advertised. In this mechanism, the current AR that the MH is visiting would be able to inform the MH of the network prefix information of ARs to which the MH is likely to handoff. If this is possible, the MH can start the Address Auto-configuration (AA) process in advance for LCoA in case of intra-map handoff, and both of LCoA and RCoA in case of inter-map handoff. After completion of the AA process, the MH sends an incomplete binding update message, which is a newly defined message and will be discussed later, to a MAP in advance, and the MAP can also perform the Duplicate Address Detection (DAD) process in advance

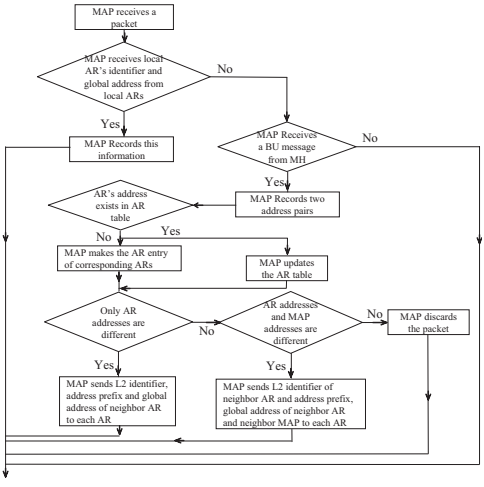


Fig. 2. Flow chart for AR discovery

using this message. Through this signaling flow, we may expect remarkable decreases of handoff latency.

In the following sections a neighbor AR discovery mechanism and the handoff procedure using the neighbor information is presented.

3.1 Neighbor ARs Discovery Mechanism

The objective of the neighbor ARs discovery is that a MH completes almost all the process required to perform a handoff before the actual handoff. The complexity and overhead of information exchange among neighbor ARs are marginal because each AR has a limited number of neighbors.

In the proposed neighbor ARs discovery scheme, MAPs play an important role for managing the neighbor ARs information. As shown in Fig. 1(b), a MAP maintains a neighbor ARs table that includes a list of ARs within its own domain and their respective neighbor ARs. Each AR in a domain periodically sends to the MAP a control message containing its L2 identifier, global address, and network prefix information. From those messages from each AR, the MAP updates the corresponding entry for the AR in the neighbor ARs table. MAPs learn the neighbor relationship among ARs from the MH when it handoffs from one AR (AR1) to another AR (AR2). Since AR1 and AR2 are neighbor ARs, the MH sends a BU message with an option field to the MAP and the MAP updates its neighbor ARs table based on the information drawn from the BU message and the option field. The option field in a BU message contains the L2 identifier, global address, and network prefix information of the previous AR. The L2 identifier of an AR is used as an index to find information of the AR in the neighbor ARs table when the current AR receives a (proxy) router solicitation message from a MH. When an entry for an AR (e.g., AR1) in the neighbor ARs table

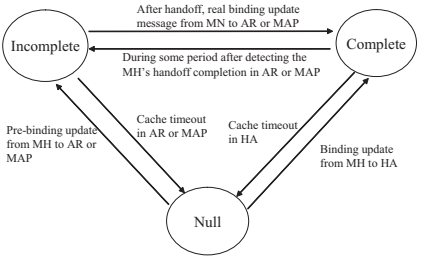


Fig. 3. State transition to MH's address

is changed, the MAP sends the changed neighborhood information to neighbor ARs of AR1. This change includes the addition of a new AR, deletion of an existing AR, or information changes of an existing AR (such as global address or advertised network prefix changes). The neighbor ARs table of a MAP also includes its neighbor MAPs information. If a MH handoffs from AR1 of MAP1 to AR2 of MAP2, then MAP2 notifies MAP1 of the neighborhood information between the two MAPs.

Figure 1(a) shows the procedure that a MAP gathers and redistributes the neighbor ARs information. A MH sends a BU message to the MAP (1-a). Each AR periodically sends control messages to the MAP and the MAP records this information (1-b). The MAP records the current and the previous addresses of ARs and MAPs from the BU message sent by the MH. The MAP compares the address pairs and derives the neighborhood relationship of the local ARs based on the information recorded before (2). If only AR addresses are different, the two ARs are located in the same MAP domain. The MAP sends the derived neighbor ARs information to each AR (3). If AR addresses and MAP addresses are different, the two ARs are in different MAP domains. In this case, the MAP sends the neighbor ARs information to its ARs and the previous MAP.

Figure 1(b) shows an example of a neighbor AR table of MAPs. In the figure, MH moves from AR1 to AR5 through AR2, AR3, and AR4. The number in a circle indicates the sequence of constructing a neighbor AR table. Initially, MAP1 has table entries of MAP1, AR1, AR2, AR3, and MAP2 has table entries of MAP2, AR4, and AR5. When the MH handoffs to AR2, MAP1 comes to know that the MH has moved from AR1 to AR2 by the BU message from the MH, and MAP1 updates AR2's and AR1's entries in the neighbor AR table (1,2). Since the neighbor AR table has been changed, MAP1 sends the neighborhood information to AR1 and AR2. When the MH handoffs to AR3, the AR3's and AR2's entries in the neighbor AR table of MAP1 are changed (3,4), and MAP1 sends the neighborhood information to AR3 and AR2. When the MH handoffs to AR4, MAP2 knows that the MH comes from AR3 whose MAP is MAP1, and thus AR4's and MAP2's entries in the neighbor AR table of MAP2 are changed (5,6), and MAP2 sends the neighborhood information to AR4 and MAP1 (7,8). Figure 2 summarizes the proposed neighbor ARs discovery scheme in a flow chart.

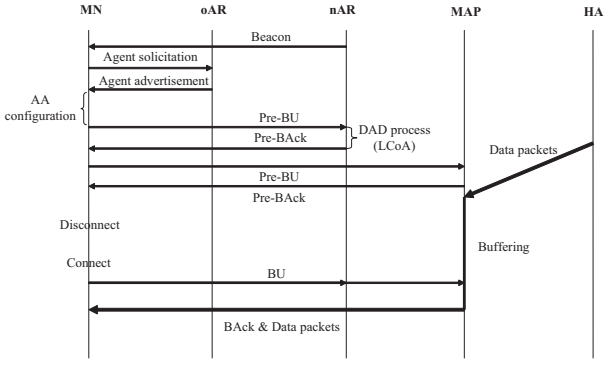


Fig. 4. Signal flows of intra-MAP domain handoff

3.2 Pre-binding Update and Forward Mechanism

In standard Mobile IPv6, after a MH moves from one subnet to another, it informs its HA and CNs of the newly configured CoA after performing the Address Auto-configuration (AA) process for that CoA. In the proposed scheme, on behalf of the MH, the new AR (for LCoA) and/or the new MAP (for RCoA) to which the MH is likely to handoff performs the DAD (Duplicate Address Detection) process to determine whether the MH's address can be used or not in that network. Because ARs and MAPs know their respective neighbor AR's address information using the neighbor ARs discovery scheme, when the MH senses that it will perform a handoff, it can transmit a (proxy) Agent Solicitation message to the current AR, and the AR can send an (proxy) Agent Advertisement message. When the MH receives this message, it performs in advance an AA process for the new CoA before the actual handoff and sends to the new AR a pre-binding update (Pre-BU) message that requests the new AR to perform a DAD process for the new CoA. The Pre-BU message format is similar to the BU message format. The AR (AR and MAP in case of inter-MAP domain handoff) performs a DAD and records the address as incomplete state in response to the MH's pre-BU message. The incomplete state of the MH's address means that it will not be used for routing of the MH until the MH sends a real BU message after the actual handoff. After L2 handoff completes, the MH sends a BU message to the new MAP via new AR. The new AR and the new MAP finish the handoff process and directly progress routing by changing the MH's address in incomplete state into in complete state without DAD process. Figure 3 shows a state transition diagram of the address that a MH uses.

Figures 4 and 5 represent the signal flows of the proposed handoff process, both intra-MAP domain handoff (Figure 4) and inter-MAP domain handoff (Figure 5). Since the signal flow patterns in both of the handoff cases are similar, we describe only the inter-MAP domain handoff in detail.

1. When a MH receives a L2 beacon message from nAR, the MH sends a (proxy) Agent Solicitation message that includes the L2 identifier information of nAR

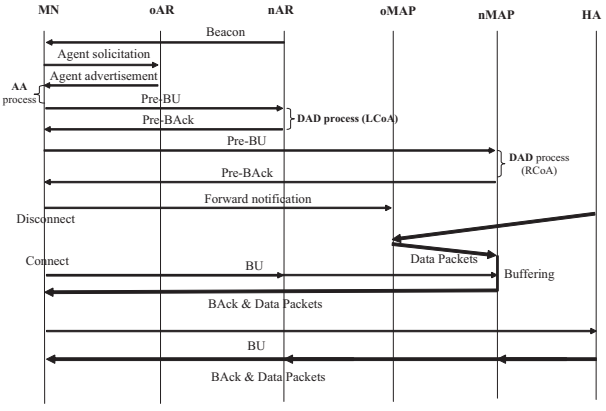


Fig. 5. Signal flows of inter-MAP domain handoff

to the current AR (oAR) to request the nAR's information (the network address prefix and global address of the nAR).

2. In response to the Agent Solicitation message, oAR checks the neighbor AR information using the L2 identifier in it. If oAR finds the requested information, it sends the MH a (proxy) Agent Advertisement message that includes the requested information (network address prefix and global addresses of nAR and nMAP).
3. The MH performs the AA process and forms new LCoA and RCoA.
4. The MH sends a Pre-BU message to nAR for DAD processing of LCoA. The nAR receiving the Pre-BU message records the MH's address as incomplete state and sends a Pre-BAck message to the MH. The MH also sends a Pre-BU message to nMAP for DAD processing of RCoA. Upon receiving it, nMAP performs the DAD process for RCoA, sends back a Pre-BAck message, and starts buffering the packet destined to the MH. During the AA and DAD processes, the MH still can receive packets from oAR. The MH also sends a forward notification message to oMAP. It requests oMAP to forward the packets arrived at oMAP to nMAP from now on.
5. The MH performs the L2 handoff.
6. As soon as the completion of L2 handoff, the MH sends a BU to nMAP via nAR.
7. The nAR and nMAP receive it and change the MH's address state into complete. nMAP sends the buffered packet destined to MH with a BAck message.
8. MH receives it and sends a BU message to HA/CN for normal routing.
9. HA/CN receives it and sends the BAck message to MH.

4 Simulation Results

We performed simulation study using ns-2 [12] to evaluate the performance of the proposed mechanism. Figure 6 illustrates the network topology used in our

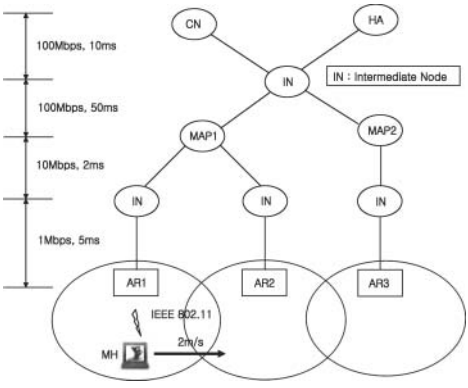


Fig. 6. Simulation Topology

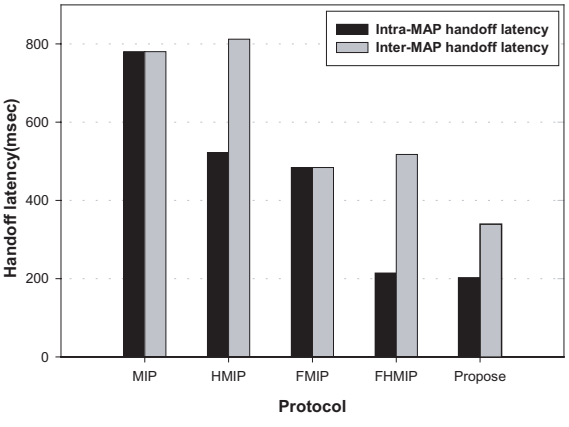


Fig. 7. *Protocol* vs. *Handoff latency*

simulation study. The total simulation time is 80 sec. and a MH moves with 2 m/s crossing contiguous cells. Each cell has a 802.11 wireless LAN air interface. The CN starts sending a packet to the MH at 10 sec. after the simulation starts and finishes sending when simulation completes. 256 bytes UDP packet is used and packet interval is 10 ms. To consider L2 trigger time, and the time required to perform AA and DAD, we assume L2 handoff latency and address resolution time are 200 ms and 300 ms, respectively, and the agent advertisement period is set to 1 s. At t=25 s, an intra-MAP domain handoff occurs from AR1 to AR2, and at t=60 s an inter-MAP domain handoff occurs from AR2 within MAP1 to AR3 within MAP2. We assume the duplex-link wired links between an AR and an intermediate node and the DropTail queue for output queue at each node.

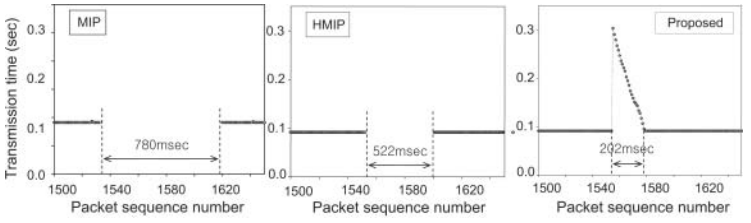


Fig. 8. End-to-end packet transmission time in the intra-MAP domain handoff

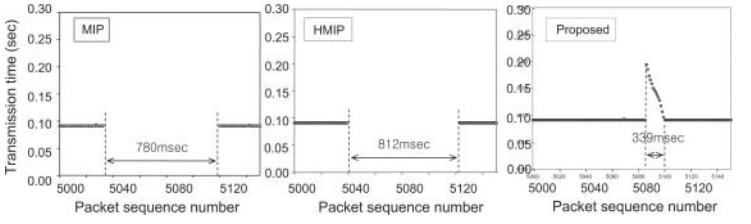


Fig. 9. End-to-end packet transmission time in the inter-MAP domain handoff

First, we measure and compare the handoff latencies for various Mobile IPv6 configurations. The handoff latency of base Mobile IP is defined as the period of time between the disconnection of the MH's wireless link and the reception of its HA's BAack (binding acknowledgement). Corresponding handoff latency for a intra-MAP domain handoff in HMIPv6 is the period between the disconnection of the MH's wireless link and the reception of its MAP's BAck. In inter-MAP domain handoffs, the handoff latency is the time from when a MH triggers link-down in the current network to when the MH receives HA's first binding acknowledgement after it receives BAck from the new MAP.

Figure 7 shows the handoff latency of five schemes - standard MIPv6 (MIP), MIPv6 with fast handoff (FMIPv6), HMIPv6, HMIPv6 with fast handoff (FHMIPv6), and the proposed mechanism. As shown in the figure, the base MIP shows the largest intra-MAP domain handoff latency. The proposed mechanism shows the minimum latency, although its latency is very comparable with that of FHMIP. For inter-MAP domain handoff, HMIP shows the largest handoff latency. Just as in intra-MAP domain handoff, the proposed mechanism shows the minimum latency. Note that the inter-MAP domain handoff improvement of the proposed mechanism is very large. It is due to the fact that the proposed mechanism performs the AA and DAD processes in advance.

Figure 8 shows end-to-end packet transmission time according to packet sequence numbers when there is an intra-MAP domain handoff. Here, end-to-end packet transmission time is the time from a CN transmit a packet to the time a MH receives the packet. We compare the three protocols - base MIP, HMIP, and the proposed mechanism. As shown in the figure, MIP shows many packet losses due to the large handoff latency, and HMIP shows decreased packet losses than base MIP through the advantages of MAP entity. The proposed mechanism

shows the best performance without packet loss. In the proposed mechanism MAPs buffer and forward packets during a handoff period, which improves the overall performance. Figure 9 shows end-to-end packet transmission time according to packet sequence numbers when there is an inter-MAP domain handoff. Compared with MIP and HMIP, the proposed mechanism shows the best performance, too.

5 Conclusions

In this paper, we propose efficient neighbor AR discovery mechanism and the handoff procedure using the neighbor information. The proposed neighbor ARs discovery scheme let each AR and MAP know its neighbor ARs or MAPs, and thus MH can perform handoff process in advance by the proposed handoff mechanism. When a MH senses that it will perform a handoff, the MH generates a new address from the AA process in advance and sends it to the corresponding AR and MAP. The AR and MAP also process the DAD for the MH's address in advance. According to our simulation study, the proposed handoff mechanism shows improved performance over existing mechanisms due to the fact that the proposed mechanism performs the AA and DAD processes in advance. Although the proposed mechanism provides improvements for both intra-MAP domain handoff and inter-MAP domain handoff, we can get more performance improvements for the inter-MAP domain handoff case.

References

[1] D. B. Johnson, C. Perkins, J Arkko: Mobility Support in IPv6. draft-ietf-mobileip-ipv6-24.txt. (June 2003, work in progress)
[2] S. Thomson, T. Narten: IPv6 stateless address autoconfiguration. RFC 2462. (December 1998)
[3] T.Narten et al.: Neighbor Discovery for IP Version 6. RFC 2461. (December 1998)
[4] Hinden, R., S. Deering: IP Version 6 Addressing Architecture. RFC 2373. (July 1998)
[5] H. Soliman, K. El-Malki: Hierarchical Mobile IPv6 mobility management(HMIPv6). draft-ietf-mobileip-hmipv6-08.txt. (June 2003, work in progress)
[6] Deering, S., R. Hinden: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460. (December 1998)
[7] K. Omae, M. Inoue, I. Okajima, N. Umeda: Performance Evaluation of Hierarchical Mobile IPv6 Using Buffering and Fast Handover. Technical Reports of IEICE. IN2002-152. (December 2002)
[8] Robert Hsieh, Zhe Guang Zhou, Aruna Seneviratne: S-MIP : A Seamless Handoff Architecture for Mobile IP. in proceeding of INFOCOM. (2003)
[9] K. Omae, et al.: Hierarchical Mobile IPv6 Extension for IP-based Mobile Communication System. Techinical report of IEICE. IN2001-178. (February 2002)
[10] Hideaki Takahashi, Ryoichi Kobayashi, Ichiro Okajima, Narumi Umeda: Transmission Quality Evaluation of Hierarchical Mobile IPv6 with Buffering Using Test Bed. in proceeding of VTC. (2003)

[11] R. Hsieh, A. Seneviratne: Performance analysis on Hierarchical Mobile IPv6 with Fast-handoff over TCP. in preceedings of GLOBECOM (2002)
[12] The Network Simulator - ns (version 2) Website. http://www.isi.edu/nsnam/ns.
[13] http://mobqos.ee.unsw.edu.au/~robert/

A Study on Availability of Mobility Databases

Ai-Chun Pang[1] and Yuan-Kai Chen[2]

[1] Department of Computer Science and Information Engineering
National Taiwan University, Taiwan, ROC
acpang@csie.ntu.edu.tw
[2] Telecommunication Laboratories
Chunghwa Telecom Co., Taiwan, ROC
ykchen@cht.com.tw

Abstract. In third generation mobile communications networks, the core network nodes such as *Home Location Register* (HLR) and *GPRS Support Nodes* are implemented on a highly scalable, real-time mobility database cluster. In this letter, we propose an analytic model to investigate the availability and performance of the parallel system for the HLR. The study indicates that the number of processors, the steady-state probability that a processor is operational, the arrival rate of HLR accesses, and the variance of the service time for a processor affect the system performance. Our study provides guidelines for designing the parallel HLR system.

1 Introduction

In third generation UMTS (Universal Mobile Telecommunications System) mobile system [3, 1], the core network nodes such as *Home Location Register* (HLR) and *GPRS Support Nodes* (GSNs) are responsible for processing several thousands of mobile user records simultaneously. For example, the HLR is accessed when a mobile terminated communication session is initiated or a location update occurs. On the other hand, a GSN is accessed when a *Packet Data Protocol* (PDP) context is created, modified, or deleted for a mobile user. The above examples are usually classified as high-performance, transaction-oriented applications. To process the records with high speed, the UMTS core network nodes are typically implemented on a highly scalable, real-time mobility database cluster. In this letter, we use HLR as an example to illustrate the availability issues of mobility databases. Same results apply to the GSNs. In a commercial HLR product [2], the mobility database cluster is implemented with loosely coupled CPUs such as Pentium processors. A unique property of mobile data access is that processing of individual records are independent of each other. Therefore, more than one processor can access the mobility database simultaneously. An abstract parallel architecture for the HLR is illustrated in Figure 1.

In this architecture, the access requests first arrive at the front-end processor. This high-speed processor dispatches the requests to the processor cluster. The processors in the cluster handle the requests in parallel. They can simultaneously

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 195–200, 2004.

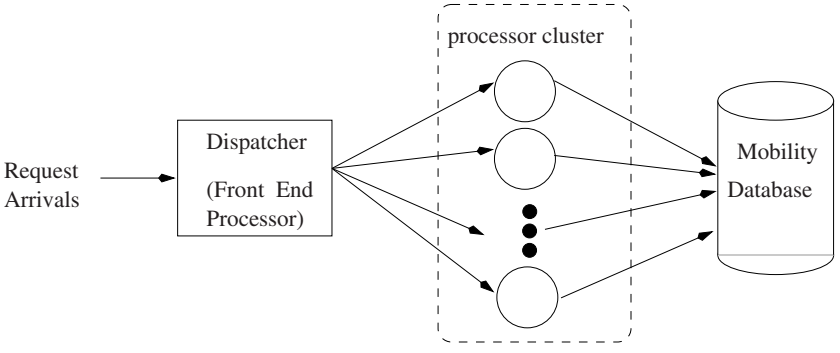


Fig. 1. Abstract Parallel Architecture for the HLR

access the mobility database without interfering each other. In this letter, we propose an analytic model to investigate the performance of the parallel HLR architecture. Section 2 describes the proposed analytic model. Section 3 uses numerical examples to show the guidelines for designing the parallel HLR system.

2 An Analytic Model

This section models the availability and performance of the parallel system for the HLR. We assume that there are n processors in the cluster. A processor is either operational or down. The *operational readiness* of a processor is modeled by alternating renewal processes. In the kth cycle ($k \geq 1$), let X_k be the operational time and Y_k be the down time. Assume that the random vectors (X_k, Y_k) are independent and identically distributed. Note that Y_k may be dependent on X_k (i.e., the length of the down time may depend on the previous operating time). If $E[X_k + Y_k] < \infty$ and $X_k + Y_k$ is *non-lattice* [1] then the operational readiness (see Theorem 3.4.4 [4]) for a processor is

$$p = \lim_{t \to \infty} \Pr[\text{processor is operational at time } t] = \frac{E[X_k]}{E[X_k] + E[Y_k]}$$

If the HLR database is an n-processor system (excluding the front-end processor), then the probability that $i \leq n$ processors are operational is

$$p(n, i) = \binom{n}{i} p^i (1-p)^{n-i} \tag{1}$$

We are interested in the following two measures of the HLR database.

– When an access (e.g., a phone call or a location update) arrives, what is the probability α_n that an n-processor HLR system is down?

[1] A nonnegative random variable is said to be lattice if it only takes on integral multiples of some nonnegative number.

– If the system is operational (i.e., $i > 0$), what is the expected response time $\bar{R} = E[R|i > 0]$ to access the HLR database?

We assume that the accesses to the HLR form a Poisson stream with rate λ. The service time t_s for a processor to complete an access has a general distribution with mean $1/\mu$. Therefore the HLR can be modeled by an $M/G/i$ queue where i is the number of operational processors. Though no closed form expression for the mean query response time $E[R_i]$ is known, $E[R_i]$ can be approximated as [5]

$$E[R_i] \simeq \frac{(1+c_v^2)\rho_i\Theta_i}{2\lambda(1-\rho_i)} + \frac{1}{\mu} \tag{2}$$

where

$$\rho_i = \frac{\lambda}{i\mu},$$

$$\Theta_i = \frac{\lambda^i}{\mu^{i-1}(i-1)!(i\mu-\lambda)} \left[\sum_{j=0}^{i-1} \frac{\lambda^j}{\mu^j j!} + \frac{\lambda^i}{\mu^{i-1}(i-1)!(i\mu-\lambda)} \right]^{-1}$$

and c_v is the coefficient of variation of the service time distribution. That is, $c_v = \mu\sqrt{Var[t_s]}$, where $Var[t_s]$ is the variance of t_s. Note that (2) is exact for $M/M/i$ and $M/G/1$.

From (1), it is clear that

$$\alpha_n = p(n,0) = (1-p)^n \tag{3}$$

From (2),

$$\begin{aligned} \bar{R} &= \left[\frac{1}{\sum_{i=1}^{n} p(n,i)} \right] \left\{ \sum_{i=1}^{n} E[R_i] p(n,i) \right\} \\ &\simeq \left(\frac{1}{1-\alpha_n} \right) \left\{ \sum_{i=1}^{n} \left[\frac{(1+c_v^2)\rho_i\Theta_i}{2\lambda(1+\rho_i)} + \frac{1}{\mu} \right] \binom{n}{i} p^i (1-p)^{n-i} \right\} \end{aligned} \tag{4}$$

3 Numerical Results and Discussions

This section investigates the performance of the parallel HLR system based on the analytic model developed in the previous section. We use some numerical examples to illustrate the effects of n (i.e., the number of processors), p (i.e., the steady-state probability that a processor is operational), λ (the arrival rate of HLR accesses) and c_v (i.e., the coefficient of variation of the service time t_s) on the output measure $\bar{R}$. Note that from (4), $\bar{R}$ represents the expected response time to access the HLR database when the system is operational (i.e., at least one processor is active).

Figure 2 plots $\bar{R}$ as functions of n, p and λ, where $c_v = 1.0$ (i.e., the service time t_s is exponentially distributed), and $p = 0.8$, 0.9 and 0.99999. In this figure,

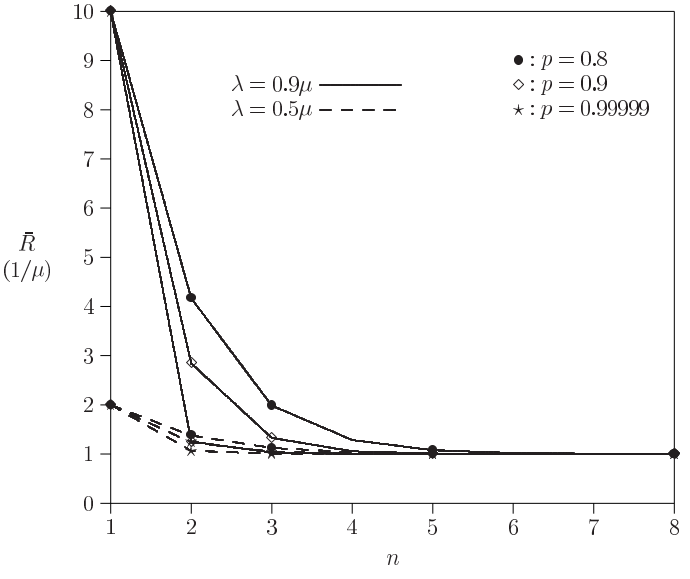


Fig. 2. The Effects of n, p and λ ($c_v = 1.0$)

the solid and dashes curves represent the cases for $\lambda = 0.9\mu$ and $\lambda = 0.5\mu$, respectively. Figure 2 shows intuitive results that $\bar{R}$ is a decreasing function of n, and is an increasing function of λ. We also observe the following non-trivial results: $\bar{R}$ is significantly affected by n when n is small (e.g., $n \leq 4$ for the case of $\lambda = 0.9\mu$). On the other hand, when n is large, $\bar{R}$ is slightly affected by n, and the value of $\bar{R}$ approximates to $1/\mu$. Specifically, for $n \geq 5$, adding extra processors will not improve the performance of the HLR. Figure 2 also indicates that increasing p results in the increase of i (i.e., the number of operational processors) and thus the decrease of the expected response time $\bar{R}$. Note that when $n = 1$, $\bar{R}$ is not influenced by the p value (because we assume that one processor is always operational).

Figure 3 plots $\bar{R}$ against c_v (i.e., the coefficient of variation of the service time distribution). In this figure, $\lambda = 0.5\mu$, and $p = 0.8$ and 0.9. The curves indicate that $\bar{R}$ increases as c_v increases. This phenomenon is explained as follows. As c_v increases (i.e., $Var[t_s]$ increases), more long and short service times are observed. An HLR access with a long service time may result in longer queuing delay of subsequent accesses even though these subsequent accesses have short service times. Thus a larger $\bar{R}$ is observed. We also observe that when $c_v \leq 1$, c_v only has insignificant effect on $\bar{R}$. On the other hand, when $c_v \geq 1$, $\bar{R}$ significantly increases as c_v increases. Furthermore, the increasing rate of $\bar{R}$ is larger for $p = 0.8$ than for $p = 0.9$.

In order to meet the telecom-grade requirement, 99.999% reliability for the parallel HLR system should be achieved. Table 1 illustrates the effect of p under the condition of 99.999% HLR reliability, where $\lambda = 0.5\mu$ and $c_v = 1.0$. In this

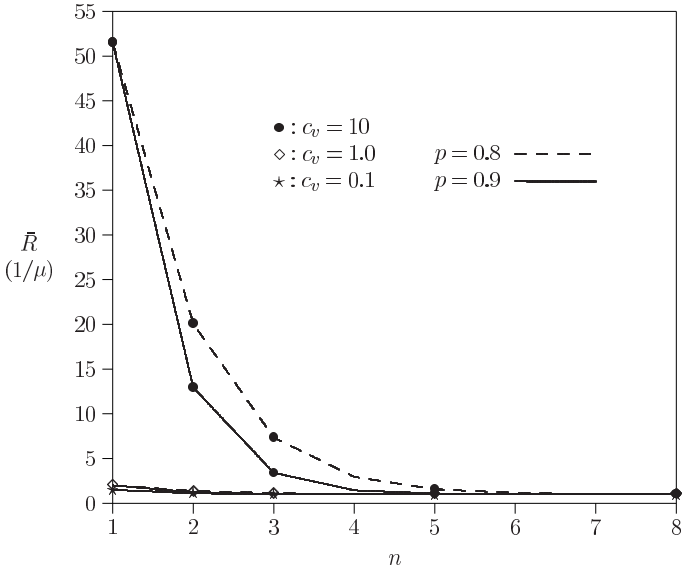


Fig. 3. The Effect of c_v ($\lambda = 0.5\mu$)

figure, two output measures are considered: the number of required processors and the expected response time. When $p = 0.8$, an 8-processor HLR system is needed to achieve 99.999% reliability, and its expected response time is about $1/\mu$. On the other hand, when $p = 0.997$, only two processors are needed, but the expected response time increases to $1.07/\mu$.

The above discussions give examples on how to select the number and the types (in terms of reliability) of processors to achieve the telecom-grade HLR performance.

Table 1. The Expected Response Time and the Number of Required Processors for Different p Values under the Condition of 99.999% HLR Reliability ($c_v = 1$, $\lambda = 0.5\mu$)

p	Number of Required Processors	$\bar{R}$ $(1/\mu)$
0.997	2	1.072250
0.99	3	1.008138
0.95	4	1.002837
0.90	5	1.001624
0.85	7	1.000252
0.80	8	1.000244

Acknowledgement

This work was sponsored in part by National Science Council under contract NSC92-2213-E-002-092, Microsoft and Intel.

References

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Description; Stage 2. Technical Specification 3G TS 23.060 version 4.1.0 (2001-06), 2001. 195

[2] Karlson, M. Ronja – A Java Application Platform. *Ericsson Review*, 77(4):224–247, 2000. 195

[3] Lin, Y.-B., and Chlamtac, I. *Wireless and Mobile Network Architectures*. John Wiley & Sons, 2001. 195

[4] Ross, S. M. *Stochastic Processes*. John Wiley & Sons, 1996. 196

[5] Ross, S. M. *Introduction to Probability Models*. Harcount/Academic Press, 2000. 197

Dynamic Bandwidth Adaptation Using Mobile IP in Hybrid Cellular Networks*

JaeWon Kang and Badri Nath

DATAMAN Lab., Department of Computer Science, Rutgers University
110 Frelinghuysen Road, Piscataway, NJ 08854-8019, USA
Tel: 732-445-2706, Fax: 732-445-6722
{jwkang,badri}@cs.rutgers.edu

Abstract. As cellular packet data services become widely deployed by the rollout of the General Packet Radio Service (GPRS) and 3G cellular networks, the packet-switched cellular network is emerging as an alternative to the existing circuit-switched cellular network. While the circuit data service provides wireless bandwidth guarantees, the packet data service offers better radio resource utilization due to its packet-switched multiplexing principle applied in the air interface. Therefore, the capability of having a data service over either the circuit-switched radio or the packet-switched radio provides a tradeoff between the perceived QoS and the radio utilization.
This paper proposes a novel framework for dynamic bandwidth adaptation that allows an ongoing data traffic to alternate between the circuit data service and the packet data service using Mobile IP in the hybrid cellular network. Our approach can be used as a tool to handle the dynamically changing load in a cell and also easily deployed in any hybrid cellular network where the circuit and packet data services coexist

1 Introduction

Radio has been considered the most scarce and valuable resource in wireless networks. Especially in commercial cellular mobile networks, the radio resource in a cell is well managed and arbitrated among active mobile stations by the collaborations of highly structured and complex link-layer protocols and various functional entities. In the existing circuit-switched cellular network, each active mobile station is assigned a dedicated unit of bandwidth called a (physical) *channel*, which might be a frequency in FDMA, a time slot in TDMA, and a code in CDMA. When used for data service, the circuit-switched cellular network, mainly designed for voice traffic transportation, has several disadvantages such as long and complicated connection setup, inefficient radio utilization for bursty data traffic, and high connection charge largely incurred by the low channel utilization. In order to address these weaknesses, the packet-switched[1] data service (or

* This research work was supported in part by DARPA under contract number N-666001-00-1-8953 and NSF grant ANI-0240383.

[1] In this paper, "packet-switched data service" or "packet data service" means a data service based on the packet-switched *multiplexing* principle applied over the air

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 201–210, 2004.

packet data service) such as General Packet Radio Service (GPRS) [1, 2] starts being deployed in the existing and 3G cellular networks. Using packet-switched multiplexing principle, several active mobile stations can be multiplexed onto a single physical channel[2], thereby increasing the channel utilization especially for bursty data traffic.

However, even though the packet data service is getting high degree of penetration, the existing circuit-switched cellular network will not disappear for the foreseeable future due to its wide deployment and the capability of reliable voice traffic transportation. The packet data service is rather integrated into the existing circuit-switched cellular network as a new bearer service. Therefore, the circuit data service will continue to be available along with the packet data service in this integrated network. We call this integrated network a *hybrid* cellular network. The GSM/GPRS cellular network is a good example of the hybrid cellular network.

The bandwidth guarantee[3] over the air provided by the circuit data service in the hybrid cellular network is of great advantage over the packet data service for some classes of applications requiring the end-to-end bandwidth guarantee because the wireless link is frequently a bottleneck of an application's entire end-to-end path due to its scarcity. Therefore, if dynamic bandwidth adaptation by alternating an *ongoing* data traffic between the circuit data service and the packet data service is provided in the hybrid cellular network, then this can used to offer different QoS to the adaptive application. In addition, since the wireless congestion in a cell can be handled by reducing the allocated bandwidth of individual connection, the dynamic bandwidth adaptation can be used as a tool to handle the dynamically changing load in a cell based on the desired tradeoff between the perceived QoS and the radio utilization.

In this paper, we propose a novel framework for dynamic bandwidth adaptation that allows an ongoing data traffic from or to the mobile station to alternate between the circuit data service and the packet data service using Mobile IP in the hybrid cellular network. While we will investigate our framework largely based on the GSM/GPRS cellular network, the proposed scheme can be easily applicable to any hybrid cellular network where the circuit data service and the packet data service coexist.

The rest of this paper is organized as follows. Section 2 describes how circuit and packet data services are established in the hybrid cellular network. Section 3 describes the proposed bandwidth adaptation scheme. Section 4 concludes the paper.

among active mobile stations. Some digital cellular networks such as GSM (Global System for Mobile communications) use packet radio, but packets from multiple active mobile stations are not multiplexed onto a channel.

[2] Each active mobile station multiplexed onto the same physical channel is assigned a *logical* channel.

[3] Due to the unstable nature of wireless link such as channel error, multi-path, and shadowing, the achieved bandwidth may be less than the nominal bandwidth.

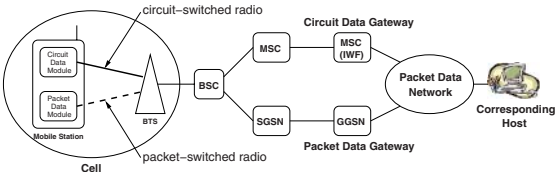


Fig. 1. Hybrid Cellular Network (GSM/GPRS network)

2 Data Services in GSM/GPRS Cellular Network

Fig. 1 shows the basic components involved for the circuit data service and the packet data service in the GSM/GPRS hybrid cellular network, omitting irrelevant nodes. In very broad terms, when used for circuit data service, a GSM network can be thought of as an ISDN network with base stations added to it to provide a wireless interface. Unlike first generation cellular systems, the modems reside in the core network, more precisely in the InterWorking Function (IWF). The IWF serves as a data service gateway to the packet data network (IP network). The IWF resides normally in the Mobile Switching Center (MSC) responsible for the routing of the calls, the tracking of the mobile users. The Base Transceiver Station (BTS) and Base Station Controller (BSC) form together the access network of the GSM/GPRS network and are shared between the circuit data service and the packet data service. The BSC is normally in charge of radio resource management of one or multiple cell sites. The BSC splits the circuit data traffic and the packet data traffic. The circuit data traffic is sent to the traditional ISDN-based GSM network while the packet data traffic is routed to the GPRS backbone network. Two new types of components are introduced in the GPRS backbone network, the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN). The SGSN, like the MSC in the GSM network, is responsible for the support of user mobility and the access control of users to the radio resources. The GGSN acts as an interface between the GPRS backbone network and the external packet data networks.

To start the circuit data service, a mobile station first needs to contact the circuit data gateway, here the InterWorking Function (IWF). The circuit data gateway can be either directly connected to the data packet network, e.g. IP network, or connected to the PSTN network that ultimately connects with some ISP for accessing the data packet network. In this paper, we assume the circuit data gateway is directly connected to the packet data network. One of the circuit data gateway's main functions is to allocate an IP address to the requesting mobile station from its IP address pool. The mobile station can get a static IP address or a dynamic IP address. In this paper, we assume the IP address assigned to a mobile station is changing on each request. This dynamic IP address allocation can be implemented through a DHCP server connected to the IWF. Once the connection between the mobile station and the IWF is established, the data transmission can start between the mobile station and the corresponding host. The physical radio channel allocated to a mobile station is dedicated to

the mobile station until the whole end-to-end data connection falls apart (the solid line in Fig. 1).

In the packet cellular network, a mobile station undergoes as similar process as in the circuit data service to initiate the packet data service. The mobile station contacts the packet data gateway, which is under the carrier's administrative domain, and asks for an IP address. After an IP address is allocated to the mobile station, the data exchange between the mobile station and the corresponding host can be initiated. Unlike the circuit-switched data service, several mobile stations are multiplexed onto a physical channel (the dashed line in Fig. 1). The allocated IP address is revoked and the packet data service should be re-initiated if the communication between the mobile station and the corresponding host is idle for some time[4].

The mobile station capable of accessing the circuit and packet-switched networks has two logical modules called *circuit data module* and *packet data module*. The routing table in the network layer of the mobile station dictates which data module the data traffic from the upper-layer application should be routed through. Depending on the these modules' connectivity, there may be three types of mobile stations in the hybrid cellular network.

1. The mobile station can be attached to both circuit and packet data services and operate both services *at the same time*.
2. The mobile station can be attached to both circuit and packet data services, but the mobile station can only operate one set of services at a time.
3. The mobile station can be exclusively attached to either circuit or packet data service.

In this paper, we assume the first type of mobile stations are roaming in the hybrid cellular network to be able to provide dynamic bandwidth adaptation. In the GSM/GPRS cellular network, for example, when a GPRS-enabled mobile station is in *Class-A mode of operation*, it supports simultaneous operation of the circuit-switched GSM and the packet-switched GPRS services. When a mobile station is connected to both circuit data service and packet data service at the same time, it consumes two logical radio channels simultaneously, which reside in separate physical channels. Since the mobile station has two IP addresses at this moment, one for the circuit data service and the other for the packet data service, it has to perform routing between these two data services in its network layer for the outgoing traffic. Depending on the routing decision, the data traffic from the upper-layer application is routed through the corresponding data module. On the other hand, the incoming data traffic through both circuit data module and packet data module is passed up to the application irrespective of the routing table.

[4] In the GPRS network, the mobile station should re-initiate the *GPRS attach* after the ready and the standby timers expire.

3 Proposed Dynamic Bandwidth Adaptation Scheme

Many bandwidth adaptation schemes have been introduced in cellular mobile networks. These schemes propose how to allocate or adjust wireless bandwidth efficiently for new, handoff, or existing connections when the load on the system changes dynamically. However, most of these schemes are very complex and difficult to deploy since they require wireless bandwidth to be split and merged at will [5, 6, 7]. This is why most of these schemes are still in the research area.

To address this deployment issue, the proposed dynamic bandwidth adaptation scheme is implemented as an add-on feature to the existing hybrid cellular network. In addition, most modifications are made onto the core network without disrupting the existing air interface protocols, so that the mobile station not conforming to the proposed scheme can bypass the add-on feature and be still compatible with our proposed framework.

Two challenging issues related to the propose scheme are to provide a transparent IP address change to the application during bandwidth adaptation and to provide fast adaptation, i.e., adaptation with low delay. To address these issues, we modify the Mobile IP and use it as a basic tool when bandwidth adaption by alternating an ongoing traffic between the circuit and the packet data service is performed. The major contributions of the proposed scheme are as follows:

1. Our scheme provide a transparent transition between the circuit and the packet data services while the connection is still in progress.
2. Our scheme is easy to deploy in cellular mobile networks, where the circuit data service and the packet data service coexist.
3. Our scheme can provide quality of service by demoting an circuit data connection to a packet data connection, which otherwise would be dropped during handoff due to its relatively large bandwidth demand.

3.1 Promotion & Demotion

In the proposed scheme, the bandwidth consumed by an ongoing data connection is adapted by being promoted or demoted by the user or the base station controller while the mobile station communicates with the corresponding host. *Demotion* forces an ongoing circuit data connection to a packet data connection that gets less bandwidth. *Promotion* is opposite.

To decide who has the capability of triggering the promotion and demotion, two mobile station modes, *user* and *system*, are defined. The transition between the user mode and the system mode is made only by the mobile station user (or application). When a mobile station is in the user mode, the current data service type is not changed until the mobile station user explicitly executes a promotion or a demotion, thereby making possible user-driven QoS control. On the other hand, when a mobile station is in the system mode, the mobile station's current data service type falls under the control of the corresponding base station controller, thereby making possible network-driven QoS control. Therefore, the base station controller will promote or demote only the mobile stations that are

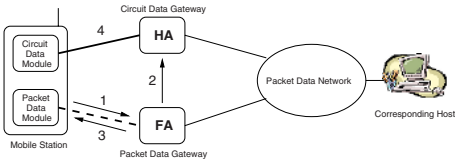


Fig. 2. Demotion in the Circuit-initiated Data Service

in the *system mode* based on the load in the cell and the desired network-wide QoS.

The procedures of promotion and demotion are slightly different whether the mobile station initiates its data service from the circuit data service or the packet data service, i.e., circuit-initiated data service or packet-initiated data service.

3.2 Circuit-Initiated Data Service

In this section, the detailed procedures of demotion and promotion are explained for the circuit-initiated data service. The circuit-initiated data service is either the circuit or packet data service that was initiated as a circuit data service and demoted or promoted zero or more times later. As noted above, when the mobile station initiates a circuit data service, i.e., performs a circuit-initiated data service, the data path is initially established through the circuit data gateway, the packet data network, and ultimately to the corresponding host. The data traffic from the corresponding host follows the reverse path as shown in Fig. 3 (a). The circuit data gateway and the packet data gateway act as a home agent (HA) and a foreign agent (FA) of Mobile IP, respectively as shown in Fig. 2. In this configuration, the demotion corresponds to the host migration from the home network to the foreign network in Mobile IP [4]. The promotion is opposite.

Demotion The demotion can be triggered either by the user or by the base station controller when the mobile station is currently in the circuit data service. The procedure is as follows.

1. The packet data module in the mobile station starts the demotion by sending to the packet data gateway a packet data service request message with the demotion bit set in the header of the request message, which specifies that the requested packet data service is for demotion (step 1 in Fig. 2). The circuit data gateway's IP address and the mobile station's current IP address are included in the packet data service request message. The address of the circuit data gateway has been obtained from the circuit data gateway when the mobile station initiated the circuit data service. The mobile station's current IP address together with its link-layer identifier[5] are recorded in the *demotion table* at the packet data gateway for later use.

[5] Gateways normally maintain the link-layer identifier (or routing number) for each active mobile station to route the incoming traffic to the right mobile station.

2. Additional IP address is allocated by the packet data gateway. At this moment, the mobile station has temporarily two IP addresses and two logical channels. To differentiate two IP addresses, we call two IP addresses a circuit IP address and a packet IP address, respectively. The traffic from or to the corresponding host is still routed through the circuit data module at this moment.
3. The packet data gateway, acting as a foreign agent (FA) in Mobile IP, contacts the circuit data gateway, whose address was provided from the mobile station, through the packet data network[6] to provide a *care-of address* that is the IP address of the packet data gateway (step 2 in Fig. 2).
4. As soon as the packet data gateway informs the circuit data gateway of its IP address, it sends an *route update* message back to the mobile station (step 3 in Fig. 2). The network layer in the mobile station updates its routing table when it receives the route update message, so that the data traffic from the upper-layer application be routed through the packet data module to the corresponding host. At this moment, the incoming traffic is still routed by the circuit data service.
5. After receiving the care-of address, the circuit data gateway delivers the incoming packets from the corresponding host using IP tunneling technique. Then the packet data gateway will strips the extra IP header and delivers the original packet to the mobile station. As in Mobile IP, the packet data gateway should not route the stripped packets based on the normal routing scheme. Instead, it searches the *demotion table* and finds the link-layer identifier of the mobile station with the IP address found in the stripped packet header. Using the link-layer identifier, it delivers the stripped packets to the mobile station. If there is no match in the demotion table, the packets are simply discarded.
6. At this point, everything seems to work correctly. However, the circuit data connection is still maintained between the circuit data module and the circuit data gateway, thereby wasting wireless bandwidth. Therefore, as soon as the tunneling from the circuit data gateway to the packet data gateway is enabled, the circuit data connection previously established for the circuit data service between the circuit data module and the circuit data gateway is disconnected by the circuit data gateway (step 4 in Fig. 2).

The data traffic before and after demotion are shown in Fig 3. An enhancement proposed in Mobile IP can be also applied here: a mobile station can act as a foreign agent, so that the tunneling from the circuit data gateway can be terminated at the mobile station.

Promotion In the circuit-initiated data service, a promotion can be triggered only after a demotion is completed (Fig 3 (b)). Basically, the promotion tries to restore the initial circuit data connection. The procedure is as follows.

[6] The packet data gateway can communicate with the circuit data gateway through a dedicated link.

1. Once a promotion is triggered, the circuit data module in the mobile station sends a circuit data service request message with the promotion bit set in the header of the request message (step 1 in Fig. 4).
2. The circuit data gateway searches its IP allocation database using the mobile station's link-layer identifier and finds the mobile station has already been allocated an IP address and the packets destined for this IP address is tunneled to the mobile station's foreign agent.
3. The circuit data gateway establishes a circuit data connection between itself and the circuit data module (step 2 in Fig. 4). Since this new connection establishment progresses in the *background*, it doesn't hurt the ongoing data traffic.
4. The circuit data gateway revokes the tunneling (step 3 in Fig. 4) and delivers the incoming packets from the corresponding host directly to the mobile station. At this moment, the packets from the mobile station are still routed through the packet data module and the packet data gateway to the corresponding host. The circuit data gateway asks the packet data gateway to de-register the mobile station.
5. The packet data gateway sends an route update message to the packet data module in the mobile station to make the traffic from the mobile station be routed through the circuit data module (step 4 in Fig. 4). Then, the packet data gateway de-registers the mobile station and redeems the allocated packet IP address.
6. The mobile station finishes the promotion by changing its routing table when it receives the route update message so that the data traffic from the upper-layer application be routed through the circuit data module.

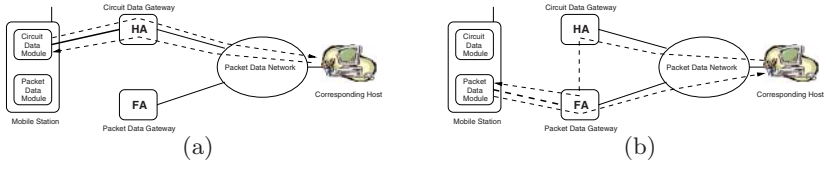


Fig. 3. Demotion (from (a) to (b)) and Promotion (from (b) to (a)) in Circuit-initiated Data Service

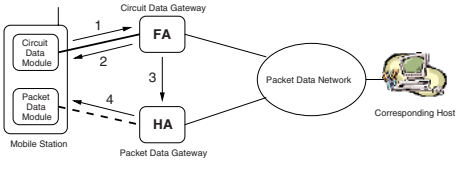


Fig. 4. Promotion in the Circuit-initiated Data Service

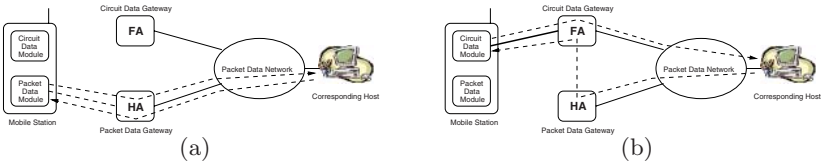


Fig. 5. Promotion (from (a) to (b)) and Demotion (from (b) to (a)) in Packet-initiated Data Service

3.3 Packet-Initiated Data Service

The packet-initiated data service is either the circuit or packet data service that was initiated as a packet data service and demoted or promoted zero or more times later. Unlike the circuit-initiated data service, the packet data gateway acts as a home agent. Therefore, it is the promotion that corresponds to the host migration from the home network to the foreign network.

Due to the lack of space, the detailed procedures of the promotion and the demotion shown in Fig. 5 are omitted. However, the promotion and the demotion in the packet-initiated data service undergo as similar procedures as in the circuit-initiated data service.

3.4 An Enhancement: Fast Demotion

Unlike the circuit-switched radio, the packet radio resource is not wasted during idle communication. The mobile station's unused radio resource share will be distributed out to the other mobile stations sharing the same physical channel. We take advantage of this to provide a fast demotion. The basic idea is to maintain the packet data connection and its context while the circuit data service is in progress, so that the demotion delay can be minimized when a demotion is requested later. Due to the lack of space, the fast demotion only in the circuit-initiated data service is explained here.

To maintain the packet data connection, the circuit data gateway should not ask the packet data gateway to de-register the mobile station in the step 4 of the promotion. Instead, the circuit data gateway asks the packet data gateway to set an internal flag called *packet connection status flag* for the mobile station, indicating whether the packet data connection is maintained or not. The circuit data gateway also maintains the tunneling information related to the mobile station. To prevent the expiration of the packet data connection after the packet data connection is maintained, the packet data module in the mobile station sends a tiny dummy packet periodically.

When the fast demotion is triggered, the outgoing traffic can be transmitted through the packet data module immediately because the packet data connection was already established. The packet data gateway bypasses the packet IP address allocation and asks the circuit data gateway to re-activate the tunneling. The circuit data gateway concludes the demotion by disconnecting the circuit data connection.

By maintaining the packet data connection, the procedures of promotion and demotion are simplified. However, the IP address allocated for the packet data connection is wasted during the circuit data service. In addition, if many mobile stations maintain the packet data connection during the circuit data service, this can limit the accomodation of new packet data service requests since the number of mobile stations sharing a physical channel is limited and these mobile stations already contribute to this number.

4 Conclusions

In this paper, we proposed a dynamic bandwidth adaptation scheme that allows an ongoing data traffic to alternate between the circuit-switched cellular network and the packet-switched cellular network using Mobile IP in the hybrid cellular network. Since our scheme is implemented as an add-on feather to the existing hybrid cellular network, it is easily deployed. The proposed scheme can be used as a tool for the network-driven QoS control to cope with the cell congestion. The base station controller may demote some of ongoing circuit data connections when the cell is overloaded and promote some of ongoing packet data connections when the cell is underloaded. The decision can be made based on the user differentiation, for example, premium or normal users.

References

[1] G. Brasche, B. Walke: Concepts, Services and Protocols of the New GSM Phase 2+ General Packet Radio Service. IEEE Communications Magazine Aug. (1997) 94–104 202

[2] 3GPP: Overall Description of the GPRS Radio Interface Stage 2. TS 03.64 V.8.0.0 (1999) 202

[3] 3GPP: General Packet Radio Service (GPRS) State 2. TS 23.060 V.4.0.0 Release 4 (2001)

[4] David B. Johnson: Scalable Support for Transparent Mobile Host Internetworking. Wireless Networks, Vol. 1, Oct. (1995) 311–321 206

[5] Y. B. Lin, A. Noerpel, D. Harasty: A nonblocking channel assignment strategy for hand-offs. IEEE ICUPC Sep. (1994) 205

[6] T. Kwon, S. Das, I. Park, Y. Choi: Bandwidth Adaptation Algorithms with Multi-Objectives for Adaptive Multimedia Services in Wireless/Mobile Networks. ACM WoWMoM Aug. (1999) 205

[7] A. K. Talukdar, B. R. Badrinath, A. Acharya: Rate Adaptation Schemes in Networks with Mobile Hosts. ACM MobiCom (1998) 205

A Dynamic Incentive Pricing Scheme for Relaying Services in Multi-hop Cellular Networks

Ming-Hua Lin[1] and Chi-Chun Lo[2]

[1] Department of Business Administration, NanHua University
32 Chung-Keng Li, Dalin, Chiayi, Taiwan, R.O.C.
mhlin@iim.nctu.edu.tw
[2] Institute of Information Management, National Chiao-Tung University
1001 Ta-Hseuh Road, Hsinchu, Taiwan, R.O.C.
cclo@faculty.nctu.edu.tw

Abstract. Applying peer-to-peer communications in cellular networks to improve performance has been an active research area in recent years. Cooperation among nodes is an important prerequisite for the success of the multi-hop cellular networks. Cost savings and service availability are the primary concerns that the network provider adopts multi-hop cellular networking technology. In this paper, we present a dynamic incentive pricing scheme to maximize the revenue of the network provider. The proposed scheme adjusts the price of the feedback incentives based on the network conditions to influence the relaying capability of the network and therefore increases the revenue of the network provider. The simulation results demonstrate that the revenue can be increased by dynamically adjusting the price of the incentives for relaying services. Moreover, the proposed pricing scheme results in more revenue but does not cause higher new call blocking probability in multi-hop cellular networks.

1 Introduction

Multi-hop cellular networking has been an active research area in recent years. In conventional cellular networks, mobile stations communicate directly with their assigned base station; in wireless multi-hop networks, mobile stations are located randomly and use peer-to-peer communications to relay their messages. Multi-hop cellular networks that combine the characteristics of both cellular and mobile ad hoc networks to leverage the advantages of each other have received increasing attention. Much research has evaluated and summarized the benefits of such a hybrid architecture [1, 3, 7, 9]:

- The energy consumption of the mobile device can be conserved.
- The interference with other nodes can be reduced.
- The number of fixed antennas can be reduced.
- The capacity of the cell can be increased.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 211–220, 2004.

- The coverage of thenetwork can be enhanced.
- The robustness and scalability of the system can be increased.

In multi-hop cellular networks, data packets must be relayed hop by hop from a given mobile node to a base station and vice-versa [1]. Cooperation among nodes is an important prerequisite for the success of the relaying ad-hoc networks. Some studies have suggested that certain incentives must be given to make the intermediate nodes willingly provide relaying services, but they do not discuss the detail of the feedback mechanism. Some research [9, 10, 11, 12, 13, 14, 15, 16] has described how to stimulate immediate nodes to forward data packets in multi-hop networks. The approaches can be classified into detection-based and motivation-based. The detection-based approach finds out misbehaving nodes and mitigates their impact in the networks. The motivation-based approach provides incentives to foster positive cooperation in ad hoc networks. Most of existing motivation-based approaches provide incentives for relaying services based on the number of the forwarding packets. The major advantage of the fixed rate is that billing and accounting processes are simple. However, the price of the incentives for relaying services is independent of the actual state of the network. Such system cannot react effectively to the dynamic and unpredictable variations of the wireless networks.

Since the intermediate nodes consume energy to provide connectivity between two nodes, it is reasonable to get some incentives for this service. In this paper, we propose a dynamic incentive pricing scheme to encourage collaboration and to maximize the revenue of the network provider that adopts multi-hop cellular networking model. Monetary incentives not only influence the motivation of the immediate nodes supporting relaying services but represent the cost of providing connection services in multi-hop cellular networks. Our focus here is on the incentive pricing scheme for revenue maximization. If the price of the incentives is too low, the number of successful connections will be small and the network provider can not get adequate profit from the relaying networks. However, if the price of the incentives is too high, the network provider can not cover the cost from the fee charged from end users. In this paper we indicate that dynamically adjusting the price of the incentives for forwarding services to affect the relaying capability of the network can make the network provider get maximum revenue and enhance the availability of services.

The rest of this paper is organized as follows. In section 2, we review the existing multi-hop cellular network models and incentive schemes for packets forwarding. Section 3 describes the detail of the proposed dynamic incentive pricing scheme for relaying services. Section 4 presents the simulation results and discussions. Finally, concluding remarks are recommended in section 5.

2 Literature Review

2.1 Multi-hop Cellular Network Model

Although many approaches in the literature have been proposed to improve the performance of cellular networks and multi-hop networks in isolation, more and more research focuses on integrating the cellular and multi-hop network models.

Aggélou et al. describe an Ad Hoc GSM (A-GSM) system that accommodates relaying capability in GSM cellular networks [2]. The authors extend the standard GSM radio interface with sufficiently flexible capabilities to support relaying. Qiao et al. present a network model called iCAR that integrates the cellular infrastructure and ad-hoc relaying technologies [3]. The proposed architecture places a number of Ad-hoc Relaying Stations (ARS) at strategic locations to relay data from one cell to another cell. Load balancing among different cells in the iCAR system not only increases system capacity, but also reduces transmission power for mobile terminals. Luo et al. propose a Unified Cellular and Ad-Hoc Network (UCAN) architecture to enhance the cell throughput. Each mobile device in the UCAN model has both 3G cellular link and IEEE 802.11-based peer-to-peer links. The 3G base station forwards packets for destination clients with poor channel quality to proxy clients with better channel quality. With the proposed greedy and on-demand protocol for proxy discovery and ad-hoc routing, the performance of a mobile user's access to the cellular infrastructure is improved by ad–hoc wireless connections [8].

Opportunity Driven Multiple Access (ODMA) is an ad hoc multi-hop protocol that the transmissions from a mobile host to the base station are broken into multiple wireless hops, thereby reducing transmission power [6, 7, 8]. ODMA is envisioned in the third-generation (3G) of cellular networks to extend the high-data-rate coverage of the cell at the boundaries. Hsieh et al. compare the performance between cellular networks and ad-hoc wireless networks [4]. They also investigate the impact of using peer-to-peer communications in cellular wireless packet data networks [5].

2.2 Incentive Scheme

Marti et al. describe two techniques to improve network throughput by detecting misbehaving nodes and mitigating their impact in ad hoc networks [10]. Since even a few misbehaving nodes can be a significant problem, they use a *watchdog* to identify misbehaving nodes and a *pathrater* to avoid routing packets through theses nodes. Although the proposed solution fosters cooperation in ad hoc networks, it does not castigate malicious nodes but rather mitigates the burden of forwarding for others.

Michiardi et al. suggest a mechanism called CORE based on reputation to enforce cooperation among nodes and prevent denial of service attacks due to selfishness [11]. Reputation directly related to the cooperative behavior of an entity is calculated based on subjective observations and indirect information provided by other members. The request from the entity with negative reputation

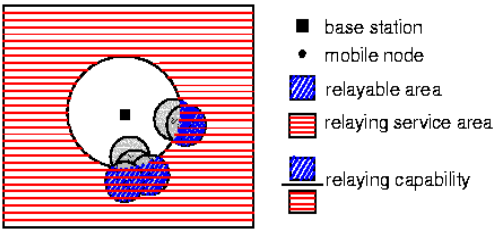


Fig. 1. Networking model

will not be executed. Buchegger et al. propose a protocol called CONFIDANT to detect and isolate misbehaving nodes, thus making it unattractive to deny cooperation [12]. Non cooperating nodes are learnt from experienced, observed, or reported routing and forwarding behavior of other nodes. Both two methods discourage misbehavior by identifying and punishing misbehavior nodes.

Buttyán et al. use a virtual currency called *nuglets* as incentives given to cooperative nodes in every transmission [13]. The proposed models do not discuss the number of nuglets should be feedback to the intermediate nodes. Buttyán et al. also propose a mechanism based on credit counter to stimulate packet forwarding [14]. Each node keeps track of its remaining energy and remaining number of ngulets. The nuglet counter is decreased when the node sends an own packet, increased when the node forwards a packet. The number of feedback nuglets depends on the number of forwarding packets in this method. In [17], Buttyán et al. present a micro-payment scheme that fosters collaboration and discourages dishonest behavior in multi-hop cellular networks. Packet originators associate subjective reward levels with packets according to the importance of the packet.

Zhong et al. propose a system to provide incentives for mobile nodes to cooperate and report actions honestly [15]. The proposed system is a pure-software solution and do not require any tamper-proof hardware at any node. A central authority is in charge of collecting receipts from the forwarding nodes and then determining the charge or the reward to each node depending on the reported receipts. The system focuses on selfish nodes and determines payment and charges from a game-theoretic perspective.

Lamparter et al. propose a charging scheme in hybrid cellular and multi-hop networks, which would be beneficial for Internet Service Provider and the ad hoc nodes and thus motivates cooperation among mobile nodes [16]. The charging scheme is based on volume-based pricing models. A fixed price per unit is rewarded for forwarding traffic irrespective of the network conditions. In [1], the authors propose an incentive mechanism based on a charging/rewarding scheme in multi-hop cellular networks [1]. The solution relies exclusively on symmetric cryptography to compliant with the limited resources of the mobile stations. Both the charge of sending the data packet and the reward of forwarding the data packet depend on the packet size in the proposed method.

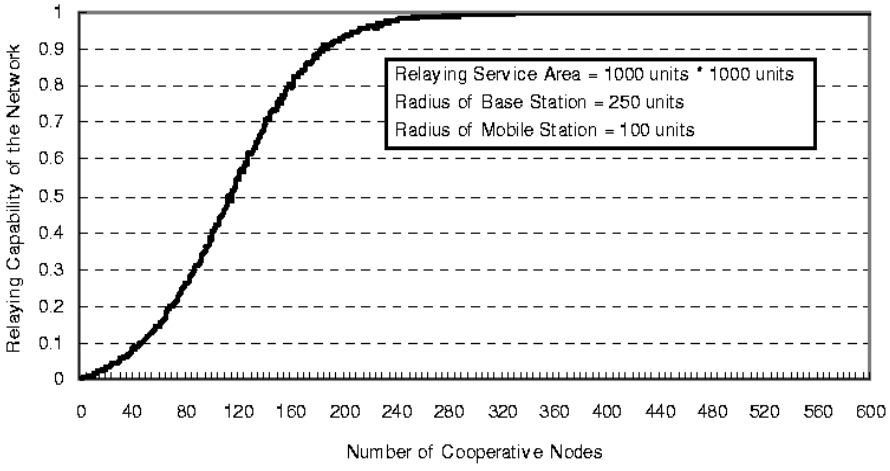


Fig. 2. The relationship between the relaying capability of the network and the number of cooperative nodes

3 Dynamic Incentive Pricing Scheme

Much research has discussed how to stimulate immediate nodes to provide relaying services in multi-hop cellular networks, but most of them use static incentive pricing schemes and do not consider the current state of the network. Since fewer base stations might be needed or more customers could be served by integrating multi-hop communications in the cellular networks, cost savings and service availability are the primary concerns that the network provider adopts multi-hop cellular networking technology. Consequently, the network provider apparently necessitates an appropriate pricing strategy to maximize its revenue based on the actual network conditions.

In this paper, we focus only on a single base-station cell as indicated in Fig. 1. We define the *relaying capability* (RC) of the network as the percentage of the relaying service area in which mobile nodes can connect to the base station through peer-to-peer communications. Cooperation among nodes plays a critical role in the success of the multi-hop cellular networks therefore the number of cooperative nodes has a significant impact on the relaying capability of the multi-hop cellular networks. We build a simulation model to observe the influence of the number of cooperative nodes on the relaying capability of the network when mobile nodes are randomly located inside the service area of the base station. As the relationship illustrated in Fig. 2, the relaying capability of the network increases as the number of cooperative nodes increases.

From above observation, we define the relaying capability of the network RC as a function of the number of cooperative nodes n, i.e., $RC = f(n)$, $f(n)$ is a function of n with the following properties:

$$0 \leq f(n) \leq 1; \quad f(n=0) = 0; \quad \lim_{n \to \infty} f(n) = 1 \ . \tag{1}$$

In cooperating groups, such as emergency and military situations, all nodes belong to a single authority and have a good reason to support each other. However, in the emerging civilian applications, the nodes do not belong to a single authority. Consequently, cooperative behaviors can not be directly assumed [15].

Monetary incentives can affect the motivation of mobile nodes providing services and is usually characterized by a supply function that describes the reaction of mobile nodes to the change of the price [18]. Here we use a general supply function as follows:

$$S[p(t)] = \begin{cases} e^{-\left(\frac{p_0}{p(t)}-1\right)^2}, & \text{when } 0 < p(t) \leq p_0; \\ 0, & \text{when } p(t) = 0 \ . \end{cases} \tag{2}$$

where p_0 is the maximum price that network provider can feedback, $p(t)$ is the price of the feedback incentives per unit of relay data at time t. In our scheme, $p(t)$ is adjusted based on the network conditions at time t. $S[p(t)]$ denotes the percentage of mobile nodes that will accept the price to forward data packets. Note that $S(p_0) = 1$, which means that the maximum price is acceptable to all mobile nodes to provide relaying services. For $p(t) = 0$, we have $S(0) = 0$, which means that no mobile node is willing to relay traffic for others if no feedback is provided for relaying service.

Let p_t denotes the price of the feedback incentives, N_t be the number of mobile nodes and RC_t be the relaying capability of the network at time t. From our observation, RC_t is a function of the number of cooperative nodes that depends on total number of mobile nodes and their willingness to support relaying services, that is,

$$RC_t = f(N_t S(p_t)) \ . \tag{3}$$

Assume K_t is the number of mobile nodes that request data transmission at time t. The availability of the relaying paths is determined by RC_t, thus the number of successful connections M_t at time t is given by

$$M_t = K_t RC_t \ . \tag{4}$$

Assume the static usage-based charging model is accepted by the end users, i.e. end users agree to pay the network provider (base station) a fixed price u, per unit of data transmitted in each hop. Let v_i be the unit of data sent by user i, h_i be the number of hops exists between user i and the base station at time t, the revenue from the service for user i is $(u - p_t)h_i v_i$. Since the base station is interested in maximizing its revenue, the corresponding maximization problem at time t is given as follows:

$$\begin{aligned} &\text{Maximize} \quad R = \sum_{i=1}^{M_t} ((u - p_t) h_i v_i) \\ &\text{Subject to} \quad M_t = K_t R C_t = K_t f(N_t S(p_t)), \\ &\qquad S[p(t)] = \begin{cases} e^{-\left(\frac{u}{p(t)} - 1\right)^2}, & \text{when } 0 < p(t) \leq u; \\ 0, & \text{when } p(t) = 0 \ . \end{cases} \end{aligned} \tag{5}$$

In order to maximize the revenue, the network provider should increase p_t to enhance the relaying capability of the network and therefore increase the number of successful connections. However, the increase in p_t will decrease the revenue $(u - p_t)h_i v_i$ from user i. Consequently, the network provider should dynamically adjust p_t based on the actual network conditions to maximize the revenue.

For $p_t = 0$, we can obtain $M_t = K_t f(N_t S(p_t = 0)) = 0$, and for $p_t = u$, we can obtain $M_t = K_t f(N_t S(p_t = u)) = K_t f(N_t) \leq K_t$. For $i = 1, ..., M_t$, the revenue $(u - p_t)h_i v_i$ received from user i decreases as p_t increases. For $p_t = 0$, we have $(u - p_t)h_i v_i = u h_i v_i$, and for $p_t = u$, we have $(u - p_t)h_i v_i = 0$.

Since both M_t and $(u - p_t)h_i v_i$ have a minimum and a maximum value over the closed interval $p_t \in [0, u]$, $R = \sum_{i=1}^{M_t} ((u - p_t)h_i v_i)$ also has a minimum and a maximum value over the same interval.

From the above discussion, the minimum value is obtained at the endpoints of the closed interval. Specifically, R is zero either when $p_t = 0$, which corresponds to the case that no successful relaying connection exits in the networks, or when $p_t = u$, which corresponds to the case that the charges from end users can not cover the cost of providing relaying services.

Let R_t^m be the maximum value of the revenue of the network provider at time t. From above analysis, we conclude that there exists at least one p_t value(s), denoted by p_t^i $(i = 0, 1, ...)$, over interval $(0, u)$ such that:

$$R(p_t = p_t^i) = R_t^m \ . \tag{6}$$

If only one p_t^i exists that satisfies (6), the optimal price of the feedback incentives is $p_t^* = p_t^0$. If more than one different values of p_t^i satisfy (6), we set the optimal price of the feedback incentives to be $p_t^* = \sup_{i \in \{0,1,...\}} \{p_t^i | R(p_t^i = R_t^m)\}$, which is the highest price of the feedback incentives that can maximize the total revenue of the network provider. The reason we select the maximum p_t^i is that the number of successful connections increases as the price of the feedback incentives increases, therefore the network provider can support relaying services with lower new call blocking probability.

4 Simulation Results

In this section, we evaluate the performance of the proposed dynamic incentive pricing scheme in terms of the revenue of the network provider and new call blocking probability. We demonstrate that the proposed pricing scheme achieves

to maximize the revenue of the network provider and decrease the new call blocking probability.

4.1 Simulation Model

The parameters used throughout our performance evaluation are as follows:

- A rectangular region of size 1000 units by 1000 units with a single base station located in the central point and various number of randomly distributed mobile nodes is used as the network topology. The radius of the base station is 250 units and the radius of each mobile node is 100 units.
- The number of active mobile nodes in the service area is varied with time. The variation of the number of the active mobile nodes during a 24-hour period used throughout our study is indicated in Fig. 3.
- We assume all nodes in the relaying area request data transmissions to the base station. The volume of the data sent is randomly distributed between 0 and 100 units.
- For finding a successful connection between each mobile node and the base station, the shortest path routing protocol is used.
- In the following numerical study, we use the supply function as following:

$$S[p(t)] = \begin{cases} e^{-\left(\frac{u}{p(t)}-1\right)^2}, & \text{when } 0 < p(t) \leq u; \\ 0, & \text{when } p(t) = 0 \ . \end{cases} \tag{7}$$

  where u is the price per unit of data transmitted in each hop charged from end users, and is also the maximum price that network provider willingly to feedback to the immediate nodes for relaying services.
- The 24-hour period is divided into 10-minute sections. At the end of each section, the optimal price p_t^* of the feedback incentives to maximize the revenue is calculated.

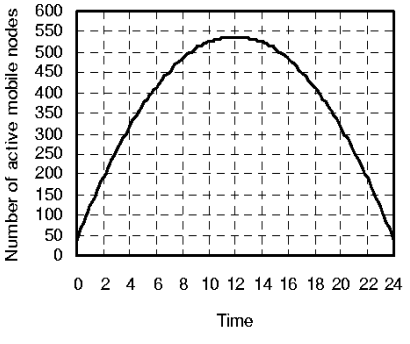


Fig. 3. Number of active mobile nodes in the simulation area

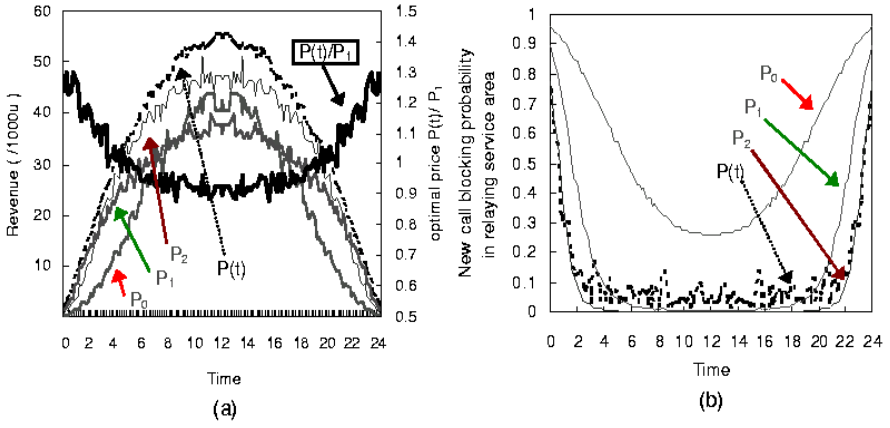


Fig. 4. Optimal price and revenue of the network provider and new call blocking probability in relaying service area for different pricing schemes

4.2 Simulation Results and Discussions

We compare the dynamic incentive pricing with the static incentive pricing. We use p_0 ($S(p_0) = 0.25$), p_1 ($S(p_1) = 0.5$), and p_2 ($S(p_2) = 0.75$) as the fixed prices, which represent that 25 percent, 50 percent and 75 percent of the mobile nodes will accept this fixed price to forward data for others. Figure 4(a) depicts how the price is adjusted according to the change of the network conditions during the 24-hour period. When the number of the cooperative nodes increases, i.e. the relaying capability of the network increases, the network provider should decrease the price $(p(t)/p_1)$ of the incentives to maximize its revenue. Figure 4(a) also illustrates the revenue of the network provider by adopting the proposed scheme and the three static pricing schemes respectively. We find that the proposed pricing scheme results in higher revenue of the network provider than the static pricing schemes. Figure 4(b) indicates the new call blocking probability in the relaying service area for different pricing schemes. We find that the proposed pricing scheme has more revenue but does not cause higher new call blocking probability than other schemes.

5 Conclusions

Cost savings and service availability are the primary concerns that the network provider adopts multi-hop cellular networking technology. In this paper, we present a dynamic incentive pricing scheme to maximize the revenue of the network provider. The proposed scheme adjusts the price of the feedback incentives based on the actual network conditions to affect the relaying capability of the network and therefore increases the revenue of the network provider. The simulation results demonstrate that the revenue can be increased by dynamically

adjusting the price of the incentives for relaying services. Furthermore, the proposed pricing scheme does not cause higher new call blocking probability than other schemes in relaying service area.

References

[1] Ben Salem, N., Buttyán, L., Hubaux, J. P., Jakobsson, M.: A Charging and Rewarding Scheme for Packet Forwarding in Multi-Hop Cellular Networks, ACM MOBIHOC 2003, June 2003. 211, 212, 214

[2] Aggélou, G. N., Tafazolli, R.: On the Relaying Capacility of Next-Generation GSM Cellular Networks, IEEE Personal Communications, pp.40-47, Feb. 2001. 213

[3] Qiao, C., Wu, H.: iCAR: an Intelligent Cellular and Ad-hoc Relay System, Proc. of IEEE IC3N, pp.154-161, Oct. 2000. 211, 213

[4] Hsieh, H.-Y., Sivakumar, R.: Towards a Hybrid Network Model for Wireless Packet Data Networks, Proc. of IEEE ISCC, July 2002. 213

[5] Hsieh, H.-Y., Sivakumar, R.: On Using the Ad-hoc Network Model in Wireless Packet Data Networks, ACM MOBIHOC 2002, June 2002. 213

[6] 3G TR 25.924 V 1.0.0. 3GPP TSG-RAN; Opportunity Driven Multiple Access, Dec. 1999. 213

[7] Rouse, T., Band, I., McLaughlin, S.:Capacity and Power Investigation of Opportunity Driven Multiple Access (ODMA) Networks in TDD-CDMA Based Systems, Proc. of IEEE ICC, pp.3202-3206, April 2002. 211, 213

[8] Luo, H., Ramjee, R., Sinha, P., Li, L., Lu, S.: UCAN: A Unified Cellular and Ad-hoc Network Architecture, ACM MOBIHOC 2003, June 2003. 213

[9] Jakobsson, M., Hubaux, J. P., Buttyán, L.: A micropayment scheme encouraging collaboration in multi-hop cellular networks, Proc. of Financial Crypto 2003. 211, 212

[10] Marti, S., Giuli, T. J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks, Proc. of ACM MOBICOM, pp.255-265, Aug. 2000. 212, 213

[11] Michiardi, P., Molva, R.: Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks, Proc. of the sixth IFIP Communications and Multimedia Security Conference, Sep. 2002. 212, 213

[12] Buchegger, S., Boudec, J. Y.L: Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Dynamic Ad-hoc Networks, ACM MOBIHOC 2002, June 2002. 212, 214

[13] Buttyán, L., Hubaux, J. P.: Enforcing Service Availability in Mobile Ad Hoc WANs, Proc. of ACM MOBIHOC, Aug. 2000. 212, 214

[14] Buttyán, L., Hubaux, J. P.: Stimulating cooperation in self-organizing mobile ad hoc networks, ACM/Kluwer MONET, Vol. 8, No. 5, Oct. 2003. 212, 214

[15] Zhong, S., Chen, J., Yang, Y. R.:Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks, Proc. of IEEE INFOCOM 2003, April 2003. 212, 214, 216

[16] Lamparter, B., Paul, K., Westhoff, D.: Charging Support for Ad Hoc Stub Networks, Journal of Computer Communication, Elsevier Science, Vol. 27, Issue 13, Aug. 2003. 212, 214

[17] Karl, H., Mengesha, S., Hollos, D.: Relaying in Wireless Access Networks, Business Briefing: Wireless Technology 2002, World Markets Research Center, Jan. 2002. 214

[18] Hou, J., Yang, J., Papavassiliou, S.: Integration of Pricing with Call Admission Control to meet QoS Requirements in Cellular Networks, IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 9, pp.898-910, Sep. 2002. 216

A Mobility-Based Mobile Multicast with Flexible Range

Seungpil Shin[1], Rhan Ha[1], and Hojung Cha[2]

[1] Dept. of Computer Engineering, Hongik University, Seoul 121-791, Korea
{spshin,rhanha}@cs.hongik.ac.kr
[2] Dept. of Computer Science, Yonsei University, Seoul 120-749, Korea
hjcha@cs.yonsei.ac.kr

Abstract. Providing multicast services to mobile hosts in wireless mobile computing is difficult due to dynamic location of the mobile host and dynamic group membership. The current multicast protocol in wired network assumes static hosts, thus it cannot be used directly to mobile computing environments. To solve this problem, several mobile multicast protocols based on Mobile IP have been proposed. Although they provide multicast service to mobile hosts, there are still problems such as tree maintenance overhead due to frequent reconstruction of multicast tree, non-optimal routing path, and service disruption, and so on. In this paper, we propose mobile multicast scheme using variable service range according to the mobility of mobile hosts and resource reservation. We evaluated the proposed scheme comparing with previous schemes by various simulation experiments. The experimental results show improvements of our scheme over previous approaches, namely remote subscription and RBMoM.

1 Introduction

In order to provide efficient multimedia services to mobile users, multicast scheme is essential for efficient use of resource. However, there are many problems to apply current multicast protocols to wireless network directly[1-3], since the current Internet multicast protocols assume static hosts in wired networks. They do not take into account dynamic change of host location. The mobile multicast protocol has to deal with not only dynamic group membership but also dynamic change of member location for efficient multicast in mobile computing. In current mobile IP, a mobile host(MH) is assigned an IP address of the mobile host's home agent(HA) in a home network. When the mobile host connects to a foreign network, it is registered at the foreign agent(FA) and some form of tunneling is established for message delivery. The current version of Mobile IP is classified into two approaches to support mobile multicast, that is bi-directional tunneling(HA-based) and remote subscription(FA-based), according to how to manage multicast group members and execute *join/leave* operations [3].

In bi-directional tunneling, the multicast delivery tree is constructed by HA. Data delivery is achieved by mobile IP tunneling via HA. When HA receives

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 221–230, 2004.

a multicast data destined for a mobile host, it sets up the tunnel between HA and FA and transmits data to FA. The main drawback of this approach is that the routing path to mobile hosts is non-optimal, hence the network bandwidth might be wasted. Moreover, HA has to duplicate and deliver the tunneled multicast data to their corresponding foreign agents and all mobile hosts. In remote subscription, a mobile host must join to the new multicast group whenever it moves to a foreign network. This scheme has the advantages of offering optimal routing paths and using efficient resources. However, when the mobile host has a high mobility, its multicast service may be very expensive because of the cost of managing the multicast tree. The overhead is the cost of reconstructing the delivery tree whenever *join/leave* operation occurs during a handoff. Furthermore, the extra delay for rebuilding a multicast tree can cause the possibility of a service disruption.

MoM [4] is the HA-based protocol to solve the tunnel convergence problem. In MoM, however, a DMSP [4] handoff problem can be caused. A DMSP handoff means that FA reselects a new DMSP. Hence, a DMSP handoff may occur in two situations. One is that a new mobile host enters a new network and new HA is more suitable for the DMSP. The other is that all the mobile hosts of the current DMSP move from the network to other networks. Multicast packets for mobile hosts will be lost during this handoff period and many multicast packets will be lost if this DMSP handoff occurs frequently.

RBMoM [5] uses service range of Multicast Home Agent(MHA) that is defined by hop count. MHA is responsible for tunneling multicast packets to the current FA. RBMoM is a hybrid scheme of the remote subscription and the bi-directional tunneling. It reconstructs the multicast delivery tree when the mobile host moves out of service range. Therefore, it reduces the overhead of multicast tree reconstruction and provides suboptimal routing path. When the mobile host moves out of its MHA service range, the first visited FA is reselected as MHA and joined the new multicast group. Then MHA forwards multicast data to mobile hosts within its service range. RBMoM fixes the service range of every MHA. It first establishes the service range according to the number of current members in the multicast group. It assumes that mobile hosts have the same mobility. However, the number of mobile hosts in a group changes dynamically and mobile hosts have the various mobilities. Therefore, the fixed service range does not reflect the real mobile computing environments and the service disruption will occur in the case of reselecting MHA when MH is moving out of the service range.

Furthermore, RSVP [6] has been proposed for efficient data transmission of multimedia services in wired network. RSVP needs just resource reservation message at connection time because host is fixed in wired network. However, RSVP does not consider host's mobility, thus RSVP cannot be directly applied to wireless network because it can not find the new location of the mobile host. MRSVP [7], an extended version of RSVP to support host mobility, avoids unnecessary resources waste by reservation failure through *passive/active* reservation. However, MRSVP exchanges reservation messages with each mobile host

to do passive reservation. Moreover, it has the scalability problem and the management overhead for each mobile host, since MRSVP continuously exchanges resource reservation for each mobile host.

In this paper, we propose the multicast scheme with variable service range according to host mobility-types [8] and resource reservation to reduce service disruption by using direction information of the mobile host through GPS(Global Position System) [9] when mobile host is moving out of its service range. In addition, our proposed scheme, VRBMoM(Variable Range-Based Mobile Multicast), manages variable service range as one group to reduce overhead of managing mobile hosts when making resource reservation. Various simulation experiments show that VRBMoM reduces the overhead of multicast tree reconstruction and minimizes service disruption time.

The rest of the paper is organized as follows. Section 2 presents a mobility-based variable service range mobile multicast scheme. In Section 3, we describe the performance evaluation and compare VRBMoM with previous techniques. Finally, Section 4 concludes the paper.

2 Variable Range-Based Mobile Multicast

In this paper, we propose a mobile multicast scheme that can reduce the reconstruction overhead of multicast delivery tree and offer more persistent service by reducing service disruption time when the mobile host moves out of the service range. We call the scheme as VRBMoM(Variable Range-Based Mobile Multicast). VRBMoM establishes MHA's service range variably according to mobilities of the mobile hosts belonging to the multicast group, and uses resource reservation for the mobile host to receive the multicast data continuously even when the host moves. RBMoM tries to reduce the tree maintenance overhead by using the service range, but it uses the fixed service range without considering the mobility types of hosts. RBMoM establishes the service range according to the number of members in the multicast group, and it assumes that mobile hosts have the same mobility. However, the number of mobile hosts changes dynamically and mobile hosts have the various mobility characteristics. Therefore, the fixed service range does not reflect the actual mobile computing environments. If mobile hosts with high mobility receive multicast service with large service range, the reconfiguration overhead of multicast delivery tree can be reduced than with small service range. On the contrary, mobile hosts with low mobility can have shorter routing path length if they receive multicast service with small service range than with large service range. VRBMoM classifies the mobilities of the mobile hosts into three mobility types, and uses the mobility types of hosts in the group to establish the variable service range of the group. Table 1 shows the mobility-types.

The pico-mobility is almost-motionless case, it means that the mobile host does not move or move slightly. The micro-mobility means that the mobile host moves little faster than pico-mobility by a vehicle, and the macro-mobility means that the mobile host moves very fast by a vehicle or the train, etc.

Table 1. Mobility Type

Mobility	Speed
`pico-mobility`	0-10km/hour : walking
`micro-mobility`	10-50km/hour : vehicle
`macro-mobility`	50-200km/hour : vehicle, train

Table 2. Notation

R_t	variable service range at time t
p	the number of hosts with pico-mobility
mi	the number of hosts with micro-mobility
ma	the number of hosts with macro-mobility
W_p	the weight of pico-mobility
W_i	the weight of micro-mobility
W_a	the weight of macro-mobility

Table 3. Service Range

$R_n = R_{n+1}$	when service range doesn't change
$R_n < R_{n+1}$	when service range increases
$R_n > R_{n+1}$	when service range decreases

The variable service range is determined by using mobility information as in Table 1 and using Equation (1). The meaning of notation using in Equation (1) is in Table 2.

$$R_t = \lceil p \times W_p + mi \times W_i + ma \times W_a \rceil \tag{1}$$

Equation (1) means that variable service range R_t is the sum of multiplication of the number of the mobile hosts and corresponding weight of the mobile hosts mobility types. The reason why each mobility type has different weight is to compute the service range properly. For example, if the numbers of mobile hosts of each mobility-type are all the same or similar in the current service range, the service range may have to be decided by macro-mobility. Similarly, if there are more mobile hosts with pico-mobility than with micro-mobility, macro-mobility in the current service range, the service range is set according to the pico-mobility. Because the macro-mobility can affect current service range greatly when the number of mobile hosts is changed due to moving of hosts, the pico-mobility has the smallest weight. Similarly, the micro-mobility has bigger weight than pico-mobility, and macro-mobility has the biggest weight.

MHA tunnels multicast data to the hosts in the service range established by Equation (1). MHA compares current service range with previous service range and adjusts the service range if necessary. The service range adjustment follows Table 3. When the previous service range is R_n and the current service range is R_{n+1}, if R_n equals to R_{n+1}, there is no need to adjust service range because

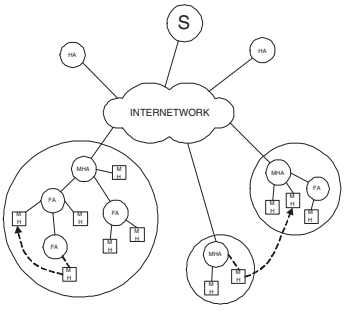


Fig. 1. Variable Service Range Multicast

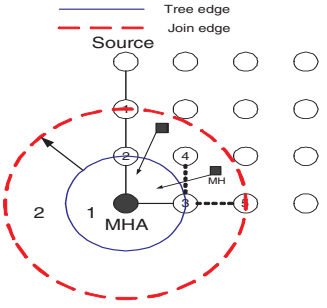


Fig. 2. Reconfiguration of multicast tree

```
/* MH registers at FA;
   FA gets the location of MHA;
   FA computes the hop distance to MHA; */
PROCEDURE VRBMoM()
  if (Distance(FA, MHA) ≥ R_n) { /* out of the current service range */
      if (MHA == HA) {
          MHA = FA;
          R_{n+1} = ⌈p × W_p + mi × W_i + ma × W_a⌉; /* computes new service range */
      }
      else if (Distance(FA, HA) ≤ R_n) {
          if (FA in the multicast_tree) {
              Attach(FA, MHA); /* FA attach to other MHA;
              R_{n+1} = ⌈p × W_p + mi × W_i + ma × W_a⌉;
          }
          else {
              MHA = FA;
              Join(MHA, multicast_tree);
              R_{n+1} = ⌈p × W_p + mi × W_i + ma × W_a⌉;
          }
      else {
          MHA = HA;
          Join(MHA, multicast_tree);
          R_{n+1} = ⌈p × W_p + mi × W_i + ma × W_a⌉;
      }
      Inform(MH, FA); /* Inform MH information to FA */
      Inform(MH, MHA); /* Inform MH information to MHA */
      Delete(MH, oldMHA); /* delete all data structures about the MH of old MHA */
      Reconfigure(R_n, R_{n+1}); /* reconfigure service range if it is necessary */
      FA join or Quit to multicast_tree;
      }
  }
  else
      Continue Service in R_n;
END PROCEDURE
```

Fig. 3. VRBMoM Algorithm

mobile hosts move slightly or the mobility of hosts cannot incur readjustment of the service range. However, if R_{n+1} is greater than R_n, there are many mobile hosts with high mobility, therefore MHA increases the service range to reduce reconstruction overhead of multicast delivery tree. On the contrary, if R_n greater than R_{n+1}, it means that there are many mobile hosts with pico-mobility. Therefore, it is efficient that MHA decreases the service range. However, some mobile hosts happen to be out of service range due to the decreased service range. It happened when the number of mobile hosts before decreasing the service range and the number of mobile hosts after decreasing the service range becomes different. If this case happens, VRBMoM keeps the current service range without decreasing the service range, thus mobile hosts in the current service range can receive multicast service continuously.

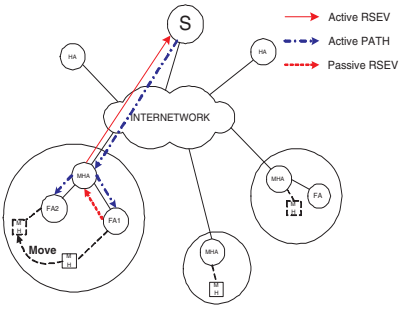


Fig. 4. Moving within service range

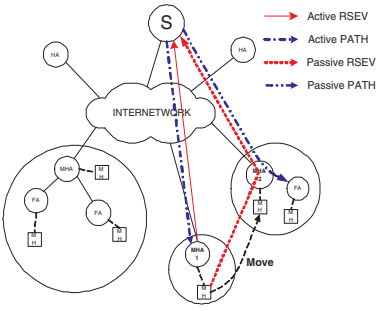


Fig. 5. Moving out of service range

Figure 1 shows that VRBMoM sets the service range differently according to the mobility types of mobile hosts. The large service range of MHA means that there are many mobile hosts with high mobility. It reduces the tree maintenance overhead when the mobile host moves within service range. In addition, the small service range of MHA means that there are many mobile hosts with pico-mobility. It reduces the routing path length by decreasing tunneling length. The dotted line shown in Figure 1 means that the mobile host moves from the current FA or MHA to the other FA or MHA.

Figure 2 shows reconfiguration of multicast tree by changing of the service range. The service range is 1 initially, and FAs that belong to current service range are FA2, FA3. At this time, if FA4 and FA5 are joined to multicast tree because the service range increases to 2, there is no change of routing path to MHA(core). VRBMoM establishes all route paths by attaching FA4 and FA5 to existent routing path. On the contrary, when the service range decreases from 2 to 1, VRBMoM reconstructs the multicast tree by deleting FA4 and FA5 in multicast tree without changing of path to MHA. That is, reconfiguration of the multicast tree due to change of service range does not change completely but locally. Figure 3 details VRBMoM algorithm.

In MRSVP [7], the sender and the receiver exchange *passive/active* message every time whenever the mobile host moves to other location. In VRBMoM, however, the sender and the receiver exchange *pasive/active* message once when the mobile host enters into the new service range and MHA changes the state from passive to active when the mobile host actually moves to the other location in the service range. Therefore, it decreases the number of message exchanges. Figure 4 shows a resource reservation in the service range, passive PATH message is not needed because it was sent to all route paths inside the service range when the mobile host joined to this service range. VRBMoM changes from passive PATH to active PATH and the sender transmits data if the mobile host moves from FA1 to FA2. Moreover, if the mobile host moves out of the service range, it makes a resource reservation using GPS [9] information of direction that is visited. At this time, the mobile host sends passive RESV message to the sender

of other location to visit, and the sender sends passive PATH message to all route paths inside the service range. Figure 5 shows resource reservation when the mobile host moves out of the service range. If the mobile host in the service range of MHA1 moves into the service range of MHA2, it sends passive RESV message to the sender via MHA2, and the sender sends passive PATH message to all locations within service range of MHA2 and set route paths to all locations. This passive state is changed to active state when the mobile host moves inside service range of MHA2, and this is similar to move within service range.

If the mobile host is in boundary service range, the mobile host sends its information to nearby MHA and VRBMoM prepares for re-set of service range or resources reservation. The decision of the boundary service range(B_j) by each mobility-type follows Equation (2).

$$B_j = R_t \times (1 - W_j) \tag{2}$$

B_j has three values according to the mobility-types shows in Table 1, and has the smaller value than the current service range. j is used to express weight of each mobility-type. B_j has the biggest value when the mobility of host is pico-mobility, and has the smallest value when the mobility of host is macro-mobility. VRBMoM makes a resource reservation when the mobile host is between B_j and R_t, because the more the number of mobile hosts with high mobility, the more the number of mobile hosts staying shortly time between R_t and B_j. Similarly, it is explained on the contrary when the mobile host has a low mobility. R_t is the current service range of MHA and W_j is the weight that explained in Table 3. If the mobile host is between R_t and B_j, the mobile host sends its information to another MHA which is expected to visit and VRBMoM prepare for re-set of service range and resource reservation. When the mobile host actually moves out of the current service range of MHA, VRBMoM re-establishes the service range and makes a resource reservation. MHA which receives information of the mobile host decides whether it resets the service range or nor. In addition, the previous service range is re-established if necessary.

3 Performance Evaluation

We have evaluated performance of VRBMoM using NS-2 [10]. The topology is based on a 10×10 mesh network in our simulation. Initially, VRBMoM selects the HA as the MHA and calculates the service range by Equation (1). We assume that each mobility weight $W_p = 0.1$, $W_{mi} = 0.3$, $W_{ma} = 0.5$, respectively. The average arriving interval of multicast packet is $pktsize * 8/rate$, and it follows an exponential distribution with $\lambda = (1/rate) * pktsize * 8$. Because the packet size is expressed in term of *byte*, we multiply 8 to *pktsize*. In addition, we assume that the traffic rate is 2.4Mbps, packet size is 512byte and the number of the group member changes from 10 to 50. The speed of each mobile host varies from 0 to 200km/h, and the distance between each node is 500m. We assume that each join to the multicast group requires 20msec, and each registration time to the MHA requires 5msec. We measured the number of reconfiguration of

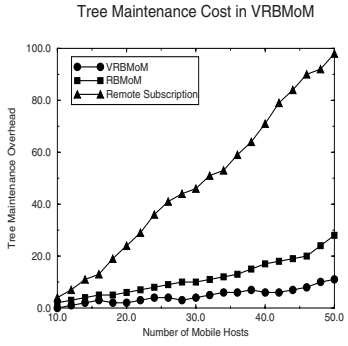


Fig. 6. Tree maintenance overhead in 10×10 network

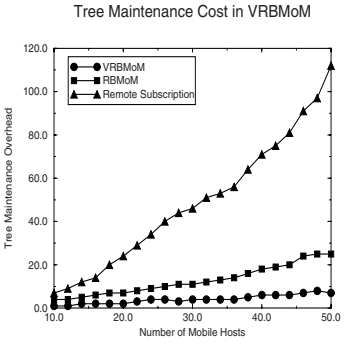


Fig. 7. Tree maintenance overhead in 20×20 network

the multicast tree and service disruption time, and routing length. We compare VRBMoM with RBMoM [5] and remote subscription [3]. The number of multicast tree reconfiguration is measured by the number of *join/leave* operations of the multicast group and the service disruption time is measured by registration time of the mobile host to the new service range. Tree maintenance overhead, service disruption time, routing path length and resource reservation overhead are experimentally measured and analyzed. Due to space limitation, only experimental results of tree maintenance overhead and service disruption time are included here.

3.1 Tree Maintenance Overhead

Figure 6 shows the tree maintenance overhead of the VRBMoM comparing with remote subscription and RBMoM. The tree maintenance overhead of the VRBMoM is lower than other two approaches as shown in Figures 6 and 7. Because the remote subscription reconfigures the multicast tree whenever mobile hosts move, the tree maintenance overhead is the most high in the remote subscription case. RBMoM has the result that tree maintenance overhead is less than the remote subscription. However, RBMoM uses fixed service range without considering mobility of mobile hosts, thus tree maintenance overhead of VRBMoM which considers mobility is less than RBMoM. Figure 7 shows the tree maintenance overhead in 20×20 network. Figure 7 shows that the tree maintenance overhead of all approaches are similar as 10×10 network. This means that tree maintenance overhead is not affected even if the network size grows in VRBMoM. The reason of such result is that tree reconfiguration is affected mostly by mobility, not by network size.

3.2 Service Disruption

Figures 8 and 9 show the service disruption time of VRBMoM comparing with the remote subscription and RBMoM. The 2.4Mbps traffic represents the multi-

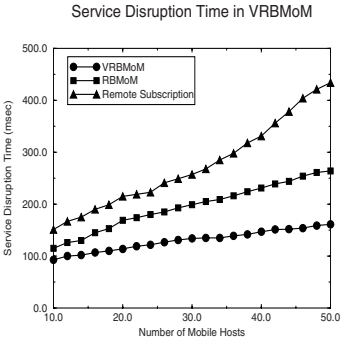


Fig. 8. Service Disruption Time of 2.4Mbps Traffic

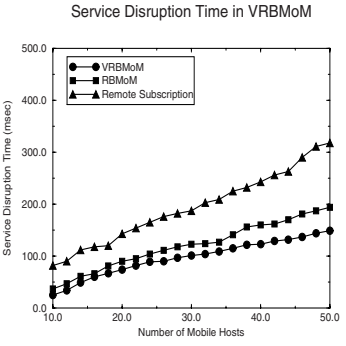


Fig. 9. Service Disruption Time of 500Kbps Traffic

media service data of video or audio, and the 500Kbps traffic represents general binary data or text data. The service disruption happens when the mobile host moves from the current service range to other new service range. If we use resource reservation technique, the service disruption time will be decreased. As shown in Figures 8 and 9, there are service disruptions in all approaches. However, because VRBMoM uses resource reservation technique, the service disruption time is less than the remote subscription and RBMoM. If we do not use resource reservation technique in VRBMoM, service disruption time will be more increased. Because mobile hosts do not receive the multicast packets until the data delivery path is established to the location to visit. Figure 9 shows the service disruption time of the 500Kbps traffic. It shows that the differences of each scheme's performance are reduced. However, we note that VRBMoM has the best performance similar as Figure 8. This is because the packets of 500Kbps traffic has shorter arriving interval than 2.4Mbps traffic. When the arriving interval of packets is shorter, the service disruption time is more reduced.

4 Conclusion

In this paper, we propose VRBMoM which adapts the service range variably according to mobility types of mobile hosts in the group and uses resource reservation to reduce service disruption in mobile computing environments. Because VRBMoM establishes variable service range adapting to mobile host mobilities, the performance is improved by reducing tree reconfiguration overhead than RBMoM. Moreover, by using resource reservation technique, it reduces service disruption time when the mobile host moves out of the service range.

VRBMoM in this paper adapts more effectively in real mobile computing environments such that mobile hosts have different speeds and move with different mobility-types. According to how RBMoM decides the service range, RBMoM shows different performance of tree reconfiguration overhead and service disrup-

tion time. However, VRBMoM adaptably controls the service range according to the host's mobility and the number of mobile hosts in the range, and uses resource reservation when the mobile host moves from the current service range to a new foreign network. Thus, VRBMoM has the advantage that reduces the tree maintenance overhead and the service disruption time than RBMoM.

In this paper, we have not considered yet the sender's mobility. The efficient multicast which also considers the sender's mobility remains to be solved. In addition, the multicast flow control is required to improve the QoS of mobile hosts.

Acknowledgement

This work was supported by HY-SDR Research Center at Hanyang University, Seoul, Korea, under the ITRC Program of IITA, Korea.

References

[1] A. Ballardie, P. Francis and J. Crowcroft, "Core Based Trees: An architecture for scalable inter-domain multicast routing", in Proceedings of ACM SIGCOMM, pp. 85-95, 1993

[2] R. Wittmann and M. Zitterbart, "Multicast Communication: Protocols and Applications", Morgan Kaufmann Publishers, 2001

[3] C. Perkins, IP mobility support, RFC2002, October 1996. 221, 228

[4] T. G. Harrison, C. L. Williamson "Mobile Multicast (MoM) Protocol: Multicast Support For Mobile Hosts", in Proceeding of ACM MOBICOM, pp. 151-160,1997 222

[5] C. R. Lin, K.-M. Wang, "Mobile Multicast Support in IP Network", in Proceeding of INFOCOM, Vol 3, pp. 1664-1672,2000 222, 228

[6] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", IEEE Network, Vol 7, pp. 8-18, September 1993. 222

[7] A. K. Talukdar, B. R. Badrinath and A. Acharya, "MRSVP: A Reservation Protocol for an Integrated Services Packet Network with Mobile Hosts", Tech. Report TR-337, Rutgers university, 1998 222, 226

[8] S. Uskela, "Mobility management in mobile internet", in Proceeding of Third International Conference on 3G Mobile Communication Technologies, pp. 91-95, 2002 223

[9] M.-H. Chiu, M. Bassiouni, "Predictive channel reservation for mobile cellular networks based on GPS measurement", in Proceeding of IEEE International Conference on Personal Wireless Communication, pp. 441-445, 1999 223, 226

[10] http://www.isi.edu/nsnam/ns/ 227

SIP Signaling Performance Evaluation for Supporting Mobility in Cellular-IP Integrated Wireless Networks

Hyun Soo Kim, Chang Ho Kim, Byeong-hee Roh, and S. W. Yoo

Graduate School of Information and Communication, Ajou University
San 5 Wonchon-dong, Paldal-Gu, Suwon,442-749, Korea
{bhroh,swyoo}@ajou.ac.kr

Abstract. In this paper, we evaluate SIP signaling performances to support macro-mobility in Cellular-IP integrated wireless networks. In order to show the efficiency of the SIP-based scheme, we analyze the handoff delay for connection re-establishment when terminals move from one administrative domain to another. The numerical results show that the SIP-based scheme can reduce the handoff delay significantly compared to MIP-based method. We also show that the SIP-based scheme can achieve much improved loss and throughput performances in delivering real-time multimedia traffic by using simulations.

1 Introduction

Recently, to provide real-time multimedia services, such as voice over IP (VoIP) that have been originally devised for the wired Internet environments, even in wireless networks is being one of the most active research areas. The mobility support is one of the most important functions to provide seamless data transfer for real-time multimedia services in wireless networks.

In order to support the mobility in wireless networks, Mobile IP (MIP) has been proposed[1]. However, it has the triangle routing problem that all packets sent to the mobile host(MH) should be delivered through its home agent(HA), causing increased load on the home network(HN) and high latency. Though routing optimization solutions[2] have been proposed in order to overcome the triangle routing problem, they would require the update of every host in the Internet. In addition, packet header overhead in MIP encapsulation is particularly significant for low bit rate VoIP services, and the use of DHCP for assigning care of address(CoA) to each MH causes additional latency in handoffs, which may not be appropriate for real-time multimedia applications.

As an alternative way, the Session Initiation Protocol(SIP)[3]-based mobility support schemes have been proposed. Wedlund and Schulzrinne showed that SIP can provide terminal, personal, session and service mobilities[4]. Unlike MIP, SIP does not require the triangle routing and the overhead due to IP encapsulation to support the mobilities. Kwon et al.[5] analytically computed and compared the handoff delays between MIP and SIP-based approaches. In their result, the

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 231–240, 2004.

handoff delay of the MIP approach is smaller than that of the SIP approach in most situations. We think that the results are mainly due to the IP address assignment from DHCP in the case of SIP approach[6]. For reducing the complexity when mobility occurs within an administrative domain, Cellular IP(CIP) interworking with MIP has been proposed[7][8]. We call the scheme MIP/CIP hereafter. Gatzounas et al.[9] proposed an mobility management architecture with the integration of SIP and CIP. However, they did not provide detailed performances of the architecture.

In this paper, we propose a SIP signaling architecture to support macro-mobility in Cellular-IP integrated wireless networks. We call the scheme SIP/CIP hereafter. Because CIP is designed for intra-domain mobility, SIP plays a key role in inter-domain mobility. The SIP/CIP can reduce the handoff delay in inter-domain mobility, so it can improve the loss and throughput performances as well. In order to show the efficiencies of the SIP/CIP, we make an analytical model for the handoff delay, then compare the delay time performances with those of MIP/CIP. The results show that the SIP/CIP outperforms the MIP/CIP unlike Kwon et al.'s results[5], in which CIP was not considered. Also, we show the effectiveness of the SIP/CIP in delivering real-time multimedia packets using simulation.

The rest of the paper is organized as follows. In Section 2, MIP-based mobility architectures will be described. In Section 3, we explain our proposed SIP/CIP architecture, then give an analytical model to compute the handoff delay. Section 4 gives experimental results, and finally, we conclude the paper in Section 5.

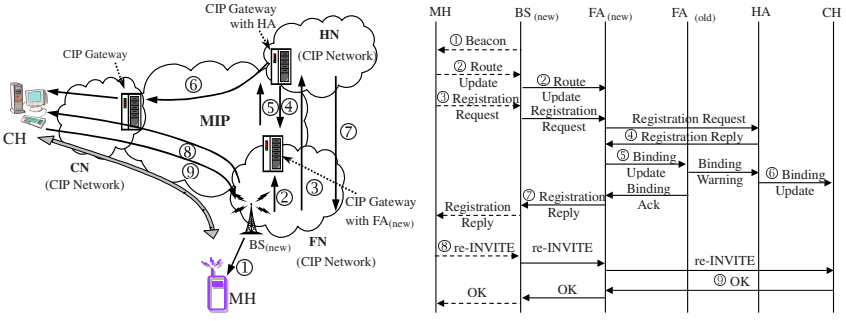
2 Related Mobility Architectures

2.1 Mobile IP with Route Optimization (MIP-RO)

MIP[1] allows a MH to move between IP subnets, while keeping communications with its corresponding host(CH). In MIP, home agent(HA) and foreign agent(FA) located in home network(HN) and foreign network(FN), respectively, play a key role in supporting the mobility. When a MH moves to a FN, it gets a temporary IP address called a CoA from a DHCP server, or uses the FA's IP address as a care of address. Then, the MH informs the HA of its CoA. After the MH's location information is registered in the HA, all packets from the CH can be routed to the MH through the FA. This is called the triangle routing, which increases load on the home network(HN) and high latency. Route optimization solutions allow packets to be routed from CH to MH directly without going to HA first[2]. For the route optimization, all hosts are required to maintain a binding cache containing CoAs of MHs. The binding cache is used for tunneling packets to MHs.

2.2 Cellular-IP Interworking with Mobile-IP (MIP/CIP)

In MIP, whenever a CoA of MH is changed, the MH must update its new CoA in the HA. If the MH moves very fast, the repeated registrations cause the network



(a) Network Architecture (b) Message Flow

Fig. 1. MIP/CIP Architecture

overheads and processing delays in handoff. Cellular IP(CIP)[7], as a micro-mobility support protocol, has been proposed to solve the above MIP's problem. CIP supports micro-mobility at layer-2. In CIP, when handoff occurs in an administrative domain, MHs are not required to register their CoA in the HA. Instead, using routing and paging caches that each CIP node maintains, the decision to reroute packets are performed. Between CIP domains, normal MIP procedures are used for macro-mobility. Any MH in CIP networks does not need to use DHCP to obtain a temporary IP address. So the integration of MIP and CIP can reduce the delay overhead for the frequent registrations.

In Fig.1(a), an architecture for mobility support in CIP networks interworking with MIP (MIP/CIP) is illustrated. CIP Gateway has similar functions as HA and FA. Fig.1(b) depicts the message flow of MIP/CIP as shown in Fig.1(a), to determine the routing path at handoff between CIP domains. ①When a MH enters a new FN, the MH receives a beacon signal from the BS in the new FN. ②The MH sends a Route Update message to the CIP gateway through the BS, along the path delivering the Route Update message a new routing path is established. ③After the routing path is determined, the MH sends a Registration Request message to the HA via FA. ④The new FA receives the Registration Reply message. ⑤When the new FA receives the Registration Reply message, the FA sends a Binding Update message to old FA and replies an ACK message to inform an entity that needs to know the new CoA and MH's home address of new binding and sends Binding Warning message to HA. ⑥HA sends the Binding Update message to CH for route optimization. ⑦The new FA received Registration Reply message sends it to the MH. ⑧After the registration is successfully performed, the MH sends a re-INVITE message to the CH to setup call again. ⑨The CH replies OK message to the MH. In case of intra-mobility, the MH sends a Route Update packet to the FA and sends an re-INVITE message to the CH without registration process such as ③, ④, ⑤, ⑥, and ⑦ because the MH's CoA is not changed though the routing path is changed.

3 Mobility Support Architecture of SIP/CIP

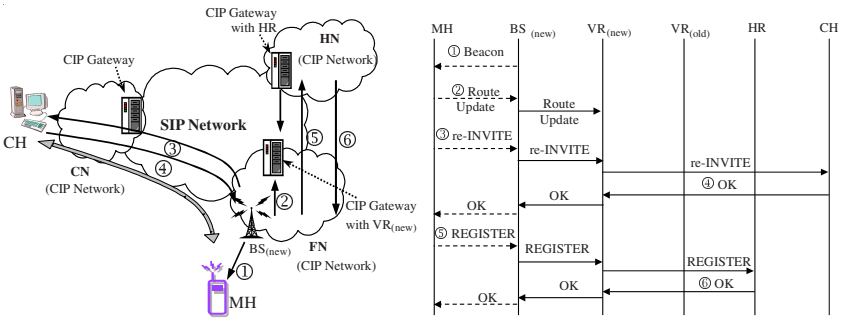
3.1 Signaling Architecture (SIP/CIP)

Session Initiation Protocol(SIP) is an application layer protocol for signaling and controlling a session consisting of multiple streams[3]. SIP basically supports not only personal mobility but also the terminal mobility using the redirect server. These ability of SIP can be used for finding and updating the location of MH.

Fig.2(a) shows the network architecture of the SIP/CIP. The location information of MH is registered in SIP redirect server instead of a HA as in MIP, and a SIP proxy server provides functions as similar as FA in MIP. SIP proxy server functions such as Home Redirect(HR) and Visited Redirect(VR) servers are implemented within the CIP gateway. When a CH sends INVITE message to a MH moved to a different FN, the CH receives "SIP 302 move temporarily" message with the current location information of the MH from SIP redirect server. Then, the CH sends INVITE message to the MH through SIP proxy server in the current domain, and gets an OK response message. After the reception of the message, packets are sent to the MH directly without triangular routing as in MIP. If the MH moved to a different administrative domain, the MH sends a REGISTER message to the SIP redirect server in HN.

Since CIP supports IP paging, the MH does not need to get an IP address from DHCP, and can use a SIP-based identifier, e.g. an email-like address of the form "user@host", in CIP networks. The information is included in the payload part of the Route/Paging Update message. In SIP INVITE messages, there exists a Contact field containing the current location information for supporting handoff. By referring the Contact field, the CH can send packets to the MH directly. SIP redirect server implemented in the CIP gateway in HN maintains the information binding SIP URL with CIP gateway address, and provides the SIP proxy server with the infor-mation for the next calls to the MH.

In Fig.2(b), we show a message flow when macro-mobility occurs in SIP/CIP architecture shown in Fig.2(a). ①When a MH enters a FN with a different administrative policy, the MH receives a beacon message from the new BS in the FN, and the MH knows that the its domain is changed. ②MH sends Route Update message to SIP VR server, which is implemented within the CIP gateway in the FN, through the new BS. ③Then, in order to update session with its CH, the MH sends a re-INVITE message to the CH via the proxy server. ④The CH replies OK message to the MH via the proxy server, and the handoff is completed. ⑤The MH sends REGISTER message to the redirect server in HN for next calls. ⑥The MH receives OK message as a response. It is noted that in MIP/CIP, the registration process must be completed before the MH sends a re-INVITE message to the CH to eliminate the triangle routing problem. On the contrary, in SIP/CIP, since the location information of the MH is included in the Contact field in SIP re-INVITE message, the handoff can be completed by sending the message. The registration process after this is only for next calls, which are regardless of the current session. From the facts, we can intuitively perceive that the routing update by SIP/CIP is performed much faster than that by MIP/CIP.



(a) Network Architecture (b) Message Flow

Fig. 2. SIP/CIP Architecture

3.2 Delay Analysis

For the analysis, we make a simple network model as shown in Fig.3, which is as similar as in Kwon et al.s'[5]. While Kwon et al. assumed that all domain and core networks use MIP or SIP, we adopt an hybrid use of CIP for micro-mobility within each domain and SIP for macro-mobility between domains. In order to show the delay performance, we will compare our SIP/CIP with MIP/CIP. This is because we would like to focus on the performances between SIP and MIP in the inter-domain handoffs. So, detailed message flow of CIP are not considered.

Let t_s be the delay for transferring a message through the wireless link between MH and BS. The delay corresponds to the time to deliver a beacon message through the wireless link. Let t_f and t_h be the delay for delivering a message between MH and FN and the delay be-tween MH and HN, respectively. Let t_{hc} be the delay between CH and HN, and t_{fc} be the delay be-tween CH and FN. Let t_{up} be the time required for sending message between FAs. We assume that the CH is located in the CN that is the home network of the CH. Let $T_{sip-inter}$ and $T_{mip-inter}$ be handoff delays for SIP/CIP and MIP/CIP, respectively, when handoff occurs between different administrative domains. First, we can obtain $T_{mip-inter}$ from Fig.1. It takes t_s for MH to receive a beacon message, t_f for MH to send a Route Update to the new FA according to CIP mechanism, t_h for MH to send a Registration Request to HA, and t_h - t_f for HA to send Registration Reply to the FA_{new}. It takes $2t_{up}$ for FA_{new} to send Binding Update message to FA_{old} and to receive ACK message from FA_{old}. It takes t_h - t_f and t_{hc} requiring for that the FA_{old} sends Binding Warning to the HA, and that the HA sends Binding update message to the CH, respectively. Then, it takes t_f in which the HA sends Registration Reply to the FA_{new}. At this time handoff procedure is completed. Then, it additionally takes $t_f + t_{fc}$ and $t_f + t_{fc}$ for MH to send re-INVITE message to the CH via the FA and for CH to reply OK message to MH via the FA for call setup, respectively. To sum up the above delay times, we can easily obtain $T_{mip-inter}$ given by

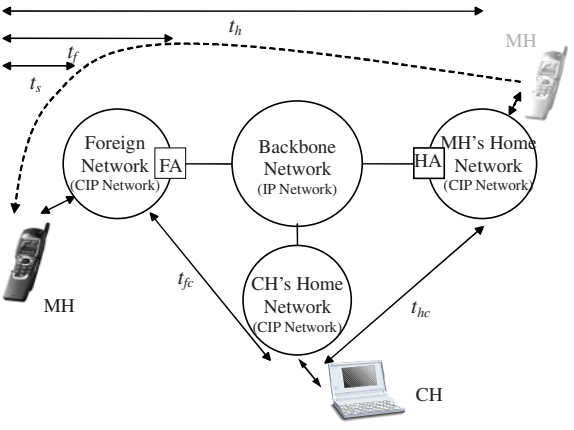


Fig. 3. A simple network model for handoff delay analysis

$$T_{mip-inter} = t_s + 3t_h + 2t_f + 2t_{up} + t_{hc} + 2t_{fc} \tag{1}$$

Similarly, we can compute the handoff delay in SIP/CIP, $T_{sip-inter}$, according to the message flow shown in Fig.2. That is, it takes t_s for receiving a beacon message, t_f for sending a Route Update according to CIP mechanism, and t_f + t_{fc} for delivering re-INVITE message from MH to CH via VR, and t_f + t_{fc} for replying OK message from CH to MH via $\text{VR}_{(new)}$. By doing so, the actual handoff procedure is completed. For next calls, it takes $2t_f$ for exchanging REGISTER and OK messages between MH and HR. Then, we have

$$T_{sip-inter} = t_s + 2t_h + 3t_f + 2t_{fc}. \tag{2}$$

For intra-domain handoffs, since a registration with HA or HR is not needed, the handoff delays for SIP/CIP and MIP/CIP, $T_{sip-intra}$ and $T_{mip-intra}$, respectively, can be obtained as follows:

$$T_{mip-intra} = t_s + 3t_f + 2t_{fc}. \tag{3}$$

$$T_{sip-intra} = t_s + 3t_f + 2t_{fc}. \tag{4}$$

4 Experimental Results

4.1 Signaling Delay Performance

For the experiments, we used the values given in [5] for the delay parameters. That is, 10ms for t_s, 12ms for t_f, and 17ms for t_h. And, we assumed that the delay within each network is constant at 5ms, and that MH is located in FN.

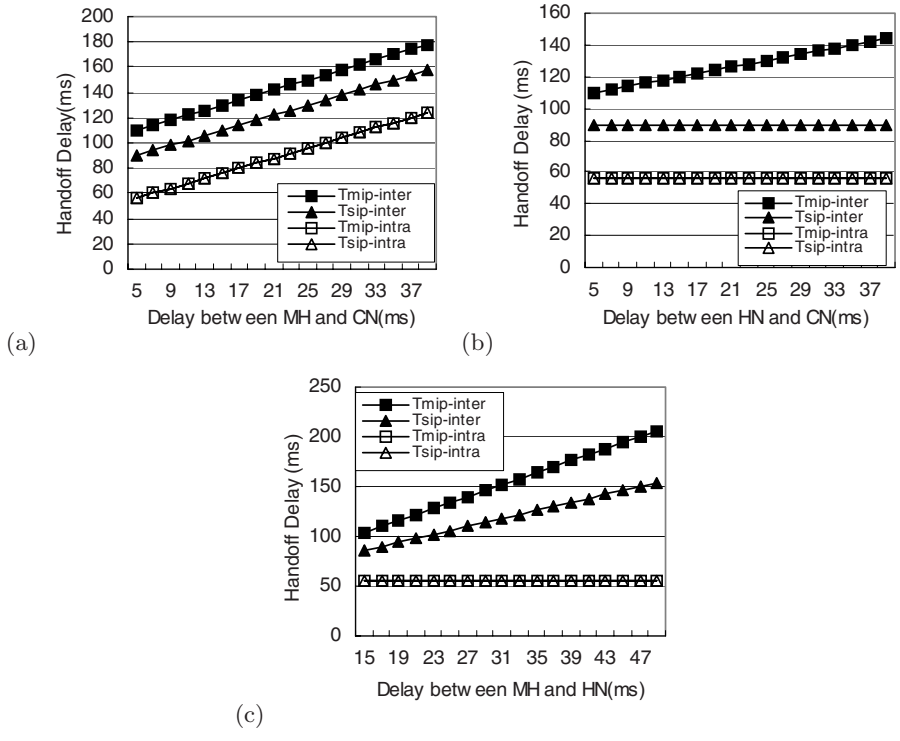


Fig. 4. Handoff delay performances varying the delays (a) between MH and CN, (b) between HN and CN, and (c) between MH and HN

In Fig.4(a), we show the handoff delay as the delay between MH and CN, t_{fc}, increases. It is noted that t_{fc} is the link delay between MH and CN, so the increase of t_{fc} means the increase of the distance or the decrease of bandwidth between them. As we can see from Fig. 6, the handoff delays increase as the delay between MH and CN increase, but in the case of macro-mobility handoff, the delays of SIP/CIP show much lower than those of MIP/CIP. While, the delays in the case of intra-domain handoffs show same values. This is because the intra-domain handoffs are treated by CIP in both approaches. Fig.4(b) shows the handoff delays when the delay between HN and CN, t_{hc}, varies. The increase of the delay between HN and CN means that the transfer delay through the IP networks or the link delays of HN and CN increase. While the handoff delay of MIP/CIP increases as the delay between HN and CN increases, that of SIP/CIP keeps at a constant level. This is because that the SIP handoff delay is independent of t_{hc} as shown in (2), while the MIP is not.

In Fig.4(c), the handoff delay varying the delay between MH an HN, t_h. The increase of the delay between MH and HN means that the transfer delay through the IP networks or the link delays of HN and FN increase. It can be also

shown that the SIP/CIP outperforms the MIP/CIP in the handoff delay when inter-domain handoff occurs.

As we can see from Fig.4, SIP shows better handoff delay performances than MIP when inter-domain handoff occurs. This comes from the following two reasons. First, MIP is more dependent on t_h than SIP as shown in (1) and (2). Second, MIP is much affected by t_{hc} while SIP is not. The two reasons mean that for handling inter-domain handoffs, the number of messages through HA or destined to HA is larger in the case of MIP than that of SIP. It is noted that the overheads for the tunneling and the binding information update to all CH in MIP are not considered. If these overhead should be considered, it is expected that the handoff delay performance of SIP/CIP can be much more improved than that of MIP/CIP.

4.2 Real-Time Data Delivery Performance

Here, we describe how the handoff delay performances evaluated in the previous subsection can affect the actual delivery of real-time data. For the experiment, we carried out the following simulation using ns-2[10]. Fig.5 shows the network model for the simulation as similar as in [11]. In the network model, there are three separate wireless networks such as HN, FN1 and FN2 with different administrative policy each other, and those wireless networks are interconnected through wired IP network. HN is the home network for MH. It is assumed that the link delays between networks are same as 5 ms and their bandwidths are 5Mbps. We also assumed that the radius of each wireless network is 75m, and there is no overlap between the wireless networks. We let CH send packets at constant rate of 500 Kbytes/sec. And, we let MH start from HN, then move through FN1 to FN2, so two handoffs occur.

Fig.6 plots each packet's delay time from CH to MH when MH's moving speed is 6m/sec. Handoffs from HN to FA1 and from FA1 to FA2 occurred at about 25 and 50 second, respectively. Fig.6 (a) corresponds to MIP/CIP case, and Fig.6 (b) to SIP/CIP case. As we can see, handoff delays of SIP/CIP are much

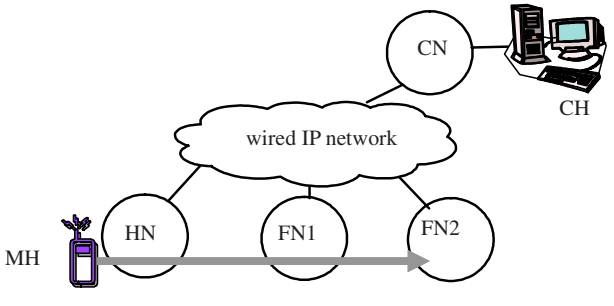
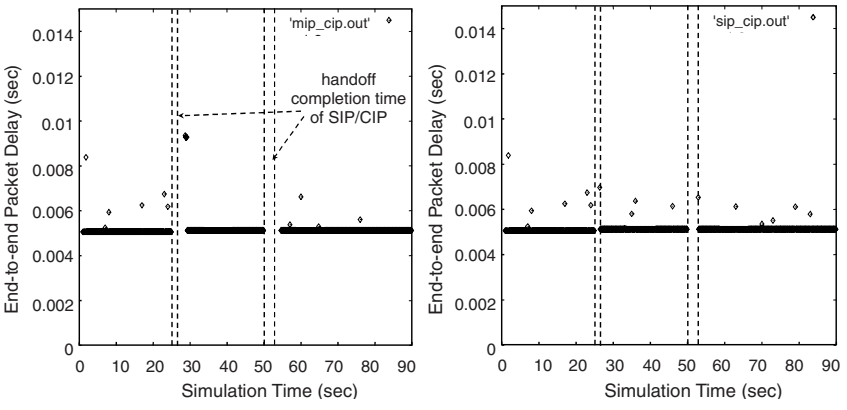


Fig. 5. Networking model of the simulation for evaluating real-time data delivery performance



(a) MIP/CIP (b) SIP/CIP

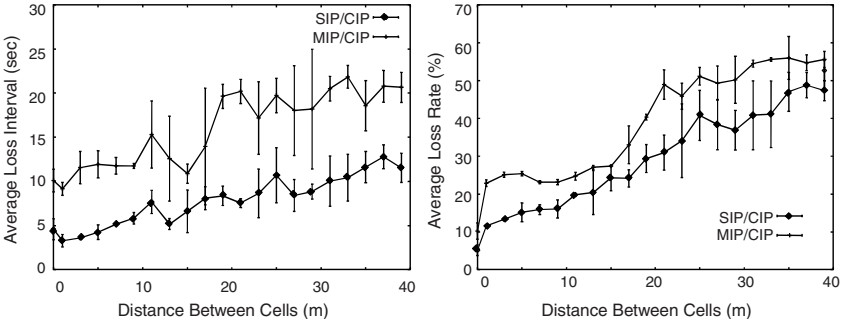
Fig. 6. UDP Traffic Delay Characteristics (Moving speed of MH=6m/sec)

smaller than those of MIP/CIP. We have shown the similar analytical results in the previous subsection. In addition, we can see the severe delays of packets just after handoffs are completed in MIP/CIP case, while the phenomenon is not appeared in SIP/CIP. Fig.7(a) shows the average packet loss ratio varying the distance between wireless network cells. The average packet loss ratio is defined as the number of lost packets over the number of total packets sent. It is noted that the environment of handoff is getting worse as the distance increases. As the distance increases, the average packet ratio also increases. However, the average packet loss ratio of SIP/CIP approach is much lower than that of MIP/CIP.

In order to illustrate the reason of results shown in the Fig.7(a), we show another results of Fig.7(b), in which the average packet loss duration varying the distance between wireless network cells is depicted. We defined the average packet loss duration as the time interval between the time of first packet loss and the time of the first packet receipt after handoff starts. We can see that the average packet loss durations of SIP/CIP are much smaller than those of MIP/CIP.

5 Conclusion

In this paper, we evaluated SIP and MIP-based mobility support architectures interworked with CIP. First, we described the detailed message flows of the architectures in handoff, and then made an analytical model to compute the handoff delay. Numerical results showed that the SIP/CIP outperformed the MIP/CIP in the viewpoints of the handoff delay. As a result of the shortened handoff delay, the SIP-based approach can fit to the provision of real-time multimedia services. We showed the effect by using simulation. It is known that CIP can be applied to support the micro-mobility well. SIP can support variety of mobilities such as personal, terminal, service, and so on. With our results, SIP can play



(a) Average packet loss rate (b) Average handoff delay

Fig. 7. Performances varying the distance between cells

an important role in the macro-mobility compared with MIP. So, we think that the SIP/CIP architecture can be a good solution to support both micro- and macro-mobilities.

References

[1] Perkins, C.: IP Mobility Support, IETF RFC 2002, (1996) 231, 232
[2] Perkins, .E, Johnson, D.: Route optimization in Mobile IP, IETF Draft,<draft-ietf-mobileip-optim-10.txt> (2000) 231, 232
[3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol, IETF RFC 3261 (2002) 231, 234
[4] Schulzrinne, H., Wedlund, E.: Application-layer mobility using SIP, ACM Mobile Computing and Commun. Rev., **4** (2000) 231
[5] Kwon, T., Gerla, M., Das, S., Das, S.: Mobility Management for VoIP Service: MIP vs. SIP, IEEE Wireless Communications, **9** (2002) 231, 232, 235, 236
[6] Schulzrinne, H.: DHCP Option for SIP Servers, IETF draft <draft-ietf-sip-dhcp-05.txt> (2001) 232
[7] Campbell, A., et al.: Cellular IP, IETF draft <draft-ietf-mobileip-cellularip-00.txt> (2000) 232, 233
[8] Carli, M., Neri, A., Picci, A.: Mobile IP and Cellular IP Integration for inter Access Network Handoff, IEEE ICC'2001 (2001) 232
[9] Gatzounas, D., Theofilatos, D., Dagiuklas, T.: Transparent Internet Mobility using SIP/Cellular IP Inte-gration, IPCN 2002, Paris France (2002) 232
[10] Network Simulator - ns (version 2), http://www.isi.edu/nsnam/ns 238
[11] Chen, H.: Simulation of Route Optimization in Mobile IP, WLN'2002 (2002) 238

Performance of Voice Traffic over Mobile Ad Hoc Network

Jisoo Kim[1], Daein Choi[1], Jungjin Park[1], Youn-Kwan Kim[2], I. Chong[3], and Hyun-Kook Kahng[1]

[1] Department of Electronics Information Engineering, Korea University
♯208 Suchang-dong Chochiwon Chungnam, Korea 339-700
{jissung,nbear,pjj,kahng}@korea.ac.kr
[2] LG Telecom
Gangnam Tower, 679, Yeoksam-dong, Gangnam-gu, Seoul, Korea 135-080
ykkim@lgtel.co.kr
[3] Information and Communications Engineering dept., Hankuk University of FS
♯207 Imun-dong, Dongdaemun-Gu, Seoul, 130-790, Korea
iychong@hufs.ac.kr

Abstract. As popularity of VoIP services in wireless network is increasing in recent, the use of VoIP services in MANET (Mobile Ad-hoc Network) is expected to grow as well. In this paper, we consider some parameters associated with QoS(Quality of Service) in MANET, and then analyze two main factors which produce severe performance degradation while each node communicates to each other in MANET environment: packet loss and disruption time. Experimental results show that packet is constantly transmitted except the time which needs for route discovery.

1 Introduction

As a cellular participant and Internet user is growing rapidly, VoIP(Voice over IP) services in wireless network becomes a matter of concern among people these days. Several issues about voice traffic transmission in wireless environment using SIP(Session Initiation Protocol) is on going. Among the growing use of VoIP in wireless network, the use of Mobile Ad-hoc Network(MANET) is expected to grow rapidly as well. [1] Several routing algorithms in MANET are presented (e.g. AODV(Ad Hoc On Demand Distance Vector)[2], OLSR(Optimized Link State Routing Protocol)[7], DSR(Dynamic Source Routing)[8], etc.). We especially focus on AODV algorithm proposed as a protocol which can perform wireless routing in MANET.

However, voice communication in MANET also produces same problem in that of wireless network. That is, problems like delay and packet loss, which is associated with QoS mechanism, is appearing in AODV algorithm.

[1] This work was supported by grant No.(R01-2002-000-00489-0) from the Basic Research Program of the Korea Science and Engineering Foundation.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 241–248, 2004.

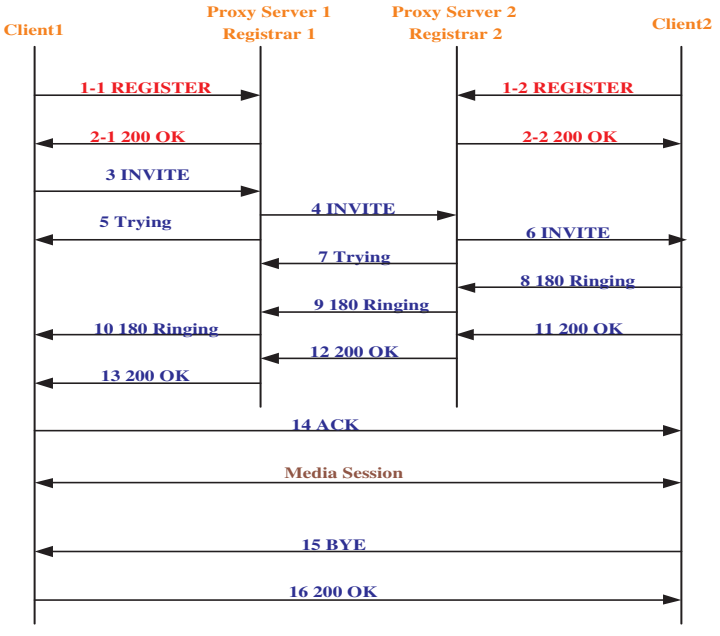


Fig. 1. SIP session setup, processing, termination

In this paper, we address this issue and experiment on voice traffic in MANET using AODV in real environment. W e also consider most significant issue in real-time traffic: packet loss and disruption time. The kernel-AODV for setting up MANET environment, sip-communicator for internet phone, and JAIN-SIP(Java Integrated Network-SIP) Proxy for server developed by NIST (National Institute of Standards and Technology) are used in our experiment.[5][6].

The rest of the paper is organized as follows, Section 2 gives a related work about SIP and AODV routing algorithm. In Section 3, Consideration about QoS of Voice Traffic in mobile ad-hoc environment is presented. Section 4 describes the packet loss and disruption time when mobile node is moving, and section 5 concludes this paper.

2 Related Works

2.1 SIP

SIP (Session Initiation Protocol) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants.[1] It consists of UA (i.e. UAC(User Agent Client), UAS(User Agent Server))and Server (i.e., proxy, redirect, and registrar). SIP uses several methods (e.g. INVITE, OK, ACK) to negotiate media type with which can coexist each other. Figure1 shows procedure of session initiation and disconnection.

Each client sends REGISTER message to Registrar in their home network for registering their location. Registrar conducts their database through location

service, and then sends OK message back which saying registration is completed. They use several methods (i.e. INVITE, OK, ACK) for session establishment. Voice applied the negotiated media type above start to transmit using RTP (Real-Time Transport Protocol). Finally, client2 sends BYE message to client1 and client1 sends OK message back to client2 to terminate the session.

2.2 AODV (Ad Hoc on Demand Distance Vector) Routing Algorithm. [2]

MANET is a collection of nodes forming a temporary network without the aid of any centralized administration, where each node communicates over wireless channels, moves freely and may join and leave the network at any time.[3] In MANET environment, the Ad hoc On Demand Distance Vector (AODV) routing algorithm is used to provide both unicast and multicast routing.

AODV is improved DSDV(Destination-Sequenced Sequenced Distance Vector) algorithm. It starts route discovery process when source node doesn't already have a valid route to a destination. It first broadcasts a RREQ(Route Request) to its neighbors. Neighbors forward the request to their neighbors, and so on until either the destination or an intermediate node with a "fresh enough" route to the destination is located. When destination is determined, source node sends data packets on route to destination based on RREP(Route Response) information.

3 Consideration

Since multimedia service like voice traffic typically require a more continuous supply of end-to-end resources, compared with non-multimedia services, QoS(Quality of Service) should be considered more.

Consideration about QoS in wireless environment has become a major part by the fact that wireless networking is unreliable and various forms of interference result in changing bandwidth availability and low effect bandwidth due to high error rates. Also, the fact that MANET is an autonomous system connected by wireless link makes that MANET should consider several metric associated with QoS.

In this section, we look around several parameters related to QoS, and focus on problems resulted by packet loss in handoff. We define "Handoff" is the function transmitting a call when node moves from one zone to another. Also, we consider several solutions for preventing disruption time generated during route setup time.

To consider QoS

- Available bandwidth : Real-time traffic such as voice and video needs high bandwidth connection
- Loss constraints: Packet loss caused by disconnection during handoff
- End-to-end delay: Transmission delay, node's latency, disruption time

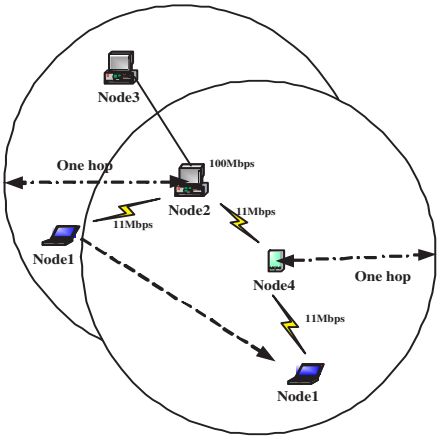


Fig. 2. Experimental architecture

Computational computing of route selection considering all of above metric must be taken into account. To analyze voice quality between mobile nodes in MANET, we focus on route discovery process during handoff.

As shown in Fig.2, we added one more active node (i.e. Node4) to make a route when node is apart from destination node. To make a new route, AODV first sends RREQ message to all neighbor nodes. In the mean time, voice traffic which already has been sent between node1 and node2 also keep sending. As a result, RREQ(and RREP) message for route setup exists with voice traffic in same link at the same time. To prevent performance degradation resulted by duplicated link use, we need to reduce the disruption time measured by the time taken until node1 get the first packet during handoff. That is, QoS mechanism providing priority service when signaling is accomplished by SIP is needed. By doing this, voice can be transmitted without delay so that efficiency will be improved.

4 Performance Evaluation

In this section, we present experimental environment. We assume that all nodes in our experimental environment have SIP-functionality which performs both UA (user agent) and Proxy Server.

As shown in figure3, Node1 and Node2 will generate registration process by themselves to upload their location. They use several methods (i.e. INVITE, OK, ACK) to establish session. Each node used to create session will carry SDP (session description) to agree on a set of media type. Node2 is only router for connection to the Node3.

After voice traffic transmission using RTP is started between Node1 and Node3, Node1 leave the range which covers one hop from Node3, and then go to another zone which cannot reach directly to the Node3. At this time Node1

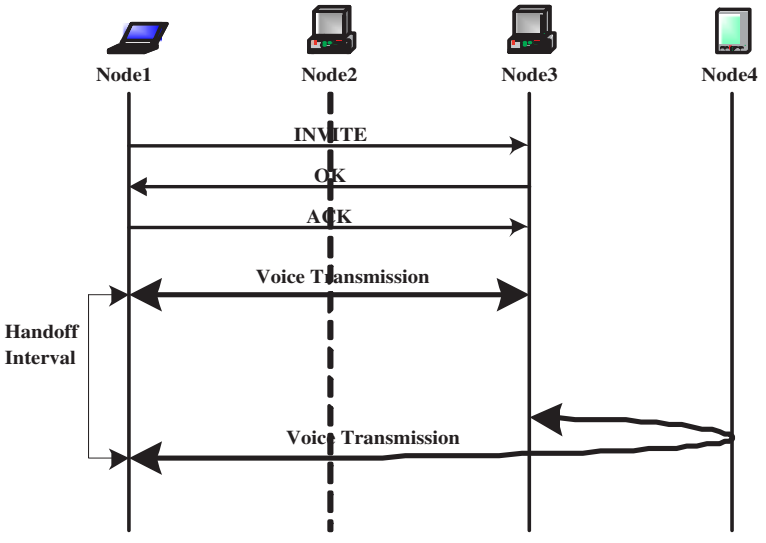


Fig. 3. Architecture of experimental network

use AODV routing algorithm to find Node3. When Node1 finds the route to the destination using RREQ and RREP, voice traffic is transmitted through Node4 according to the route information.

4.1 Analytical Model

The experimental model is based on the specification of the SIP and AODV[1][2]. We install kernel AODV to let each node use AODV routing algorithm, SIP-communicator for phone, and JAIN-SIP Proxy for server provided by NIST[5][6].

Figure 3 shows the experimental environment scenario.

Each node produces a fixed length packet, that is (520 bytes: payload of 480 bytes and a header (RTP+UDP+IP) of 40 bytes). Media sample file for Voice traffic measurement is wave file - PCM (Pulse Code Modulation) coding scheme. Node1 keep moving at 0.397 meters/sec through network domain.

Following one is Node's specification which is use for experiment.

- Node1: Pentium III 800MHz / 256M
- Node2: Pentium III 866MHz / 256M
- Node3: Pentium III 433MHz / 348M
- Node4: Intel 400MHz Processor with XScale technology, 48MB (FLASH ROM) / 64MB (SDRAM)

Wired connection (i.e. connection between Node2 and Node3) is modeled as a 100Mbps bandwidth, and bandwidth of wireless connection are set to 11Mbps. Practical length of distance covered by notebook and PDA (i.e. one hop) is 105 meter and 240 meter, respectively while length of distance covered by notebook and PDA is 160, 304.8 in specification.

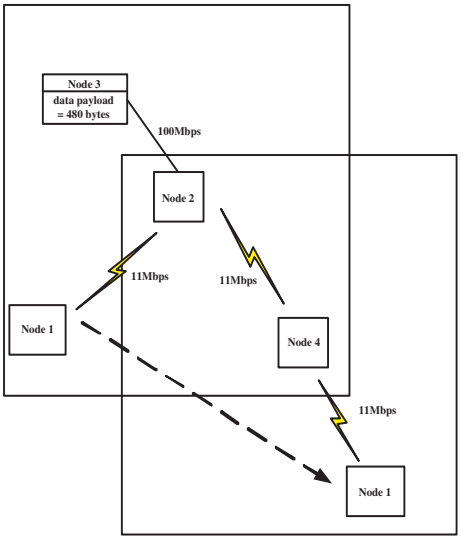


Fig. 4. The experimented network topology

4.2 Experimental Results

Figure 5(a) illustrates a sequence number of packet versus time during whole experimental period. Node1 continue receiving RTP packet uniformly with constant slope. Packet loss can be sometimes produced as shown in each time from 51 to 58, from 73 to 90, and from 182 to 188 second since node1 keep moving. However, as node1 is apart from node3, it has difficulty in communication with node3 directly (i.e. node1 moves to the place which is located in more than one hop from node3). At time between t $\approx$ 277 and t $\approx$ 327, route discovery is processed by node1, and lots of packet loss is generated. We assume that the length of distance covered by notebook and PDA is different and therefore handoff time is larger than normal time (i.e. handoff time in experiment is about 50 sec. while normal handoff time is 3 sec.).

Figure 5(b) illustrates the packet loss status more precisely when node1 is far from node3 enough to change the route. Since node1 cannot communicate with node3 directly, long constant value of sequence number (i.e. 4476, 5081) is presented for that time. At t $\approx$ 291 sec., packet receives first packet from node3 among the time of handoff resulting 16 second of disruption time. Once node1 sets the new route with node3 at t $\approx$ 328, node1 starts to transmit node3 voice traffic through node4 with the route obtained by AODV algorithm.

Node1's average receiving number of packets is about 12.6 packets. Following formula presents receiving rates of packets while node is moving.

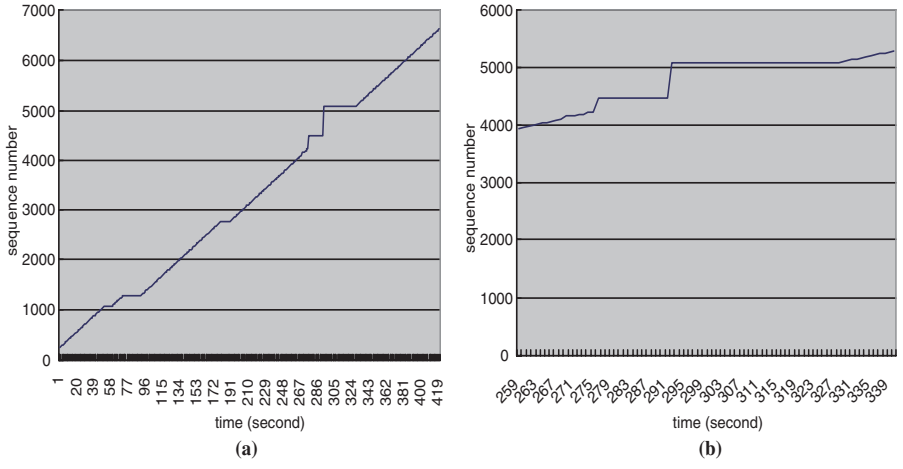


Fig. 5. (a) Sequence number of packets vs. time during whole communication (b) Sequence number of packets vs. time during handoff

Receiving rate (Kbyte / second)
$\equiv$ (received packet number $*$ payload length) $\div$ 1000

While average receiving rate of node1 is 7.8 (Kbyte/sec) during the time node1 communicates with node3 without route discovery, Average receiving rate of node1 with handoff is about 7.14 (Kbyte/sec) for whole experimental time.

Packet loss rate is 3.57%. It can be produced by following formula.

Packet loss rate (%) $\equiv$ (lost packet number) $\div$ (all packet number received from destination node) $*$ 100

5 Conclusion

Analyzing the voice traffic in Mobile Ad-hoc network is very important to provide better multimedia service. We have described some protocols (i.e. SIP and AODV) and see how they can work in experimental environment. We also provide some consideration concerning about QoS of voice traffic. We address the importance of QoS mechanism in multimedia service, and examine more precisely about the factor (such as transmission delay, packet loss, disruption time) which is very sensitive in multimedia service. As we have seen, since MANET environment doesn't consider QoS mechanism, we have mentioned about priority service which sets precedence of transmission.

In this experiment, packet loss and throughput in whole packet transmission is about 3.57

Routing algorithm of voice traffic in MANET environment is very important because of the characteristic of real-time traffic (i.e. delay sensitivity), and several consideration about QoS mechanism (such as AODV, OLSR, etc.) have been presented.

In future work, we extend routing algorithm enough to meet the requirement of QoS related to voice traffic in application layer.

References

[1] J. Rosenberg, H. Schulzrinne, E. Schooler, M. Handley, G. Camarillo, A. Johnston, J. peterson, R. Sparks, "Session Initiation Protocol", RFC 3261 in IETF, June 2002

[2] C. Perkins, E. Belding-Royer, S. das, "Ad hoc On-Demand Distance vector (AODV) routing", RFC 3561 in IETF, July 2003

[3] H. Khlifi, A. Agarwal, J. Grogoire, "a framework to use SIP in Ad-hoc networks" in IEEE CCECE, Canadian Conference on, Volume 2, 2003

[4] N. Banerjee, K. Basu, "Hand-off delay analysis in sip-based mobility management in wireless networks" Parallel and Distributed Processing symposium, 2003

[5] `http://www-x.antd.nist.gov/proj/iptel`

[6] `http://w3.antd.nist.gov/wctg/aodv_kernel/`

[7] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol", RFC 3626 in IETF, in October 2003

[8] David B. Johnson, David A. Maltz, Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", draft-ietf-manet-dsr-09.txt, April 2003.

The Two-Tiered Proxy System for Seamless Multimedia Service in Mobile Computing Environment

Jung-Rock Kim, Jang-Woon Back, Kyungshik Lim, and Dae-Wha Seo

Electrical Engineering and Computer Science, Kyungpook National University
1370 Sankyuk-Dong, Buk-gu, Dae-gu, Korea
{kjrock,kutc}@palgong.knu.ac.kr
kslim@knu.ac.kr
dwseo@ee.knu.ac.kr

Abstract. In this paper, we propose the two-tiered proxy system that offers seamless media service to mobile nodes in mobile computing environment. We also propose the enhanced prefetching strategy. Our proxy system can provide seamless multimedia streaming service and supports user mobility efficiently. We show that the initial delay of media service, handoff delay, and network traffic are reduced in our simulation.

1 Introduction

Wireless Internet services rapidly enriched from late 90's. Multimedia streaming service is more important because the performance of mobile hosts has been improved and their functions have been diversified.

Generally, a media service in wireless network is difficult because of low network bandwidth and user mobility. Many research tried to solve this problem using the proxy system in a mobile computing environment[1-2]. Because the size of media data is rather bigger than other data, it is difficult to cache media data[3]. The mobility of the mobile node can cause multiple transmissions of media data and handoff delay.

In this paper, we propose the two-tiered proxy system (TOPS) to solve problems that can be occurred at the multimedia service in the wireless environment. And we also propose the enhanced prefetching strategy to support seamless streaming service with less handoff latency and no packet loss.

The rest of this paper is deployed as follows. We explain the structure of TOPS in chapter 2 and the enhanced prefetching strategy in more detail in chapter 3 and the simulation result in chapter 4. Finally, we conclude this paper in chapter 5.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 249–256, 2004.

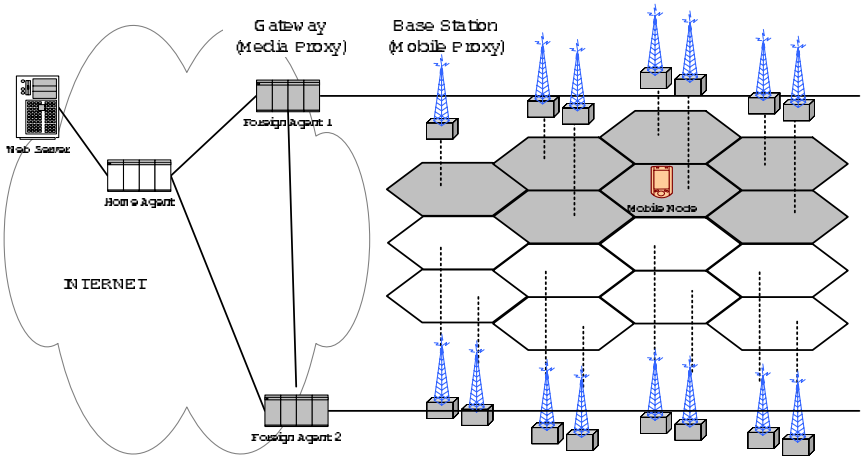


Fig. 1. Two-tiered proxy system architecture

2 Two-Tiered Proxy System

2.1 Two-Tiered Proxy System Environment

In this paper, we propose a two-tiered proxy system (TOPS) architecture which offers seamless media services to the mobile nodes. Figure 1 shows the architecture of TOPS.

TOPS is composed of two proxies. The media proxy (MeP) is responsible for media caching, transcoding, filtering about the multimedia service. MeP is located nearby a gateway. The mobile proxy (MoP) senses the bandwidth change of the wireless network and manages mobile node's profile. MoP is located in the base station. By dividing proxy architecture into two layers, we can scatter the loads of one proxy into two proxies. The traffic and the handoff delay, which are caused by the mobility of the mobile node, are reduced.

MoP traces the movement of the mobile node, monitors the wireless bandwidth, and manages the information of user preference. MoP determines the service quality that is given to the mobile node. When the mobile node requests a media service, QoS information (network bandwidth, the mobile node characteristics and user preference) is delivered to MoP. Then MoP sends a profile of the mobile node and a service request message to MeP. MeP transcodes the requested media data on the basis of requested service level, and transfers transcoded media data to MoP. Then, MoP transmits the transcoded media data to mobile nodes based on the wireless network condition.

TOPS network stack is depicted in figure 2. A home agent (HA) has an encapsulation module. A Foreign Agent (FA) has a media proxy and a decapsulation module. A base station (BS) has a mobile proxy, a prefetching module and a beacon module. A mobile host (MH) has a client agent, route decision module and beacon module.

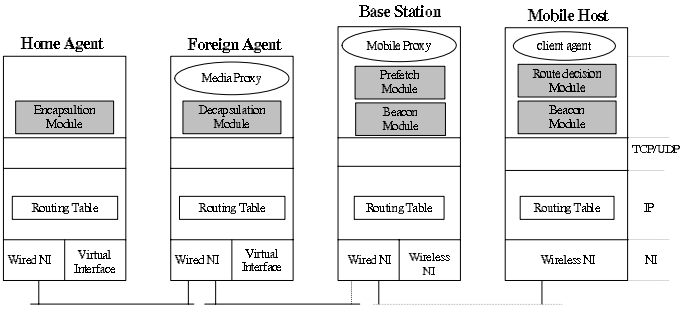


Fig. 2. Two-tiered proxy system network stack

When a MH resides in foreign network, a home agent encapsulates multimedia packets for mobile host (MH). Encapsulated packets are transmitted to the FA at which network MH is located. A decapsulation module of FA extracts multimedia data from packets that are transmitted from HA, and gives multimedia data to the MeP. MeP trascodes the media data based on QoS information and transforms the media data to the layered encoded multimedia data. MeP transmits the transcoded multimedia data to MoP in BSs associated with mobile host (MH). BSs that are associated with the MH mean the BS forwarding packets to MH or the neighborhood BS.

MoP in the forwarding BS transmits the transcoded packet to the MH, while a prefetch module in neighbor BSs store the transcoded packets. The beacon module in BS keeps track of the roaming MH. Each BS periodically broadcasts a beacon message to mobile hosts. The beacon module in MH informs the BS about the current location of MH and communication states and the route decision module decides a forwarding BS and a buffering BS based on the received signal strength from BS and communication states.

2.2 Media Proxy (MeP)

The main functions of the MeP are media filtering, transcoding, and media caching. MeP consists of a control module, a transcoding module, a caching module, a door module and database (Figure 3).

A control module manages an each module of MeP. When an user request message arrives at a control module, MeP interprets the message and determines transcoding and caching policy. According to the decided policy, the control module transmits the control message to each module. The transcoding module converts the media data to the suitable form for the environment of the mobile node. That is, the transcoding module converts the media data to other media data format which the mobile node supports and encodes the media data to the layered encoded multimedia data in order that MoP controls the multimedia data transmit rate according to network condition. Caching module determines the caching policy. The advantage of the caching is to prevent the increment

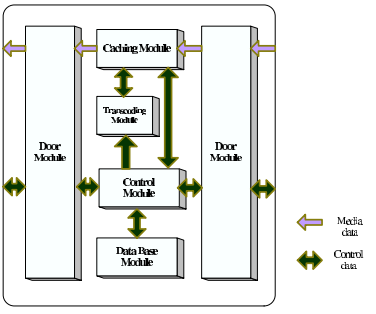


Fig. 3. Media Proxy Block diagram

of network traffic, to reduce the burden of web servers, to serves rapidly media streaming service to the mobile nodes. A door module charges the communication with outside nodes. The door module packetizes media data and sends them to outside nodes. And the door module reassembles media data from the packets coming from the outside. In database, there are mobile node's information and wireless state information, user preference.

2.3 Mobile Proxy (MoP)

The mobile proxy (MoP) is at an access network. The MoP provides QoS to users by monitoring the user preference and wireless network bandwidth. It provides users other QoS mechanisms except for media data transcoding mechanism. Then a mobile proxy delivers QoS information to a media proxy and this information is reflected immediately in the transcoding task.

Basically, the transmission module plays the role to send media data to the mobile node. Transmission module sends media data in the method that is modified from existing transmission method and is suitable to wireless network. That is, layered encoded Multimedia data that is encoded by MeP is used. Network monitoring module monitors currently usable network resources. Because the resource of the wireless network changes rapidly, it is needed to offer a changed media service according to the resource state of the wireless network through the continuous resource observation.

2.3.1 Transmission Mechanism

The bandwidth of the wireless network changes all the time. If we transmit data in the wired transmission method, the probability of data loss or error occurrence is high. To solve this problem, MoP has transmission module and transmits layered encoded media data to MHs according to wireless network state in order to maximize the delivered quality of popular streams to interested MHs. Figure 4 shows the transmission scheme between MoP and a mobile node.

To provide seamless media services to the mobile node, the transmission module transmits media data to the mobile nodes at a transmission rate which is

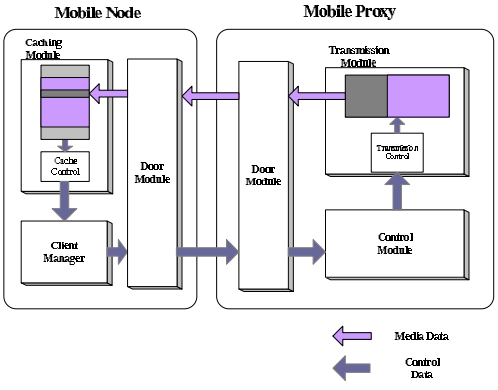


Fig. 4. Transmission scheme between MoP and mobile node

offered from MeP. MoP cooperates with a caching module of the mobile node and transmits media data at adaptive transmission rate. The transmission module of the mobile node transfers transmission error information to the MoP. Then MoP makes a transmission module retransmit the data without errors.

MoP delivers media data to the mobile nodes. If the loss and error occur in media data transmitted to the mobile nodes, the mobile nodes deliver the information of damaged data to client manager. Then client manager delivers a retransmission request message of damaged data to the MoP. Control module of MoP delivers request message to its transmission module if it takes a retransmission message. The transmission control module of transmission module analyzes the retransmission message and sends the requested media data which are buffered.

3 Prefetching Strategy in TOPS

We propose two-tiered proxy Prefetching strategy. Figure 5 depicts a TOPS Prefetching strategy diagram.

To reduce a handoff delay, a mobile node sends a proxy handoff message to MoP with registration message. MoP registers a mobile node and delivers this proxy handoff message at MeP. MeP confirms if a mobile node is registered. Namely, MeP distinguishes inside domain's movement. If a mobile node is registered, MeP continues current service after sending an ACK message. If a mobile node was not registered, MeP sends at the same time a request message to former MeP and the web server that stores the original media data which a mobile node wants to get. The message from former MeP is used to grasp the location of media data to be sent to a mobile node. The media data are sent to the mobile node through tunneling of mobile IP. If MeP itself can do a media service to a mobile node, after mobile node's handoff. MeP sends a service break message to former MeP and does a media service directly to a mobile node. MeP which

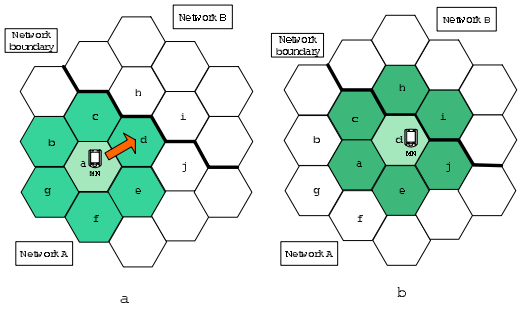


Fig. 5. TOPS Prefetching strategy

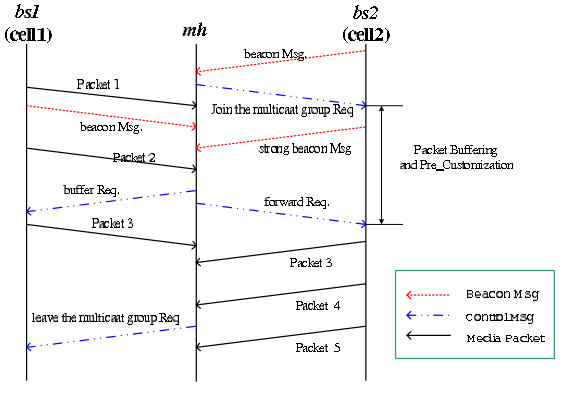


Fig. 6. TOPS handoff signal flow

takes a service break message releases registration of a mobile node and sends registration release message to MoP in its network.

Figure 6 depicts a handoff protocol between BSs and a MH when the MH moves from the cell1 to the cell2.

MH periodically receives beacon messages from BSs(BS1, BS2). As MH approaches the cell 2, MH gets stronger signals than before. If MH receives the stronger signal than the threshold of signal strength, MH names BS2 for a buffering base station. At same time, MH sends a message to join BS2 in the multicast group. Then the BS2 joins the MH's multicast group and receives packets from the home agent. BS2 decapsulates the received media packets, and transcodes media data through the transcoding proxy, and then stores the transcoded media data packets in the buffer. After than, if the BS2's signal strength is stronger than the BS1's signal strength, MH decides the forwarding BS with BS2 and requests the packet forwarding. BS2 immediately sends the transcoded packet which is stored in buffer when BS1 enters the cell2. At this time, BS1 receives the buffer request and becomes the buffering BS. If MH goes away from BS2 and receives the weaker signal than the lower threshold, MH

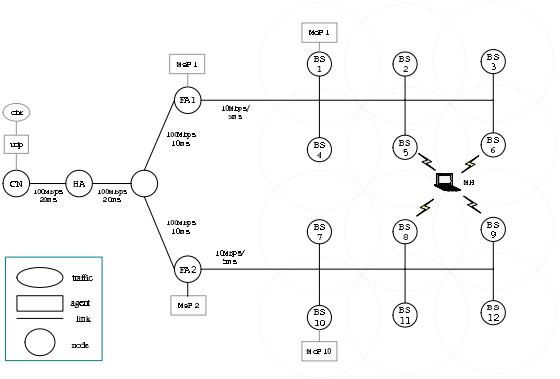


Fig. 7. Simulation Test-bed

sends a message that asks BS1 to leave the multicast group. BS1 no longer receives the message associated with the MH.

When a MH performs the handoff, a buffering BS immediately forwards MH the packets stored in buffer. Accordingly, the additional forwarding from an old base station is not required. Also, since the route update between a home agent and BSs is not required, multimedia service has no packet loss and minimum handoff delay.

4 Simulation

In this section, we evaluate the performance of the two proxy systems (General proxy and TOPS), based on simulation. We use the network simulator (NS-2) to examine the proposed system's performance. Figure 7 show the simulation test-bed.

We use one multimedia server (CN), one intermediate node, one HA, two FAs and 12 BSs for simulation. BSs which belong to FA1 are from BS1 to BS 6. The other BSs belong to FA2. We use several FA to estimate the multimedia service quality in mobile node during handoff. The link between media server (CN) and HA, between HA and intermediate node, is set as 100Mbps transmission rate and 20ms transmission delay. The link between intermediate node and FA is set as 100Mbps and 10ms delay. In FA area, the link is set 100Mbps and 5ms delay. Wireless links is set as 10Mbps transmission rate and 10ms delay. Analyzing the media service quality of MN which is moving around, we evaluate the each proxy system.

Table 1 shows that TOPS is better performance than the General proxy, that located in foreign network, in terms of average transmission rate, average handoff latency, RMS jitter of transmission delay and number of packet loss.

In case of using TOPS, the number of packet loss is 0 and average handoff latency is lower than that of general proxy. So, the average transmission rate and

Table 1. The Characteristics of the transmitted data

	Avg. Transmission rate (bps)	Avg. handoff latency (ms)	RMS jitter (1-sigma)	No. of Pkt loss
General proxy	458,389	12-13	0.04450	5-6
TOPS	458,677	5-8	0.04101	0

jitter in TOPS is lower than those in general proxy. Consequently, multimedia service in TOPS is better than it in general proxy.

TOPS is on center in its management domain. So, if MN moves around the domain, the data retrieval time is lower than general proxy.

5 Conclusion

This paper surveys the state-of-the-art in providing media streaming service and mobility support to mobile nodes in the wireless network.

In standard Mobile IP, in order to send media data to the mobile nodes, the media data have to pass through the mobility agents, which are foreign agent or home agent, and a proxy server. Through this procedure, data delivery latency and packet loss increases. So, frequent handoff procedure makes multimedia service worse.

In our proposed TOPS, using transmission mechanism, layered encoded multimedia and the enhanced prefetching strategy, data delivery latency and packet loss decreases. So, although MHs take frequent handoff or wireless network state becomes worse, the multimedia service quality is a slightly worse. So, optimal multimedia service is given to MH.

Finally, we show that TOPS improves media streaming service in the mobile communication environment. In the simulation result, the service overhead is low relative to typical mobile communications.

References

[1] Dong-Hoon Nam, Seung-Kyu Park, "Adaptive multimedia stream presentation in mobile computing environment" TENCON 99. Proceedings of the IEEE Region 10 Conference, Dec 1999 Page(s): 966 -969 vol.2

[2] Xueyan Tang, Fan Zhang; Chanson, S.T "Streaming media caching algorithms for transcoding proxies" Parallel Processing, 2002. Proceedings. International Conference on, 2002 Page(s): 287 -295

[3] Soam Acharya and Brian Smith. "MiddleMan: A Video Caching Proxy Server" NOSSDAV'00

Auto-Networking Technologies for IPv6 Mobile Ad Hoc Networks

Jaehoon Jeong, Jungsoo Park, and Hyoungjun Kim

Protocol Engineering Center, ETRI, 161 Gajeong-dong, Yuseong-gu
Daejeon 305–350, Korea
{paul,pjs,khj}@etri.re.kr
http://www.adhoc.6ants.net/

Abstract. This paper presents auto-networking technologies for IPv6 mobile ad hoc networks. The auto-networking technologies consist of IPv6 unicast address autoconfiguration, IPv6 multicast address allocation, secure multicast DNS, and service discovery. These technologies are based on IPv6's inherent autoconfiguration facility, which can provide ad hoc users with automatic networking in IPv6 ad hoc environment.

1 Introduction

Wireless networks are categorized into two classes; (a) Infrastructured wireless network and (b) Infrastructureless wireless network. As infrastructured wireless network, there are wireless lan (WLAN), celluar networks (e.g., 3GPP and 3GPP2) and so on. The current Internet services can be provided through these infrastructured wired and wireless networks. A representative network of infrastructureless wireless network is ad hoc network. Mobile Ad Hoc Network (MANET) is the network where mobile nodes can communicate with one another without preexisting communication infrastructure such as base station or access point. When mobile nodes are necessary to communicate in the environments such as battlefield and disaster relief communication where are separated from the Internet, they need to construct a temporary and infrastructureless network. Recently, according as the necessity of MANET increases, the development of ad hoc routing protocols for multi-hop MANET has been being led very strongly by IETF MANET working group [1]. Also, ad hoc multicast routing protocols for multicast service, such as video conferencing, DNS service, and service discovery in MANET have been being developed. With this trend, if IPv6 that has lots of good functions such as stateless address autoconfiguration for address configuration is adopted well in MANET, users in MANET will be able to communicate more easily through the zeroconfiguration that provides easy configuration [2, 3].

This paper suggests four auto-networking technologies for automatic networking in IPv6 mobile ad hoc network. The first is IPv6 unicast address autoconfiguration through which a unique unicast address is configured in mobile node. The second is IPv6 multicast address allocation through which a unique

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 257–267, 2004.

multicast address is allocated to application that needs a new multicast address. The third is secure multicast DNS that every ad hoc node takes part in DNS service, such as name-to-address translation. The last is service discovery based on multicast DNS, which allows ad hoc users to discover the service information that is necessary to connect to or join the service when the name, transport protocol (e.g., TCP or UDP) and domain for the service are given.

The remainder of the paper is organized as follows. In Section 2, related work is presented. The auto-networking architecture and components are described in Section 3. we describe four auto-networking technologies in detail, in Section 4. We descibe our MANET testbed and the experiment of the auto-networking technologies in Section 5. Finally, in Section 6, we conclude the paper with future research work.

2 Related Work

IETF Zeroconf working group has defined the technology by which the configuration necessary for networking is performed automatically without manual administration or configuration in the environments, such as small office home office (SOHO) networks, airplane networks and home networks [3]. This technology is called zero-configuration or auto-configuration. The main mechanisms related to the autoconfiguration technology are as follows; (a) IP interface configuration, (b) Name service (e.g., Translation between host name and IP address), (c) IP multicast address allocation, and (d) Service discovery.

3 Auto-Networking Architecture

Mobile nodes in MANET play the role of host and router simultaneously. Each node should run a common ad hoc routing protocol for multi-hop routing. These nodes are connected dynamically through ad hoc routing. IPv6 address configuration in each node should precede ad hoc routing. However, because ad hoc network has dynamic topology according to time, DHCPv6 for stateful address autoconfiguration or Neighbor Discovery (ND) for stateless address autoconfiguration [4, 2] are difficult to adopt in ad hoc network. This paper suggests IPv6 unicast address autoconfiguration that considers the resolution of address duplication which can be caused by MANET partition and mergence. Also, for auto-networking in IPv6 MANET, it proposes other automatic configuration and network services, namely IPv6 multicast address allocation, secure multicast DNS and service discovery.

Fig. 1 shows the protocol stack supporting auto-networking in IPv6 MANET. Four auto-networking techologies including unicast address autoconfiguration are implemented in application layer. We assume ad hoc routing protocols for unicasting and multicasting are executed. We use IPv6 AODV and MAODV for unicasting and multicasting respectively [5, 6, 7].

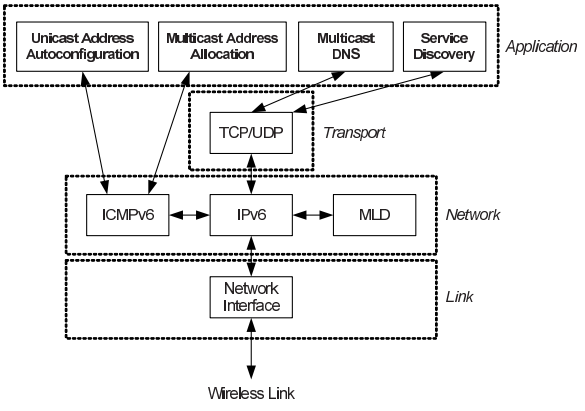


Fig. 1. Protocol Stack supporting Auto-Networking

4 Auto-Networking Technologies

4.1 IPv6 Unicast Address Autoconfiguration

IPv6 unicast address of ad hoc node can be autoconfigured by IPv6 address autoconfiguration for ad hoc networks [8, 9]. The configuration of address is comprised of three steps; (a) selection of random address, (b) verification of the uniqueness of the address and (c) assignment of the address into network interface. The duplication address detection (DAD) proposed in this paper not only checks address duplication during the initialization of address configuration, but also checks and resolves the address duplication, detected by intermediate nodes, during route discovery. Also, during the resolution of address conflict, the sessions using the conflicted address can be maintained until the sessions are closed.

IPv6 DAD for ad hoc network proposed in [9] cannot solve the the duplication of address by MANET partition and mergence because it is time-based DAD [10]. This autoconfiguration can be complemented with Weak DAD [10]. So, our DAD is a hybrid scheme of combining the time-based DAD and Weak DAD [8]. First of all, let's define time-based DAD as Strong DAD like in [10]. Strong DAD is used to check if there is address duplication in a connected MANET partition within a bounded time. Weak DAD is used to find the address duplication occurring when two or more MANET partitions are merged.

Procedure of Address Autoconfiguration IPv6 Unicast Address Autoconfiguration works like the state transition diagram of Fig. 2. States and events are descibed in Table. I and Table. II respectively. The IPv6 unicast address autoconfiguration consists of two phases. The first phase is to autoconfigure an IPv6 address in network interface by Strong DAD. The second phase is to detect the address duplication during routing process by Weak DAD. In Fig. 2, state 1

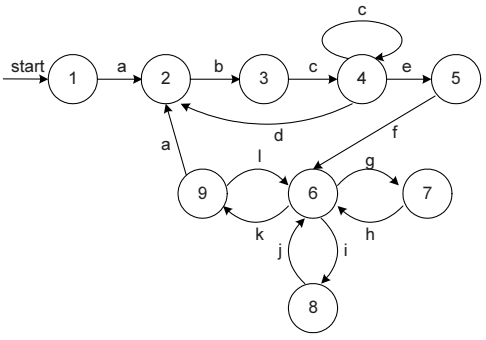


Fig. 2. State Transition Diagram of IPv6 Unicast Address Autoconfiguration

Table 1. State Description

State	Description
State 1	Node has no address.
State 2	Node has a temporary address as its own source address.
State 3	Node has a tentative address.
State 4	Node is verifying the uniqueness of the tentative address.
State 5	Node has verified the uniqueness of the tentative address.
State 6	Node is ready to process messages related to address and routing.
State 7	Node is processing AREQ (Address Request) message.
State 8	Node is processing RREQ (Route Request) message.
State 9	Node is processing AERR (Address Error) message.

through state 5 belong to the first phase and state 5 through state 9 belong to the second phase.

Strong DAD works as follows. Because this paper does not consider the global connectivity to the Internet, it assumes that MANET is a temporary network isolated from the Internet and the scope of addresses used in MANET is not global, but local. We use "fec0:0:0:ffff::/64", MANET_PREFIX, as MANET exclusive prefix [9]. This prefix will be replaced with another one for ad hoc network that will be determined by IPv6 working group.

Among the MANET_PREFIX, "fec0:0:0:ffff::/96", MANET_INIT_PREFIX, is used for temporary unicast address during Strong DAD [9]. The low-order 32 bits of the temporary address are configured with 32-bit pseudo random number. MANET_PREFIX is used for actual unicast address. The address that will be actual unicast address is a tentative address of which the uniqueness of the address has not been verified in MANET yet. The uniqueness is verified through Strong DAD and the low-order 64 bits of the tentative address is EUI-64 Identifier derived from MAC address. When the tentative address has already been used by another node, another new 64-bit pseudo random number is selected for the low-order 64 bits of the tentative address.

Table 2. Event Description

Event	Description
Event a	Node selects a temporary address.
Event b	Node selects a tentative address.
Event c	Node sends AREQ message for checking the uniqueness of tentative address and waits for AREP message indicating address duplication.
Event d	Node receives AREP (Address Reply) message for the tentative address.
Event e	Node has received no AREP after sending as many AREQ messages as the predefined number.
Event f	Node assigns the verified address in network interface.
Event g	Node receives an AREQ message.
Event h	Case 1 : Node forwards the AREQ message. Case 2 : Node discards the AREQ message. Case 3 : Node sends an AREP message to the source node.
Event i	Node receives an RREQ message.
Event j	Case 1 : Node forwards the RREQ message. Case 2 : Node discards the RREQ message. Case 3 : Node sends an AERR message indicating address duplication.
Event k	Node receives an AERR message indicating address duplication.
Event l	Node discards the AERR message.

In the last step of Strong DAD, state 5, when an actual unicast address is configured in network interface of mobile node, the temporary source address is not used any more as the source address.

During the ad hoc routing in state 6, Weak DAD detects the address duplication. Key is used for the purpose of detecting duplicate IPv6 addresses, which is selected to be unique by mobile node. When mobile node receives routing control packet, it compares the pairs of address and key contained in the control packet with those in the routing table or cache [8, 10]. RREQ (Route Request) and RREP (Route Reply) messages for route discovery in IPv6 AODV contain key for each address. When it detects the address duplication, it notifies the node having the duplicate address of the address duplication. For the message format for IPv6 unicast address autoconfiguration, there are three messages for address autoconfiguration; (a) Address Request (AREQ) message, (b) Address Reply (AREP) message, and (c) Address Error (AERR) message [8]. This message format can be used commonly for these three AREQ, AREP and AERR messages with 8-bit different type vaules. “Code” field is 8-bit unsigned integer, which has 0 or 1 as code value for message type. Code value 1 in AERR message indicates that the peer node’s address has been changed. In the other cases, code value is always 0. “Identifier” field is 32-bit unsigned integer, which is used to prevent duplicate AREQ message from being flooded. “Originator IPv6 Address” field contains the IPv6 address of the sender of ad hoc address autoconfiguration message. “Requested or Duplicate IPv6 Address” field contains the requested IPv6

address in AREQ and AREP messages, or the duplicate IPv6 address in AERR message.

AREQ message is used for the purpose of checking if a tentative address address is duplicate in the connected MANET partition during Strong DAD. AREP message is used so that the notification of address duplication is delivered to the node under Strong DAD by another node that receives an AREQ message and detects the address duplication with its own address. AERR message is used as the notification of address duplication in order that during processing control packets related to routing, a node finding the adress duplication can notify the originator node that has sent AREQ message of address conflict.

We define a new ICMPv6 message for IPv6 ad hoc address autoconfiguration instead of extending the current ND protocol, so that we separate IPv6 ad hoc address autoconfiguration from IPv6 stateless address autoconfiguration based on ND protocol. The address autoconfiguration in the current ND protocol is suitable only for fixed or mobile IPv6 networks of link-local scope, not for ad hoc network of site-local scope. Therefore, our address autoconfiguration works in multi-hop ad hoc network instead of IPv6 ND, only when node runs as ad hoc mode.

Maintenance of Upper-Layer Sessions under Address Duplication When address duplication happens and the duplicate address is replaced with another, the sessions above network layer can be broken. So, the survivability of upper-layer sessions using the duplicate address should be guaranteed.

In order to allow data packets related to the sessions using the duplicate address to be forwarded to destination nodes for a while, after sending error message (i.e., AERR message) to the node related to the duplicate address, the intermediate nodes that have perceived address duplication continue to forward on-the-fly data packets associated with the sessions using the duplicate address, on the basis of virtual IP address which is the combination of IP address and key, until the route entry for the duplicate address expires [8]. The node that receives an AERR message autoconfigures a new IPv6 address through Strong DAD and makes the new address used by the old upper-layer sessions that used the duplicate address as well as by new upper-layer sessions from this time forward. The node informs the peer nodes of the change of address by sending AERR messages with code 1. The "Originator IPv6 Address" field contains the duplicate address and the "Requested IPv6 Address" field contains a new address to be used for the communication. After receiving the AERR message, the peer node sends its packets to the node through IP tunneling. The destination address in outer IP header is the new IP address of the node that announced duplicate address and that in inner IP header is the duplicate IP address of the node. When the node receives tunneled packet from the peer node, it decapsulates the packet and delivers the data in the packet to upper layer. Both the node and peer nodes maintain the information of duplicate address and use it for processing IP tunneling.

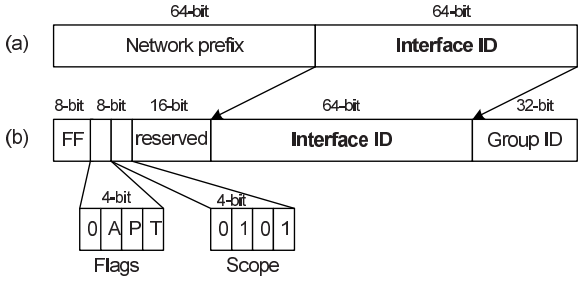


Fig. 3. Generation of IPv6 Interface ID-based Multicast Address. (a) is the format of IPv6 ad hoc unicast address and (b) is the format of IPv6 ad hoc multicast address

4.2 IPv6 Multicast Address Allocation

IPv6 multicast address allocation allows a unique multicast address allocated to application that needs a new multicast address, such as SDR (Session Directory Tool) that is one of the famous MBone tools. The main idea of the multicast address allocation proposed in this paper is based on Interface Identifier (ID) of IPv6 unicast address of which uniqueness has been already verified. So, the allocation of unique multicast addresses is possible for ad hoc node itself, without another multicast address allocation server.

Format of Multicast Address The format of site-local unicast address and that of site-local multicast address are shown in Fig. 3. A unique site-local scoped multicast address is formed as follows; So that we indicate that the multicast address of Fig. 3 (b) is based on interface ID, namely, Interface ID-based multicast address, P-bit (Interface bit) is set to 1. In order that we indicate the address is used temporarily, T-bit (Temporary bit) is set to 1. Also, we define a new bit, A-bit (Ad Hoc bit), so as to indicate this multicast address is one used in ad hoc network. So, A-bit is set to 1. Because the scope of the address is site-local, the Scope field is set to 5 which is the decimal number for binary value "0101". The 16-bit reserved field is set to zero. The Interface ID field of the multicast address is set to the value of that of the unicast address, the low-order 64 bits of site-local scoped unicast address configured by IPv6 ad hoc address autoconfiguration. Because the uniqueness of the unicast address's interface ID has already been verified, the uniqueness of a multicast address based on interface ID is guaranteed without the procedure of verifying the uniqueness of the multicast address. Each node can generate a unique multicast address by selecting an unused 32-bit random number for Group ID field by itself without any help of multicast address allocation server. Therefore, this mechanism for multicast address allocation is suitable for MANET where dedicated server is difficult to deploy for some services.

When ad hoc node receives an AERR message, one of ICMPv6 messages, indicating address duplication, it does not allocate multicast addresses until

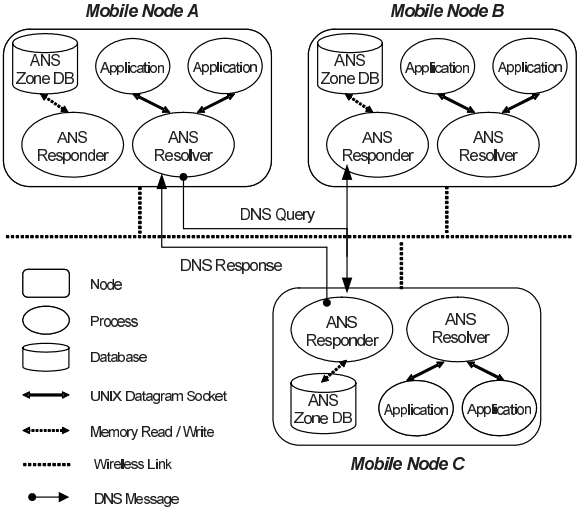


Fig. 4. DNS Name Resolution through Ad Hoc Name Service System (ANS)

a new unicast address is set up in its network interface. After a new unicast address is configured in network interface, the ad hoc node starts to allocate multicast addresses on the basis of the new Interface ID.

4.3 Secure Multicast DNS

We developed Ad Hoc Name Service System for IPv6 MANET (ANS) that provides the name resolution and service discovery in IPv6 MANET which is site-local scoped network [11]. Every network interface of mobile node can be configured automatically to have site-local scoped IPv6 unicast address by IPv6 ad hoc address autoconfiguration. ANS System consists of ANS Responder that works as DNS name server in MANET and ANS Resolver that performs the role of DNS resolver for name-to-address translation. Mobile node registers an AAAA type DNS resource record of combining its unicast address and host DNS name with DNS zone file of its ANS Responder (ANS Zone File). Fig. 4 shows the architecture of ANS System for name service in MANET and DNS name resolution through ANS. Each mobile node runs ANS Responder and Resolver. An application over mobile node that needs the name resolution can get the name service through ANS Resolver because ANS provides the applications with the library functions for name resolution through which they can communicate with their ANS Resolver through UNIX datagram socket.

In Fig. 4, ANS Resolver of mobile node A sends DNS query in ANS multicast address, "ff05::224.0.0.251" or "ff05::e000:00fb", which all ANS Responder should join for receiving DNS query [11]. When ANS Responder receives DNS query from ANS Resolver in other mobile nodes, after checking if it is responsible for the query, it decides to respond to the query. When it is responsible for the

query, it sends the appropriate response to ANS Resolver in unicast. In Fig. 4, mobile node C responds to DNS query of mobile node A.

Authentication of DNS Message In order to provide secure name service in ANS, it is necessary to authenticate DNS messages. We can use IPsec ESP with a null-transform or the secret key transaction authentication for DNS (TSIG) [12], which can be easily accomplished through the configuration of a group pre-shared secret key for the trusted nodes. In ANS, we implemented the authentication of DNS message on the basis of TSIG resource record [11]. All ANS Resolvers and Responders in a trusted group should share a group secret key for TSIG authentication. Whenever ANS Responder responds to DNS query, it sends DNS response message including TSIG resource record that has the hashing value of the DNS response based on the group's secret key. With TSIG resource record, ANS Resolver can decide if the response is valid or not.

4.4 Service Discovery

Service discovery allows ad hoc users to discover the service information that is necessary to connect to or join the service when the service name, transport protocol (e.g., TCP or UDP) and domain where the service is placed are given. We developed service discovery based on secure multicast DNS and DNS SRV resource record [13, 14]. We assume that mobile node running multicast or unicast service can register a DNS SRV resource record for each service with its ANS Zone File [13].

Procedure of Service Discovery For service discovery, a client sends DNS SRV query to get the information of a service via site-local multicast through ANS Resolver. The server that can serve the queried service responds to the client's query delivers the data of SRV resource record to the client via site-local unicast. When the client receives the response of SRV query, it checks whether the service is unicast or multicast. If the service is unicast, the client tries to connect to the server by the server's IPv6 address, transport protocol and port number. If the service is multicast, the client makes the multicast address related to the multicast service and joins the multicast group with the multicast address and UDP port number of the service [13].

5 Experiment in IPv6 MANET Testbed

We have implemented IPv6 AODV and MAODV as ad hoc unicast and multicast routing protocols, which have been extended for the support of IPv6, on the basis of NIST AODV [5, 6, 7]. These ad hoc routing protocols have been implemented in Linux kernel 2.4.18 version. Also, we have developed IPv6 Wireless Mobile Router (WR) for MANET testbed, which is a small box with IEEE 802.11b interface and embedded linux of kernel version 2.4.18 [7]. In order that we can set

up multi-hop MANET testbed and handle the topology easily, we have made the box regulate the signal range by controlling Rx and Tx power level of the wireless interface. In addition, we have implemented MAC filtering in device driver of wireless interface in order to filter adjacent node's packet in MAC level. With the Rx/Tx power control and MAX filtering, we can handle MANET topology at more liberty. With this MANET testbed, we tested the operation of MANET routing protocols and auto-networking technologies.

6 Conclusion

In this paper, we propose an architecture of auto-networking services in IPv6 mobile ad hoc nework. The services consist of four technologies; (a) IPv6 unicast address autoconfiguration, (b) IPv6 multicast address allocation, (c) Secure multicast DNS, and (d) Service discovery. These allow ad hoc users to communicate with one anther in easy and convenient way. As our future work, we will add more security functions to our auto-networking technologies in order to provide securer service against the various security attacks. Also, we will develop interworking between MANET and Internet so as to provide ad hoc users with global connectivity.

References

[1] Manet working group, `http://www.ietf.org/html.charters/manet-charter.html` 257
[2] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998. 257, 258
[3] Zeroconf working group, `http://www.ietf.org/html.charters/ zeroconf-charter.html` 257, 258
[4] T. Narten, E. Nordmark and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998. 258
[5] C. Perkins, E. Belding-Royer and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003. 258, 265
[6] E. Belding-Royer and C. Perkins, "Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing", draft-ietf-manet-maodv-00.txt, July 2000. 258, 265
[7] ETRI Ad Hoc Project, `http://www.adhoc.6ants.net/` 258, 265
[8] Jaehoon Jeong et al., "Ad Hoc IP Address Autoconfiguration", draft-jeong-adhoc-ip-addr-autoconf-02.txt, February 2004. 259, 261, 262
[9] C. Perkins et al., "IP Address Autoconfiguration for Ad Hoc Networks", draft-ietf-manet-autoconf-01.txt, November 2001. 259, 260
[10] Nitin H. Vaidya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks", MobiHoc 2002, June 2002. 259, 261
[11] Jaehoon Jeong, Jungsoo Park and Hyoungjun Kim, "DNS Name Service based on Secure Multicast DNS for IPv6 Mobile Ad Hoc Networks", ICACT 2004, February 2004. 264, 265
[12] P. Vixie, O. Gudmundsson, D. Eastlake and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000. 265
[13] Jaehoon Jeong, Jungsoo Park and Hyoungjun Kim, "Service Discovery based on Multicast DNS in IPv6 Mobile Ad-hoc Networks", VTC 2003-Spring, April 2003. 265

[14] A. Gulbrandsen, P. Vixie and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000. 265

New Binding Update Method in Mobile IPv6

Heshmatollah Khosravi, Hiroaki Fukuda, and Shigeki Goto

Waseda University, 3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan
{khosravi,fukuda,goto}@goto.info.waseda.ac.jp
http://www.goto.info.waseda.ac.jp

Abstract. In the current mobile IP standard, the home agent (HA) of a mobile node (MN) is located in the home link. If a mobile node (MN) is moved away from the home link, it takes time for MN to make a registration and binding update at the home agent (HA). It also generates extra traffic in the Internet because the Binding Update should be refreshed after the lifetime expires.
This paper proposes a new method for distributing multiple home agents (HAs) geographically. By applying this method, a mobile node (MN) can find a home agent (HA) which is nearest to it. It facilitates fast registration and short latency time. It also reduces the traffic of transaction. This technique is simple to apply. However, it is very effective. We demonstrate the capability of this method through working experiments.

1 Introduction

Mobile IPv6 [1] is a feasible mechanism for implementing static IPv6 addressing [2] known as the *home address* for a mobile node. Mobile IP allows packets sent to the home address to be delivered to the mobile node. It also hides any address changes from the transport and application layers. It enables a mobile node (MN) to roam between different networks.

In Mobile IPv6, each mobile node (MN) is identified with a static home address, regardless of the current point of attachment to the Internet. The home address is stored by the home agent (HA), located in the home link. When a MN is moved to a foreign link, it is addressable by a *care of address* (CoA), in addition to its home address. The care of address provides information about the current location of the MN. The CoA should be registered at the HA, when changed. The mapping or association of the care of address and the home address is called *binding*. A mobile node (MN) sends a binding update containing its new CoA. The Binding Update is shown in Fig. 1.

The binding information is valid only for the life time (420 sec) [3]. It should be updated after the life time is expired. Several packets should be sent and received between the MN and the HA for updating. These packets starts with ICMP Home Agent Address Discovery Request Message, ICMP Home Agent Address Discovery Reply Message, ICMP Mobile Prefix Solicitation Message, ICMP Mobile Prefix Advertisement Message, Authentication and Registration, and ends with Binding Update Acknowledgment packet [1].

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 267–276, 2004.

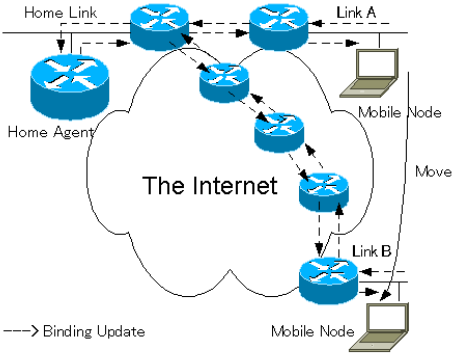


Fig. 1. Binding Update

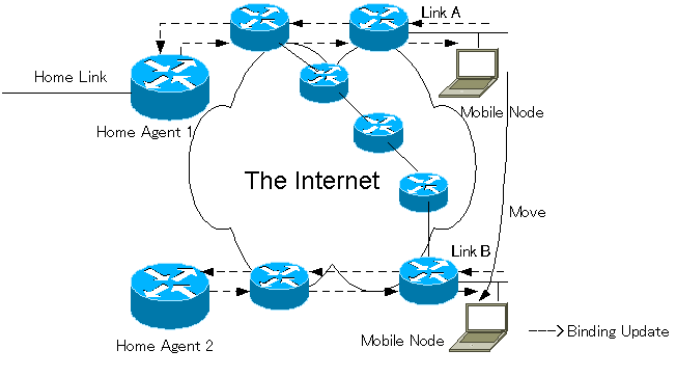


Fig. 2. New Binding Update Method

There is a problem in the current Mobile IP standard. If the MN is located at a foreign link such as link A, or link B shown in Fig. 1. The MN does not care about the distance to the HA. It may be far away from the home link, or it may be close to the HA. MN uses the same method for Binding Update. This is not efficient because when the MN is far away such as in link B, the Binding Update takes much time. It also generates extra traffic over the network.

In order to resolve this problem, this paper proposes to increase the number of HAs, and place them at different locations in the Internet. We explain our proposal in more detail in Section 3 of this paper. In this proposal, all the HAs would have the same *anycast* address. There has been known that anycast can be used in Mobile IPv6 [4, 5, 6]. However, our method is different from the existing one. We propose to modify the current protocol. By this new method, a MN can finds and registers at the nearest HA. The selection of HAs is dynamic. It is based on *Dynamic Home Agent Address Discovery* (DHAAD) [1] and uses IPv6 anycast [4].

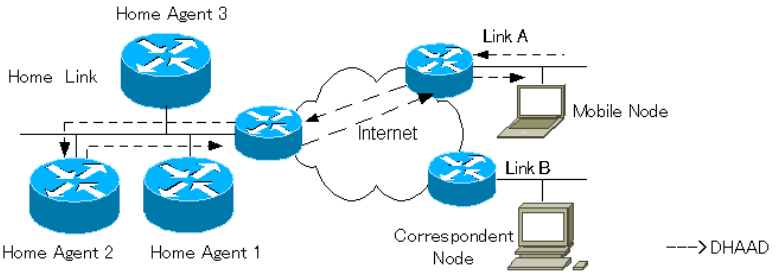


Fig. 3. DHAAD: Dynamic Home Agent Address Discovery

In Fig. 2, when MN moves to link B, it does not need to register at the primary home agent (HA1) in the home link, which is located at several hops away. Instead, it can register at HA2 more efficiently because it is closer to MN. This makes a short latency time in Binding Update and also it reduces the traffic in the Internet.

This technique can be applied over the Internet efficiently. In section 2, we present the Mobile IPv6. Section 3 mentions about our proposal and shows how it works. Section 4 deals with the experiments. Section 5 discusses the alternative approaches. And finally, Section 6 concludes this paper.

2 Current Mobile IPv6 and the Problems

Mobile IP [1, 11] allows a mobile node to move around the Internet dynamically. Mobile IP works at the network layer and deals with the routing of datagrams. It can handle the mobility among different media (LAN, WLAN, dial-up links, etc.). Mobile IPv6 is much improved from Mobile IPv4. Mobile IPv6 allows a MN to roam among different subnets freely while transparently maintaining the present connection and be reachable to the rest of Internet. Mobile IPv6 identifies each MN by its static home address regardless of the point of attachment.

When MN moves from the home link, it receives a care of address (CoA) from the foreign link. A CoA can be derived from the receipt of router advertisement in *stateless address auto-configuration* [7], or be assigned by a DHCP server in *stateful address auto-configuration* [10]. MN registers its current address at a HA on the home link. The HA intercept the packets, and forward them to the MN. This mechanism is transparent for the transport layer (e.g. TCP, UDP) and the application layer.

In Mobile IPv6, a MN can discover HA dynamically. It uses Dynamic Home Agent Address Discovery (DHAAD) protocol. Fig. 3 shows the information flow of DHAAD. DHAAD uses IPv6 anycast.

It is possible to have multiple Home Agents in the current IPv6 standard. When MN sends a Binding Update to the *Home Agents anycast address*. One of the Home Agents receives the Binding Update request. The HA should reject

Table 1. Example: Home Agent list

Home Agent List	Preference Number
HA1	−2
HA2	5
HA3	3

the Binding Update from the mobile node. Instead, it returns a list of all the home agents with their preference number, as shown in Table 1. The MN sends a Binding Update to the HA that has the highest number, i.e. HA2 in Table 1.

Then, the MN should use HA2 as the home agent. It is pre-determined statically. Although the multiple home agents realize back-up agents, it does not realize the right selection of home agents based on the distance in the network.

3 New Method for Mobile IPv6

Our new method uses anycast to find the nearest home agents among several HAs. The HAs are geographically distributed. And they belong to the same anycast group. An anycast address can be assigned to the multiple home agents (HAs). A HA is a member of the anycast group if the anycast address is assigned to one of the interfaces of the HA. A packet sent to an anycast address is delivered to the closest member in the group. The *closeness* is measured by the distance in routing protocol. We have already shown the diagram of the new idea in Fig. 2. All the HAs have the same anycast address. A mobile node can find the *nearest* HA by the DHAAD (Dynamic Home Agent Address Discovery) mechanism. According to the IPv6 anycast protocol, the DHAAD packet sent to an anycast address is routed to the nearest HA. The actual route is determined by the distance in the network [5].

3.1 Modified Protocol for Home Agent Registration

When a mobile node (MN) detects that it has moved from one link to the other, it receives a new care of address (CoA). In the current Mobile IPv6 draft, the MN will send the Binding Update to the HA if it keeps the address of HA in the Binding Cache. In our proposal, MN deletes Home Agent record in the Binding Cache when it moves from one link to the other. MN should find a new HA. It finds the nearest HA by using anycast. This scheme has no single point of failure. If one of the HAs is down, the MN can find the second nearest one.

In Section 4, we will explain the working example of Binding Update, like the following. A MN moves from one foreign link to another link.

```
receive a new prefix 2001:0200:0001:0006::
found a new router fe80:0001::0206:5bff:fe49:24cc(2001:0200: 0001:0006::)
CoA has changed to 2001:0200:0001:0006:0209:6bff:fefa:7b78
location = 2
```

```
BU timer stopped.
Binding cache removed.
no home agent.  start ha discovery.
BU timer started.
no home agent.  start DHAAD.
MIP6_BU_PRI_FSM_STATE_WAITA
MIP6_BU_PRI_FSM_STATE_BOUND
```

where the MN receives a new care of address, and start finding a home agent (HA) by DHAAD.

In our proposal, all the preference numbers in Table 1 are set to zero. This is a modification to the current protocol. When the nearest HA receives the Binding Update request, it does not reject the request. It does not send the list of all the Home Agents to the MN either, but sends its own address only.

If the nearest HA to a MN is not the primary HA in the home link, the HA sends the binding update to the primary HA. The *corresponding node* (CN) sends the first packet to the home address at the primary HA. The packet will be forwarded to the care of address (CoA) by the primary HA.

3.2 Implementation

There have been several implementation of IPv6 protocol which have Mobile IPv6 capability. We use KAME [3] implementation for FreeBSD for our experiments. We modify the source code to implement our new method. There are two major changes in our KAME code.

1. The source code is changed so that MN removes the previous HA record from the binding update list when it moves to the other link. This modification is necessary because it prevents the MN to send the Binding Update to the old HA. It enables MN to find the nearest HA by using DHAAD.
2. The source code is changed so that HAs do not send the HA list with the preference number in response to the DHAAD requested from a MN. Instead it sends its own address to the MN. The modification allows the MN to get the nearest HA properly.

Module name in KAME	where to use	New function
mip6_mncore.c	mobile node	Clear cache after moving
halist.c	home agent	Respond to MN with its own address

The above table describes two modules in KAME which are modified by us. These modifications are tested in the environment described in the following section.

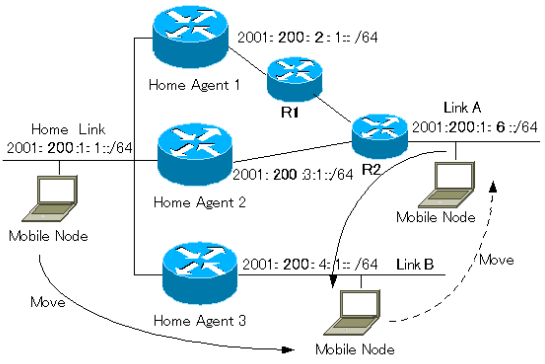


Fig. 4. Topology of the Testbed

4 Experiments and Evaluation

Fig. 4 shows the configuration of our testbed. For simplicity, all HAs are connected to the same link (home link). The home link has the address of 2001:200:1:1::/64. There are three home agents (HAs). They have the same anycast address of 2001:200:1:1:fdff:ffff:ffff:fffc. The IP address of HA1 is 2001:200:1:1::1, HA2 is 2001:200:1:1::3 and HA3 is 2001:200:1:1::4. A mobile node (MN) has the home address of 2001:200:1:1:209:6bff:fefa:7b78. There are two foreign links A and B. They have prefix addresses of 2001:200:1:6::/64 and 2001:200:4:1::/64, respectively. The MN moves among the home link, link A and link B.

4.1 Standard Mobile IPv6 Operation

The following list shows the HAs.

```
lladdr          gaddr            flags   pref    lifetim ltexp
fe80::240:5ff:f 2001:200:1:1::1  --H       0     1800    1647
fe80::240:5ff:f 2001:200:1:1::4  --H       0     1800    1648
fe80::290:27ff: 2001:200:1:1::3  --H       0     1800    1648
```

After moving to link B from the home link, MN sends the following message. It shows a new prefix of IPv6 address, and the care of address (CoA) is changed. However, MN still uses the same home agent, HA1.

```
receive a new prefix 2001:0200:0004:0001::
found a new router fe80:0001::02b0:d0ff:fe07:791b(2001:0200: 0004:0001::)
CoA has changed to 2001:0200:0004:0001:0209:6bff:fefa:7b78
BU timer started.
```

If MN moves to link A, it still uses HA1 according to the current standard of mobile IPv6. In case MN does not keep any record in the HA list, it will search for a new HA by DHAAD protocol.

In our testbed, MN sends ICMP type-#150 which is the Home Agent Address Discovery Request message to the anycast address of 2001:200:1:1:fdff:ffff:ffff:fffe. Then, the nearest home agent, HA2, receives the packet and replies with ICMP type-#151 which is the Home Agent Address Discovery Reply message. The reply message contains the Home Agent list.

4.2 New Mobile IPv6 Operation

In the new method, MN first deletes the HA list and it tries to find the nearest HA. For example, MN moves from the home link to link B. It sends the following message. It successfully find the nearest HA3.

```
receive a new prefix 2001:0200:0004:0001::
found a new router fe80:0001::02b0:d0ff:fe07:791b(2001:0200:0004:0001::)
CoA has changed to 2001:0200:0004:0001:0209:6bff:fefa:7b78
location = 2
no home agent.  start ha discovery.
BU timer started.
no home agent.  start DHAAD.
MIP6_BU_PRI_FSM_STATE_BOUND
```

If MN moves to link A from link B, it deletes the previous HA record from the Binding Update list. It also removes the Binding cache and starts DHAAD. It finds the nearest HA, i.e. HA2.

```
receive a new prefix 2001:0200:0001:0006::
found a new router fe80:0001::0206:5bff:fe49:24cc(2001:0200: 0001:0006::)
CoA has changed to 2001:0200:0001:0006:0209:6bff:fefa:7b78
location = 2
BU timer stopped.
Binding cache removed.
no home agent.  start ha discovery.
BU timer started.
no home agent.  start DHAAD.
MIP6_BU_PRI_FSM_STATE_WAITA
MIP6_BU_PRI_FSM_STATE_BOUND
```

The log file is the same record shown in Section 3.1.

4.3 Comparing Binding Update Time

We compare the time for Binding Update from the MN to HAs. MN moves among the home link, link A and link B. MN is located on link A. We force MN to perform the Binding Update with HA1 by intentionally shut down HA2.

The total time for Binding Update to HA1 is 1.438417 sec. The time for Binding time to HA2 is 0.996083 sec. The time for HA3 is 0.996699 sec. The Binding Update time to HA1 is 44% longer than the time for HA2 or HA3.

Home Agent	Binding Update time (Sec)	Location of MN
HA1	1.438417	Link A
HA2	0.996083	Link B
HA3	0.996699	Link B

There are two routers (R1 and R2) between the MN and HA1. These routes cause delay time in packet forwarding. The Binding Update time to HA1 is longer than the Binding Update time to HA2 and HA3.

The result shows that it is meaningful to select the right home agent because we can make the time for binding update much shorter.

5 Discussion

This section compares our new method with alternative methods for the Binding Update. Method A is a standard protocol. It has only one home agent (HA). Method B is defined in the current standard Mobile IPv6 protocol. It has multiple home agents. They are located at the same home link. They are addressed by a common unicast address in IPv6. There is a HA which is selected by anycast mechanism of IPv6. The HA is not simply used, because it rejects the request from a mobile node (MN). The HA responds to the MN by sending a list of home agents (HAs) with preference numbers. In Method B, the home agent which has the highest preference number is selected finally. The merit of method B is backup and load balancing.

Method C is our new mechanism. It uses distributed multiple home agents. It uses anycast in IPv6. The HAs belong to the same anycast group. If one HA is selected by the anycast mechanism, it is used as a primary home agent. It does respond to the MN. It sends the address of the home agent, not the list of all the home agents.

Method	Number of HAs	Address	Response	Reply	Location of HAs
A	single	Unicast	Accept	One HA (fixed)	centralized
B	multiple	Anycast	Reject	List of HAs	centralized
C	multiple	Anycast	Accept	One HA (selected)	distributed

The new method (C) simply realizes the selection of nearest HA to a MN. It is a modification of Method B. We should change the protocol to accept the request, and send the proper address of the HA. We have realized the protocol as a part of KAME implementation of IPv6.

Our new method requires additional time for DHAAD. However, if a MN moves locally within a site, it does not need DHAAD. The new method increases the number of HAs so that we can utilize the nearest HA to a MN.

Internet Draft [21] describes Hierarchical Mobile IPv6 (HMIPv6). It is an efficient method for local moving within a site. MAP (Mobile Anchor Point) is used to make local moving more efficient with a faster handoff. On the other

hand, our method deals with global mobility of MIPv6. We have a plan to investigate the new method to improve the local mobility.

6 Conclusion

We proposed the new method as an enhancement to Mobile IPv6. It solves the problem that exists in the current standard for Binding Update. The new method increases the number of Home Agents. Their locations are distributed. It helps a Mobile Node (MN) to apply DHAAD dynamically by IPv6 anycast. MN can find the nearest Home Agent. MN makes a registration and Binding Update at the HA. It realizes short latency in binding update time and reduces the extra traffic in the network. The new method decreases the latency time of binding update. It also reduces the traffic in the Internet. This scheme has no single point of failure.

References

[1] Johnson, D., Perkins, C., Arkko J.: Mobility Support in IPv6, draft-ietf-mobileip-ipv4-24.txt, Jun 2003. 267, 268, 269
[2] Deering, S., Hinden, R.: RFC 2460 – Internet Protocol, Version 6 (IPv6) Specification, Dec 1998. 267
[3] Wide Kame Project, `http://www.kame.net` 267, 271
[4] Johnson, D., Deering, S.: RFC 2526 – Reserved IPv6 Subnet Anycast Addresses, Mar 1999. 268
[5] Hinden, R., Deering, S.: RFC 2373 – IP Version 6 Addressing Architecture, Jul 1998. 268, 270
[6] Hagino, J. I., Ettikan, K.: An analysis of IPv6 anycast, Sep 2000. `http://playground.iijlab.net/I-d/draft-itojun-ipv6-anycast-analysis-00.txt` 268
[7] Thomson, S., Narten, T.: RFC 1462 – IPv6 Stateless Address Autoconfiguration, Dec 1998. 269
[8] Narten, T., Nordmark, E., Simpson, W.: RFC 2461 – Neighbor Discovery for IP Version 6 (IPv6), Dec 1998.
[9] Conta, A., Deering, S.: RFC 2473 – Generic Packet Tunneling in IPv6 Specification, Dec 1998.
[10] Droms, R.: RFC 2131 – Dynamic Host Configuration Protocol, Mar 1997. 269
[11] Perkins, C.: RFC 3344 – IP Mobility Support for IPv4, Aug 2002. 269
[12] Kent, S., Atkinson, R.: RFC 2401 - Security Architecture for Internet Protocol, Nov 1998.
[13] Ferguson, P., Senie, D.: Ingress Filtering in the Internet, `ftp://ftp.ietf.org/internet-drafts/draft-ferguson-ingress-filtering-03.txt`
[14] Methfessel, M., Dombrowski, K. F., Langendorfer, P., Frankenfeldt, H., Babanskaja, I., Mathaei, I., Kraemer, R.: Vertical Optimization Of Data Transmission For Mobile Wireless Terminal, IEEE Wireless Communication, Dec 2002.
[15] Xie, J., Akyildiz, Ian F.: A Novel Distributed Dynamic Location Management Scheme for Minimizing Signaling Costs in Mobile IP, IEEE Transmission on Mobile Computing, Vol. 1, No. 3, July – Sep 2002.

[16] Das, S., Mcauley, A., Dutta, A., Misra, A., Chakraborty, K., Das, S. K.: IDMP: An Intradomain Mobility Management Protocol for Next-Generation Wireless Network, IEEE Wireless Communication, Jun 2002.
[17] Salkintzis, A. K., Fors, C., Pazhyannur, R.: WLAN-GPRS Integration for Next-Generation Mobile Data Network, IEEE Wireless Communication Oct 2002.
[18] Montes, H., Gomez, G., Cuny, R., Paris, J. F.: Deployment of IP Multimedia Streaming Services In Third-Generation Mobile Networks, IEEE Wireless Communication, Oct 2002.
[19] Katabi, D., Wroclawski, J.: A Framework for Scalable Global IP-Anycast (GIA), ACM SIGCOMM Computer Communication Review, Volume 30 , Issue 4 pp. 3–15, Oct 2000.
[20] Engel, R., Peris, V., Saha, D.: Using IP Anycast for load distribution and Server Location, `http://noah.cs.ccu.edu.tw/SARP/anycast-gi98.pdf`
[21] Soliman H., Castelluccia C., El-Malki K., and Bellier L.: Hierarchical Mobile IPv6 mobility management (HMIPv6), draft-ietf-mobileip-hmipv6-08.txt, Jun 2003. 274

Dynamic Agent Advertisement of Mobile IP to Provide Connectivity between Ad Hoc Networks and Internet*

Jin-Woo Jung[1], Doug Montgomery[1], Kyungshik Lim[2], and Hyun-Kook Kahng[3]

[1] National Institute of Standards and Technology
100 Bureau Drive, Stop 8920, Gaithersburg, MD 20899, USA
[2] Computer Science Department Kyungpook National University
Taegu, 702-701, Korea
[3] Department of Electronics Information Engineering, Korea University
♯208 Suchang-dong Chochiwon Chungnam, Korea
jjw@korea.ac.kr

Abstract. Although the Ad Hoc On-Demand Routing Protocol (AODV) is well designed for the ad hoc network, it does not deal with Internet connectivity. While some of solutions are proposed for integrating the ad hoc networks with global Internet, there are some limitations and drawbacks. In this paper, we propose the dynamic agent advertisement to reduce the control packets overhead and the power consumption due to a redundant packet processing. We use ns2 to compare the proposed approach with the existing solutions in terms of the overhead and throughput for packet transmission.

1 Introduction

In recent, the proliferation of mobile communications and multimedia services has made mobility support on the Internet an important issue. This trend has led to the introduction of new protocols to Internet. In this paper, we examine a recently Ad Hoc On-Demand Routing Protocol (AODV) and the Mobile IPv4 (MIP) standard from the perspective of Internet connectivity over wireless environments [1,2].

MIP, supporting a macro-mobility at network layer, is the oldest and probably the most widely known mobility management proposal. The MIP allows the mobile node to move around the world with Internet connectivity. The AODV is generally viewed as a stand-alone network, where communication is only supported between nodes in the specific ad hoc network. The lack of connectivity to the wired infrastructure enables simple management and deployment, but limits the applicability of ad hoc network to scenarios that require connectivity outside the ad hoc network.

From the view point of Internet mobility, some of MIP-based proposals for providing Internet connectivity in ad hoc environments are proposed. The challenge to enable such support stems from the need to provide good connectivity

* This research was supported by University IT Research Center Project.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 277–286, 2004.

in a dynamic, resource poor (i.e. limited power and bandwidth) environment[3]. However, the proposed solutions have several drawbacks over integrating environments. In this paper, we propose the dynamic agent advertisement to provide good connectivity with reduced overhead and to achieve good performance when mobile nodes move a new subnet. In our approach, we suggest the mobility agent located at boundary between ad hoc network and fixed network advertises its information to ad hoc network with two different lifetimes and two different advertisement scopes. The proposed approach combines the advantages of both MIP and AODV. The remainder of this paper is organized as following. Section II examines the tradeoffs between using advertisement and solicitations. In Section III, we describe our protocol and optimizations. The performance of our approach is detailed in Section IV and Section V concludes the paper.

2 Related Works

2.1 Connectivity for Mobile Ad Hoc Networks

There are several researches on extending MIP capabilities to an ad hoc network for supporting global roaming and Internet connectivity. These approaches are broadly divided into three classes in aspects of movement detection and finding mobility agent.

- First one mainly depends on the periodic advertisements from mobility agents (called as gateway): This approach provides good connectivity, but imposes a high overhead, especially when not all the nodes in the ad hoc network require Internet connectivity.
- Second one primarily uses solicitation method: such approaches have the overhead of maintaining connectivity to external traffic patterns but negatively impact the mechanisms necessary for MIP such as agent discovery and movement detection.
- The hybrid approaches utilize solicitation and advertisement signaling between a MANET node and the gateway. It still has traffic overhead and delay for registration.

Although the existing solutions provide Internet access for ad hoc networks, the solutions continue to suffer from several drawbacks as specified following. The most significant of which is high overhead of foreign agent advertisement messages (flooding). MIP relies on link-layer broadcasts to provide foreign agent information to interested nodes. However, these broadcasts can prove to be extremely expensive in a ad hoc network where a broadcast translates to the packet being flooded throughout the network. To reduce the flooding of advertisements, some schemes increase the beacon interval (i.e. the interval between successive advertisement floods). However, these increased interval cause to degrade the handoff performance of MIP due to delayed movement detection. Second, some protocols have an overhead for route maintenance. If not all the nodes in ad hoc

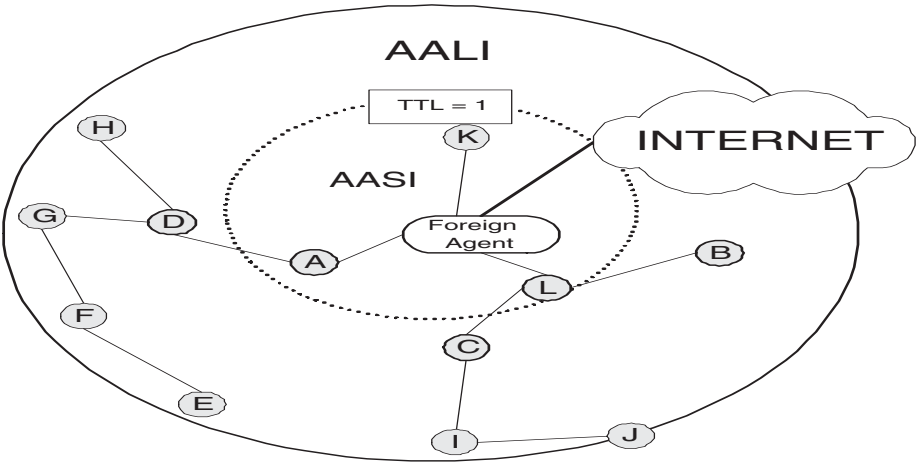


Fig. 1. Two types of agent advertisement

network require connectivity, the AODV's route maintenance can have a negative on the AODV due to excessive flooding overhead. Third, it is not useful that Agent advertisements set up reverse routes to the mobile node over frequently moving.

3 Dynamic Agent Advertisements

In this paper, we propose on demand Internet connectivity in ad hoc environments for achieving low handoff delay and low burden. The proposed approach is based on MIP for connecting an ad hoc network, in which AODV is used, to the Internet. While the purpose of the registration process in MIP is to update the binding information between the home address of mobile node and its current care-of address, the aim of AODV is to inform MIP in the mobile node, with having active route, of changing subnets (or a new AODV network). The proposed approach uses two kinds of advertisement in foreign agent so that the AODV's scarce resources are not further burdened with MIP overhead and the required information is received on demand.

- Agent Advertisement message with Short beacon Interval (AASI): All values in this message are identical to agent advertisement message in MIP (its advertisement scope is limited to one hop from mobility agent).
- Agent Advertisement message with Long beacon Interval (AALI): The message is identical to agent advertisement message of MIP, except with longer registration lifetime and larger time-to-live value (i.e. registration lifetime is over 300sec and TTL is set to 200).

In fig 1, the proposed model consists of three components: the foreign agent, inner nodes, and outer nodes. The foreign agent is responsible for routing packets

between the ad hoc network and the Internet. While AASIs are periodically flooded within one hop from the foreign agent, AALIs are broadcasted through the whole ad hoc network. All nodes within one hop from foreign agent are notated as inner nodes (e.g. A, K, L in fig 1), and all nodes outside one hop range from foreign agent are depicted as outer nodes (e.g. B, C, D, etc in fig 1). In this paper, we assume that nodes in an ad hoc network that want Internet access use their home address for all communication and register with a foreign agent.

3.1 Agent Discovery

In the traditional MIP the movement detection is accomplished by receiving agent advertisement. Unfortunately none of the move detection methods provided MIP is suitable because of the on demand property and the multihop nature of ad hoc networks [6][7]. In this paper, the mobile node can find out a new mobility agent with one of following three methods.

- If the mobile node has one more active route, it can identify a new mobility agent with the extended RREP (e-RREP, Section 3.5).
- If the mobile node is located within one hop from a foreign agent, it can detect a new mobility agent by receiving AASI.
- If the mobile node has no active route, it can identify a new mobility agent with unicast solicitation message.

The movement detection in the proposed approach is archived by using routing update events and the extension of RREP.

When the event which the route for the destination is updated is occurred, the AODV compares the existing mobility agent (or gateway) with a fresh mobility agent included in the received RREP (e-RREP).

If the mobile node receives a new e-RREP including the different mobility agent information with the existing mobility agent, it first determines the hop count of the new e-RREP is smaller than the hop count for the existing mobility agent. If the hop count of the new e-RREP is smaller than the existing hop count, it immediately updates its mobility agent. Otherwise, it must wait some period. It updates the mobility agent if it does not receive another e-RREP message during waiting time. The mobile node without ongoing communication, it is valid that the mobile node is non-sensitive to the handoff delay. In this paper, the mobile node which has no active route finds out a new mobility agent with the unicast agent solicitation. If the mobile node is stayed in one hop from mobility agent, it can detect its movement with AASI. This procedure is the same as in that of the traditional MIP.

3.2 Registration

Once a node has recognized that a new mobility agent is available in a new ad hoc network, a node should register with a home agent via the new mobility

agent. The registration procedure is almost the same as in the traditional MIP, except that the registration request may have to traverse multiple hops before reaching the foreign agent (and vice versa for the registration reply). A mobile node generates Registration Request message with the same method in standard MIP[1]. The foreign agent and home agent process the registration message as specified in [1] by recording the new care-of address for the mobile node. The foreign agent can utilize the AODV route discovery procedure to rediscover a route to the mobile node for delivering the Registration Reply message to the mobile node.

3.3 Routing

A mobile node does not initially know whether the destination node is within the ad hoc network, or whether it is reachable through the foreign agent. It therefore creates e-RREQ packet and broadcasts this packet to the whole ad hoc network.

This e-RREQ packet can be replied by one more nodes among three components:

- If an intermediate node receiving the e-RREQ have a valid route to the destination, it returns an e-RREP message including mobility agent information to the originator.
- If a mobility agent receives the e-RREQ, it first checks its visitor entry for the destination. If such an entry does not exist, it returns an e-RREP message with a large hop count to the originator (i.e. 200). That is, the mobility agent replies with a proxy ARP to the originator.
- If the destination node within an ad hoc network receives the e-RREQ packet, it replies an e-RREP packet including agent's information to the source.

If the destination exists within the ad hoc network, an e-RREP can be returned by both a mobility agent and the destination node (or an intermediate node which have a valid route to the destination node). In this case, if the mobile node receives e-RREP with large hop count from the mobility agent, it must store this route but wait some period because it is possible for the mobile node to receive an e-RREP from the foreign agent before it receives a route reply from the destination node (or a intermediate node) within the ad hoc network. Therefore, the mobile node should retain this route, and utilize it only after it has concluded that the destination is not located in the ad hoc network. Otherwise the mobile node receives a e-RREP from the destination, it can transmit data packets to that destination.

3.4 Operations for Ad Hoc Mobile Node

In this paper, mobile nodes in ad hoc network are divided into inner nodes and outer nodes. While AASI is only broadcasted to inner nodes, AALI is advertised

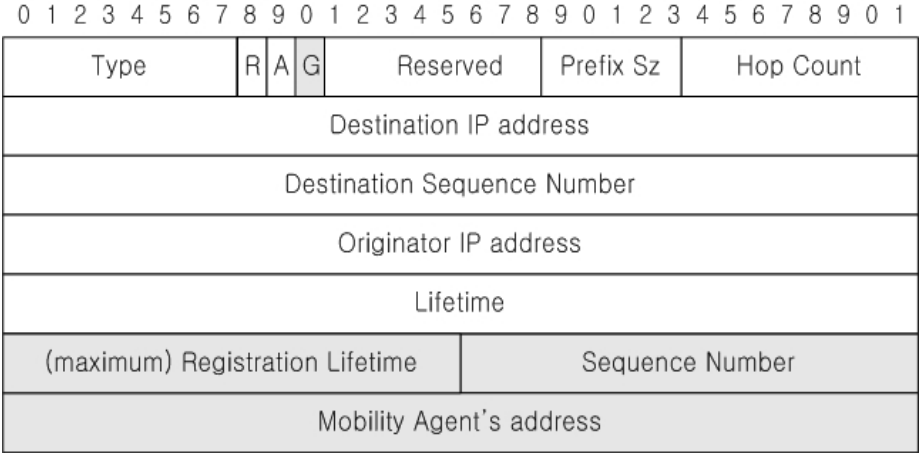


Fig. 2. Extension of Route Reply Message

to both inner nodes and outer nodes. When a mobile node receives an AALI, it records the IP address of the foreign agent, together with the sequence number of the Agent Advertisement, in its foreign agents list. It then assigns long registration lifetime to that entry. That is, outer node should not change this entry until it receives another agent advertisement from a new foreign agent or a Router Error (RERR) message from any inner node. Since an AASI is broadcasted within one hop from the foreign agent by the foreign agent, outer nodes can not maintain foreign agent information timely fashion. For solving this limitation, inner nodes are responsible for monitoring and detecting the reachability of the foreign agent. If an inner node does not receive an agent advertisement within an advertised lifetime of AASI, it sends unicast-solicitation to foreign agent. If an inner node can not receive any agent advertisements after broadcasting three successive solicitations, it must broadcast Route Error message for foreign agent in the whole ad hoc network. If an inner node moves outer range of mobility agent, it will receive an unicasted agent advertisement with long lifetime from mobility agent.

3.5 New Message Types

When an intermediate node or a mobility agent receives e-RREQ, it must reply with e-RREP including mobility agent information. Fig 2 shows the extension of RREP specified in the standard AODV. Unless otherwise noted, the parameter values for the standard AODV are the same as those suggested in [2]. We extend the RREQ message and the RREP message for showing which it can support the proposed algorithm or not. For RREQ message, we add the 'C' bit in reserved filed in the traditional AODV standard for showing that a node supports the extensions for the Internet Connectivity. For the RREP message, we add the 'G'

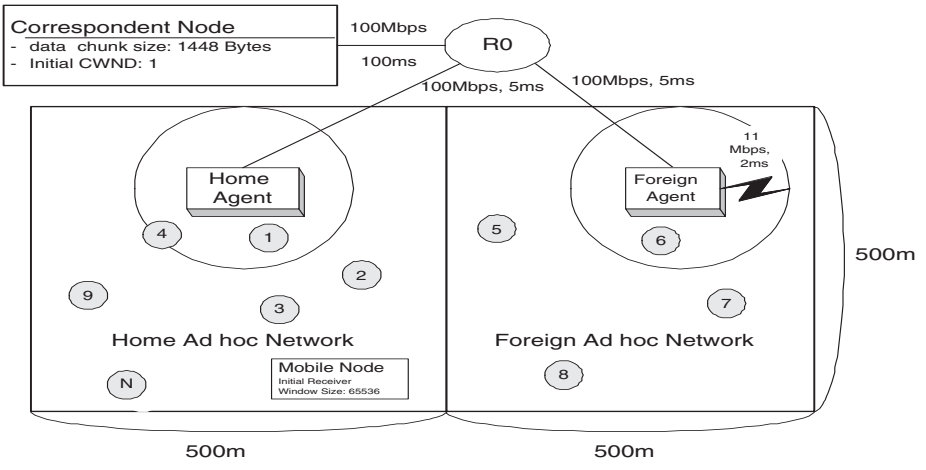


Fig. 3. Simulation Topology

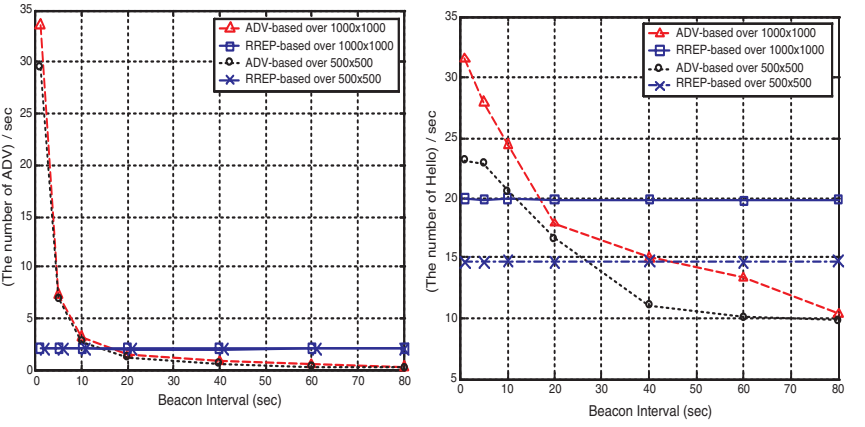
bit in reserved field in the standard AODV for indicating the RREP includes information of a new mobility agent and extend the message to include mobility agent information(see shaded portion of fig 2). The mobile node in ad hoc network can detect the subnet changing based this information.

4 Performance Evaluations

In this section we present a summary of our simulation results. The protocol is implemented in the NS-2 simulator with mobility extensions[4]. For generating random scenario, we use CMU's node-movement generator included in ns-2 2.1b9a. Unless otherwise noted, parameter values for MIP and AODV are the same as those suggested in [1,2], respectively.

4.1 Simulation Model

There are home agent and foreign agent running both AODV and MIP. There is one correspondent node on the wired network connected to both wireless domains through R0. Fig 3 illustrates this network configuration. There are three constant bit rate (CBR) traffic sources distributed randomly within each ad hoc network. The destination of each of the data sessions is the correspondent node in the wired network. The CBR data packets are 120 bytes and the sending rate is 66 packets per second. All mobile nodes move according to the random waypoint mobility model [5]. The Mobile node speeds are randomly distributed between zero and twenty milliseconds. The pause time is consistently 10 seconds. For our proposed approach, we assume that AASI is broadcasted with one second of interval and AALI is flooded with 180 seconds of interval.



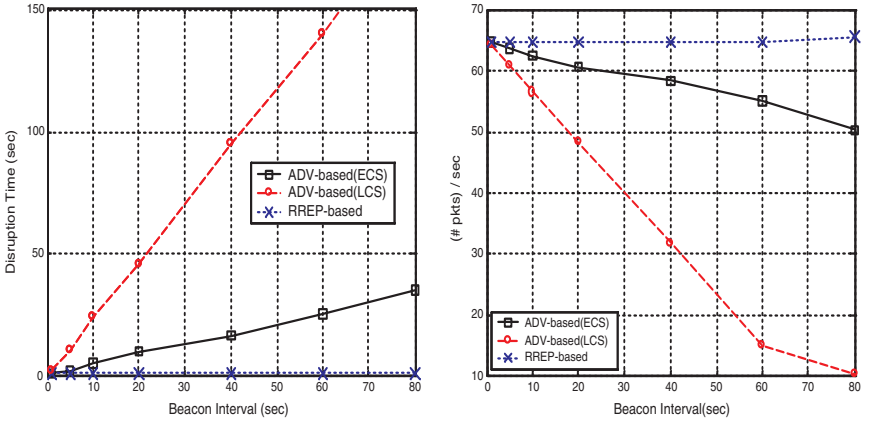
(a) MIP overhead (b) AODV overhead

Fig. 4. Mobile IP overhead and AODV overhead

4.2 Simulation Results

Fig 4(a) shows the difference in MIP overhead between the broadcast and the proposed approaches. Control packets may not be consuming large amount of bandwidth, but they may be too much to interfere with the transmissions. The MIP overhead is calculated with the number of advertisements per second and is counted on a per-hop basis, meaning that a packet that travels five hops is counted five times.

For the advertisement based approach (ADV-based), the number of beacon message is flooded within the whole network decreases as the beacon interval increases. In the ADV-based approach, two mobility agents flood each network periodically with their agent advertisements. According to the Fig 4(a), the MIP overhead of the ADV-based approach is sharply decreased at 5s and the number of advertisement per sec is one at 40s interval. The MIP overhead of the proposed approach is almost fixed to 2.1 packets per second because it limits the scope of flooding to one hop. Fig 4(b) shows the AODV overhead for each approach. In this simulation, we assume just six traffic source within the whole network and the interval for hello message is one second. If the number of active route rises, the AODV overhead simultaneously increases. For the ADV-based approach, the short beacon intervals cause the more redundant active route due to agent advertisement and result in increasing the AODV overhead (that is, all nodes in ad hoc network don't want to Internet connection). This also contributes to the degradation of throughput. The AODV overhead of our proposed approach is about 20 messages per second because its hello message overhead depends on the number of inner node and is independent of the beacon interval. Fig 5(a) illustrates the disruption time due to the handoff. To show distinct results, we assume the network partition (10m) between home ad hoc network and foreign



(a) disruption time (b) throughput

Fig. 5. Disruption Time due and Throughput at handoff

ad hoc network. Because the move detection methods provided by MIP highly affects the throughput. we compare the proposed RREP-based approach with the two basic solutions proposed in MIP.

- Lazy Cell Switching(LCS) is that a node registers its location with home agent as soon as it detects the unreachability to foreign agent. Other method proposed in MIP is Eager Cell Switching (ECS).
- ECS is that it assumes movement along a straight line and detects its movement as soon as a new agent advertisement.

The disruption time of LCS sharply rises as the beacon interval increases because of its waiting period. For example, an agent advertisement beacon period of 5 seconds results in an agent advertisement lifetime of 15 seconds. In worst case a node would wait 15 seconds. The increment of ECS is slower than that of LCS because it detects its movement on receiving a new agent advertisement. In the contrast, the proposed approach is fixed low disruption time because it is independent of the beacon interval. Fig 5(b) demonstrates the UDP throughput at receiver side for each method. Throughput is calculated such that, it is the number of the successfully received packet within the period, which starts when a source opens a communication, and which ends when the simulation stops. For both ECS and LCS method, While the movement detection time increases as the beacon interval increases, their performance is degraded as the beacon interval increases. However, the proposed approach is not depends on the beacon interval and its movement detection is determined by the routing update delay of AODV.

5 Conclusion

MIP and AODV protocols in a MANET can work together to support Internet mobility. In this paper, we propose dynamic agent advertisement of MIP to provide a good connectivity between ad hoc networks and global Internet while keeping flooding overhead costs low. Regarding MIP in mobile ad hoc networks, our proposal includes a new movement detection scheme using routing protocol. The proposed approach can support faster handoff than the movement detection of MIP because it detects its movement as soon as the route destined to the destination is updated. For network overhead, because the dynamic agent advertisement limit the periodical advertisement to one hop, it archives provide internet connectivity with lower burden than the overhead of the advertisement based approach. However, the proposed approach has some limitations when all nodes want to connect Internet or when the route of a node is changed frequently. This limitations is caused by the overhead per RREP because the frequent route change increases the number of RREP. Through the simulation results, we can see that the proposed approach achieve the best throughput for all case by using optimized route and eliminating the spurious packet transmissions. We believe that the work presented here is an important step towards supporting the connectivity between ad hoc networks and Internet.

Acknowledgements

This research was supported by University IT Research Center Project

References

[1] C. Perkins, "IP Mobility Support", RFC 3344 in IETF, August 2002
[2] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561 in IETF, July 2003
[3] P. Ratanchandani, R. Kravets, "A Hybrid Approach to Internet Connectivity for Mobile Ad Hoc Networks", IEEE Wireless Communications and Networking Conference(WCNC) 2003, 2003
[4] ns2 home pages, `http://www.isi.edu/nsnam/ns`
[5] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C Hu, and J. Jetcheva. "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols", Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM'98), pages 85-87, October 1998
[6] U. Jonsson, F. Alriksson, T. Larsson, P. Johansson, and G. Q. Maquire Jr., "MIP-MANET - Mobile IP for Ad Hoc Networks", Proceedings of the 1st Workshop on Mobile Ad hoc Network and Computing(MobiHOC'00), Boston, Massachusetts, August 2000.
[7] Y. Sun, E. M. Belding-Royer, and C. E. Perkins, "Internet connectivity for ad hoc mobile networks", International Journal of Wireless Information Networks special issue on Mobile Ad Hoc Networks(MANETs): Standards, Research, Applications, 2002.

A Transport Layer Mobility Support Mechanism

Moonjeong Chang[1], Meejeong Lee[1], and Seokjoo Koh[2]

[1] Dept. of Computer Engineering, Ewha Womans University, Seoul 121-791, Korea
{mjchang,lmj}@ewha.ac.kr
[2] Protocol Engineering Center, ETRI, Daejeon 305-350, Korea
sjkoh@etri.re.kr

Abstract. Recently, mobile SCTP (mSCTP) has been proposed as a transport layer approach for supporting mobility. mSCTP is based on the 'multi-homing' feature of Stream Control Transmission Protocol (SCTP), and utilize the functions to dynamically add or delete IP addresses of end points to or from the existing connection in order to support mobility. In this paper, we propose a mechanism to determine when to add or delete an IP address, utilizing the link layer radio signal strength information in order to enhance the performance of mSCTP. We also propose a mechanism for a mobile node to initiate the change of data delivery path based on link layer radio signal strength information. The simulation results show that the performance of proposed transport layer mobility support mechanism is competitive compared to the traditional network layer mobility supporting approach. Especially, when the moving speed of mobile node is fast, it shows better performance than the traditional network layer approach.

1 Introduction

For the next-generation Internet, one of the essential requirements of the users is being connected to the network while roaming. Mobile IP is the proposed standard of IETF for supporting mobility based on IP [1]. Various protocols to enhance the performance of Mobile IP have also been proposed [1, 2, 3, 4, 5, 6]. These protocols, including Mobile IP, take a common stance in the sense that they all deal with mobility at the network layer. If mobility is handled at the network layer, transport connections may remain transparent to the user movement. Mobility support at the network layer requires special entities such as HA (Home Agent) and FA (Foreign Agent) to be deployed in the network, and this involves overhead and inefficiency such as tunneling and/or triangle routing [1, 2, 3, 4, 5, 6].

Recently, mobility support at the transport layer protocol has been discussed in the specification of some of the newly emerging transport protocols such as Stream Control Transmission Protocol (SCTP) or Datagram Congestion Control Protocol (DCCP) [7, 8]. Especially with the SCTP, an extension named mobile SCTP (mSCTP), which facilitates mobility has been drafted in [9].

SCTP is a new IETF standard track general purpose transport protocol for the Internet. Similar to TCP, SCTP provides a connection oriented reliable service, and a connection between two SCTP endpoints is called as an association.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 287–296, 2004.

One of the major features that SCTP provides is multi-homing. Multi-homing allows an endpoint of an SCTP association to be mapped to multiple IP-addresses. Among those addresses, one is chosen as the 'primary path' and is used as the destination for normal transmission. The other addresses are used for retransmissions only. A sender may change the primary path if the number of successive retransmissions in the current primary path is over a certain threshold. New primary path is randomly selected among the available active IP addresses mapped to the receiving end of SCTP association.

Multi-homing feature of SCTP provides a basis for mobility support since it allows a mobile node (MN) to add a new IP address, while holding the old IP address already assigned to itself. On top of SCTP multi-homing feature, mSCTP utilizes ADDIP and DELETEIP functions which enables dynamically adding and deleting an IP addresses to and from the list of association end points in the middle of association [10]. If an MN obtains a new IP address when it moves into a new subnet, the mSCTP at MN sends out ADDIP message to inform the mSCTP at correspondent node (CN) of the new IP address to be added to the list of end point addresses for the association. mSCTP at MN also informs the mSCTP at CN to delete the IP address of previous subnet from the address list by sending out DELETEIP message. The SCTP association, therefore, can continue data transmission to a moved new location without aid from the network layer.

In the current specification of mSCTP is, though, at a very primitive stage, and it mearly illustrates the basic requirements and suggestions to utilize ADDIP and DELETEIP to support session mobility. Some essential issues, such as when and by which criteria the primary path to be changed or the addition and deletion of the IP addresses mapped to the SCTP association should occur in order to deal with handover seamlessly, are yet left for future elaboration. Without these issues being defined, the current mSCTP cannot practically handle mobility. For example, without appropriate mechanism to determine when to change and how to select the primary path, a serious oscillation problem, which may degrade the performance to the minimum, could occur during handover. In this paper, we identify these loosely defined or missing aspects of the current mSCTP definition and propose a transport layer mobility supporting scheme which addresses all of those aspects. Through extensive simulations, the proposed transport layer mobility supporting scheme is tested and the performance is compared with the traditional TCP over Mobile IP.

The rest of this paper is organized in the following way. Section 2 gives a detailed explanation on the operation of proposed scheme. Simulations and its numerical results are presented in section 3. Finally, section 4 concludes the paper.

2 An mSCTP Enhancement Scheme

When MN moves into a new subnet, layer 2 (L2) handover and new IP address acquisition should happen. The proposed scheme assumes that L2 handover and

the acquisition of a new IP address is achieved in the same as they are done with Mobile IP. For IPv4, it is assumed that the new IP address is obtained by DHCP (Dynamic Host Configuration Protocol) or CCoA (Co-located Care-Of Address) is deployed; for IPv6, the new IP address is assumed to be obtained by Stateless Address Auto configuration [11]. Typically, IP address acquisition starts after L2 handover in the new subnet is completed. In [11, 12], it is proposed to proceed IP address acquisition and L2 handover simultaneously in order to reduce the handover latency. The proposed scheme can work with either of these two cases.

When handover happens, mSCTP at MN should perform ADDIP for the new IP address and DELETEIP for the old one. In the proposed scheme, mSCTP at MN performs ADDIP as soon as the signal strength of the new access router exceeds the signal strength threshold value that enables communications (hereinafter, it is called L2 handover threshold). Once an IP address is added, DELETEIP for that address is not triggered until the signal strength from the corresponding access router becomes lower than the L2 handover threshold. With these policies, an SCTP association of the proposed scheme maintains the MN's IP addresses corresponding to all of the accessible subnets, and furthermore an accessible IP address is added to the SCTP association as early as possible. The main purpose of these policies regarding adding or deleting end point IP addresses is to maximize the chance that an end point IP address is ready when it is needed for handover.

When handover happens, primary path also needs to be changed. The current mSCTP does not specifically mention about how to change the primary path for handovers. If the way that SCTP uses to change the primary path is adopted in mSCTP, CN should experiences multiple data packet losses for each handover before it finally determines to change the primary path. In order to prevent these losses, the proposed scheme makes the mSCTP at MN to trigger primary path change toward the mSCTP at CN when handover happens. Furthermore, mSCTP at MN triggers this request before DELETEIP for the current primary path occurs in order to avoid a time interval during which no primary path exists for data transmission. Similar to issuing an ADDIP or a DELETEIP, mSCTP at MN uses L2 radio signal strength information for primary path changes. If the radio signal strength of the primary path becomes lower than a certain threshold (hereinafter it is called primary change threshold), primary path is replaced. The threshold value for this purpose is set slightly higher than L2 handover triggering threshold in order to have the primary path change occur before DELETEIP of the primary path. While satisfying this condition, the primary change threshold should be as low as possible in order to reduce primary path changes. In addition, the proposed scheme let mSCTP at MN determine a new primary path utilizing the L2 radio signal strength information of the wireless subnet, and inform it to mSCTP at CN. Among the accessible subnet, the one providing strongest radio signal is selected as the new primary path in order to minimize the possible oscillation.

The functions of the proposed scheme as described above are implemented in a logical block named AMM (Address Management Module). Fig.1 presents

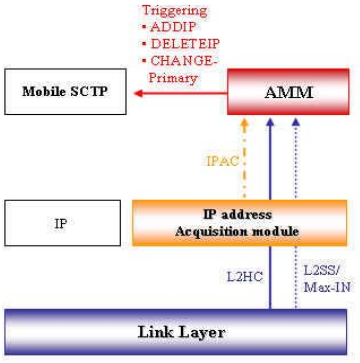


Fig. 1. Signaling in proposed scheme

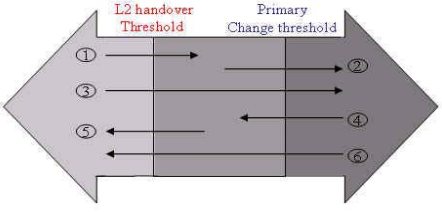


Fig. 2. L2 Signal strength change

the interaction between AMM and the rest of mSCTP, IP address acquisition module, and link layer respectively. Receiving signals from the link layer and the IP address acquisition module, AMM determines when to trigger ADDIP, DELETEIP, and primary path change and informs it to mSCTP. mSCTP at MN then interact with peer mSCTP at CN to change the end point mapping or the primary path for the SCTP association.

Link layer sends out following three types of signals to AMM whenever a corresponding event happens:

1. L2HC (L2 Handover Completion): the L2 handover is completed for the interface specified in the signal.
2. MaxIN (Interface with Maximum signal strength): the interface providing maximum signal strength has been changed to the one specified in the signal.
3. L2SS (L2 Signal Strength): one of the L2 signal strength changes shown in Fig. 2 has occurred for a certain interface; the signal specifies the interface and the types of signal strength change (S) whose value is determined according to Table 1.

IP address acquisition module sends out IPAC (IP address Acquisition Completion) signal when an IP address acquisition for an interface is completed. The IPAC signal indicates the interface ID as well as the acquired IP address.

Table 1. S field value of L2SS

Signal Strength Change	S field of L2SS
1	1
2, 3	2
4	3
5	4
6	5

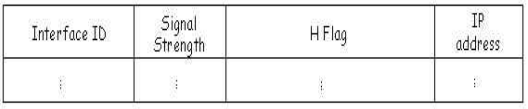


Fig. 3. Address Table in AMM

In order to store the information collected from the signals from the link layer and the IP address acquisition module, AMM maintains an Address Table as shown in Fig. 3. The SS (Signal Strength) field of the Address Table indicates the current signal strength of the interface, and the meaning of the value of this field is shown in Table 2. This field is updated from the S value of L2SS signal. As shown in Fig. 2 and Table 1, S value of L2SS can be used to induce in which state the signal strength of the given interface belongs to. The H flag in the Address Table indicates whether the L2 handover is completed for the corresponding interface. Receiving L2HC signal for a certain interface, H flag of corresponding entry in the Address Table can be set. The IP address field of the Address Table is filled when IPAC signal for the corresponding entry comes in from the IP address acquisition module. In addition to Address Table, AMM also maintains information such as the interface corresponding to the current primary path and the interface with maximum signal strength.

mSCTP at MN starts ADDIP for a certain IP address when both the L2 handover and the IP address acquisition of the corresponding interface are completed. That is, for a certain entry of the Address Table, AMM triggers mSCTP to start ADDIP for the corresponding interface when both the IP address field and the H flag are set upon receiving L2HC or IPAC signals.

When the received L2SS is for the current primary path interface and its S field is 3 or 5, AMM should trigger mSCTP to start primary path change. Before this triggering, AMM checks whether there is an alternative interface ready to be used as the new primary path. If one is found, it immediately triggers mSCTP to start the replacing primary path with that interface. In order for an interface to be a primary path interface, it should satisfy the following three conditions:

1. It is the interface with maximum signal strength and the signal strength is greater than the 'primary change threshold'. Note that there could be a case that even the interface with maximum signal strength may not provide the signal strength higher than the primary change threshold.

Table 2. The vaule of SS field in the Address Table

SS	Signal Strength ρ
0	$\rho <$ L2 handover threshold
1	L2 handover threshold $< \rho <$ Primary change threshold
2	Primary change threshold $< \rho$

2. Link layer handover is completed.
3. IP address acquisition is completed.

If there is no such interface, AMM just updates the SS field of the Address Table to be 0 or 1 depending on the value of S field in L2SS, and postpones triggering the primary change. Afterwards, as L2SS, L3HC, or IPAC signals are received at AMM, an interface satisfying all three of the above conditions could show up. When SS=0 or 1 for the primary path interface, AMM triggers mSCTP to start the primary path change as soon as an interface satisfying all three conditions of the primary path interface shows up.

If AMM receives an L2SS signal with S=4 or 5 for a certain interface, AMM triggers mSCTP to start DELETEIP for that interface. Before this triggering, AMM checks whether the interface corresponds to the current primary path. If it is, an alternative interface to serve as the primary path should be searched. If there is no interface ready to replace the primary path, DELETEIP triggering should be postponed. In this case, whenever primary path change can be triggered afterwards, DELETEIP for the current primary path interface should be triggered together.

The primary path change and DELETEIP are triggered together if primary path change happens after the MN completely moves out of the cell overlapping area. In this case, the acknowledgements for the outstanding packets that are transmitted through the previous primary path may not come back to CN, resulting in the disruption of steady arrival of acknowledgements. If the arrival of acknowledgements is disrupted, not only that the fast retransmit cannot be applied but also the opening of the receiving window is not informed to the sending side. Furthermore both flow and error control may erroneously trigger timeouts as well as window reduction. In order to avoid the performance degradation cause by these, we make the mSCTP at MN to inform this to situation the mSCTP at CN by setting one of the flags in the SCTP ASCONF chunk which encapsulates the primary path change and DELETEIP requests [10]. Receiving the ASCONF chunk with this particular flag set, the mSCTP at CN transmits a probe packet to MN. According to the SACK for the probe packet, mSCTP at CN immediately starts the retransmission of gaps without waiting for further acknowledgements. When CN starts transmitting data to the new primary path, the data transmission window is controlled by slow start in all cases.

3 Simulation

In this section, we present the simulation model and compare the performance of proposed scheme and TCP over Mobile IP (hereafter, TOM) through the numerical results of the simulation. For the performance comparison purpose, TOM is specifically chosen since it is the representative data transmission framework based on network layer mobility support. The comparison of proposed scheme to the original mSCTP is not performed since plain mSCTP cannot cope with the mobility on its own due to the reasons explained in the introduction.

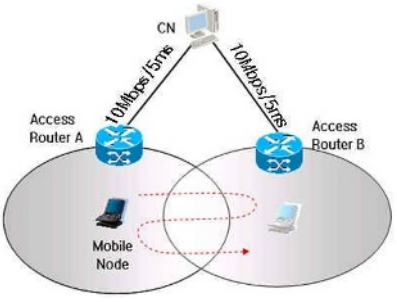


Fig. 4. Simulation Network Model

3.1 Simulation Model

The simulation was implemented using ns-2 simulator proposed by U.C. Berkeley. For the proposed scheme, the ns-2 SCTP node module implemented in [13] is patched. The simulation was run on RedHat Linux 7.3 with the v2.4.18 kernel.

Fig. 4 shows the network model used in our simulations. The wireless channel is assumed to be 802.11b WLAN with 2Mbps capacity and negligible propagation delay. All of the wired links are assumed to have 10Mbps link capacity with 5ms of propagation delay. The coverage radius of each wireless cell is assumed to be 300 meters, and the distance between two neighboring cells is 520 meters. Therefore, the longest distance across the overlapping area is 80 meters. In order to take account for the impact of handover to the performance, we made MN move between two access routers in turn at constant moving speed during the whole simulation time.

As for the performance metric, the elapsed time for MN to download the 140Mbytes of file form the CN is measured, and it is denoted as file transfer time. Handover latency, which is defined as the length of time interval between the instance receiving the last packet from the old path and the instance receiving the first packet from the new path when handover happens, is also measured. The performance of proposed scheme and TOM are measured with these two performance metrics for various moving speed of MN and the path acquisition time. The path acquisition time is defined as the time to complete both the L2 handover and the IP address acquisition for a wireless subnet.

3.2 Simulation Results

Fig. 5 and 6 show the file transfer time and the handover latency respectively for changing path acquisition time. The moving speed of mobile node is set to 30m/sec for this experiment. If the path acquisition time is very short (when it is 1 second in our experiment), the performance of TOM is better than the proposed scheme.

In this case, both TOM and mSCTP can start transmitting data to the new path while MN is transiting the cell overlapping area, and the chance for MN

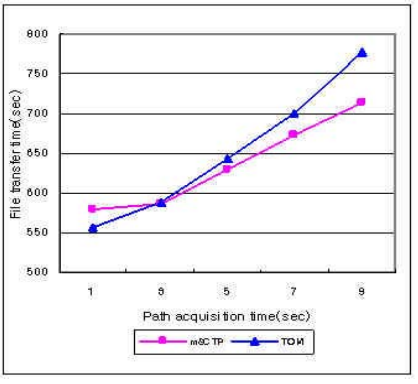


Fig. 5. Signaling in proposed scheme

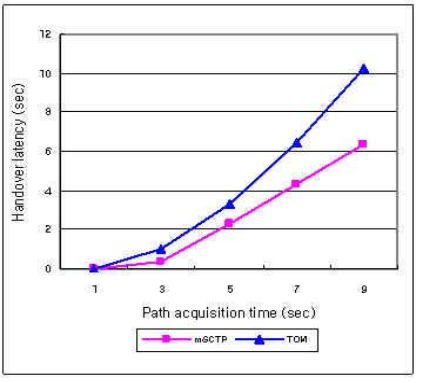


Fig. 6. L2 Signal strength change

to successfully receive all the data transmitted through the old path before it leaves the overlapping area is very high. That is, the impact of handover is minimal in this case. Under these circumstances, the performance of proposed scheme is worse than TOM due to SCTP's higher header overhead as well as the impact of slow start used in mSCTP. Note TCP in TOM is not aware of the handover, and maintains the congestion window size of the previous path, which is higher than the initial window size of the slow start in most of the cases, when it starts transmitting on the new path. On the other hand, mSCTP always starts transmitting to the new path with the initial window size of the slow start. As the path acquisition time becomes longer, the time to start transmitting data through new path is delayed and as a result the amount of data, which are transmitted through the old path and not being able to be delivered to MN while it is transiting the overlapping area, increases. That is, amount of losses caused by handover increases. Moreover, changing the data delivery path may not even happen while MN is transiting the overlapping area if the path acquisition time becomes larger than the MN's overlapping area transiting time. The amount of losses caused by handover grows even larger in this case. Since TCP in TOM is not aware of handover, it reduces the transmission window if handover causes packet losses. Furthermore, if timeout occurs due to the losses during handover, transmission through the new path may not start even after handover is completed due to the retransmission timeout interval. On the other hand, the proposed mSCTP enhancement schemes makes mSCTP to start transmitting data to the new path as soon the handover is completed. Therefore, the proposed mSCTP enhancement scheme always shows smaller handover latency as presented in Fig. 6. Mainly due to the impact of handover latency, it also shows shorter file transfer time than TOM when path acquisition time is larger than 3 seconds as presented in Fig. 5.

Fig. 7 and 8 show the handover latency and the file transfer time respectively for different moving speed of MN. For this experiment, both the parallel and the sequential approaches, in terms of processing the L2 handover and the IP

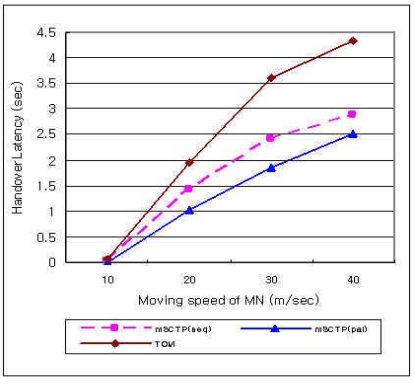


Fig. 7. Signaling in proposed scheme

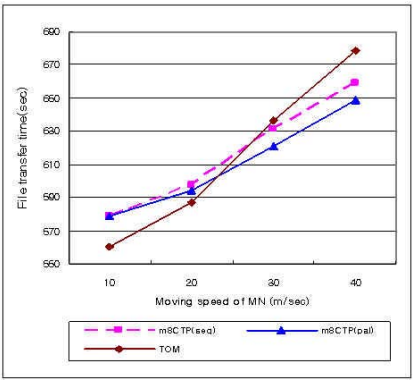


Fig. 8. L2 Signal strength change

address acquisition, are applied for the proposed scheme in order to investigate the impact of path acquisition time to the performance of the proposed scheme. These two different cases are specified as mSCTP(seq) and mSCTP(pal) in the figures.

As the moving speed of MN becomes faster, handover latency increases in both the proposed scheme and TOM. If two MNs with different moving speed start transiting the cell overlapping area at the same time, the faster MN should escape from the overlapping area earlier, i.e., the faster MN stops receiving packets from the previous path earlier. Since the path acquisition time is not affected by the moving speed of MN, the time to start receiving packets through the new path is almost the same regardless of the moving speed. Therefore, handover latency, which is defined as the length of time interval between the instance receiving a packet from the old path for the last time and the instance receiving a packet from the new path for the first time, becomes larger as the moving speed becomes faster. Due to the reason explained for Fig. 6, the proposed scheme always has shorter handover latency than TOM for all moving speeds.

In Fig. 8, it is shown that the proposed scheme outperforms TOM when the moving speed of MN is over 20m/sec. As clearly illustrated in Fig. 5, the relative performance gain of proposed scheme compared to TOM, with respect to the file transfer time, becomes greater as the ratio of path acquisition time to cell overlapping area transiting time becomes larger. Since the cell overlapping area transiting time becomes smaller as the moving speed becomes faster, the proposed scheme shows better performance than TOM when the moving speed of MN is relatively faster.

With respect to both the handover latency and the file transfer time, the proposed scheme performs relatively better when the L2 handover and the IP address acquisition proceed in parallel. This result is symmetric to the performance comparison between TOM and fast handover [11].

4 Conclusion

Recently, mSCTP has been proposed as a transport layer approach to support mobility. We propose an enhancement scheme of mSCTP, which utilizes the link layer radio signal strength information and specifically addresses the following aspects:

- Adding or deleting IP addresses for handover
- Initiating the change of data delivery path from MN in case of handovers
- Selecting a new primary path by MN
- Reducing handover latency by explicit signaling and probing at the transport layer

The simulation results show that the proposed scheme is very competitive compared to the traditional network layer mobility support mechanism. Expecially, when the moving speed of mobile node is fast, it shows better performance.

References

[1] C.Perkins: IP Mobility Support for IPv4, RFC3344 (2002)
[2] A. Campbell, J.Gomez, S. Kim, Z. Turanyi, C-Y Wan, A. Valko: Comparision of IP Micro-Mobility Protocols, IEEE Wireless Communication Magazine (2002)
[3] C. Perkins:Mobile IP Regional Registration", Internet-Draft <draft-ietf-mobile-ip-reg-tunnel-04.txt> (2001)
[4] R.Ramjee, et al.: IP-Based Access Network Infrastructure for Next-Generation Wireless Data Network: HAWAII, IEEE Personal Communications (2000)
[5] A. T. Campbell, et al.: Design, Implementation, and Evaluation of Cellular IP, IEEE Personal Communications (2000)
[6] A. Misra, et al.: IDMP-Based Fast Handoffs and Paging in IP-Based 4G Mobile Networks", IEEE Communications Magazine (2002)
[7] R.Stewart, Q. xie, et al.: Stream Control Transmission Protocol, RFC 2960 (2000)
[8] E. Kohler, M. Handley, S.Floyd, and J. Padhye: Datagram Congestion Control (DCCP), Internet-Draft < draft-ietf-dccp-spec-04 > (2003)
[9] M. Riegel, M. Tuxen: Mobile SCTP, Internet-Draft, < draft-reigel-tuexen-mobile-sctp-03> (2003)
[10] R. Stewart: Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration, Internet-Draft, < draft-ietf-tsvwg-addip-sctp-08.txt> (2003)
[11] N. Montavont, T. Noel: Handover Management for Mobile Nodes in IPv6 Networks, IEEE Communication Magazine (2002)
[12] A. Singh, et al.: Fast handoff L2 trigger API, Internet-Draft < draft-signh-l2trigger-api-00> (2002)
[13] `http://pel.cis.udel.edu/download`

Secured Anonymous ID Assignment Support for LIN6

Masahiro Ishiyama[1], Mitsunobu Kunishi[2], Michimune Kohno[3], and Fumio Teraoka[4]

[1] Communication Platform Laboratory
Corporate R&D Center, Toshiba Corporation
masahiro@isl.rdc.toshiba.co.jp
[2] Graduate School of Science and Technology, Keio University
kunishi@tokoro-lab.org
[3] Sony Computer Science Laboratories, Inc.
mkohno@csl.sony.co.jp
[4] Graduate School of Science and Technology, Keio University
tera@tera.ics.keio.ac.jp

Abstract. Although mobility support protocols such as Mobile IPv6 and LIN6 are essential for a real mobile computing environment, there is an privacy issue: these protocols have to disclose an identity of the node to receive the benefit of mobility support. In this paper, we attempt to address this issue by assigning an identity to a mobile node dynamically and securely without disclosing the statically-assigned ID of the node in the LIN6 protocol. In our method, a mobile node generates an ephemeral public/private key pair and decides a LIN6 ID that is given by a hash of the public key. This LIN6 ID is called "anonymized LIN6 ID". Then the mobile node requests to assign this ID dynamically to the Mapping Agent that maintains location information of the ID. The Mapping Agent issues a shared secret for updating the location information to the mobile node by using the public key. A mobile node can discard the ID or request a new ID whenever the node wants, thus it is hard to track the mobile node with the anonymized LIN6 ID. We also discuss the characteristics of anonymity and the potential of DoS attack in our proposed method.

1 Introduction

Mobile computing is becoming more common with advances in the performance of PDAs and the growing popularity of mobile access devices. A great number of people access the Internet with their mobile terminals nowadays. Demands for a protocol that supports node mobility are emerging. To support communication with mobile nodes, especially for the Internet Protocol Version 6, Mobile IPv6[1] and LIN6[2] have been proposed. Since these protocols provide mobility in Layer 3 of the OSI model, there is no impact on the existing IPv6 applications, and expectations for various mobile-computing applications are mounting. On the other hand, there is increasing concern about privacy issues. Current mobility protocols such as Mobile IPv6 cannot provide anonymous communication

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 297–306, 2004.

because they require an identifier that is defined in the protocol, e.g. Home Address, and have to disclose an identity to the correspondent node in order to receive the benefit of mobility support. However there is a growing demand for the ability to hide one's identity from a correspondent node, especially in the personal communication area (e.g. telephones).

This paper discusses a method that provides anonymity to LIN6 without losing support for transparent mobility. In the proposed method, a mobile node can use randomly chosen LIN6 ID as its ephemeral LIN6 ID, and the validity of using the ephemeral LIN6 ID can be verified without disclosing any other identities of the mobile node that is using the ephemeral LIN6 ID.

2 Overview of LIN6

In this section, we give an overview of the LIN6 protocol. See [2] for details. LIN6 is a new protocol that supports mobility for IPv6, which is based on a Location Independent Network Architecture (LINA) [2] and employs separation of identifier and locator to support node mobility. In LIN6, correspondent node is identified by its globally unique identifier, which is called LIN6 ID, on upper layers such as transport or application layer, and locator is used to route a packet to the network interface in network layer.

2.1 Addressing Model

In LIN6, all mobile nodes are assigned a 64-bit global unique node identifier that is called *LIN6 ID* and are identified by this LIN6 ID. LIN6 introduces two types of network addresses: *LIN6 address* as a locator and *LIN6 generalized ID* as an identifier, which are based on *Embedded Addressing Model* proposed by LINA. In Embedded Addressing Model, a node can embed the identifier in the locator. In LIN6, a LIN6 address is used to deliver a packet and LIN6 Generalized ID is used to identify a correspondent node. On both address formats, LIN6 ID is used for the lower 64 bits. On the other hand, the upper 64 bits of a LIN6 address is a network prefix which represents a current location of the node, and the upper 64 bits of a LIN6 Generalized ID is the *LIN6 prefix*.The LIN6 prefix is a predefined fixed value and it is expected to be well known to all LIN6 nodes in advance. These two address formats are compatible with the Aggregatable Global Unicast Address (AGUA) [3] that is the current IPv6 address format, and thus LIN6 can use the existing IPv6 infrastructure.

2.2 LIN6 Communication Model

This section describes the overview of LIN6 communication model, referring to Fig. 2. A mobile node registers its LIN6 ID and current location to an agent that manages location information of mobile nodes. The association of a LIN6 ID of the mobile node with the current location of the mobile node is called *mapping*, and the agent is called Mapping Agent (MA) (Fig. 2 (0)). An MA maintains

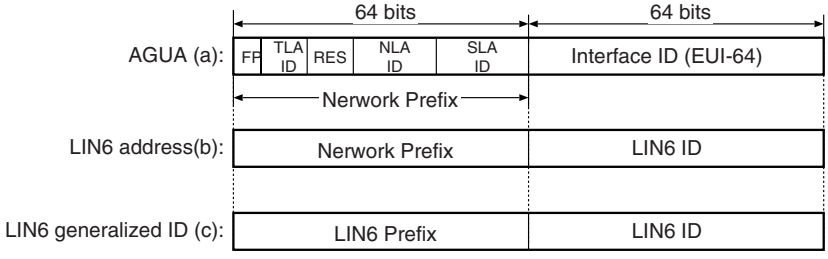


Fig. 1. The format of Aggregatable Global Unicast Address, LIN6 address and LIN6 Generalized ID: In AGUA, the upper 64 bits represent the location of a subnetwork and the lower 64 bits represent the identifier of an interface, not of the node. In LIN6 address, although the upper 64 bits are the same as for AGUA, the lower 64 bits represent the node identifier, LIN6 ID. In Generalized LIN6 ID, the upper 64 bits are *LIN6 prefix* which is well-known, predefined fixed value

a mapping, and *Designated Mapping Agents* that are the Mapping Agents that maintain the mapping of a particular node identifier. That is, "Designated Mapping Agents of the node A signifies those Mapping Agents maintain the mapping of node A. A mobile node registers a new mapping with its designated MA when the node changes its location on the network. MA1 replies with mapping when it is requested to provide a mapping for a mobile node that it maintains. When a correspondent node (CN) starts to communicate with a mobile node (MN), CN sends a mapping request message to the designated MA of MN. The designated MA responds by providing a mapping (Fig. 2 (1,2)). Now CN can send a packet to MN because CN knows the LIN6 address, i.e. current location, of MN (Fig. 2 (3)). When MN responds to CN, MN obtains the mapping of CN in a similar way, and then MN can send a packet to CN. A mobile node identifies the correspondent node by LIN6 generalized ID in upper layers and uses LIN6 address for packet delivery. This separation guarantees a transparent mobility on the Network Layer.

2.3 Finding Designated Mapping Agents

To find designated Mapping Agents of a particular mobile node, LIN6 uses the Domain Name System (DNS). LIN6 makes use of DNS to maintain the relation between the mobile node and its Mapping Agents. A new DNS record MA is introduced to register the address of the dedicated Mapping Agents of the mobile node with the DNS database. A correspondent node can acquire the IPv6 addresses of designated MAs for the mobile node by querying DNS, specifying the LIN6 node's LIN6 generalized ID as the key, as in the case of reverse lookup. For example, to find designated MAs of a mobile node that has a LIN6 ID: `0001:4afe:dcba:9876`, a node asks for a type MA resource record for the domain name `6.7.8.9.a.b.c.d.e.f.a.4.1.0.0.0.lin6.net`.

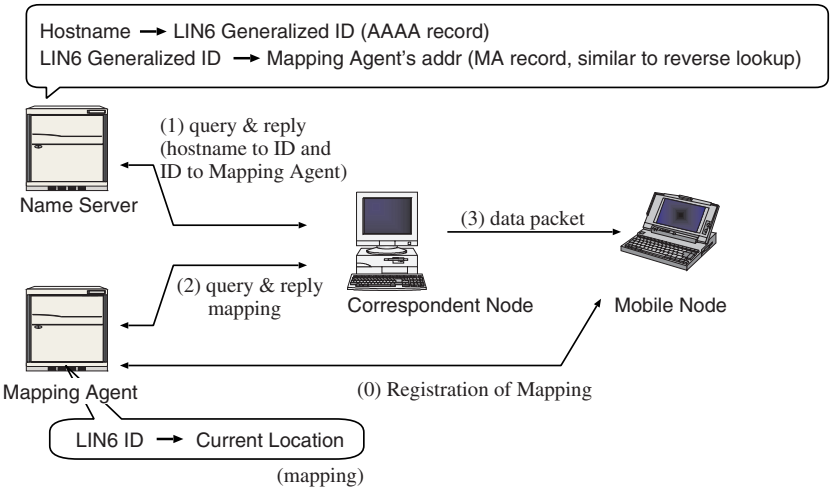


Fig. 2. LIN6 communication model: Information of a location of a mobile node (MN), called mapping, is maintained by a Mapping Agent (MA). Mobile nodes register its mapping to a designated MA of the MN. Correspondent node requests a mapping from the designated MA of the MN

3 Proposed Method

In addition to the LIN6 protocol, we propose a method that enables a "caller" node to communicate with a correspondent node without disclosing a LIN6 ID that identifies the mobile node and without losing a transparent mobility feature. In this paper, we call a mobile node that sends the first packet in order to start communicating with another node the *caller node*. The opposite node is called the *callee node*.

In this case, the caller node first resolves a mapping of the callee node by using the LIN6 ID of the callee node. Then the callee node resolves a mapping of caller node when the callee node sends a packet to the caller node by using the LIN6 ID that is known from the packet that the caller node has sent. From this point of view, we can find the difference of the role of LIN6 ID between caller and callee nodes. That is, the LIN6 ID of the callee node represents the identifier of the node with which the caller node intends to communicate for caller, but the LIN6 ID of caller node represents the identifier of the node to which the caller node has to return packets for the callee node. In consideration of this difference, we propose a method that enables a mobile node to use two different types of LIN6 ID. The current LIN6 protocol assumes that LIN6 ID is assigned to the node statically by an authority. Our proposed method introduces a new type of LIN6 ID that can be dynamically assigned to a mobile node in addition to the statically-assigned LIN6 ID. There is no guarantee of the permanent one-to-one relationship between a node and a LIN6 ID when a LIN6 ID is dynamically assigned. That is, it is possible that a particular LIN6 ID represents a different

mobile node in a finite time interval. This characteristic can provide anonymity to mobile nodes in LIN6 because a node cannot assume that it is the same node that claims a LIN6 ID which is the same as the previous correspondent node claimed. In other words, a caller node can start an anonymous communication by using this dynamically allocated LIN6 ID. A caller node that wants to start a new communication chooses a LIN6 ID randomly first and requests allocation of the LIN6 ID, and then starts a new communication with the randomly-chosen LIN6 ID. When the mobile node closes the communication, the mobile node may request release of the LIN6 ID in order to terminate the association of the mobile node with the LIN6 ID. We call this dynamically assigned LIN6 ID that is randomly chosen by a mobile node itself *anonymized LIN6 ID*.

We have to consider the following two issues in assigning anonymized LIN6 IDs to mobile nodes.

- Authentication to use an anonymized LIN6 ID
- Finding a designated Mapping Agent of the anonymized LIN6 ID

We discuss those issues using the following notation: ID_x is a LIN6 ID of x, (PK_x, SK_x) is (public, private) key pair of x, H(x) is a hash of x, Sig_x is a signature using key SK_x. Ni is a caller mobile node and Nr is a callee mobile node. We also assume that Ni knows the LIN6 ID ID_{Nr} of Nr.

3.1 Authentication of Anonymized LIN6 ID at the Mapping Registration

Since anonymized LIN6 ID does not have a permanent association with a particular mobile node, we cannot expect the mutual trust between a mobile node and the designated MA of the anonymized LIN6 ID. Thus we cannot use the existing authentication mechanism of existing LIN6 protocol. However, if there are no authentication mechanisms on the mapping registration, an attacker can hijack any communication that is using anonymized LIN6 ID by sending an illegitimate mapping. For example, when Ni communicates with a correspondent node using anonymized LIN6 ID ID_{rnd}, an attacker can send a mapping registration that includes ID_{rnd} and current location of the attacker. Then all the traffic will go to the attacker after the correspondent node refreshes the mapping of ID_{rnd}. Consequently, we need a new authentication mechanism between a mobile node and a Mapping Agent in order to use anonymized LIN6 ID securely.

3.2 A New Registration Method for Anonymized LIN6 ID

We describe the registration procedure of anonymized LIN6 ID, referring to Fig. 3.

We assume that there is a Certificate Authority (CA) that authorizes placing a Mapping Agent and this CA has a public/private key pair (PK_w, SK_w) and PK_w is well-known. We also assume that every Mapping Agent has its own

public/private key pair that is signed by the CA and all of the LIN6 nodes know PK_w, the public key of the CA.

A mobile node Ni that is going to start an anonymous communication generates its "ephemeral" public/private key pair (PK_a, SK_a). An anonymized LIN6 ID ID_a of the mobile node is $H(PK_a)$ (Fig. 3 (1)). Then Ni finds an address of a designated Mapping Agent of ID_a, which is denoted by MA_a. (Fig. 3 (2)). We will discuss the method to find a designated Mapping Agent of an anonymized LIN6 ID in section 3.3. At this point, Ni knows the address of MA_a and sends an allocation request message to MA_a (Fig. 3 (3)) as follows:
[ID_a, PK_a, T_{Ni}, {ID_a, PK_a, T_{Ni}}Sig_a]
T_{Ni} is a timestamp of Ni.

MA_a verifies {ID_a, PK_a, T_{Ni}}Sig_a and verifies that $ID_a = H(PK_a)$. If they are correct, MA_a checks whether ID_a is available or not (Fig. 3 (4))). If it is available, MA_a generates a symmetric key K for mapping registration of ID_a, and MA_a responds by providing an allocation succeeded message (Fig. 3 (5,6)) as follows:
[{K}PK_a, T_{Ni}, L, {{K}PK_a, T_{Ni}, L}Sig_m, PK_m, {PK_m}Sig_w]
PK_m is the public key of MA_a, {PK_m}Sig_w is the signature which is issued by the Certificate Authority, and L is the term of validity of shared secret K.

We assume that all of the LIN6 nodes know PK_w, and thus Ni can verify {PK_m}Sig_w. This signature authorizes PK_m, and so Ni can verify {{K}PK_a, T_{Ni}, L}Sig_m. Then Ni obtains shared secret K by decrypting {K}PK_a. At this point, Ni can use ID_a and send a registration request message that includes message authentication codes using K (Fig. 3 (8)).

When the term of validity of K expires, Ni forfeits the privilege of using ID_a. However, Ni may request MA_a to extend the time by using K before the expiration. In addition, even after the expiration, Ni may request the same anonymized LIN6 ID ID_a whenever Ni needs as long as Ni keeps the public/private key pair. Note that it is difficult enough for the other nodes to override ID_a even after the expiration because a node that want to override ID_a has to find a public/private key pair (PK_o, SK_o) that satisfies $H(PK_o) = ID_a$.

3.3 Finding a Designated Mapping Agent

The other issue of the dynamic allocation of LIN6 ID is how to find the MAs. Current LIN6 protocol uses existing DNS mechanism to find designated MAs. In the case of statically-assigned LIN6 ID, MAs can be assigned with the assignment of LIN6 ID. That is, when allocation of a new LIN6 ID is requested, at that point, MAs can be selected and can be assigned to the requested LIN6 ID. With this method, it is appropriate that a new DNS record is added when a new LIN6 ID is requested. However, with dynamically assigned LIN6 ID, it is difficult to use the existing method because we cannot predict the LIN6 ID that will be requested. In this case, we have to assign all the IDs that may possibly have been requested to MAs previously. On the other hand, it is efficient to use small number of MAs when the number of users of this anonymized LIN6 ID is small, and increase MAs with the growth of users. However, if MAs are previously assigned, it is

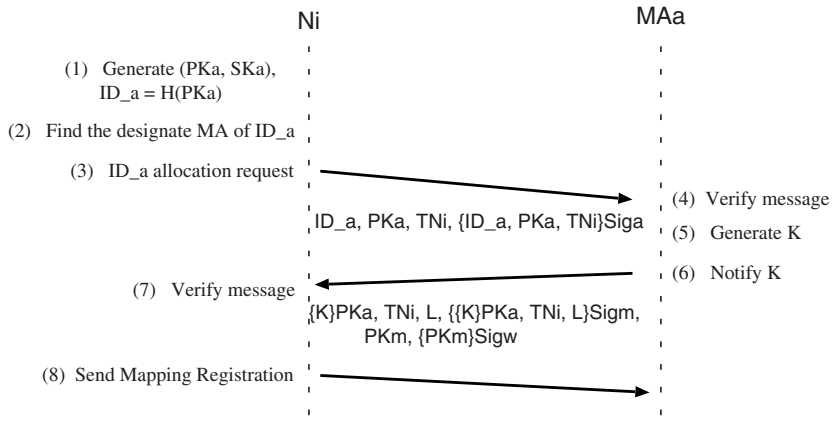


Fig. 3. Registration procedure of anonymized LIN6 ID: A mobile node Ni that is going to start an anonymous communication generates its "ephemeral" public/private key pair (PK_a, SK_a) and determines anonymized LIN6 ID which is given by $ID_a = H(PK_a)$. Then Ni sends it to MA_a with PK_a. MA_a determines shared secret K and sends {K}PK(a) to Ni. The validity of MA_a is verified by the signature of CA, which is done by a well-known public key

complicated to increase MAs because there are a huge number of records which must be updated.

We can use Wildcard Resource Records (Wildcard RRs) of DNS[4] to resolve this issue. Wildcard RRs provide a way to synthesize multiple resource records and enable coverage of all possible names that are the same except for the subdomains by a single resource record. For example, `*.0.lin6.net` will apply to domain names `1.0.lin6.net`, `2.1.0.lin6.net`, etc. Thus we can bind multiple LIN6 IDs to a mapping agent without defining a huge number of resource records and can remove a complication from maintaining a DNS database. However, several issues remains. Because Wildcard RRs can be applied for each label, they can only synthesize a subtree. It imposes a restriction on allocating MAs. In addition, the management of DNS is basically not easy. For example, in [5] the authors report that about 36% of DNS traffic consists of errors such as lookup packets with no answer or packets that indicate an error condition, and these errors are mainly caused by a configuration error of DNS. This implies that there are difficulties in managing DNS servers to keep integrity with a large number of domain names.

On the other hand, we can use another mechanism to find designated MAs, rather than using existing DNS infrastructure. For example, we can introduce an infrastructure which is based on Distributed Hash Table (DHT) for finding designated MAs. DHT is a technology based on a hash table that enables the table to be spread across many nodes. Several implementations of DHT-based systems have been proposed such as Chord[6], CAN[7] and Tapestry[8]. A DHT mechanism is very suitable for our proposed method because an anonymized

LIN6 ID itself is a hash value, and so we can use a DHT-based system directly for finding designated MAs of a particular anonymized LIN6 ID.

4 Consideration

4.1 Characteristics of Anonymity in the Proposed Method

In our proposed method, a node can use statically-assigned LIN6 IDs and anonymized LIN6 IDs simultaneously. This guarantees that the node can keep or start communications using a statically-assigned LIN6 ID while it is communicating with other nodes using anonymized LIN6 ID and vice versa. On the other hand, when a node N uses a statically-assigned LIN6 ID ID_s, and if the node uses an anonymized LIN6 ID ID_a simultaneously and there is a node E that can tap all the traffic of N, E can derive that these two nodes, ID_s and ID_a, are on the same subnetwork from source or destination addresses in packets but cannot derive that ID_s and ID_a are the same node. However, when N moves, E can derive that ID_s and ID_a are the same node from the moving pattern of N. In addition, Let us consider if N and E are connected to the same link. In this case, there is a possibility that E can derive a relationship between ID_s and ID_a from the link-layer address of N even if N does not move. However many different attacks are possible when the nodes are on the same link, and so this issue is beyond the scope of this paper.

Similarly, let us consider that N does not use statically-assigned LIN6 ID but it uses anonymized LIN6 ID ID_{a1}. When N stops using ID_{a1} and begins to use a new anonymized LIN6 ID ID_{a2}, E can only see that a node using ID_{a1} stops its communication, and a new node using ID_{a2} appears on the same network. However, if E has statistical data, we cannot deny the possibility that E can find a correlation among ID_{a1}, ID_{a2} and another ID that N used.

However, these issues arise only when the attacker can eavesdrop on all the communications of the target node and it is basically difficult. Thus, these issues do not constitute a fundamental problem for our proposed method.

4.2 Selection of ID

When a node initiates a communication, it can choose which ID it will use and a user may add usage preferences. For example, a user may want to use a statically-assigned LIN6 ID when a node connects to the enterprise server to which the user belongs and may want to use an anonymized LIN6 ID to communicate with another servers. This selection can be done in IPv6 stack basically in a similar way to the IPv6 source address selection[9], and so modifications to applications are also unnecessary.

However, there are several applications in which a correspondent node is requested to reveal an IPv6 address that the node uses. FTP[10] is an example. In this case, the application has to respond by providing an appropriate address. That is, when the application opens a session using an anonymized LIN6 ID

and it is requested to reveal the current IPv6 address the node is using, it should provide the anonymized LIN6 ID that the node used to open the session, or another anonymized LIN6 ID at least, but it may not return a statically-assigned LIN6 ID. This sort of application will need modifications to support such ID selection feature.

4.3 Denial of Service Attack

Protecting against Denial of Service Attack (DoS) is becoming more important on today's Internet. In this section, we focus on the possibility of a DoS attack on the parts that our proposed method adds to the current LIN6 protocol, especially from the aspect of computational power exhaustion attack because our method uses an asymmetric cryptosystem.

For a mobile node, no serious DoS issues arise due to the additional features. Although a node has to verify signatures when it receives a response from a designated MA (Fig. 3 (6)), a node can discard while it is not requesting a new LIN6 ID. The time slot for an attacker is very short and the attacker has to know the anonymized LIN6 ID that the target node requests.

On the other hand, a Mapping Agent has to verify a signature whenever it receives an allocation request message (Fig. 3 (3)). Thus an attacker can send a large number of bogus allocation request messages and this has a potential for exhausting the MA's computational resources. To protect against this attack, we have to control a rate of verification of allocation request messages. To resolve this issue, "client puzzles"[11] can be applied. This method forces each client to solve a cryptographic puzzle for each service request, and the server provides a service only if the client solved the puzzle. The puzzle costs the client a lot of computation, although the verification of the answer costs very small computation. The puzzle can be difficult enough because a request for an allocation of anonymized LIN6 ID is done before a node starts a communication and the overhead of this allocation procedure does not affect the performance of the communication itself. That is, the overhead of this allocation procedure affects neither throughput of data transfer nor delay. Thus, the client puzzles method is suitable for protecting MAs against this kind of DoS attacks.

5 Conclusion

This paper proposed an additional method that provides anonymity support to the LIN6 protocol without losing support for transparent mobility. A mobile node generates an ephemeral public/private key pair and uses the hash value of the public key as "anonymized" LIN6 ID. Our proposed method provides an authentication scheme that protects against impersonation and the mobile node does not need to disclose any other identities. We also discussed characteristics of anonymity in our method and the possibility of a DoS attack. We showed the client puzzles method is suitable for protecting against a DoS attack on Mapping Agents.

Among several candidate topics for future work are the prototype implementation of this framework and performance evaluation on real IPv6 networks.

References

[1] Johnson, D. B., Perkins, C., Arkko, J.: Mobility Support in IPv6. (2003) Internet-draft. 297
[2] Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H., Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks. IEICE Transactions on Communications **E84-B** (2001) 297, 298
[3] Hinden, R., O'Dell, M., Deering, S.: An IPv6 Aggregatable Global Unicast Address Format. (1998) RFC 2374. 298
[4] Mockapetris, P.: Domain names - concepts and facilities. (1987) RFC 1034. 303
[5] Balakrishnan, H., Jung, J., Sit, E., Morris, R.: Dns performance and the effectiveness of caching. In: Proceedings of the ACM SIGCOMM Internet Measurement Workshop '01, ACM (2001) 303
[6] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceeding of ACM SIGCOMM, ACM (2001) 303
[7] Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. In: Proceedings of ACM SIGCOMM. (2001) 303
[8] Zhao, B. Y., Kubiatowicz, J., Joseph, A. D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. In: Technical Report UCB CSD 01-1141, University of California at Berkeley, Computer Science Department. (2001) 303
[9] Draves, R.: Default Address Selection for Internet Protocol version 6 (IPv6). (2003) RFC 3077. 304
[10] Postel, J., Reynolds, J.: File Transfer Protocol. (1985) RFC 959. 304
[11] Juels, A., Brainard, J.: Client puzzles: A cryptographic defense against connection depletion attacks. In: Proceedings of Network and Distributed System Security (NDSS '99). (1999) 151–165 305

Distributed Collision-Free/Collision-Controlled MAC Protocols for Mobile Ad Hoc Networks with Hidden Terminals

Chi-Hsiang Yeh

Dept. of Electrical and Computer Engineering, Queen's University
Kingston, Ontario K7L 3N6, Canada
chi-hsiang.yeh@ece.queensu.edu

Abstract. In this paper, we introduce the *multiple access collision prevention (MACP)* scheme for MAC-layer collision control without relying on busy tone. MACP is the first and only distributed MAC scheme that can achieve collision freedom for both control messages and data packets, even in the presence of hidden terminals without relying on spread spectrum with extremely large spreading factors. The proposed position-based prohibiting mechanism is particularly suitable for future high-speed networks with non-negligible propagation delay, and can mitigate the additive prohibiting signal problem without unnecessary idle time as previous approaches based on backoff.

1 Introduction

Although the MAC protocols of IEEE 802.11 and IEEE 802.11e [2] work well in single-hop WLANs, several problems will arise when they are applied to ad hoc networks. In particular, high collision rate constitutes a major issue in mobile ad hoc networks, which will degrade the throughput, delay, QoS and fairness capability, and increase energy consumption. Moreover, repeated collisions lead to higher delay and packet dropping, which may prevent some TCP-based and/or real-time applications from working properly.

Dual busy tone multiple access (DBTMA) [1] is the only previous MAC protocol proposed in the literature thus far that can achieve 100% collision-free transmissions for data packets (under certain mathematical model assumptions). However, its control messages may be collided and will thus still cause unbounded delay for repeatedly colliding nodes. Also, DBTMA requires busy tone that consume extra energy and dual transceivers per node that may lead to higher hardware cost.

In this paper, we propose a new paradigm, called the *multiple access collision prevention (MACP)* scheme, for medium access control in ad hoc networks and single-hop/multihop wireless LANs without relying on busy tone. Appropriately designed MACP protocols, including the MACP/NCK protocol, can achieve 100% collision-free control/data packet transmissions in a fully distributed manner. MACP only requires one transceiver, but dual transceivers per

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 307–319, 2004.

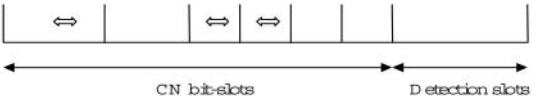


Fig. 1. A competiton round for MACP/NCK. The CN is 101100 based on the n-choose-k codes where $n = 6$ and $k = 3$

node can improve the performance by reducing the durations of prohibiting slots enhancing the effectiveness for competition, as well as transmitting data packets in dual/multiple channels concurrently if so desired.

2 The MACP with n-choose-k (MACP/NCK) Protocol

In MACP protocols, we incorporate *distributed multihop binary countdown* [7] before transmitting control messages that require collision freedom or collision control. In single-channel MACP, prohibiting slots (e.g., CN bit-slots and detection slots), control messages, and data packets are mixed together in the same physical channel. If all nodes are synchronized to the same types of slots, then it will not cause problems. However, if the network is asynchronous [7], then certain accompanying mechanisms are required to avoid collisions and/or interference between different types of signals. Spread spectrum techniques such as the *spread spectrum scheduling* S^3 *scheme* [6] are possible solutions that can reduce the power levels for sending the prohibiting and detection signals (and/or control messages) so that data receptions protected by CTS messages will not collided by them. Other techniques are possible and will be reported in the future. In particular, group competition (see Section 3) may be employed to partition nodes (or transmissions of nodes with specified power levels) into appropriate groups so that nodes/transmissions of the same group can avoid causing such interference/collision problems. In the rest of the paper, we only describe the competition procedure for *separate-channel synchronous MACP* for simplicity, where all participating nodes are synchronized and start the competition round at the same time, and the prohibiting signals, control messages, and data packets are transmitted in different physical channels.

In *separate-channel synchronous MACP/NCK*, the simplest version of MACP/NCK, an intended transmitter (for control messages or small data packets) uses a binary number that have exactly k 1-bits and $n - k$ 0-bits as its *competition number (CN)*. Typically k can be selected as $\lfloor n/2 \rfloor$ or $\lceil n/2 \rceil$. An example for CNs based on the 5-choose-3 coding is provided in Fig. 1a. During prohibiting bit-slot i, $i = 1, 2, ..., n$, the intended transmitter that has value 1 for its i-th bit transmits a short *prohibiting signal* at power level sufficiently high to be detected by (most/all) other nodes within its prohibitive range. We define the *prohibiting threshold* as the signal strength required for the received strength to be detected and recognized as a prohibiting signal. Note that there is a lower bound on the prohibiting threshold for all transmissions in any nodes, but the prohibiting threshold can be adjusted when interference control (see Section 4.3)

is employed. On the other hand, a node whose i-th bit is 0 keeps silent and senses whether there is any prohibiting signal that has strength above its prohibiting threshold during bit-slot i. If the silent competing node finds that bit-slot i is not idle (i.e., there is at least one nearby competitor whose i-th bit is 1), then it loses the competition and keeps silent until the end of the current round of competition. Otherwise, it survives and remains in the competition. If a node survives all the n prohibiting bit-slots, it becomes a candidate for the winner within its prohibitive range.

All active nodes that require to receive RTS/CTS control messages but are not in the competition can serve as *mutually hidden terminals detectors* to eliminate mutually hidden candidates that will transmit collided control messages (and/or small data packets) to them. A hidden terminal detector listens to the channel to determine whether the prohibiting signal strength received during each bit-slot is above the *control coverage threshold* and the *control-to-control (C2C) interference threshold*, where the control coverage threshold is the minimum signal strength required for a control message to be received successfully, and the C2C interference threshold is the minimum signal strength required for a control message to be collided by the signal. The hidden terminal detector counts the number of bit-slots with received strength above the control coverage threshold during the current competition round. If the number is at least k, then the node becomes a *valid mutually hidden terminals detector*. It also counts the number of bit-slots with received strength above the C2C interference threshold during the current competition round, including the bit-slots with received strength above the control coverage threshold. If a valid mutually hidden terminals detector hears more than k such bit-slots, then there are mutually hidden nodes involved in the competition and the candidate(s) whose coverage range(s) cover the valid mutually hidden terminals detector must be one of them. Even though other mutually hidden node(s) might have lost the competition, the valid mutually hidden terminals detector will send an *objecting-to-send (OTS)* [6] short signal during the following *mutually hidden terminals detection slot* to block such candidate(s) from transmitting their control messages (or small data packets). The candidate(s) then has to backoff before participating in competition again. The contention windows for such candidates are exponentially increased whenever they are blocked by OTS short signals, but will be reduced the minimum value or a normal value [4] when the transmission is successful. If a candidate winner does not receive any OTS short signals, then it becomes a winner and will be eligible to transmit its control message (or small data packet).

When the competition numbers are unique, there can only be at most one winner within the prohibitive range of the winner. As a result, the control message to be transmitted will not be collided. Since all control messages can be received by all active nodes without collisions, all schedules are known by nearby nodes so that no one will request for conflicting schedules. Hence, collision freedom can be achieved in MACP/NCK (when transmission errors due to unreliable wireless channels are negligible). Note that to enforce such collision-free

property, a node lost a competition has to remain silent and observe the control messages for a sufficiently long time according to the *observe before transmit discipline* [6]. However, when collision-freedom is not necessary, this requirement can be relaxed in MACP/NCK. Note also that the requirement for CN format can also be relaxed by choosing k 1-bits from $n' \leq n$ positions in CNs only (e.g., the last n' bit positions).

Note that different thresholds should be used in the preceding procedure for correctness and efficiency of the protocol. Note also that the duration for bit-slots should be selected to be sufficiently large so that multipath signals and echoes will not cause mistakes in such detecting and counting procedures. Moreover, the durations for bit-slots can be larger than that for the prohibiting signals, especially for the first few bit-slots. A competitor with CN bit value equal to 1 for the corresponding bit-slot will then randomly select an appropriate time instant within the bit-slot to start its prohibiting signal transmission. This can mitigate the *additive prohibiting signal exposed terminal (APSET) problem* that may block far-away nodes unnecessarily when the density of competitors is high. If the short signals are transmitted with spectrum spectrum, the APSET problem can be further mitigated. An appropriate flow control mechanism should also be employed to reduce the number of competitors at the first place in order to reduce the number of concurrent competitors at the first place, reducing the required durations for such bit-slots and thus competition overhead. Similarly, the duration for the mutually hidden terminals detection slot should be even larger, especially in a dense network, since many nearby valid mutually hidden terminals detectors may decide to transmit their detection signal during the same slot.

In addition to the n-choose-k codes used in MACP/NCK, we can employ other codes for CNs to obtain different classes of MACP protocols. In particular, binary *additive error detectable codes (AEDC)* [7] are binary codewords that are guaranteed to be changed to a non-codeword as long as any 0-bit is changed to 1, given that none of the 1-bits are changed to 0. n-choose-k codes are a special class of AEDC, while *binary 0-count mapping (BZM)* represents a general scheme that can convert *any* binary codes into AEDC (see Fig. 1b for an example and [7] for more details). Various other codes may also be used to achieve respective advantages. For example, a binary code can be extended with a CRC code, another type of error detection codes, or a short n-choose-k code. The resultant protocols are not collision-free, but can reduce the length of CNs to reduce the associated control overhead. We can also cascade several codes of the same and/or different types to gain their combined strengths. For example, a CN can start with a PP-slot (see Subsection 4.1) for mitigating APSET, followed by a binary code for higher efficiency in competition, and then ended with a short NCK code plus a HTD slot for further competition and to prevent the rare occurrence of mutually hidden winners. The MACP protocols resulting from these codes are usually similar to MACP/NCK, except that the required hidden terminal detection mechanisms may need to be modified. Some examples can be found in [7]. As another example, a declaration slot or encoded collision detection

slot can be added for candidate(s) to transmit a single short declaration signal. If a valid mutual hidden terminal detector detects multiple declaration signals that are not multipath signals or echoes, it will send an OTS signal to prevent control message collisions at its location. When CNs are not guaranteed to be unique, the PP mechanism (see Subsection 4.1) can be employed to improve the performance. Also, if a valid mutually hidden terminals detector can recognize the ID of the candidate to be blocked, it can send coded OTS signal to block that candidate alone if such a capability is supported. Another approach, to be referred to as the *parrot approach*, is to have active nodes or mutual hidden terminal detectors repeat the prohibiting signals they received or send a short signal at the end of competition with the CN they appear to have heard. Some of these subclasses of MACP will be investigated in details in the future.

3 Extensible MACP with Group Competition

In the *group competition mechanism* [7], only nodes (or transmissions of nodes with specified power levels) belonging to the same level-3 group are allowed to compete at the same time. Nodes belonging to the same level-2 group use the same activation mechanism to invoke other nodes, while nodes (or transmissions of nodes with specified power levels) belonging to the same level-1 group use the same group CN for competition and are encouraged to compete at the same time. Note that a node may have membership in multiple groups at the same level. Note also that level-1 groups consist of members that can transmit their control messages or data packets at the same time without causing collisions. The term "group competition" mainly refers to the fact that members belonging to the same level-1 group can concurrently participate in competition with the same group CN. As a result, "groups" are referring to level-1 groups in this paper if the level is not explicitly specified. Other hierarchical structures and grouping strategies are possible for group competition, but are outside the scope of the paper.

There are a number of advantages for employing the group competition mechanism. First, the typical number of winners within a typical prohibitive range will be significantly increased (e.g., considerably greater than 1). As a result, the overhead for competition is now shared by many winners and can thus be considerably reduced per winner. Second, many level-1 group members within a typical prohibitive range may win the competition and become eligible for transmissions of control messages at the same time. As a result, the spacial reuse can be considerably improved for the transmissions of RTS/CTS control messages, significantly reducing the control overhead. Third, when many level-1 group members are competing at the same time, the chance of having hidden terminals will be significantly reduced. As a result, in E-MACP/GC, the HTD mechanism becomes optional, which can reduce the competition overhead and considerably simplify the design of asynchronous E-MACP/GC protocols. Fourth, by appropriately adjusting the prohibitive ranges, both the hidden and exposed terminal problems can be resolved or at least mitigated without

having to rely on RTS/CTS dialogues. This can considerably reduce the control overhead, especially when data packets are not large. Fifth, many level-1 group members within a typical prohibitive range may win the competition and become eligible for transmissions of data packets at the same time (when Data-ACK two-way handshaking is employed), solving or at least mitigating the exposed terminal problem of CSMA/IC [8]. As a result, when RTS/CTS handshaking is is replaced by, or combined with, sensitive CSMA (with lower carrier sensing threshold) [7] or prohibition-based competition as in CSMA/IC [8], the spacial reuse can be significantly improved as compared to sensitive CSMA or CSMA/IC without group competition. Sixth, group competition naturally leads to effective coordination between level-1 group members, which enables effective group scheduling [7] among nodes that are eligible to transmit data packets or control messages at the same time. This further improves the spacial reuse relative to conventional distributed MAC protocols that typically have little or no coordination.

To take advantages of the aforementioned characteristics, we develop *extensible MACP with group competition (E-MACP/GC)*, a collision-controlled MAC protocol where the RTS/CTS dialogues and the hidden terminal detection mechanism are optional even in multihop networking environments. In E-MACP/GC, the lengths of CNs may be adaptive to traffic conditions, the tolerance to collisions for the associated transmissions, and the local history of performance. When the traffic is very light, group competition or even the competition itself can be skipped if so desired, and be reactivated when needed. A node or group can thus start with no or shorter CNs, and increases the CN length when the collision rate is high. The RTS/CTS and/or HTD mechanisms can also be turned on when found necessary (e.g., after a number of collisions or based on other observations/information). Different from IEEE 802.11/11e, however, E-MACP/GC does not suffer from the hidden or exposed terminal problem even when the RTS/CTS and HTD mechanisms are not turned on, as long as level-1 groups are sufficiently dense. More details concerning group competition, the invocation of group competition among members, as well as group scheduling can be found in [7].

4 Accompanying Mechanisms for MACP

4.1 The Position-Based Prohibiting (PP) Mechanism

In BROADEN [9] and the MACP protocols presented in this paper thus far, we utilized binary countdown with the *on/off prohibiting mechanism*. To mitigate the APSET problem, we can insert one or several larger competition slots based on position-based prohibition. We can also design *MACP with position-based prohibition (MACP/PP)* protocols that rely on position-based prohibition for all its competition slots. When the PP-slot is sufficiently large, we can also use a single PP-slot for competition, possibly followed by a yield period (with sensitive carrier sensing) before transmitting a control message or data packet.

The position-based prohibiting mechanism is similar to the on/off prohibiting mechanism except that we use position-based prohibiting slots (PP-slots) with durations larger than bit-slots, and there are competitions for signals within the same PP-slot. More precisely, there is one or multiple PP-slots in a competition round, where the durations typically decrease (or sometimes remain the same). There is a guarding period (lower bounded by the maximum-allowed propagation delay) at the end of a PP-slot (as in a bit-slot) so that the prohibiting signal transmitted in a PP-slot will not be heard with non-negligible strength in the subsequent PP-slots or bit-slots by any nodes in the network. The prohibiting signals can then be transmitted at any desirable position in the remaining part of a PP-slot, according to the viewpoint of the transmitter.

A competitor first decides whether it is preparing to send a prohibiting signal in the following PP-slot. If it does, it selects an appropriate position in the PP-slot either randomly according to an appropriate probability distribution or following certain rules (e.g., according to its priority, urgency, and/or ID). It will then listen to the channel before its time to turn around for transmitting its own prohibiting signal. (As a result, if there is an additional receiver for sensing, the turn-around time can be avoided and considerably increasing the efficiency of MACP/PP.) If it did not hear anything above the appropriate prohibiting threshold, it will transmit a short prohibiting signal at the selected position according to its own clock and viewpoint of the competition frame; however, if it detects any prohibiting signals before the selected position, it loses the competition and will wait for the next competition round or back off for a longer time. A competitor that survives all the prohibiting slots become a candidate, and will become a winner eligible transmitting a control message (or small data packet) if it does not receive any OTS short signals from valid mutual hidden terminal detectors.

When the PP-slot(s) is/are followed by NCK bit-slots the hidden terminal detection mechanism for NCK can be applied to the NCK part. For MACP/PP, we can augment several special NCK bit-slots, called *HTD-code bit-slots*, for the purpose of hidden terminal detection. Such HTD-code bit-slots may be considerably smaller than typical PP-slots or bit-slots since the former do not need to be lower bounded by the maximum-allowed propagation delay. Instead, as long as the prohibiting signal can decay below the threshold for HTD in the subsequent HTD-code bit-slots, the duration is acceptable. HTD-code bit-slots based on AEDC can also be used alone as a means for *wireless collision detection*. Some special requirements inclue that the total duration for these bit-slots should be lower bounded by twice the maximum-allowed propagation delay (plus some adidtional time). Also, at least one of the first few bits and one of the last few bits must be equal to 1. The approach based on an additional declaration slot or coded collision detection slot (see the end of Section 2) may also be employed for hidden terminal detection. Note that in MACP/PP, the declaration slot will also employ the PP mechanism for competition, in addition to the purpose of mutually hidden winners detection.

Note that we can use backoff control (similar to IEEE 802.11/11e) as a means to conduct flow control in order to mitigate the APSET problem. However, when backoff control is the only mechanism to reduce the attempt rate, radio resources cannot be efficiently utilized because the radio channel will typically stay idle for a non-negligible portion of time. However, in MACP/PP or other MACP protocols augmented with position-based prohibition, backoff control can be employed to reduce the typical number of competitors to a constant number considerably greater than 1, and then use the first or first few PP-slots to eliminate most of the competitors. This way the APSET problem can be resolved without noticeable idle times for the ratio channel, considerably increasing the radio efficiency.

4.2 Fairness and Prioritization in MACP

We develop several strategies that can improve fairness and prevent starvation in MACP networks based on its strong differentiation capability. A directly applicable approach is to allow nodes or packets that have been treated unfairly to climb up one or a couple of priority levels when desired, or to use a more favorable probability distribution to select the random number part of CNs [4, 5]. To assess the unfairness or the urgency, we can either exchange the performance information locally with nearby nodes, or use one or the composite measure of several performance metrics, such as delay, queue length, granted bandwidth, discarding ratio, blocking rates, the status of last attempt, service quality, and the number of trials, collisions, or failed transmissions. We then calculate the *urgency index* and the *efficiency index*, and determine the CN in combination with other parameters such as priority by either mapping to the *urgency part* of CNs (with one to typically several bits) or to choose an appropriate probability distribution to randomly select CNs. By appropriately combining the delay or countdown time with the location information, we can in fact generate unique CNs without having to rely on other ID assignment mechanisms. We can also set the *continuation bit* [7] to 1 when a node lost a competition (possibly under with certain accompanying conditions) as a simple mechanism to achieve fairness, especially under the single-hop environments. More details will be reported in the near future.

To enhance fairness and QoS in an efficient and economic manner, we propose to utilize the *multiple ID scheme (MIS)* that assigns multiple IDs to a node. The IDs for a node are typically spread all over the possible domain so that there will not be nodes that only have smaller IDs and suffer from unfairness or even starvation. As a result, MIS naturally solves the inherent unfairness problem of binary countdown [3] and CSMA/IC [8]. To support prioritization in the proposed MIS approach, a node simply uses larger IDs for higher-priority transmissions, and smaller IDs for lower-priority transmissions. Similarly, to support adaptive fairness [5] in MIS, a node can choose to use a relatively large ID after it has been treated unfairly, or randomly select a smaller ID as a courtesy to yield to other nodes when it has been well treated. In MACP, several additional bits typically need to be reserved for IDs just in case some prohibitive ranges become very dense. As a result, there will be many unused IDs that can be well

utilized in MIS without additional competition overhead in typical operating environments, Thus, in contrast to previous approaches [5, 8, 7] that requires additional bits for prioritization and fairness, MIS can support fairness and QoS without increasing CN lengths, leading to economic implementations. Moreover, this scheme will experience smaller collision rate as compared to CSMA/IC [8], PIC [5] and PRIC [5] when there are duplicate IDs, since multiple concurrent competitors that possess the same ID(s) may not choose the same ID at the same time. When a region becomes denser, a node can release some IDs to other newcomers. If variable-length CNs are supported, a node typically owns several ranges of IDs, and can split a range and release part of it to other newcomers. The MIS approach also works well with E-MACP/GC and power control (see Subsection 4.3).

For MACP/PP and MACP protocols augmented with position-based prohibition, the priority can be implied by the probability distribution used to randomly select the positions for prohibiting signals in a way similar to the random selection of the random number part of CNs. However, in MACP/PP, the random selection is redone for every PP-slot, possibly with different probability distribution to optimize the performance.

4.3 Interference/Power-Control in MACP

When power control is employed the coverage ranges for many RTS control messages may be orders of magnitude smaller than the maximum control coverage range, given sufficient density and appropriate power-controlled routing and MAC protocols [6]. When interference control is employed [6], the coverage ranges for CTS control messages associated with low-power transmissions may also be considerably smaller than the maximum control coverage ranges by allowing higher tolerance to interference. However, in naive implementations of MACP, the prohibitive ranges for those control messages can only be slightly smaller than the associated maximum prohibitive ranges, leading to unacceptable overhead in terms of spacial usage and power consumption for competition.

In this paper, we extend the proposed interference control [6] to the transmissions of prohibiting signals. By allowing a higher tolerance threshold for reception of control messages, the prohibitive ranges can be considerably reduced, thus reducing the transmission power for the associated prohibiting signals and allowing more winners to transmit control messages or data packets within the same area. Note that competitors that employ interference control need to adjust their prohibiting thresholds accordingly so that they will not be prohibited unnecessarily by far away nodes that use higher transmission power levels for prohibiting signals. The proposed approach can also be combined with the *differentiated channel discipline* [6] for power control supports to make it even more efficient. When both power control and interference control are employed, the size of prohibitive ranges will change considerably from transmission to transmission. As a result, the optimized lengths for the random parts of CNs (for collision control) or minimum lengths for IDs (for collision freedom) are considerably different for different transmissions. Thus, variable-length CNs [7] and

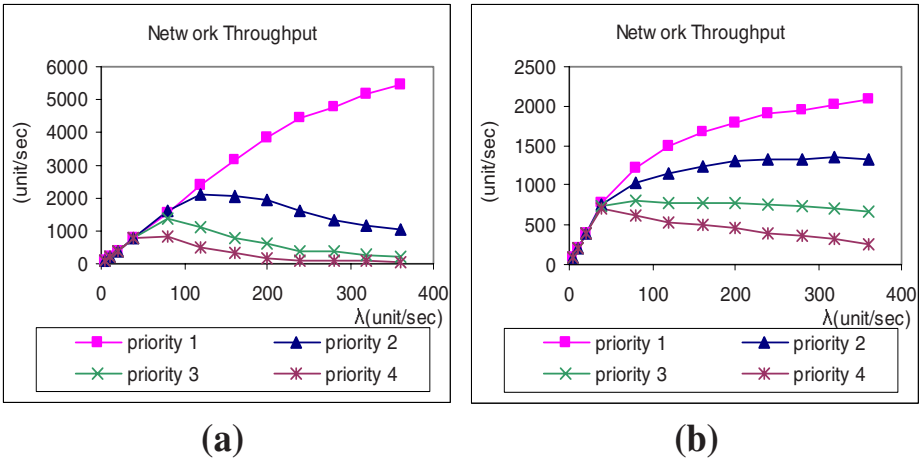


Fig. 2. Throughput differentiation among 4 priority classes in (a) MACP and (b) IEEE 802.11e

variable-length MIS are is particularly suitable to interference/power-controlled MACP.

Other potential mechanisms for MACP and related protocols include *coded interference signaling* and *detached competition*. In coded interference signaling, intermittent prohibiting signals will be recognized as codes to convey important information or instructions when none of the other more efficient approaches (such as spread spectrum) are working or supported. *Periodic prohibiting signaling* can also be used to prohibit other nodes (especially standard IEEE 802.11/11e nodes) from transmitting based on their inherent carrier sensing mechanism. This is particularly efficient when done by an MACP agent near the standard IEEE 802.11/11e node, but is also useful when radio channel characteristics (such as severe multipath effect) prevent correct decoding of signals even when interference of similar levels are not negligible. Detached competition utilizes part of the CNs or coded interference signaling to indicate the specified time for the transmissions of control messages after winning a competition. Detached competition may improve spacial reuse for transmissions of control messages, though the same issue can also be addressed by other techniques such as group competition. More accompanying mechanisms for MACP will be investigated and reported in the near future.

5 Performance Evaluation

In this section, we compare the differentiation capability of IEEE 802.11e and MACP that uses CNs with 2 priority bits followed by a 6-bit random number bits [5].

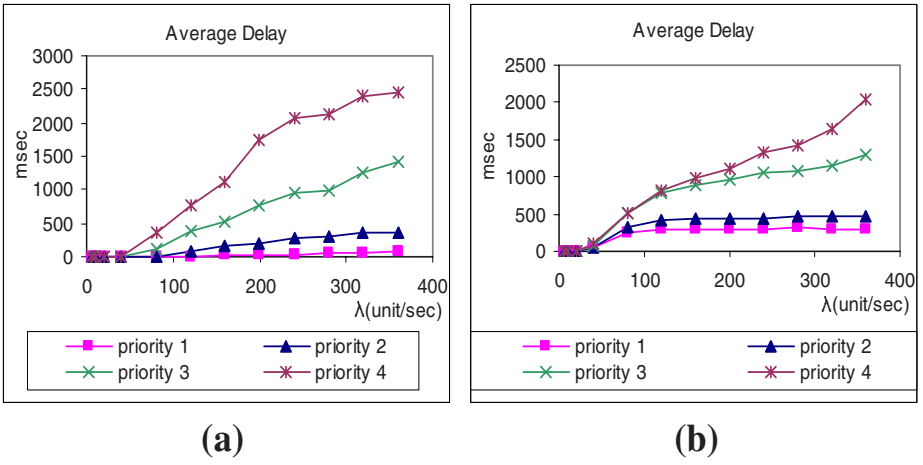


Fig. 3. Delay differentiation among 4 priority classes in (a) MACP and (b) IEEE 802.11e

In our simulation experiments, the total channel capacity is set to 20 Mbps for both protocols in comparison. In MACP, the control channel occupies 4 Mbps, while the data channel occupies 16 Mbps. There are 80 nodes within a 400×400-unit grid area. The transmission radius is set to 100 grid units. The mobility model is random waypoint with 1 unit/sec moving speed and 4 seconds pause time on the average. In the simulations, the length of control messages including RTS and CTS messages is set to 20 Bytes, while data packets are set to be 2000 Bytes. A node in any of these protocols has maximum queue length equal to 10 for class-1 and class-2 packets, maximum queue length equal to 20 for class-3 packets, and maximum queue length equal to 60 for class-4 packets. When a queue is full, new arrivals will be blocked.

From Figs. 2 and 3, we can see that the differentiation capability of MACP is considerably stronger than that of IEEE 802.11e. Moreover, when IEEE 802.11e is employed, the throughput and delays of all the 4 priority classes begin to be differentiated simultaneously when the respective saturation point is reached, and even the highest-priority class suffers from severe degradation in throughput and delay. As a comparison, in MACP, different priority classes are degraded one after another as the network load is increased, rather than simultaneously at the saturation point. In particular, the highest-priority class is hardly degraded in terms of throughput and average delays even when the network load is 3 times higher than that of the saturation point.

6 Conclusions

In this paper, we present a family of MAC protocols, called MACP, based on on/off and/or position-based prohibition, hidden terminal detection, and group

competition. MACP can control collision rate or achieve 100% collision freedom for both control messages and data packets in mobile ad hoc networks and single-hop/multihop wireless LANs.

In [3, p. 261], Tanenbaum states that

> *Binary countdown is an example of a simple, elegant, and efficient protocol that is waiting to be rediscovered.*
>
> *Hopefully, it will find a new home some day.*

We believe that we find one such home for the binary countdown mechanism, where it leads to the first fully-distributed MAC scheme, MACP, that can achieve collision freedom for control messages and data packets in the presence of hidden terminals (without relying on spread spectrum with extrmemly large spreading factors).

Acknowledgements

We would like to acknowledge Tiantong You for conducting simulations and drawing the figures used in this paper. Also, the continuation bit for CNs is a modification and multihop extension of the so-called "loser bit" (for single-hop WLANs) in a course term paper submitted by Andrzej Antoszkiewicz.

References

[1] Haas, Z. J. and J. Deng, "Dual busy tone multiple access (DBTMA)-performance evaluation," *Proc. IEEE Vehicular Technology Conf.*, 1999, pp. 314-319. 307

[2] IEEE 802.11 WG, *Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems LAN/MAN Specific Requirements – Part 11: Wireless Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, IEEE 802.11e/D4.0, 2002. 307

[3] Tanenbaum, A. S., *Computer Networks, 4th Edition,* Prentice Hall, N. J., 2002. 314, 318

[4] Yeh, C.-H., "QoS differentiation mechanisms for heterogeneous wireless networks and the next-generation Internet," *Proc. IEEE Int'l Symp. Computer Communications*, June/July 2003. 309, 314

[5] Yeh, C.-H. and T. You, "A QoS MAC Protocol for Differentiated Service in Mobile Ad Hoc Network," *Proc. Int'l Conf. Parallel Processing,* Oct. 2003, pp. 349-356. 314, 315, 316

[6] Yeh, C.-H., "The heterogeneous hidden/exposed terminal problem for power-controlled ad hoc MAC protocols and its solutions," *Proc. IEEE Vehicular Technology Conf.*, May 2004, to appear. 308, 309, 310, 315

[7] Yeh, C.-H., "A new scheme for effective MAC-layer DiffServ supports in mobile ad hoc networks and multihop wireless LANs," *Proc. IEEE Vehicular Technology Conf.*, May 2004, to appear. 308, 310, 311, 312, 314, 315

[8] You, T., C.-H. Yeh, and H. Hassanein, "CSMA/IC: A new class of collision-free MAC protocols for ad hoc wireless networks," *Proc. IEEE Int'l Symp. Computer Communications,* Jun./Jul. 2003, pp. 843-848. 312, 314, 315

[9] You, T., C.-H. Yeh, and H. Hassanein, "A new class of collision-prevention MAC protocols for ad hoc wireless networks," *Proc. IEEE Int'l Conf. Communications,* May 2003, pp. 1135-1140. 312

Part II

QoS, Measurement and Performance Analysis

Interaction between TCP Reno and TCP Vegas in End-to-End Congestion Control

Aun Haider[1], Harsha Sirisena[2], and Krzysztof Pawlikowski[3]

[1] Department of Electrical and Computer Engineering
University of Canterbury, Christchurch, New Zealand
hai22@student.canterbury.ac.nz
[2] Department of Electrical and Computer Engineering
University of Canterbury, Christchurch, New Zealand
sirisehr@elec.canterbury.ac.nz
[3] Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand
krys@cosc.canterbury.ac.nz

Abstract. This paper addresses incompatibility issues between TCP Reno and TCP Vegas. In order to investigate the bandwidth sharing between both versions of TCP in heterogenous network environments, analytical expressions for throughput and Jain's fairness index are derived. Further, based on the Explicit Congestion Notification (ECN) technique and the Random Early Detection (RED) algorithm, we propose a new algorithm to be incorporated in TCP Vegas for securing its compatibility with TCP Reno. The new form of TCP Vegas is simulated using the discrete event simulator NS-2. Simulation results show that our new algorithm can effectively restrict TCP Reno from unfairly grabbing the bandwidth share of TCP Vegas.

1 Introduction

TCP Reno was first implemented in 1990 as an improved form of TCP Tahoe in 4.3 BSD [1]. Presently, it is the dominant version of TCP in the current Internet and other computer networks, including computational grids; see for example [2], [3] and [4]. Additionally, a significant deployment of TCP New Reno and TCP SACK has been observed recently in the Internet, [5]. However, these new protocols retain the basic end-to-end congestion control algorithms of TCP Reno, hence their interaction with other versions of TCP would be similar to that of TCP Reno.

The congestion control mechanisms of TCP Reno are designed in such a way that the sender decreases its congestion window upon detection of packet losses, when duplicate acknowledgements are received or timeout occurs. Thus, TCP Reno is a typical example of the *reactive* congestion control approach where a sender increases its congestion window until there is a packet loss. Another major problem with TCP Reno is its bias against connections with longer round trip times (R). This bias of TCP Reno has been observed both theoretically and in simulation studies, see [6] and [7].

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 321–331, 2004.

An alternative *proactive* approach to the congestion control problem has been employed in TCP Vegas [8]. In this approach congestion is detected before it actually happens and remedial measures, such as decrease in congestion window, are taken before packets start to get lost. In the end-to-end paradigm the detection of congestion at end hosts can be done by observing a change in the round trip time or a change in the rate of ACK arrivals. TCP Vegas detects the onset of congestion by observing changes in the round trip time. Using this approach TCP Vegas can overcome the problems of TCP Reno in a homogenous environment where all end hosts are using TCP Vegas. It gives a higher throughput and has no bias against connections with longer round trip times; see [9] and [10].

In the case of TCP Reno, packets can experience long queuing delays and jitter if longer queues are formed due to buffers of larger capacity. On the other hand, TCP Vegas maintains a very small backlog of packets in the queue, which results in a short queuing delay, and once TCP Vegas reaches its equilibrium state then the delay jitter also becomes very small.

In heterogenous environments (TCP Reno competing with TCP Vegas) TCP Reno users will normally get higher bandwidth than TCP Vegas users. Over time, the TCP Vegas users can be deprived of their fair share of bandwidth. Therefore the main problem with deploying TCP Vegas is its difficulty in co-existing with the dominant TCP Reno type protocols. This incompatibility issue of TCP Vegas and TCP Reno is addressed in this paper.

The rest of the paper is organized as follows. A review of previous work is given in section 2. The congestion avoidance mechanism of TCP Vegas is summarized in section 3. Analysis of TCP Reno and TCP Vegas interaction is presented in section 4. A new algorithm consisting of TCP Vegas with ECN and the RED algorithm is presented in section 5. The simulation results are presented in section 6. Finally we present our conclusions in section 7.

2 Review of Previous Work

In [9], emulation techniques were used to study the interaction of TCP Vegas and TCP Reno in heterogenous environments. These experiments showed that in head-to-head transfers TCP Reno will get 50 % more throughput than TCP Vegas. This incompatibility issue has been investigated in [10], [11], [12], [13] and [14], using both analytical and simulation techniques. It has been found that TCP Vegas shows better fairness and higher throughput than TCP Reno in homogenous network scenarios, but it cannot perform well in heterogenous scenarios, when competing with TCP Reno.

To improve the performance of TCP Vegas in heterogenous environments comprising of TCP Reno and TCP Vegas, use of the RED routers was suggested in [14]. It was observed through simulations that the relative performance of TCP Vegas against TCP Reno depends upon the values of max_{th} and min_{th} of RED: the throughput of TCP Reno increases and that of TCP Vegas decreases at higher values of thresholds. The idea of using RED algorithm for improving the fairness between TCP Reno and TCP Vegas was also studied in [15], in

which the authors varied the max_{th} parameter of RED, to get more bandwidth for TCP Vegas. The flaw in their scheme is that, as the max_{th} of RED is lowered the link utilization decreases, because then RED drops more packets, and so will reduce the overall throughput. Thus, it is not a very efficient scheme.

In [16], two solutions were presented to make TCP Vegas compatible with TCP Reno. They propose to use TCP Vegas with Stabilized RED (SRED) algorithm routers as well as to modify the basic TCP Vegas algorithm. It is suggested that TCP Vegas should behave similar to TCP Reno. When the former is not getting its fair share of bandwidth, the operation of TCP Vegas is divided into two modes of different aggressiveness: (a) moderate and (b) aggressive mode. Moderate mode is similar to normal TCP Vegas while aggressive mode is similar to TCP Reno. The main weakness of these algorithms is the arbitrary choice of the different parameters. Another solution to the incompatibility issue is to use the Random Exponential Marking (REM) algorithm whose main limitation is the required global knowledge of a tuning parameter.

In order to overcome the unfairness and low throughput characteristics of TCP Vegas while competing with TCP Reno, we propose an ECN-based solution for improving TCP Vegas throughput in section 5.

3 Congestion Avoidance Mechanism of TCP Vegas

The congestion avoidance mechanism of TCP Vegas was proposed in [8]. A TCP Vegas sender computes the minimum round trip time, called the base round trip time R_b, for a given connection during a period of no congestion, which is usually the round trip time of the last packet sent before the router queue builds up. Let the current congestion window of TCP Vegas equal W_v. It is updated by the following congestion avoidance algorithm:

1. Given the base round trip time R_b, compute the expected throughput as $T_{exp} = \frac{W_v}{R_b}$.
2. Given the actual round trip time R, calculate the actual throughput, $T_{act} = \frac{W_v}{R}$.
3. Estimate the backlog D in router buffers as $D = (T_{exp} - T_{act}) \cdot R_b$.
4. Using two constants $\alpha = 1$ and $\beta = 3$, update the congestion window, W_v, by applying the following set of rules: if $D < \alpha$ then $W_v \leftarrow W_v + 1$, if $\alpha \leq D \leq \beta$ then $W_v \leftarrow W_v$ and finally if $D > \beta$ we have $W_v \leftarrow W_v - 1$.

We will be concentrating only on the congestion avoidance phase of TCP Vegas and TCP Reno due to its fundamental role and importance. In the next section we will analyze the interaction between TCP Vegas and TCP Reno.

4 Analysis of TCP Vegas and TCP Reno Interaction

Let us assume a heterogenous network environment where TCP Reno and TCP Vegas are sharing a common bottleneck link of capacity C packets/s, having two

way propagation delay of τ_p and operating in the congestion avoidance phase. A Droptail router having service rate of μ ($\mu \leq C$) packets/s is being used in the bottleneck link. In this section we analyze and derive expressions for the throughputs and fairness between TCP Reno and TCP Vegas.

4.1 Throughputs of TCP Reno and TCP Vegas

Let queue size is $q(t)$ packets and congestion windows of TCP Vegas source and TCP Reno source are $W_v(t)$ and $W_r(t)$ packets, respectively. The based round trip time R_b, and the current round trip time $R(t)$, are given by:

$$R_b = \tau_p + \frac{1}{\mu}, \quad R(t) = \tau_p + \left(\frac{q(t)+1}{\mu}\right). \tag{1}$$

Further, as derived in [17], during the congestion avoidance phase, for the constant random packet loss probability p, the congestion window W_r of TCP Reno is governed by $W_r = \sqrt{\frac{8}{3p}}$. It can be generalized as $W_r(t) = \frac{K_r}{\sqrt{p(t)}}$. Thus, the expression for throughput of TCP Reno, $T_r(t)$, is given by:

$$T_r(t) \equiv \frac{W_r(t)}{R(t)} = \frac{K_r}{\sqrt{p(t)}} \cdot \frac{1}{R(t)} = \frac{K_r}{\sqrt{p(t)}} \cdot \frac{\mu}{\mu \cdot \tau_p + q(t) + 1}. \tag{2}$$

Now, during the congestion avoidance phase of TCP Vegas, we can calculate:

$$W_v(t+1) = \begin{cases} W_v(t)+1 & \text{if } \alpha \cdot \left(\frac{q(t)+\mu\tau_p+1}{q(t)}\right) > W_v(t), \\ W_v(t) & \text{if } \alpha \leq D \leq \beta, \\ W_v(t)-1 & \text{if } \beta \cdot \left(\frac{q(t)+\mu\tau_p+1}{q(t)}\right) < W_v(t), \end{cases} \tag{3}$$

and then throughput of TCP Vegas, $T_v(t)$, can be obtained as $T_v(t) = \frac{W_v(t)}{R(t)}$, where $W_v(t)$ is given by (3) and $R(t)$ by (1). One can see that as we increase the values of α and β in TCP Vegas, the products $\alpha \cdot \left(\frac{q(t)+C\tau_p+1}{q(t)}\right)$ and $\beta \cdot \left(\frac{q(t)+\mu\tau_p+1}{q(t)}\right)$ will become larger in magnitude. This will keep $W_v(t)$ increasing, which will cause an increase in throughput. Thus, in a heterogenous environment, TCP Vegas can also get higher throughput similar to that of TCP Reno, if the values of its α and β parameters are appropriately increased.

4.2 Fairness between TCP Reno and TCP Vegas

In a heterogenous environment packet loss will occur if the overall input traffic rate exceeds the bottleneck link capacity C. Thus, there will be no packet loss observed when $\frac{W_v(t)}{R(t)} + \frac{W_r(t)}{R(t)} \leq C$. At the equilibrium point, we get $W_v(t) = R(t) \cdot C - W_r(t)$ i.e. $W_v(t) = R(t) \cdot C - \sqrt{\frac{K_r}{p(t)}}$, which after substituting (1) can be written as $W_v(t) = q(t) + \tau_p \cdot C + 1 - \frac{K_r}{\sqrt{p(t)}}$. This gives us the expressions for

congestion windows of TCP Vegas and TCP Reno in heterogenous environments. Given TCP Reno's throughput, $T_r(t)$, and TCP Vegas's throughput, $T_v(t)$, we get the Jain's Fairness Index (JFI) as:

$$JFI = \frac{\{T_r(t) + T_v(t)\}^2}{2 \cdot \{T_r(t)^2 + T_v(t)^2\}}. \tag{4}$$

Simplification of equation (4) yields:

$$JFI = 0.5 + \left\{ \frac{W_r(t) \cdot W_v(t)}{W_r(t)^2 + W_v(t)^2} \right\}. \tag{5}$$

Substituting expressions of $W_r(t)$ and $W_v(t)$ into (5), we get:

$$JFI = 0.5 + \left[\frac{\left(\frac{K_r}{\sqrt{p(t)}}\right) \cdot \left(R(t) \cdot C - \frac{K_r}{\sqrt{p(t)}}\right)}{\left(\frac{K_r}{\sqrt{p(t)}}\right)^2 + \left(R(t) \cdot C - \frac{K_r}{\sqrt{p(t)}}\right)^2} \right]. \tag{6}$$

In order to plot variations of JFI with bandwidth-delay product, we substitute $K_r = \sqrt{\frac{8}{3}}$ and assume that $p(t) = p$ and $R(t) = R$ in (6), to get the following expression:

$$JFI = 0.5 + \frac{1.632RCp^{-0.5} - 2.666p^{-1}}{5.333p^{-1} - 3.265RCp^{-0.5} + (RC)^2}. \tag{7}$$

JFI as a function of p and RC, from (7), is shown in Figure 1. We can also have an alternative expression for JFI, in terms W_r and RC, as:

$$JFI = 0.5 + \left\{ \frac{W_r \cdot RC - W_r^2}{2W_r^2 - 2W_r \cdot RC + (RC)^2} \right\}. \tag{8}$$

JFI as a function of W_r and RC is depicted in Figure 2.

5 Improving TCP Vegas/Reno Compatibility with RED Based ECN

In the case of homogenous environments of TCP Vegas connections, after the initial transient period the round trip times of all connections will reach a steady state value. It will not change significantly because TCP Vegas keeps a bounded number of packets in queue buffer oscillating between α and β. Hence if the round trip time of a TCP Vegas connection does not change then we can infer that all other connections are TCP Vegas type. Thus, we will not have any problem with unfairness among different connections.

On the other hand, in a heterogenous scenario of TCP Vegas and TCP Reno connections, TCP Reno will keep on increasing its congestion window

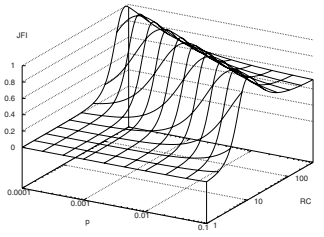


Fig. 1. JFI variation with drop probability p and bandwidth delay product RC (packets)

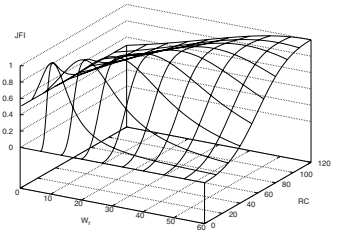


Fig. 2. JFI variation with W_r (packets) and bandwidth delay product RC (packets)

until a packet loss is recorded, thus creating congestion. This causes an increase in queueing delays and results in an increase of round trip time as measured by TCP Vegas. Ultimately it culminates in the reduction of congestion window of TCP Vegas. Thus, in order to avoid the unfair distribution of bandwidth the agressivenss of TCP Vegas should be increased.

We propose to employ a RED router with ECN ([18], [19]), to increase the aggressiveness of TCP Vegas in the above scenario by increasing the values of its α and β parameters. At the onset of congestion due to increased traffic load, the RED router at the bottleneck link will start marking packets before dropping them at the time of severe congestion. Keeping in view of above facts we propose the new ECN-based TCP Vegas/RED algorithm. Let us assume that we deal with TCP Vegas congestion avoidance mechanism operating with a given values of α and β. For example, following [8], let $\alpha = 1$ and $\beta = 3$. Then, our algorithm (named as TCP NVegas) is given by the following four steps:

1. Compute the new round trip time, R_n, after the arrival of each ACK.
2. Compare R_n with the previous round trip time R_o

   and then, after each round trip time interval:

   if $(R_n > R_o)$
   $\{\alpha \leftarrow \alpha + 1;$
   $\beta \leftarrow \beta + 1;\ \}$
   elseif $(R_n \leq R_o)$
   $\{\alpha \leftarrow \alpha + 0;$
   $\beta \leftarrow \beta + 0;\}$

3. If ACK has ECN echo bit high, set $\alpha \leftarrow 1$ and $\beta \leftarrow 3$ (or assume other designated values) and set the congestion window of both TCP Vegas and TCP Reno connections reduced to half of their existing values.
4. Goto step 1 and repeat the above procedure.

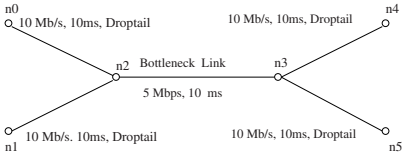


Fig. 3. Topology of the simulated network

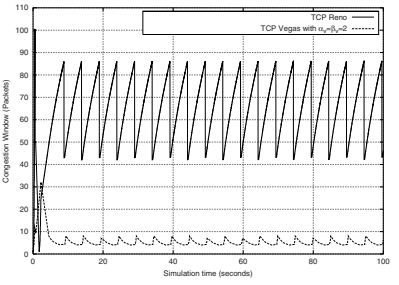


Fig. 4. Congestion windows of TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 2$) connections

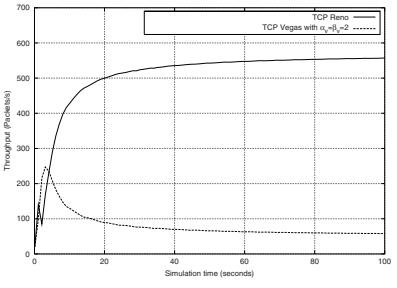


Fig. 5. Throughputs of TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 2$) connections

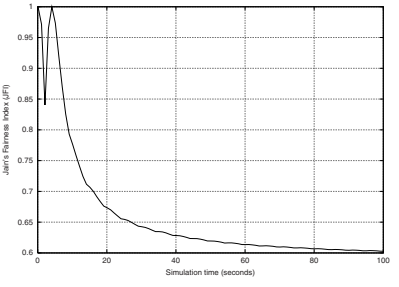


Fig. 6. JFI, TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 2$)

In the next section we report three simulation experiments that confirm our above proposal experimentally.

6 Simulations

We simulated a network shown in Figure 3. The links between nodes n0 to n2, nodes n1 to n2, nodes n3 to n4 and nodes n3 to n5 have capacity of 10 Mbps and propagation delay of 10 ms. TCP Reno send data between nodes n0 and n4 and TCP Vegas or TCP NVegas send data between nodes n1 to n5. The bottleneck link between nodes n2 to n3 had capacity C of 5 Mbps and propagation delay of 10 ms. Droptail and RED routers with ECN were employed at the bottleneck link. In order to analyze the performance of TCP Vegas, TCP NVegas and TCP Reno we use congestion window variations, JFI and throughput as performance measures.

6.1 Experiment 1

In this experiment we increased the values of α and β parameters of TCP Vegas to determine their effects on congestion window, throughput and JFI of TCP

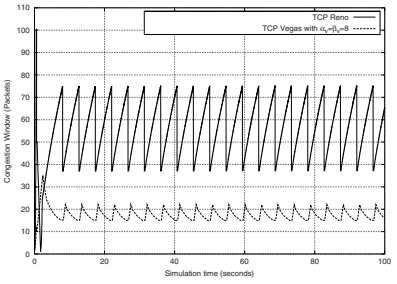


Fig. 7. Congestion windows for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 8$) connections

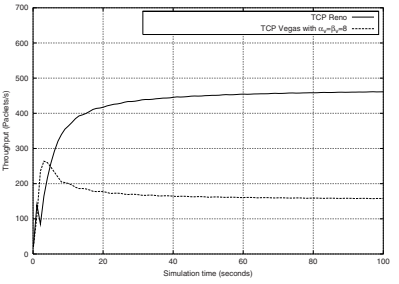


Fig. 8. Throughputs for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 8$) connections

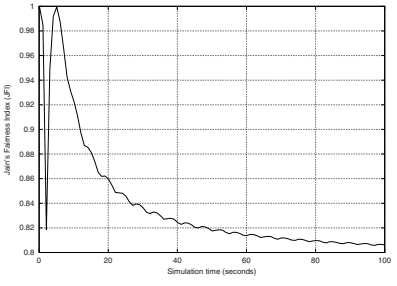


Fig. 9. JFI for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 8$) connections

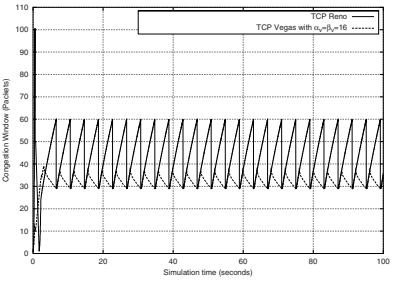


Fig. 10. Congestion windows for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 16$) connections

Vegas and TCP Reno. We started from $\alpha = \beta = 2$ and gradually increased their values until $\alpha = \beta = 32$. The congestion window plots for TCP Vegas and TCP Reno are given in Figures 4 7, 10 and 13. The throughputs of TCP Vegas and TCP Reno for different values of α and β are shown in Figures 5, 8, 11 and 14. Next, we plot JFI between TCP Vegas and TCP Reno, as calculated by using equation (4), in Figures 6, 9, 12 and 15. This experiment confirms that fairness between TCP Vegas and TCP Reno can be improved by increasing the values of the α and β parameters of TCP Vegas. This forms the experimental basis of our new algorithm, formulated in section 5.

6.2 Experiment 2

In this experiment we employed a RED router with ECN at the bottleneck link and the α and β parameters of TCP Vegas were increased automatically when congestion information from the RED router was received via ECN (i.e. used TCP NVegas). We started to gradually increase the values of α and β from the minimum value of $\alpha = \beta = 2$ as assumed in the previous experiment. The congestion window, throughput and Jain's fairness index are plotted in

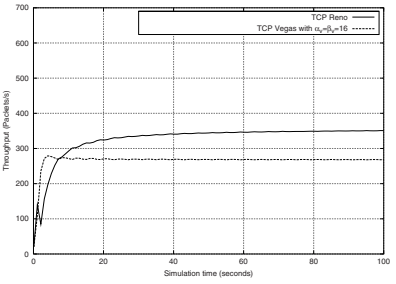


Fig. 11. Throughputs for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 16$) connections

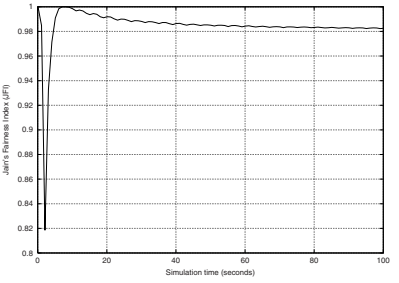


Fig. 12. JFI for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 16$) connections

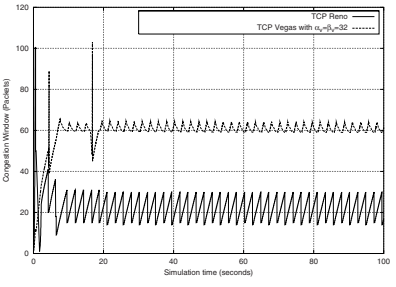


Fig. 13. Congestion windows for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 32$) connections

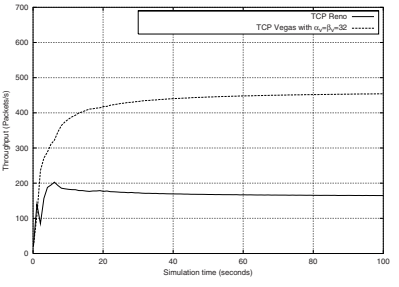


Fig. 14. Throughputs for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 32$) connections

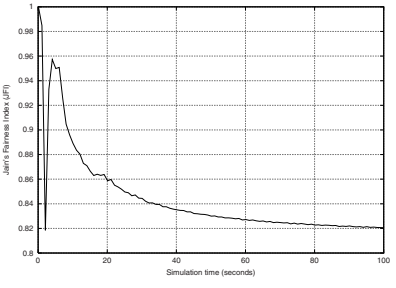


Fig. 15. JFI for TCP Reno and TCP Vegas ($\alpha_v = \beta_v = 32$) connections

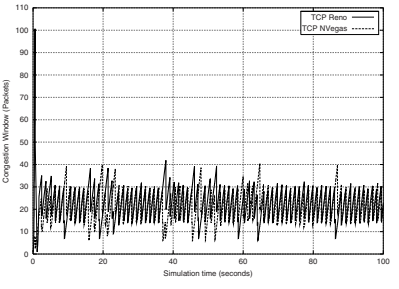


Fig. 16. Congestion windows of TCP Reno and TCP NVegas with RED router based ECN at the bottleneck link

Figures 16, 17 and 18, respectively. The congestion windows of both TCP NVegas and TCP Reno are very close to each other, except at some transient epochs e.g. just after $t = 0$ s and at about $t = 9, 17, 19$ s. The throughputs of both TCP versions are also close to each other thus giving a better JFI.

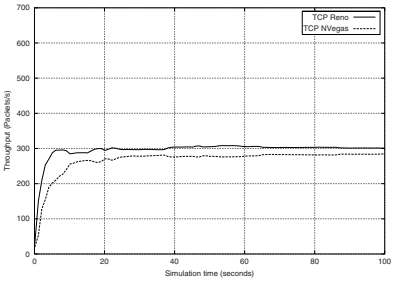


Fig. 17. Throughput of TCP Reno and TCP NVegas with RED router based ECN at the bottleneck link

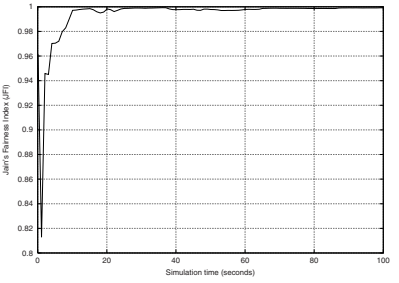


Fig. 18. JFI with RED router based ECN

7 Conclusions

We considered the incompatibility issue between TCP Reno and TCP Vegas. We have analyzed a heterogenous scenario in which TCP Reno and TCP Vegas connections are competing for bandwidth of a bottleneck link. We showed analytically that the Jain Fairness Index decreases with an increase in congestion window of TCP Reno. In order to increase the aggressiveness of TCP Vegas in heterogenous scenarios we suggest to increase the α and β parameters of TCP Vegas. First, they were increased manually and later the process of increasing α and β was automated by using ECN based congestion signals and the difference in new and old round trip times.

Therefore, we have shown that bandwidth sharing (thus JFI) between TCP Vegas and TCP Reno can be improved by making TCP Vegas more aggressive by increasing its α and β parameters. We have also been able to show that TCP NVegas (ECN-based TCP Vegas) can successfully compete with TCP Reno while improving the resulting JFI. However, TCP NVegas algorithm requires detailed mathematical analysis to ensure its better quantitative tuning. It forms part of our ongoing research work which will be presented in a forthcoming paper.

References

[1] Stevens, W. R.: TCP/IP Illustrated: The Protocols. Volume 1. Addison-Wesley (1994) ISBN 0-201-63346-9, Information available at `http://www.kohala.com/start/tcpipiv1.html`. 321

[2] Paxson, V.: Automated Packet Trace Analysis of TCP Implementations. Proceedings of the ACM SIGCOMM'97 **27** (1997) 167–179 Cannes, France. 321

[3] Paxson, V.: End-to-end Internet Packet Dynamics. IEEE/ACM Transactions on Networking **7** (1999) 277–292 321

[4] Weigle, E., chun Feng, W.: A Case for TCP Vegas in High-Performance Computational Grids. Proceedings of the IEEE International Symposium on High Performance Distributed Computing (2001) 152–158 321

[5] Padhye, J., Floyd, S.: TCP Behavior Inference Tool, TBIT. Information and Code available at `http://www.icir.org/tbit/` (2002) 321
[6] Lakshman, T. V., Madhow, U.: The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. IEEE/ACM Transactions on Networking **5** (1997) 336–350 321
[7] Henderson, T. R., Sahouria, E., McCanne, S., Katz, R. H.: On Improving the Fairness of TCP Congestion Avoidance. Proceedings of the IEEE GLOBECOM'98 **1** (1998) 593–544 321
[8] Brakmo, L. S., O'Malley, S. W., Peterson, L. L.: TCP Vegas: New Techniques for Congestion Detection and Avoidance. ACM Computer Communication Review **24** (1994) 24–35 322, 323, 326
[9] Ahn, J. S., Danzig, P., Liu, Z., Yan, L.: An Evaluation of TCP Vegas: Emulation and Experiment. ACM Computer Communication Review **25** (1995) 185–195 322
[10] Bonald, T.: Comparison of TCP Reno and TCP Vegas via Fluid Approximation. Technical Report RR-3563, INRIA, France (1998) `ftp://ftp-sop.inria.fr/pub/rapports/RR-3563.ps.gz`. 322
[11] Boutremans, C., Boudec, J. Y. L.: A Note on the fairness of TCP Vegas. Proceedings of International Zurich Seminar on Broadband Communications (2000) 163–170 322
[12] Lai, Y. C., Yao, C. L.: The Performance Comparison between TCP Reno and TCP Vegas. Proceedings of seventh IEEE International Conference on Parallel and Distributed Systems (2000) 61–66 322
[13] Low, S. H.: A Duality Model of TCP and Queue Management Algorithms. Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management (2000) Monterey CA, USA. 322
[14] Mo, J., La, R., Anantharam, V., Walrand, J.: Analysis and Comparison of TCP Reno and Vegas. Proceedings of the IEEE INFOCOM 1999 **3** (1999) 1556–1563 Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 322
[15] Lai, Y. C.: Improving the Performance of TCP Vegas in a Heterogeneous Environment. Proceedings of the Eighth International Conference on Parallel and Distributed Systems, ICPADS 2001 (2001) 581–587 322
[16] Hasegawa, G., Kurata, K., Murata, M.: Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet. Proceedings of International Conference on Network Protocols (2000) 177–186 323
[17] Mathis, M., Semke, J., Mahdavi, J., Ott, T.: The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. ACM Computer Communication Review **27** (1997) 67–82 324
[18] Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking **1** (1993) 397–413 326
[19] Ramakrishnan, K., Floyd, S., Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. Network Working Group, RFC 3168 (2001) Category: Standards Track. 326

Enhancements to the Fast Recovery Algorithm of TCP NewReno*

Dongmin Kim, Beomjoon Kim, Jechan Han, and Jaiyong Lee

High Performance Multimedia Network Lab.
Department of Electrical & Electronic Engineering, Yonsei University
134 Shinchon-Dong Seodaemun-Gu, Seoul, Korea
{danny,jyl}@nasla.yonsei.ac.kr

Abstract. Transmission control protocol (TCP) suffers significant performance degradation over wireless networks where packet losses are not always caused by network congestion. In order to prevent frequent retransmission timeout (RTO), which is the main reason for the degradation, we propose enhancements that make it possible for a TCP sender to recover packet losses occurred during fast recovery period. The proposed scheme consists of two algorithms called Duplicate Acknowledgement Counting (DAC) and Extended Fast Recovery (EFR). Simulation results show that the proposed scheme improves the throughput of TCP NewReno by reducing the number of RTO.

1 Introduction

Transmission control protocol (TCP) is widely used as the transport layer protocol in the Internet. Since it is designed on the assumption that it may be used on wireline networks where packet loss probability is negligibly low [1], TCP regards all packet losses as network congestion.

The throughput of TCP, therefore, can suffer over wireless networks that are characterized by bursty and high channel error probability [2], [3]. Performance degradation of TCP over wireless links has two main reasons. First, unnecessary congestion control caused by non-congestion packet losses prevents the congestion window from growing enough [4]; it results in low transmission speed of a sender. Second, retransmission timeout (RTO) occurs frequently when multiple packets are lost in a window. Especially, when a RTO takes place, a sender cannot transmit data till it is expired but should start to transmit data in slow start. In addition, since the value of RTO is doubled every time a retransmission is sent, it may have few chances to decrease due to successive RTOs, which leads to low utilization of the limited resource. In [10], the authors show that roughly 56% of retransmissions sent by a busy web server are sent after RTO expires, whereas only 44% are handled by fast retransmit. Therefore, it is a very

* This work was supported by grant No. R01-2002-000-00531-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 332–341, 2004.

important issue whether the packet losses may be recovered without RTO or not.

Although TCP NewReno can recover multiple packet losses without RTO [5]–[8], it has a problem that RTOs take place frequently if packet losses occur during fast recovery. During fast recovery, the sender transmits retransmissions of lost packets and new packets that are included by the slide and inflation of the window.[1] If a retransmission is lost, a RTO always cannot be avoided [8]. According to [9], 4% of timeouts are caused by lost retransmissions. Not all but almost all of new packet losses during fast recovery cannot be recovered by fast retransmit.

In this paper, we propose a scheme that makes it possible for a TCP NewReno sender to recover packet losses during fast recovery. The proposed scheme operates on the basis of two algorithms called Duplicate Acknowledgement Counting (DAC) and Extended Fast Recovery (EFR). DAC can recover retransmission losses based on the number of duplicate acknowledgements (ACKs), and EFR recovers new packet losses by extending fast recovery till they are recovered by retransmissions. The proposed scheme requires simple changes only to TCP implementation at the sender and is perfectly consistent with current TCP specification.

The remainder of the paper is organized as follows. We describe the behaviors of DAC and EFR by showing two examples in Section 2. Section 3 presents the simulation environments. Section 4 contains the simulation results and their discussion. Finally, our conclusions are summarized in Section 5.

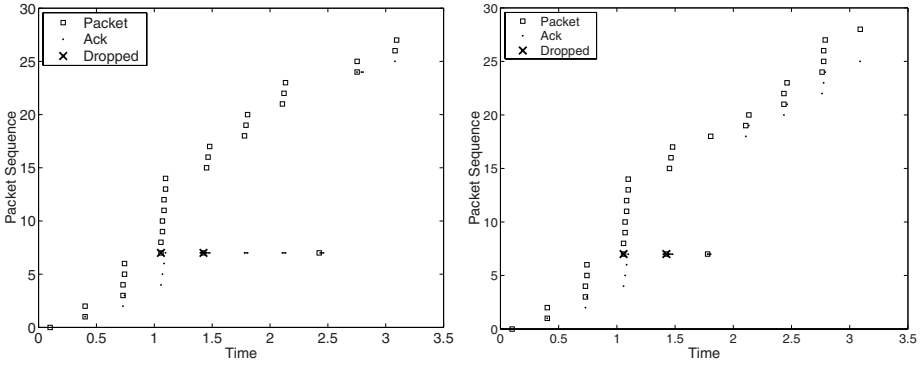
2 Description of the Proposed Scheme

TCP congestion control consists of four algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. The initial fast recovery algorithm of TCP Reno has been modified in the TCP NewReno implementation, and our proposed scheme is applied to the renewed fast recovery algorithm.

2.1 Duplicate Acknowledgement Counting (DAC)

For DAC operation, the sender keeps some variables to store the expected number of duplicate ACKs for a packet loss if its retransmission is not lost again. The congestion window size just before the first fast retransmission is stored to another variable denoted by s_{cwnd}. During fast recovery, the sender counts the number of duplicate ACKs for a packet loss. If it receives more duplicate ACKs for the packet loss than the stored value, it determines that its retransmission is lost again and retransmits it without waiting for its RTO. We denote the expected number of duplicate ACKs for the ith lost packet in a window by DAC_i. When the first fast retransmit is performed, the sender is not aware of how many packets are lost in a window but only knows that at least one packet is lost. Therefore, DAC_1 is always equal to $s_{cwnd} - 1$.

[1] In the rest of this paper, we call these packets "new packets".



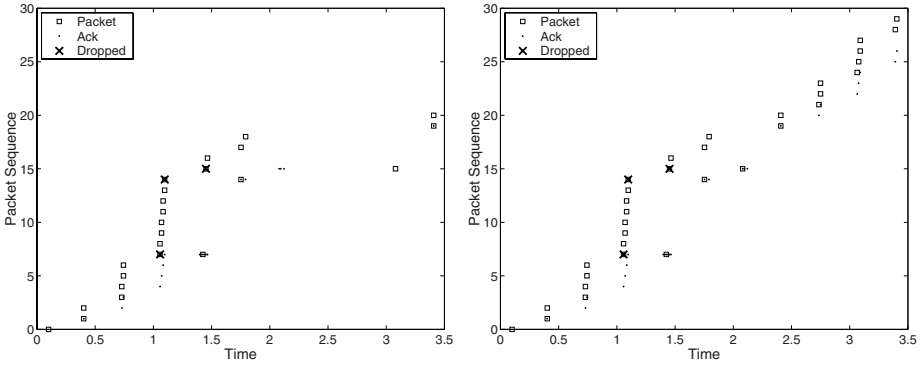
(a) TCP NewReno (b) TCP NewReno using DAC

Fig. 1. Comparison of the loss recovery behaviors for a single retransmission loss (x-axis: time, y-axis: packet sequence number)

In fig. 1, we compare the sender's behavior of TCP NewReno with and without DAC when a packet is lost and its retransmission is also lost. Each transmitted packet is marked by a square on the graph. Packets dropped are indicated by an "x" on the graph for each packet dropped. Received ACK packets are marked by a smaller dot. At 1.1 second, the sender transmits eight packets 7–14 with the congestion window of eight and packet 7 is lost.[2] When the sender receives the third duplicate ACK for packet 7 at 1.42 second, it performs fast retransmit and sets its *usable window* [8] to $7(=\lfloor 8/2 \rfloor + 3)$ and DAC_1 to $7(= 8 - 1)$. Four more duplicate ACKs for packet 7 inflate the *usable window* to eleven, and three new packets 15–17 are allowed to be transmitted. Since the retransmission of packet 7 is lost, the sender receives three more duplicate ACKs by packets 15–17, which inflate the *usable window* by three so that newly included packets 18–20 are transmitted. Since the receiver cannot deliver a normal (i.e., non-duplicate) ACK as long as packet 7 is not received, the sender cannot exit fast recovery till RTO of packet 7 is expired. The RTO of packet 7 expires at 2.42 second, causing a retransmission and putting the sender into slow start.

In the case of using DAC, the sender sets $DAC_1 = 7(= 8 - 1)$ when it fast retransmits packet 7. When the sender receives the eighth duplicate ACK by packet 15, which is greater than DAC_1, it can retransmit packet 7 with the congestion window of two$(=\lfloor 4/2 \rfloor)$. Loss in two successive windows of data, or the loss of a retransmission, should be taken as two indications of congestion and, therefore, congestion window and slow start threshold must be lowered twice in this case [6]. The first ACK for a new packet 18 as a result of the

[2] For convenience, packets are numbered with unique sequence numbers starting with zero. We use these numbers to specify the packets in the rest of this paper.



(a) TCP NewReno (b) TCP NewReno using EFR

Fig. 2. Comparison of the loss recovery behaviors for a single new packet loss during fast recovery (x-axis: time, y-axis: packet sequence number)

receiver receiving the second retransmitted packet 7 brings the sender out of fast recovery with the congestion window of two, and the sender continues in congestion avoidance. With DAC, almost all of 4% of timeouts caused by lost retransmissions could be recovered during fast recovery without RTOs.

2.2 Extended Fast Recovery (EFR)

When the sender of TCP NewReno performs fast retransmit, it sets the value of *recover* [7] to the highest sequence number of packet that has already been transmitted. During fast recovery, the sender can determine whether it is a partial ACK or not by comparing the sequence number that an ACK takes to the stored value in *recover*. In the original implementation of the fast recovery algorithm of TCP NewReno, the value of *recover* is not updated during fast recovery. Therefore, if a new packet whose sequence number is greater than *recover* is lost, its loss cannot be detected without another three duplicate ACKs. If three duplicate ACKs can be received,[3] the new packet loss can be recovered by fast retransmit. In this case the sender shrink its congestion window twice.

In the case of using EFR, the sender updates *recover* every time it retransmits a lost packet during fast recovery. Therefore, even if there is a new packet loss, the sender always receives a partial ACK, thereby, it can retransmit the lost packet without aborting fast recovery. In this case, the congestion window and slow start threshold should be reduced one more time as in the same way of DAC.

[3] It depends upon not only the number of packet losses and congestion window size but also the position of the newly dropped packet in the *usable window*.

In fig. 2, we compare the sender's behaviors of TCP NewReno using EFR to the original TCP NewReno. Two packets, packet 7 and packet 14, are lost in a window and a new packet, packet 15, is also lost during fast recovery. Before the sender retransmits the first lost packet 7, the evolutions of the sender's window can be explained in the same way as DAC in fig. 1. When the sender fast retransmits the packet 7, it sets *recover* to 14 and transmits two new packets, packet 15 and packet 16, by the inflation of the *usable window*. At about 1.75 second, the sender receives a partial ACK for the second lost packet 14 and transmits the retransmission of packet 14. At this time a new packet 17 is also transmitted with the window of four($=\lfloor 8/2 \rfloor$). One more duplicate ACK by packet 16 inflates the *usable window* by one, so that packet 18 is transmitted. The first ACK for the newly lost packet 15, which is greater than *recover*, takes the sender out of fast recovery with the congestion window of four. In congestion avoidance, the sender receives only two duplicate ACKs for packet 15 by packets 17–18. Consequently, the sender cannot trigger a fast retransmit for packet 15 but waits for its RTO expiry.

In the case of using EFR, the sender updates *recover* to 16 after the retransmission of packet 14. Note that packet 15 and packet 16 have already been transmitted. At 2.08 second, the first ACK for the newly lost packet 15 by the retransmission of packet 14 is received. Since its sequence number is still smaller than the updated *recover*, the sender regards the ACK as a partial ACK and transmits packet 15 immediately. Since loss in two successive windows of data should be taken as two indications of congestion, the sender reduces the congestion window and slow start threshold to two ($=\lfloor 4/2 \rfloor$) [6]. The ACK for packet 19 as a result of the receiver receiving the retransmitted packet 15 brings the sender out of fast recovery, and the sender continues in congestion avoidance with the congestion window of two. Even if two or more new packets are lost, EFR can recover them without RTO in the same way.

3 Simulation Environments

Using *ns* simulations [11], we evaluate the performance of the proposed scheme. In our simulations, a sender and a destination establish a single TCP connection over a link of 10Mbps and 50ms, where packets are lost in random with packet loss probability p. It is assumed that packet losses are independent to each other and each packet has the same size of 1 Kbytes. Since all TCP variants have almost the same throughput when packet loss probability is quite small or large, simulations are performed for packet loss probabilities from 10^{-3} to $3 \cdot 10^{-1}$. Since the size of an ACK packet is considerably small compared to data packet, an ACK packet is assumed not to be lost or compressed. During simulations are running, the source transmits 10^6 packets and the congestion window can grow up to W_{max}, which is the upper bound value advertised by the receiver at the connection setup time.

We evaluate the performance of the proposed scheme from the aspects of the normalized throughput, fast recovery probability, timeout probability, and response time, which are defined as follows:

i) The normalized throughput is the ratio of the throughput for each packet loss probability to the throughput when p is 10^{-3}.

ii) The fast recovery probability is the ratio of the number of retransmissions recovered by not olny fast retransmit and partial ACK but also DAC and EFR to the number of total packet losses.

iii) The timeout probability is the number of retransmissions after RTO to the number of total packet losses.

iv) The response time is the sender-side elapsed time between the first packet sent and the last ACK received.

4 Simulation Results

In fig. 3, we compare the proposed scheme to TCP NewReno in the normalized throughput, fast recovery probability, and retransmission timeout probability obtained from simulations. As expected, the throughput of each TCP decreases for increasing packet loss probability. It reflects that more packet losses are not likely to be recovered by retransmissions for larger packet loss probability. The normalized throughput in fig. 3-(a) and fast recovery probability in fig. 3-(b) make it clear; they shows a similar shape. Since the proposed scheme can avoid unnecessary RTOs due to retransmission losses and new packet losses occurred during fast recovery, its fast recovery probability is higher than TCP NewReno, which leads to slight throughput improvement. The insignificant improvement can be explained by the fact that the results in these figures are obtained in steady-state so that the effect of a RTO may be averaged throughout a long TCP connection. However, if a new packet loss or a retransmission loss occurs in a short TCP connection (e.g., Web), the proposed algorithm may offer even greater benefit.

When packet loss probability is small, few packets are lost; a retransmission or a new packet is also not likely to be lost for such a low packet loss probability.

It is the reason that the improvement of the proposed scheme (a dashed line with circles) rather increases as packet loss probability increases till it arrives at a certain value. When packet loss probability arrives at the value of $5 \cdot 10^{-2}$, the improvement shows its highest value of about 10%. As packet loss probability increases continuously, the congestion window cannot keep its size large enough due to frequent loss recovery events. Therefore, fast retransmit does not succeed very often so that the improvement decreases close to zero. Note that our proposed scheme does not work if a fast retransmit is not triggered.

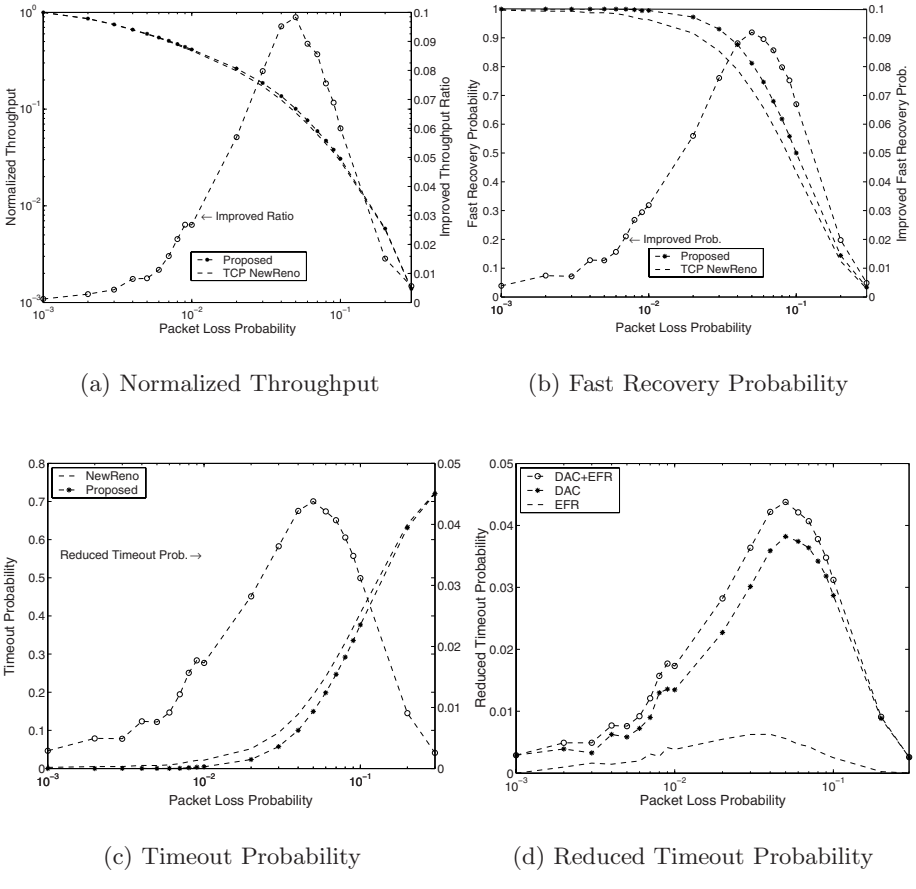


Fig. 3. Comparing the proposed scheme to TCP NewReno in the normalized throughput, fast recovery probability, and retransmission timeout probability ($W_{max} = 32$)

In fig. 3-(c), we show the timeout probability of the proposed scheme and TCP NewReno. The overall shape of the timeout probability and reduced timeout probability can be explained in the similar way as in fig. 3-(a) and fig. 3-(b); as packet loss probability increases, the timeout probability also increases. It can be seen that roughly 4% of timeouts can be avoided by using the proposed scheme for packet loss probability of $5 \cdot 10^{-2}$. It results from preventing RTOs by DAC and EFR. Fig. 3-(d) shows each improvement that DAC and EFR[4] offer to

[4] Since a new packet loss does not always cause a RTO, we consider only the packet losses that are going to cause RTO without EFR as 'valid' retransmission for EFR. Note that a new packet loss in current fast recovery may be recovered by fast retransmit if three duplicate ACKs can be received in congestion avoidance following the current fast recovery period.

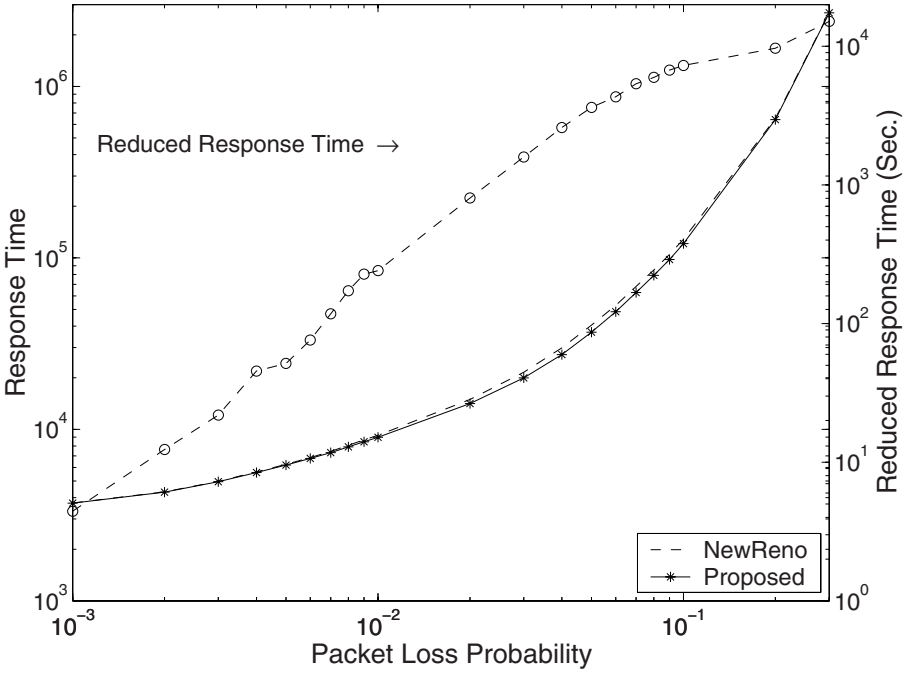


Fig. 4. Response time and reduced response time of the proposed and the TCP NewReno ($W_{max} = 32$)

the timeout probability. Unlike retransmission losses, which always invoke RTO, all of new packet losses do not cause RTO. Therefore, DAC offers greater benefit than EFR.

In fig. 4, we show the response time of the proposed scheme and TCP NewReno to transmit 10^6 packets. The difference between two lines of the proposed scheme and TCP NewReno is indicated by a dashed line with circles. As mentioned earlier, the difference is not significant compared to the response time because a RTO duration is quite short compared to total duration. However, from the view of RTT, so many RTT has been reduced by means of our proposed scheme.

Fig. 5 shows the fast recovery probability of the proposed and TCP NewReno when the value of W_{max} varies. For TCP NewReno, the increment of W_{max} from 8 to 128 makes no significant differences to the fast recovery probability. It means that TCP NewReno cannot take any advantage of a large W_{max} from the aspect of loss recovery since the fast recovery probability mainly depends on the congestion window size and the number of losses in it rather than the value of W_{max}. Note that a single packet loss can be recovered by fast retransmit if only the window size is larger than four, and, if the first packet loss can be recovered by fast retransmit, even if multiple packets are lost, RTO can be avoided. On the other hand, a large W_{max} of 128 makes slight difference to the recovery

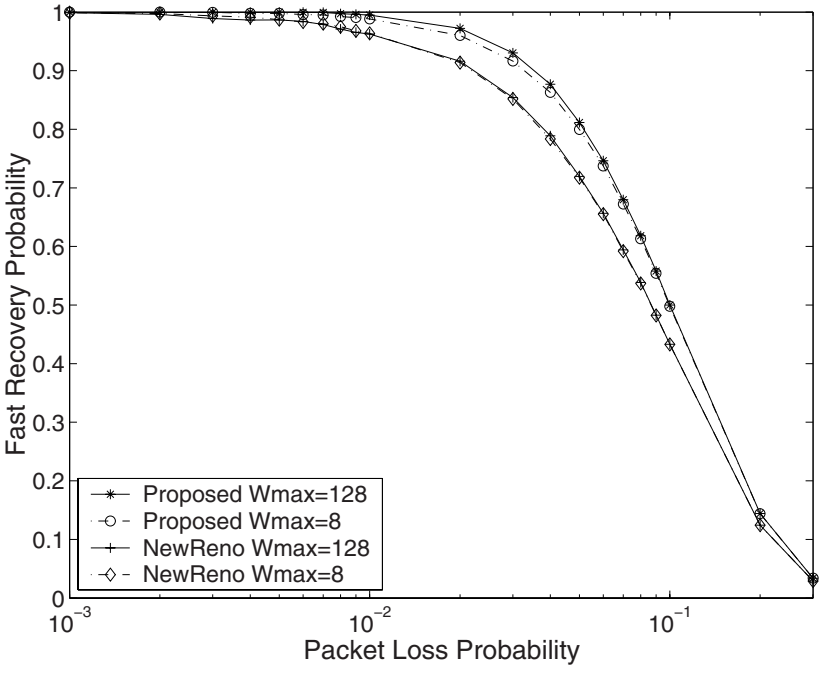


Fig. 5. Relation between the fast recovery probability and W_{max}

probability of the proposed scheme. The difference comes from the different number of new packet losses for different size of W_{max}. Suppose that the first packet in a window is lost. If the window size is equal to 8, the sender transmits $3(= \lfloor 8/2 \rfloor + 7 - 8)$ new packets after fast retransmit. Whereas, if the window size is equal to 128, the sender transmits $63(= \lfloor 128/2 \rfloor + 127 - 128)$ new packets after fast retransmit. Inferring that the more new packets corresponds to the more new packet losses for the same packet loss probability, EFR may have more chances to recover new packet losses when the window size is equal to 128. However, as packet loss probability increases, the congestion window decreases before it reaches its maximum value so that it does not show such a large difference. Though congestion window size is kept small in the presence of packet losses, the fast recovery probabilities of proposed scheme with $W_{max} = 8$ are greater than that of TCP NewReno with $W_{max} = 128$. It reveals that proposed scheme works well with small congestion window size. Note that W_{max} does not affect DAC because it is operated by the number of duplicate ACKs.

Since the proposed scheme is invoked by the receiving ACKs which are not delivered if a network is in congestion state, and reduce the congestion window by half when they work as recommended in [6], TCP with proposed scheme may not be aggressive and does no harm on fairness of other competing TCP flows. Therefore, if sender surely knows which packet is lost, retransmission before

a RTO is a reasonable and proper operation. In case of using proposed scheme for the transfer of short files, it will not lead to additional congestion. Adding the proposed scheme makes TCP more robust to random transmission errors.

5 Conclusions

In this paper, we have proposed a scheme that enhances the fast recovery algorithm of TCP NewReno by avoiding unnecessary RTOs due to packet losses occurred during fast recovery. By using the proposed scheme, a sender of TCP NewReno can detect and recover almost all of retransmission losses and new packet losses without RTO. Simulation results show that the proposed scheme increases the ratio of retransmissions to packet losses, which results in the slight improvement of the steady-state throughput of TCP NewReno, without violating the congestion control principles. Since a retransmission loss or new packet loss is not a common event, its improvement seems to be insignificant. However, considering the effect of RTO on TCP performance, it is rather a great improvement with no large modifications. Especially, the proposed scheme may offer large benefit for a short TCP connection such as web transfer, which remains as one of our future works.

References

[1] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz: A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. IEEE/ACM Transactions on Networking, vol. 5. no. 6, pp. 756–769, 1997. 332

[2] M. Zorzi, A. Chockalingam, and R. R. Rao: Throughput Analysis of TCP on Channel with Memory. IEEE Journal on Selected Areas in Communications, vol. 18. no. 7, pp. 1289–1300, 2000. 332

[3] C. Barakat, E. Altman, and W. Dabbous: On TCP Performance in A Heterogeneous Network: A Survey. IEEE Communications Magazine, pp. 40–46, 2000. 332

[4] J. Pan, J. W. Mark, and X. Shen: TCP performance and its improvement over wireless links. IEEE GLOBECOM'00, pp. 62–66, 2000. 332

[5] Janey C. Hoe: Improving the Start-up Behavior of a Congestion Control Scheme for TCP. ACM SIGCOMM'96, 1996. 333

[6] M. Allman, V. Paxson, and W. Stevens: TCP Congestion Control: RFC 2581, 1999. 334, 336, 340

[7] S. Floyd and T. Henderson: The NewReno Modification to TCP's Fast Recovery Algorithm: RFC 2582, 1999. 335

[8] K. Fall and S. Floyd: Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. ACM Computer Communication Review, vol. 26. no. 3, pp. 5–21, 1996. 333, 334

[9] Dong Lin and H. T. Kung: TCP Fast Recovery Strategies: Analysis and Improvements. IEEE INFOCOM'98, pp. 263–271, 1998. 333

[10] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm, and R. H. Katz: TCP Behavior of a Busy Internet Server: Analysis and Improvements. IEEE INFOCOM'98, 1998. 332

[11] `http://www.isi.edu/nsnam/ns/index.html` 336

Route Reinforcement for Efficient QoS Routing Based on Ant Algorithm*

Jae Seuk Oh, Sung-il Bae, Jin-ho Ahn, and Sungho Kang

Dept. of Electrical and Electronic Engineering, Yonsei University
132 Shinchon-Dong, Seodaemoon-Gu, 120-749, Seoul, Korea
{jay3204,sominaby}@soc.yonsei.ac.kr
{anet0727,shkang}@yonsei.ac.kr

Abstract. In this paper, we present a new method to calculate reinforcement value in QoS routing algorithm for real-time multimedia based on Ant algorithm to efficiently and effectively reinforce ant-like mobile agents to find the best route toward destination in a network. Simulation results show that the proposed method realizes QoS routing more efficiently and more adaptively than those of the existing method thereby providing better solutions for the best route selection.

1 Introduction

As Internet expands, the demand for real time and quality of services (QoS) in a network increases. The quality of services are sensitive to the network's characteristics such as bandwidth, delay, delay jitter, packet loss and cost depending on the type of applications. Furthermore, the use of multiple metrics is needed to better characterize a network and to support a wide range of QoS requirements [1].

Ant algorithm is a routing algorithm, which is inspired by the trail following behavior of real ant colonies, and this algorithm realizes an adaptive and social behavior of ants of finding the best route to the food source from the nest by indirect communications between ants using a chemical substance called *pheromone* [2,3]. However, thus far, there haven't been many researches that are done to realize QoS routing based on Ant algorithm, leaving a large room for improvements and explorations.

The main purpose of this paper is to propose a new method to calculate reinforcement value reflecting all the necessary QoS metrics to better realize an adaptive behavior of Ant algorithm for real-time applications.

This paper is organized as follows. In section 2, QoS routing algorithm based on Ant algorithm is introduced. In section 3, detailed description of the proposed method of reinforcement calculation under the QoS routing algorithm described in section 2 is introduced. In section 4, experimental results are presented; and at last, conclusion from this research is drawn out in section 5.

* This work was supported by the Brain Korea 21 Project in 2003.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 342–349, 2004.

2 Ant Algorithm Based QoS Routing

There are T sets of ants where every set is consisting of M types of ants belonging to M different call requirements, where each ant type must find the best path to its destination that satisfies all the requirements. Furthermore, the properties of pheromone deposits of each ant type are different from each other, so that an ant selects a route relying only on the pheromone deposited by the same type of ant on the path.

To speed up the process, this algorithm makes some adjustments: considering the delay jitter constraint outside the ant algorithm, and filter the topology of the network by cancelling the edges that do not satisfy the bandwidth constraint [1]. The steps needed to take in the algorithm are described below.

1. If actual end-to-end delay jitter is greater than the constraint, routing fails. 2.Filter out links that do not satisfy the bandwidth constraint. 3. Initialize the amount of pheromone deposits. 4. Send out a set of ants of every type one at a time at a constant interval toward corresponding destinations, while choosing its path by repeatedly applying the state transition rule. 5. When a path to the destination has found, use the local updating rule to adjust the amount of pheromone deposits. Repeat step 4 and 5 until all sets of ants finish the steps. 6. Select the globally best ant of each type. 7. Apply the global updating rule to update pheromone deposits. 8. Repeat step 4 through 7 until the accuracy requirement is satisfied [1].

In order to follow the above-mentioned steps of Ant algorithm based QoS routing algorithm, the state transition rule and the pheromone-updating rule are proposed. Under the state transition rule proposed in [1], a d type of an ant at node r selects a next node s to travel according to following rule

If $q \leq q_0$, then

$$\rho_d(r,s) = \begin{cases} 1, \max(\text{phero(d,r,s)}),\ s \in J_d(r) \\ 0, \text{otherwise} \end{cases} \tag{1}$$

otherwise,

$$\rho_d(r,s) = \begin{cases} \frac{phero(d,r,s)}{\sum_{u \in J_d(r)} phero(d,r,u)},\ s \in J_d(r) \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{2}$$

where, q_0 represents a constant value that lies between 1 and 0, which is use to compare with q, a randomly chosen number between 1 and 0.

The pheromone-updating rule is further divided into two sub-rules: local updating rule and global updating rule, the concept was first proposed in [3]. Under the local updating rule, suppose a d type ant at node r chooses a neighbor node s as the next node to travel, the amount of pheromone $phero(d,r,s)$ is adjusted in accordance with equation (3), otherwise no pheromone amount gets adjusted

$$phero(d,r,s) = (1-a_0) \cdot phero(d,r,s) + a_0 \cdot cons \tag{3}$$

where, a_0 is a value between 0 and 1, and $cons$ is a constant.

The global updating rule is used when the globally best path to the destination is found. Once the globally best path is determined, pheromone amounts of edges between all nodes in the globally best path are adjusted in accordance with equation (4), and pheromone amounts of all the other edges get adjusted by equation (5)

$$phero(d,r,s) = (1-a_1)\cdot phero(d,r,s) + a_1 \cdot F \tag{4}$$

$$phero(d,r,s) = (1-a_1)\cdot phero(d,r,s) \tag{5}$$

where, a_1 is a value between 0 and 1, and F is the cost function, and it plays the same role as the reinforcement value in AntNet [2]. Furthermore, the value of F is calculated by following equations:

$$F = -F_1 + F_2 \tag{6}$$

where,

$$F_1 = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} LC_{ij} \cdot P_{ij}^{d} \tag{7}$$

$$F_2 = A \cdot \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} H(Z_{ij}) + B \cdot H(Z_2) + C \cdot H(Z_3) \tag{8}$$

where, N is the node number, A, B and C are positive real coefficients. Here, F_1 represents the total cost of the route selected by an ant; and F_2 represents the QoS constraints.

$$H(Z) = \begin{cases} 0, \text{ where } Z < 0 \\ Z, \text{ otherwise} \end{cases} \tag{9}$$

$$Z_{ij} = P_{ij}^{d} \cdot LB_{ij} - B_w \tag{10}$$

$$Z_2 = D_w - (\sum \sum LD_{ij} \cdot P_{ij}^{d} + \sum N_i^d \cdot ND_i \tag{11}$$

$$Z_3 = \prod_{i=1}^{N} (1 - N_i^d \cdot NL_i) - (1 - L_w) \tag{12}$$

Here, $P_{ij}^{d} = 1$ if an edge between node i and node j is in the d type ant selected route, otherwise $P_{ij}^{d} = 0$. $N_i^d = 1$ if node i is the node in the d type ant selected route, otherwise $N_i^d = 0$. Symbols LB_{ij}, LC_{ij} and LD_{ij} are the available bandwidth, cost and delay of an edge between node i and node j respectively, and L_w, B_w and D_w represent link, bandwidth and delay constraints respectively.

3 Proposed Method to Reinforce Route

The equations described in the previous section do not realize the adaptive behavior well. Unlike [1], the proposed reinforcement calculation uses ratios between QoS measurements and QoS constraints. The proposed method to calculate the reinforcement value is as follows.

If($F_2 \geq 1$),

$$F = F_2 - k(\frac{F_1}{F_2}) \tag{13}$$

else

$$F = F_2 - k \cdot F_1 \tag{14}$$

$$F_1 = \sum_{i=1}^{N} \sum_{j=1, j \neq i} LC_{ij} \cdot P_{ij}^{d} \tag{15}$$

$$F_2 = A \cdot min(Band_{ij}) + B \cdot Dly + C \cdot PLR + D \cdot DJ \tag{16}$$

where, F is the cost function or the reinforcement; F_1 is the total cost of the route; F_2 is the QoS constraints; and k is weight constant for cost to indicate its importance compare to other QoS metrics.

$$Band = \begin{cases} \frac{B_{mea}}{B_w}, & \text{where } B_{mea} \geq B_w \\ (\frac{B_{mea}}{B_w})^2, & \text{where } toleration - rate \leq \frac{B_{mea}}{B_w} < 1 \\ 0, & \text{where } \frac{B_{mea}}{B_w} < toleration - rate \end{cases} \tag{17}$$

$$Dly = \begin{cases} \frac{D_w}{\sum D_{mea}}, & \text{where } \sum D_{mea} \leq D_w \\ (\frac{D_w}{\sum D_{mea}})^2, & \text{where } toleration - rate \leq \frac{D_w}{\sum D_{mea}} < 1 \\ 0, & \text{where } \frac{D_w}{\sum D_{mea}} < toleration - rate \end{cases} \tag{18}$$

$$PLR = \begin{cases} \frac{L_w}{1-\prod(1-L_{mea})}, & \text{where } \prod(1 - L_{mea}) \geq (1 - L_w) \\ (\frac{L_w}{1-\prod(1-L_{mea})})^2, & \text{where } toleration - rate < \frac{L_w}{1-\prod(1-L_{mea})} < 1 \\ 0, & \text{where } \frac{L_w}{1-\prod(1-L_{mea})} < toleration - rate \end{cases} \tag{19}$$

$$DJ = \begin{cases} \frac{J_w}{\sum J_{mea}}, & \text{where } \sum J_{mea} \leq J_w \\ (\frac{J_w}{\sum J_{mea}})^2, & \text{where } toleration - rate \leq \frac{J_w}{\sum J_{mea}} < 1 \\ 0, & \text{where } \frac{J_w}{\sum J_{mea}} < toleration - rate \end{cases} \tag{20}$$

Equation (16) calculates the amount of positive influence, the QoS measurements have on the reinforcement calculation by considering the goodness of each QoS measurement compare to its constraint [5]. The goodness of each QoS measurement is calculated using equations (17) through (20).

Equations (17) through (20) also have a term "*toleration rate*". Toleration rates are set individually for each QoS metric with values between 0 and 1. Each toleration rate represents percentage of negative discrepancy that the QoS

metric can tolerate. However, in the case of having no paths satisfying the QoS constraints, toleration rate can be used to find a path that provides a decent level of quality of service but with some service degradation.

Equations (13) and (14) are the top level calculation of reinforcement. Equation (13) subtracts the total amount of goodness of all QoS measurements by the fractional value of the ratio between the total cost of the chosen path and the total amount of goodness of all QoS measurements. Equation (14) is for when F_2 less than 1.

In addition to the reinforcement calculation, the global update rule is also proposed to get the best out of the proposed reinforcement calculation.

$$phero(d, r, s) = (1 - a_1) \cdot phero(d, r, s) + c \cdot F \tag{21}$$

where, c is coefficient of value between 0 and 1.

Equation (21) is used to adjust pheromone amount on the globally best route, and equation (22) shown below is a new method to adjust pheromone amount of less qualified paths that were chosen in the process.

$$phero(d, r, s) = (1 - a_1) \cdot phero(d, r, s) - d \cdot (F - F_{others}) \tag{22}$$

Here d is a weight coefficient and F_{others} is the reinforcement value of a less qualified path. Since the reinforcement value under the proposed method is relative to the delay size, if there exist some routes that have reinforcement values almost as good as the globally best route, then the pheromone amounts of the nearly good routes reduce at a rate close to $(1 - a_1) \cdot phero(d, r, s)$.

4 Simulation

Fig. 1 depicts the topology of a network system, which consists of eight nodes and twelve edges connecting the nodes. This topology is adopted from the simulation environment used in [1] to ensure that the simulation results of the existing method in this experiment agree with the results presented in [1].

The values in the parenthesis near each node in Fig. 1 represent node delay, packet loss rate and node delay jitter respectively; whereas, the values in the parenthesis near each edge represent link cost, bandwidth and link delay respectively.

The QoS requirements are set to $B_w = 70$, $D_w = 8$, $L_w = 10^{-5}$ and $J_w = 3$. For the simulation of [1], the simulation parameters are set to T (sets of ants) = 8, M (call requests) = 3, $a_0 = 0.069$, $a_1 = 0.079$, $cons = 0.32$, $q_0 = 0.20$, $A = 0$, $B = 10$ and $C = 15$. For the simulation of the proposed method, most of the simulation parameters are set exactly same as in the existing method; however, since the calculation method is different from the existing method, some new parameters are introduced. And those parameters are set to $C = 0.01$, $D = 0$, $k = 1$, $c = 0.079$, $c_1 = 0.02$, $\alpha = 1$ and $\beta = 1$; and these are set relative to the values of parameters used in [1].

At first, several unicast routing requests: node 0 to node 5, node 1 to node 5, node 1 to node 6, node 2 to node 6, node 2 to node 7, node 5 to node 7, and

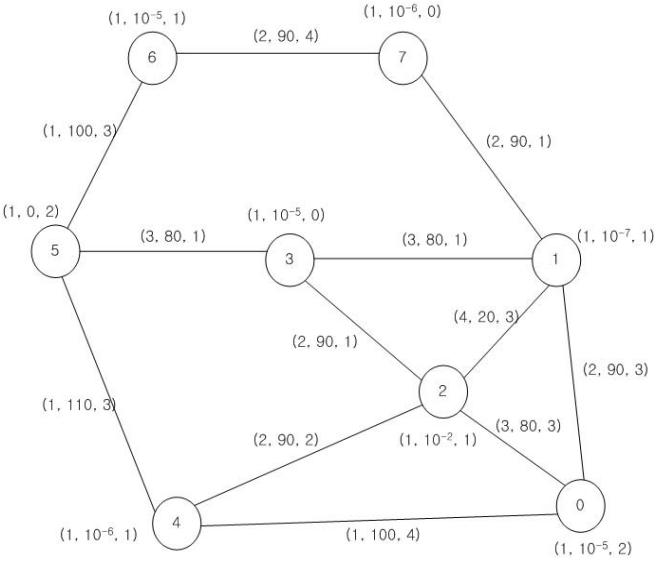


Fig. 1. Network Topology model and its parameters used for simulation[1]

node 4 to node 7 are assumed. By simulation, the globally best routes are found for each method as shown in Table 1.

If we take a good look at Fig. 1, for the routing request (0,5), there exists a route $0 \rightarrow 4 \rightarrow 5$ that satisfies all the QoS requirements and has the cost of 2; however, the reason why the route $0 \rightarrow 1 \rightarrow 3 \rightarrow 5$ or $0 \rightarrow 2 \rightarrow 3 \rightarrow 5$ is selected as the best route is because we set the delay term to be more sensitive than the cost.

Looking at Table 1, we can see that the proposed method finds the best routes faster than the existing method. Under the proposed method, faster convergences are achieved for all the routing requests, where the reinforcement values of some the routing requests are considerably less than those of the existing method. This is due to the adaptive behavior that is realized in the proposed method of the global update using equations (21) and (22).

Next, an experiment is performed to see how the proposed method dynamically adjusts the reinforcement value, by comparing the simulation results of one with small delay constraint and the other with large delay constraint, thereby identifying the advantages of the proposed method.

To simulate one with small delay constraint, the QoS information on edges between node 5 and 6, and node 6 and 7 are changed to (1, 100, 0) and (2, 90, 1) respectively; and to simulate one with large delay constraint, the QoS information on edges between node 0 and 1, 1 and 3, 2 and 3, 1 and 7, 5 and 6, and 6 and 7 are changed to (2, 90, 93), (3, 80, 91), (2, 90, 91), (2, 90, 90), (1, 100, 1) and (2, 90, 1) respectively. Simulation results under the existing method and the proposed method are shown in Table 2.

Table 1. Average iterations to find the globally best route

Routing Requests (s,d)	Existing method of [1]		Proposed method w/ TR=0.99		Proposed method w/ TR=1	
	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.
(0,5)	$0 \rightarrow 1/2 \rightarrow 3 \rightarrow 5$	62.2	$0 \rightarrow 2 \rightarrow 3 \rightarrow 5$	31.4	$0 \rightarrow 1/2 \rightarrow 3 \rightarrow 5$	36.9
(1,5)	$1 \rightarrow 3 \rightarrow 5$	27.2	$1 \rightarrow 3 \rightarrow 5$	25.2	$1 \rightarrow 3 \rightarrow 5$	25.5
(1,6)	$1 \rightarrow 7 \rightarrow 6$	36.8	$1 \rightarrow 7 \rightarrow 6$	31.4	$1 \rightarrow 7 \rightarrow 6$	31.5
(2,6)	$2 \rightarrow 3 \rightarrow 5 \rightarrow 6$	48.7	$2 \rightarrow 3 \rightarrow 5 \rightarrow 6$	49.7	$2 \rightarrow 3 \rightarrow 5 \rightarrow 6$	29.2
(2,7)	$2 \rightarrow 3 \rightarrow 1 \rightarrow 7$	30.6	$2 \rightarrow 3 \rightarrow 1 \rightarrow 7$	28.5	$2 \rightarrow 3 \rightarrow 1 \rightarrow 7$	28.3
(5,7)	$5 \rightarrow 3 \rightarrow 1 \rightarrow 7$	32.6	$5 \rightarrow 3 \rightarrow 1 \rightarrow 7$	27.7	$5 \rightarrow 3 \rightarrow 1 \rightarrow 7$	27.1
(4,7)	N/A	N/A	$4 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 7$	27.5	$4 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 7$	27.8

Table 2. Simulation result of routing request (1,5) under two different delay constraints

	Existing method of [1]		Proposed method	
	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.
Under Small Delay Constraint ($D_w = 8$)	$1 \rightarrow 3 \rightarrow 5$	27.63	$1 \rightarrow 3 \rightarrow 5$	25.22
Under Large Delay Constraint ($D_w = 98$)	$1 \rightarrow 3 \rightarrow 5$	27.47	$1 \rightarrow 3 \rightarrow 5$	34.53

Assuming the routing request of (1,5), the globally best route should be $1 \rightarrow 3 \rightarrow 5$ since it satisfies the constraints and has the smallest delay measurement. However, the route $1 \rightarrow 7 \rightarrow 6 \rightarrow 5$ also satisfies the constraints and has the end-to-end delay of 4 and 94, just one unit of delay more than $1 \rightarrow 3 \rightarrow 5$ in both cases. If we look at Table 2, the average iterations of two different cases under the existing method are nearly identical to each other. However, the average iterations of two different cases under the proposed method show some discrepancy.

The reason for such discrepancy is due to how the reinforcement values are calculated in the methods. Under the existing method, amount of impact a unit of delay exerts on the reinforcement value, is constant regardless of delay size. However, in the proposed method, such fact is realized, and that is why it takes more iterations to converge when the delay size and the constraint are large.

One last experiment is done using network topology with edges between node 0 and 1, 1 and 3, 2 and 3, 1 and 7, 5 and 6, and 6 and 7 having QoS information of (2,90,93), (3,80,91), (2,90,99), (2,90,90), (1,100,1) and (2,90,10) respectively. Assuming the routing request (1,5), the simulation result is compared with that of the previous experiment under large delay constraint.

If we compare the results in Table 2 with Table 3, the average iteration of [1] is the same, while the average iteration of the proposed method is reduced. The

Table 3. Simulation result of the last experiment

Route Requests (s,d)	Under large delay constraint			
	Existing method of [1]		Proposed method	
	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.
(1,5)	$1 \rightarrow 3 \rightarrow 5$	27.47 (27.47)	$1 \rightarrow 3 \rightarrow 5$	28.53 (34.53)

reason for such reduction in the number of iterations under the proposed method is due to equation (22), which reflects more of an adaptive behavior. Since the reinforcement value under the proposed method is relative to the delay size, if there exists some routes that have reinforcement values almost as good as the globally best route, then the pheromone amounts of the nearly good routes reduce at a rate close to the previous experimental results.

5 Conclusions

This paper has presented with an adaptive method to reinforce route in QoS routing algorithm based on Ant algorithm. Under the proposed method, the reinforcement value is calculated using the fractional ratios between the measurements and the constraints, which in turn provides with the reinforcement value relative to the measurement's size and the constraint's range. And with further changes in the global update equations, the proposed method realizes more adaptive behaviors than the existing method, providing faster convergence time.

References

[1] Zhang, S., Liu, Z.: A QoS Routing Algorithm Based on Ant Algorithm. 25^{th} Annual IEEE Conference on Local Computer Networks. (2000) 574–578
[2] Dorigo, M., Di Caro, G.: AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificail Intelligence Research, Number 9. (1999) 317–365
[3] Quadros, G., Monteiro, E., Boavida, F.: A QoS Metric for Packet Networks. Proc. of SPIE International Symposium on Voice, Video, and Data Communications Conference (1998)
[4] Dorigo, M., Gambardella, L. M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation. (1997) 53–66
[5] Stutzle, T., Dorigo, M.: ACO algorithms for the quadratic assignment problem. New Ideas in Optimization. McGraw Hill. (1999) 33–50
[6] Chu, C., Gu, J., Hou, X., Gu, Q.: A Heuristic Ant Algorithm for Solving QoS Multicast Routing Problem. Evolutionary Computatioin, 2002. CEC'02. Proceedings of the 2002 Congress on, Vol. 2. (2002) 12–17

A Study of Internet Packet Reordering*

Yi Wang[1], Guohan Lu[2], and Xing Li[1]

[1] Department of Electronic Engineering, Tsinghua University
Beijing, 100084, P. R. China
wangyi@ns.6test.edu.cn
xing@cernet.edu.cn
[2] China Education and Research Network (CERNET)
Beijing, 100084, P. R. China
lvguohan@tsinghua.org.cn

Abstract. Packet reordering is a well-known phenomenon that the order of packets is inverted in the Internet. Previous studies indicates reordering can affect the performance of both the network and the packets receiver. Nevertheless, they get different results about the prevalence of reordering in the Internet. In this paper, we firstly present a methodology for single-point reordering measurement at a TCP receiver, including the algorithm and its implementation. Then we show the results of our three-week observation of reordering from a set of 10,647 Internet Web sites in China. In addition, we discuss a method to distinguish reordering and loss by making use of the distribution of their time lag and packet lag. Finally, we study the pertinence of sites experiencing reordering according to the network topology and propose a novel and relatively reliable approach to infer reorder-generating spots in the Internet.

1 Introduction

Packet reordering in the Internet is a well-known phenomenon. It is deceptively simple that packets can be reordered due to multi-path routing or parallelism at the routers. However, packet reordering is practically challenging to study, since it is a silent problem leaving little to no trace.

The IP layer in the Internet just provides a "best effort" datagram service. Although TCP is a reliable higher-layer protocol, packet reordering can affect it's performance and the efficiency of packet receiver for several reasons:

– **Causes Unnecessary Retransmission:** When the TCP receiver gets packets out of order, it sends duplicate ACKs to trigger fast retransmit algorithm at the sender [12]. These ACKs (3 or more) makes the TCP sender infer a packet has been lost and retransmit it. If the temporary sequence number gap is caused by reordering, then the duplicate ACKs and the fast retransmission are unnecessary and a waste of bandwidth.

* This work is supported by Cisco University Research Program.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 350–359, 2004.

- **Limits Transmission Speed:** When fast retransmission is triggered by duplicate ACKs, the TCP sender assumes it is an indication of network congestion. It reduces its congestion window (*cwnd*) to limit the transmission speed, which needs to grow larger from a "slow start" again. If reordering happens frequently, the congestion window is at a small size and can hardly grow larger. It results in a limited speed of packets transmission, and hence a throughput degradation [14].
- **Reduce Receiver's Efficiency:** Since the TCP receiver has to hand in data to the upper layer in order, when reordering happens, the receiver has to buffer all the out-of-order packets until getting all packets in order. Meanwhile, the upper layer gets data in burst rather than smoothly, which also reduces the system efficiency as a whole.

As the load of Internet grows, packet transmission equipments that do not guarantee FIFO are more and more used. It is worthwhile to know the frequency and magnitude of packet reordering in the Internet. Previous studies get discrepant results of the prevalence of reordering [1, 2, 3]. The reason partially lies in the methodological differences, and partially lies in the fast growth of the Internet. What is more, previous work on reordering mainly focused on the causes, dynamic characters and improving TCP performance in the face of reordering beyond measurement. No work has been done about correlation between reordering and the network topology to our knowledge. In this paper, we design a methodology which is not only suitable to common measurement of reordering, but also convenient for studying the above correlation.

The remainder of the paper is organized as follows. In Sect. 2, we review the related work. We propose our measurement methodology in Sect. 3, followed by Sect. 4 that shows our measurement results. Our novel approach to infer reorder-generating spots in the Internet is presented in Sect. 5. Finally, Sect. 6 concludes the whole paper.

2 Related Work

Previous studies of packet reordering can be approximately divided into two categories: general measurement study, which includes measurement methodology and experiment in the Internet; and specific topics on reordering, such as the causes, measurement techniques and metrics, improvement of TCP performance in the face of reordering, etc.

The first category of study is the fundamental of understanding packet reordering. Paxson's 1997 study [1] is based on a series of measurements taken between 35 Internet sites by transferring 100Kbyte TCP bulks on 1994 and 1995 separately. Paxson reports during two measurement periods, 36% and 12% of sessions experienced at least one reordering event respectively, and 2.0% and 0.26% of packets were reordered. Bennett et al's study work [3] in the year 1997-1998 at MAE-East network use a different approach in which they measure reordering by sending back-to-back ICMP-ping packets and evaluate the response. They

report that over 90% of packets were reordered during their two measurements of 140 Internet hosts. While Jaiswal et al's 2002 measurement in Sprint IP backbone observe a relatively lower rate of reordered packets of approximately 5% [2]. Instead of measuring end-to-end probe traces at the sender or receiver, they measure reordering at a single point within the backbone.

In the second category of study, Bennett et al's 1999 paper [3] attributes most reordering to "local parallelism". Liu's 2002 paper [5] does further discussion about the packet level parallelism. Bellardo and Savage describe a set of measurement techniques that can estimate one-way end-to-end reordering rates [4]. There are also some studies on modifications to TCP aiming better tolerance of reordering [8, 9, 10, 11].

3 Methodology

As mentioned in Sect. 1, what we are interested in is not only the frequency and magnitude of packet reordering in the Internet, but also the relationship between reordering and network topology. We propose a novel single-point and easy-implemented measurement methodology that can meet both the two aims without requiring either the control of both ends of the connections (e.g., see [1]) or the privilege of accessing the backbone (e.g., see [2]).

3.1 Measurement Environment

Our measurement uses a host in the CERNET[1] as the measurement point. We choose 10,647 web sites in China[2] as our data source, since the WWW (to be more precise, the HTTP on port 80) is the most widely used service in the Internet according to [13]. To the 10,647 web sites, we firstly did web page crawling (using *wget*) and measured forward-path[3] reordering twice a day from May 3 - May 12, 2003. Then we divided these sites into two categories: *Reorder Sites* (591 sites that experienced reordering at least once) and *Ordinary Sites* (10,056 sites that experienced no reordering). From May 16 - June 5, 2003, we did consecutive measurement comparison between the two categories. Every 3 hours, we measured reordering by crawling web pages from all the Reorder Sites. At the same time, we randomly crawled Ordinary Sites of the same number. When reordering was observed from an Ordinary Site, it was moved into the Reorder Sites category before the next measurement began.

3.2 Measurement Environment

Generally, we say that a packet is out-of-sequence if its Seq (TCP sequence number) is less than that of a previous received packet in the same connection. An

[1] China Education and Research Network

[2] These web sites are routed inside China according to the IP blocks routed inside China on March 2003 announced by CERNIC (`http://www.nic.edu.cn`).

[3] Define it as the direction from the web sites to the measurement host. The opposite direction is called the backward-path direction.

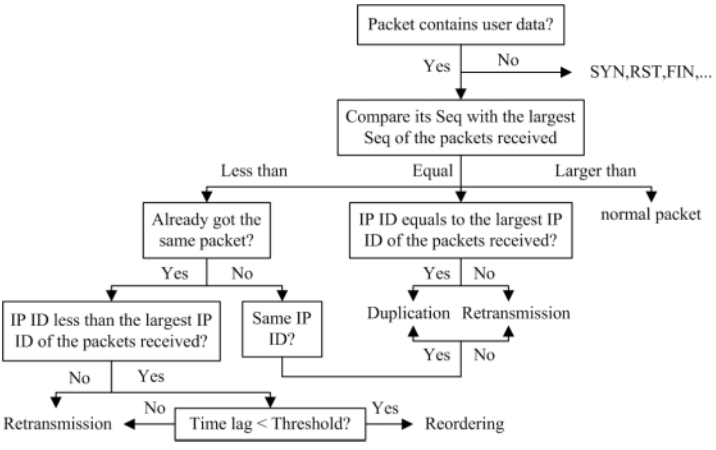


Fig. 1. Process of the reorder-judging algorithm

out-of-sequence packet could be the result of retransmission, network duplication or network reordering. These three causes are essentially different (see [2]). In this paper, not like many of previous studies, we focus on the network reordering which is mainly caused by parallelism within a router or a route change. Figure 1 shows the reorder-judging algorithm we proposed at the TCP receiver, which can distinguish most of out-of-sequence packets for different causes. It is based on Seq (TCP sequence number), IP ID and the time lag between packets [6]. Since the IPID of TCP wraps to 0 when monotonically increases to 65535, it is possible that the IPID of a retransmitted packet from a busy site is less than the previous lost one's. However, the time lag of the retransmitted packet must be much larger than a reordered packet because of the fast retransmission algorithm. So we set a threshold of 300ms to the time lag to distinguish reordered packet and retransmitted packet with IPID wrapped[4].

4 Results

In this section, we show the results of our measurement and discuss the approach to distinguish reordering and loss.

4.1 Measurement Data

In the three-week period (May 16 – June 5, 2003), we traced 208 thousand connections with totally 3.3 million data packets. 3.197% of all the packets were reordered. 5.79% of all 10,647 web sites (that is, 616 Reorder Sites) experienced

[4] Our measurement data shows it is a rare instance and the time lags of this kind of retransmit-ted packet are usually larger than 500ms. Figure 4 in Sect. 4.2 also shows 300ms is long enough for almost all the reordered packets to arrive.

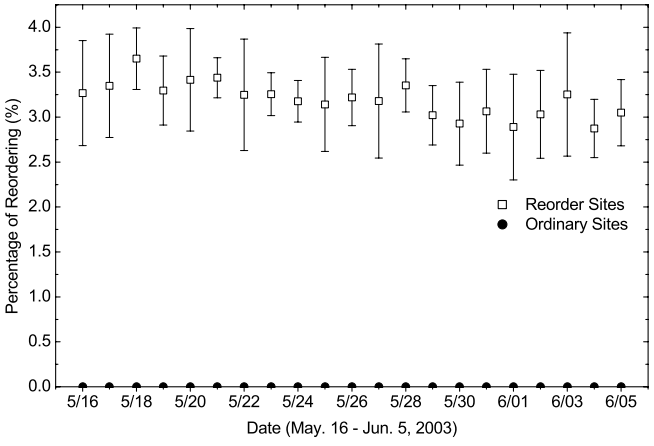


Fig. 2. Distribution of packet reordering from May 16 – June 5, 2003

Table 1. Reordering frequency of Reorder Sites

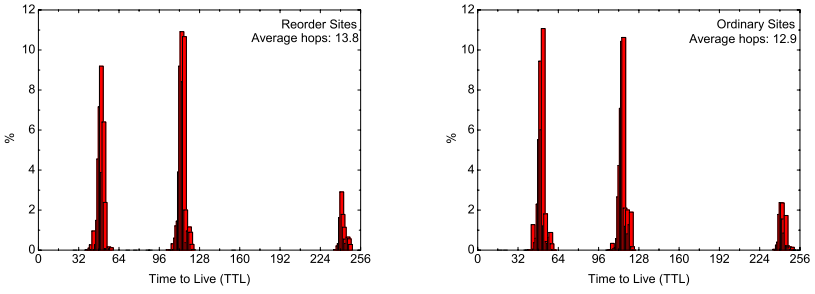
Reordering Freq.	>90%	80%–90%	70%–80%	60%–70%	50%–60%
Number of Sites	66	50	31	22	38
Percentage(%)	10.71	8.12	5.03	3.57	6.17
Reordering Freq.	40%–50%	30%–40%	20%–30%	10%–20%	0%–10%
Number of Sites	39	55	65	72	178
Percentage(%)	6.33	8.93	10.55	11.69	28.90

reordering at least once. Figure 2 shows the distribution of packet reordering over the entire duration of our measurement.

The discrepancy of reordering rate between the two site categories is huge and relatively steady. Reordering rate of Reorder Sites is between 2.39%–4.27% with a mean of 3.197%, while the reordering rate of Ordinary Sites is always below 0.14% with a mean of 0.017%. This discrepancy indicates that reordering is strongly site-dependent and occurs mainly in some certain parts of the Internet.

Table 1 summarizes the reordering frequency of the 616 Reorder Sites. About 20% of the Reorder Sites are with a reordering frequency higher than 80%. These sites contribute the bulk of reordering in our experiment.

Figure 3 shows the distribution of TTL (time-to-live) values of Reorder Sites and Ordinary Sites. Since TTL values are usually set to a few well-known values such as 64, 128 and 256, we can easily infer the distance from a web site to the measurement host in terms of the number of router hops. We find Reorder Sites (with average hops value 13.8) tend to be farther away from the measurement host than the Ordinary Sites (with average hops value 12.9).



(a) Reorder Sites, average hops: 13.8 (b) Ordinary Sites, average hops: 12.9

Fig. 3. Distribution of TTL

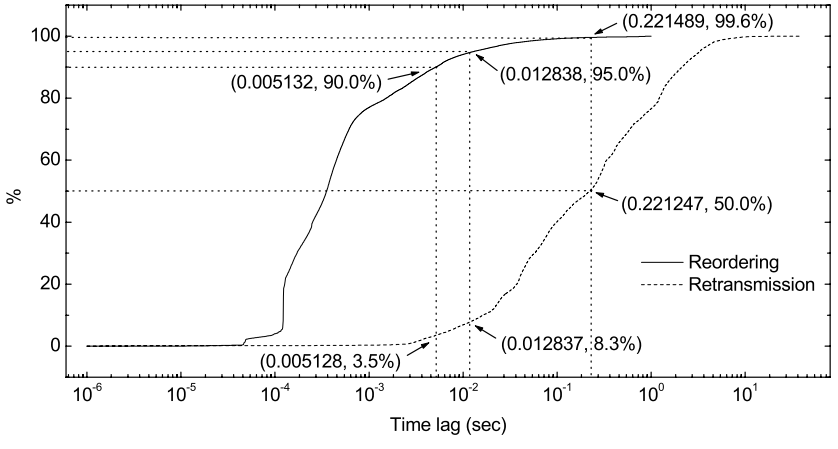
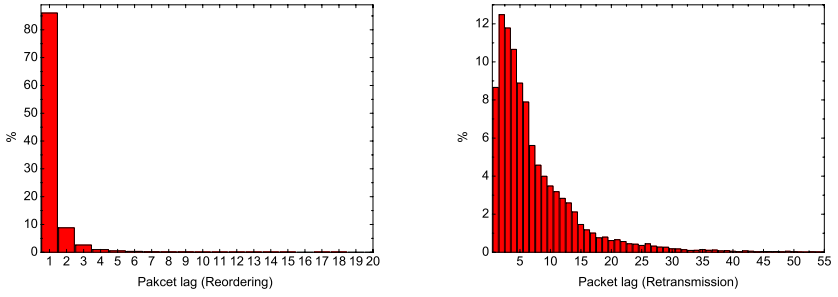


Fig. 4. Cumulative distribution of time lag of reordering and retransmission

4.2 Distinguishing Reordering and Loss

One of the main reasons that packet reordering affects the TCP performance is that, TCP would mistake reordering for loss when meeting sequence hole at the receiver. Since loss can not be confirmed until retransmitted packet arrived, we studied the time lag and packet lag of both packet reordering and retransmission. What we found indicates that we can distinguish them by setting certain threshold.

Figure 4 shows the cumulative distribution of time lag of both reordering and retransmission. 90% of reordered packets arrive at the receiver with time lag less than 5.1 ms, while only 3.5% of retransmitted packets arrive then. We find 12.8 ms is a relative good threshold in our experiment: 95% of reordered packets arrived but only 8.3% of retransmitted packets arrive.



(a) Reordering (b) Retransmission

Fig. 5. Distribution of packet lag

Figure 5 shows the packet lag of reordering and retransmission. 86.5% of reordered packets left behind only 1 packet and 95.3% of reordered packets left behind within 2 packets. Packet lag of retransmission has a dispersed distribution and a larger mean. About 78.8% of retransmitted packets left behind with a lag of 3 or more packets.

From the discussion about packet lag, we can evaluate the impact of reordering on TCP performance. Since over 95% of reordered packets are with a packet lag less than 3, there is only a little probability that reordering would trigger the TCP fast retransmit algorithm, which is consistent with [7]. However, as to some Internet paths suffering from serious reordering, knowledge of both time lag threshold and packet lag threshold can help improve TCP performance by distinguishing reordering and loss precisely.

5 Reordering and Network Topology

There are mainly two approaches to deal with packet reordering in the Internet. The one is to improve the TCP on end-hosts, making TCP more robust to reordering. The other is to improve the routers in the Internet which is the main cause of reordering. In this section, we discuss the approach to infer reordering-generating spots in the Internet.It is a prerequisite for the latter topic, on which little research is published.

A packet often goes through many routers from the sender to the receiver. When a packet arriving at the receiver reordered, we can not find out the reordering-generating spot without further information. However, if a router is a reordering-generating spot, all the packets transmitted by it may be reordered in theory. In our measurement, all the forward-paths from remote web sites to the local measurement host form a tree, in which the measurement host is the root, the routers are the middle nodes and the remote web sites are the leaves. If a router in the tree generates reordering, all its leaves may be affected. So we can infer

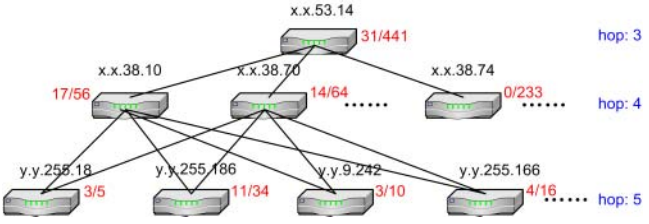


Fig. 6. Inferring the reorder-generating spots in the routing tree (Condition 1)

reorder-generating spot by studying the pertinence of leaves found as Reorder Sites. We use *traceroute* to gather the route information from the measurement host to all the 10,647 web sites. Then we generate a backward-path route tree in which the measurement host is also the root. Since the route architecture of CERNET is mainly symmetric, forward-path tree and backward-path tree within it could be approximately treated as the same. We introduce a metric R_r (reorder ratio) to every router as the main parameter for the reorder-generating spot judgment, which is defined by (1):

$$R_r = \frac{R}{T} \tag{1}$$

where, R is the number of Reorder Sites that go through the router and T is the total number of sites that go through it.

If a router has one of the below two characters, it is probably a reorder-generating spot in the network:

- The router's R_r is by far higher than its previous-hop's and other routers' of the same hop. Its next-hop routers also have got high and close R_r.
- All of the router's previous-hop routers have got high R_r, and its next-hop routers also get high R_r.

As Fig. 6 shows, the R_r of the 4th hop routers x.x.38.10 and x.x.38.70 are obviously higher than the 3rd hop x.x.53.14íúFirstly, since none of the 233 sites which goes through the 4th hop router x.x.38.74 experienced reordering, it is impossible that the reordering-generating spot locates at the 3rd hop or its previous ones. Moreover, the routers of 5th hop have got high and close R_r. So the two IP of 4th hop are probably the same router with "local parallelism". In fact, x.x.38.10 and x.x.38.70 are the same equipment in CERNET Super Computer Center with two gigabit paths connected to the 3rd router x.x.53.14. The several 5th hop routers are all star connected to a GSR (Gigabit Switch Router). So we come to the conclusion that the reorder-generating spot in Fig. 6 locates between the 3rd router x.x.53.14 and the 4th hop routers with IP x.x.38.10 and x.x.38.70.

As Fig. 7 shows, the 4 routers of 8th hop and the 6 routers of 10th hop which connects to the two 9th hop routers have got high and close R_r[5]. Since

[5] The 10th hop router with IP: y.y.139.214 is an exception, which contains too little data and could be neglected.

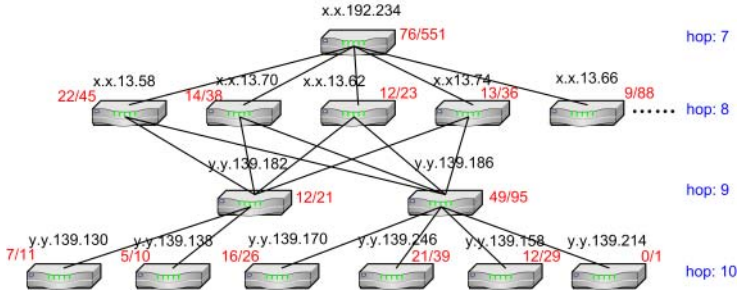


Fig. 7. Inferring the reorder-generating spots in the routing tree (Condition 2)

the R_r(0.102)of 8th hop router 202.96.13.66 is much lower than the other four R_r(with an mean of 0.435), routers previous to hop 7 can not be the reorder-generating spot. On the other hand, all the 6 routers of 10th hop are connected to the 9th routers with a single path. It is very unlikely that they all generate reordering at the same time and result in the relationship of in Fig. 7. So it is most probably that the two IP close to each other belongs to the same router with "local parallelism". And the 8th hop and 9th hop routers along with the multi-paths between them are probably the reorder-generating spot in Fig. 7. Although the routers in Fig. 7 are not in CERNET, the approach above has general significance as long as we know the forward-path route tree. It might not exactly locate where the reorder-generating spot is, but it does can help us a lot to exclude reduce the scope. Further study may introduce multi-point observation to gather more route information.

6 Conclusions

This paper provides an insight into both the attributes of packet reordering in the Internet itself, and the relationship between reordering and network topology. Firstly, a measurement methodology with a single-point reorder-judging algorithm is proposed, which is suitable for the above two purposes. Then, measurement results of 208 thousand connections with totally 3.3 million data packets are presented, in which about 3.2% of all the packets are observed reordered. We find that reordering is not prevalent in the entire Internet but significantly site-dependent. We also note that certain threshold can be found to effectively help distinguish reordering and loss on some heavily reordering paths. Nevertheless, the distribution of packet lag of reordering and retransmission implies that in most cases reordering will not trigger the fast retransmission algorithm, thus will not affect the TCP performance seriously. Moreover, we propose a novel and relatively reliable approach to infer reorder-generating spots in the Internet by studying the pertinence of reordering sites and the routing tree. It is a first step of our work to analyze the relationship between packet reordering and network topology. Currently, deployment of multi-point reordering measurement

with more precisely data crawling is considered to overcome the limitations of single-point observation at the TCP receiver, such as the exact forward-path route tree is unknown and the possible difficulty of inferring reorder-generating should there be a reorder-generating spot very close to the root of the route tree.

References

[1] Paxon, V.: End-to-end Internet Packet Dynamics. IEEE/ACM Transactions on Networking, Vol. 7, Issue 3. (1999) 277–292 351, 352

[2] Jaiswal, S., Iannaccone, G., Diot, C., et al.: Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone. Sprint ATL Technical Report TR02-ATL-071121 (2002) 351, 352, 353

[3] Bennett, J. C. R., Partridge, C., Shectman, N.: Packet Reordering is Not Pathological Network Behavior. IEEE/ACM Transaction on Networking, Vol. 7, Issue 6. (1999) 789–798 351, 352

[4] Bellardo, J., Savage, S.: Measuring Packet Reordering. Department of Computer Science and Engineering, University of California at San Diego (2002) 352

[5] Liu, H.: A Trace Driven Study of Packet Level Parallelism. Proc. International Conference on Communications (ICC), New York, NY (2002) 352

[6] Lu, G. H., Li, X.: On the Correspondency between TCP Acknowledgment Packet and Data Packet. Proc. Internet Measurement Conference (IMC), Miami Beach, FL, USA (2003) 285–294 353

[7] Iannaccone, J., Jaiswal, S., Diot, C.: Packet reordering inside the Sprint backbone. Sprint ATL Technical Report TR01-ATL-062917 (2001) 356

[8] Floyd, S., Henderson, J.: The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582 (1999) 352

[9] Floyd, S., Mahdavi, J., Mathis, M., et al.: An Extension to the Selective Acknowledgement (SACK) Option for TCP. RFC 2883 (2000) 352

[10] Zhang, M., Karp, B., Floyd, S., Peterson, L.: Improving TCP's Performance under Reordering with DSACK. Technical Report TR-02-006, International Computer Science Institute (2002) 352

[11] Blanton, E., Allman, M.: On Making TCP More Robust to Packet Reordering. ACM Computer Communication Review, Vol. 32, Issue 1. (2002) 352

[12] Stevens, W.: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001 (1997) 350

[13] McCreary, S., Claffy, K.: Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange. `http://www.caida.org/outreach/papers/2000/AIX0005/` 352

[14] Laor, M., Gendel, L.: The Effect of Packet Reordering in a Backbone Link on Application Throughput. IEEE Network, Vol. 16, Issue 5. (2002) 351

Policy-Based Differentiated QoS Provisioning for DiffServ Enabled IP Networks*

Si-Ho Cha[1], Jong-Eon Lee[2], WoongChul Choi[2],
Jae-Oh Lee[3], and Kuk-Hyun Cho[2]

[1] Dept. of NMS, Network Business Group, WarePlus Inc., Korea
sihoc@wareplus.com
[2] Department of Computer Science, Kwangwoon University, Korea
{jelee,khcho}@cs.kw.ac.kr
wchoi@daisy.kw.ac.kr
[3] Department of Computer Engineering
Korea University of Technology and Education, Korea
jolee@kut.ac.kr

Abstract. This paper proposes and implements a policy-based QoS management platform for DiffServ enabled IP networks, which specifies QoS policies to guarantee QoS requirements. The proposed platform integrates the function of policy-based management and QoS monitoring by extending the original IETF policy-based management architecture. High-level QoS policies are represented as valid XML documents and are translated to EJB beans in the EJB-based policy server of the platform. The policy distribution and the QoS monitoring are processed using SNMP. This paper also describes the implementation of Linux-based DiffServ routers with DiffServ MIB and analyzes the experimental results using a video streaming system.

1 Introduction

The best-effort service model in current IP networks does not provide the QoS requirements of QoS-sensitive services. To solve this problem, the IETF (Internet Engineering Task Force) proposed two models of IntServ [1] and DiffServ [2]. IntServ model is based on per-flow resource reservation and admission control through RSVP (Resource Reservation Protocol). The main disadvantage of IntServ is that the required information of flow states and the QoS treatments in a core IP network raise severe scalability problems. DiffServ model, on the other hand, supports aggregated traffic classes rather than individual flows and provides different QoS to different classes of packets in IP networks. However, it is possible to lead to serious QoS violations without a QoS management support. From this reasoning, a QoS management system that can manage differentiated QoS provisioning is required. There are several research projects for QoS guarantees using the policy-based management technology going on, but only few of

* The present Research has been conducted by the Research Grant of Kwangwoon University in 2003.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 360–369, 2004.

them detail the design and implementation issues of a QoS management platform with policy concepts.

Therefore, we propose and implement a policy-based QoS management platform for DiffServ enabled IP networks, called SEMA-Q. The objective of the policy-based QoS management is to manage the QoS of connections using high-level policies that describe the behavior of the network in a way as independently as possible of network devices and topology [3]. The amount of QoS management task can be reduced by using policies because one policy can be used for many policy targets that are various network nodes. The QoS management procedures of the SEMA-Q consist of the functions of topology discovery, definition and validation of high-level policy, conversion of high-level policy to low-level policy, deployment of low-level policy, and QoS monitoring. In the SEMA-Q, high-level QoS policies are represented in valid XML documents and are translated to EJB beans in an EJB-based policy server of the SEMA-Q. A policy consists of a policy ID, a source or a source group, a destination or destination group, a router or a router group, a application type, a time period, and a service level. The service level is set to one of the values of premium, gold, silver, or bronze. The premium service is provided using an Expedited Forward (EF) PHB, whereas the gold and the silver service are provisioned to Assured Forwarding (AF) PHB groups of a DiffServ network. The bronze service is offered using the best-effort service of a network. We configure Linux-based DiffServ routers [4] and implement SNMP agents with DiffServ MIB on the DiffServ routers. One note is that the standard protocol for policy distribution of the IETF PBM architecture uses COPS (Common Open Policy Service) [5]. While most DiffServ routers support SNMP only, few routers support COPS, therefore, current implementation of the SEMA-Q uses SNMP to distribute QoS policies. However, because the SEMA-Q employs component-based platform, COPS can be easily included in the platform.

This paper is structured as follows. Section 2 discusses the architecture and components of the proposed SEMA-Q. Section 3 presents the implementation of the Liuux DiffServ router and the SEMA-Q and the experimental results in the video streaming system. Finally in section 4 we conclude the paper.

2 Design

2.1 SEMA-Q Architecture and Components

The architecture of SEMA-Q is shown in Fig. 1. The SEMA-Q conforms to the Model-View-Controller (MVC) architecture. Therefore, it is highly manageable and scalable, and provides the overall strategy for the clear distribution of objects involved in managing service. There are two main components in the architecture: a Web server and an EJB-based policy server. A Web server is responsible for the presentation logic of the SEMA-Q. An EJB-based policy server is responsible for the business logic of the SEMA-Q. The SEMA-Q provides a Web-based interface for a network administrator to create and revise high-level QoS policies to be enforced on the DiffServ network. The SEMA-Q uses Java Servlets, JSP template/pages, and XML technologies to provide for the administrator's view.

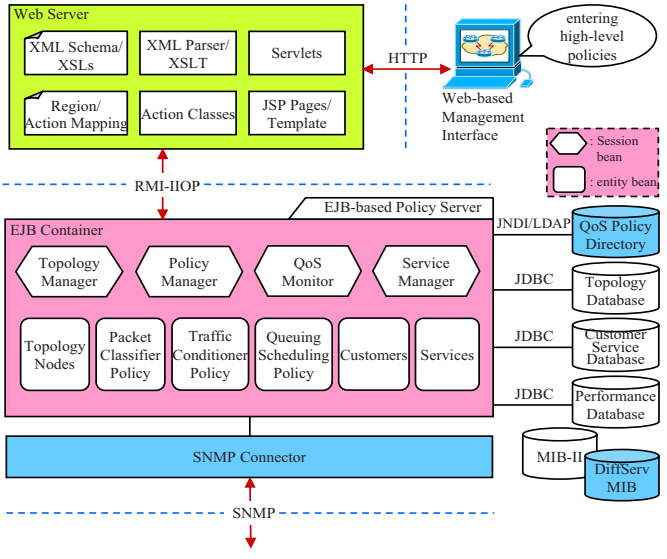


Fig. 1. The architecture of SEMA-Q

As illustrated in Fig. 1, there are several functional components in the EJB-based policy server. The SEMA-Q uses the following components to discover network topology and each router type.

- The topology node (TN) bean is an entity bean containing the information of a network topology and each router type. The information is retrieved using SNMP MIB-II.
- The topology database (TD) stores the topology information and each router type retrieved from a DiffServ network through SNMP.
- The topology manager (TM) bean is a session bean responsible for discovering the topology information and each router type and storing them into the TD and setting up the TN beans according to the retrieved information.

The SEMA-Q uses the following components to translate XML documents, high-level QoS policies, into EJB beans, low-level QoS policies, and deploy them to the DiffServ network.

- The packet classification policy (PCP) bean is a part of low-level QoS policy entity beans that classifies packet flows and assign class identifiers to them. The PCP beans are deployed to edge routers and control the inbound traffics.
- The traffic conditioning policy (TCP) bean is a part of low-level QoS policy entity beans that meters the classified packets to check whether they conform to a traffic profile and performs marking, dropping, and/or shaping packets according to the metering results. The TCP beans are deployed to edge routers and control the outbound traffics.

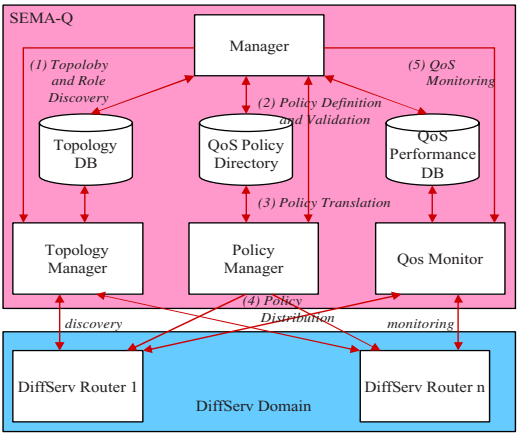


Fig. 2. QoS management process of SEMA-Q

- The queuing and scheduling policy (QSP) bean is a part of low-level QoS policy entity beans. It performs queuing, scheduling, and/or dropping packets. The QSP beans are deployed to core routers to control the outbound traffics.
- The QoS policy directory is a directory storing the a part of low-level QoS policy entity beans.
- The policy manager (PM) bean is a session bean that is responsible for translating high-level QoS policies into low-level QoS policy beans and setting the values of DiffServ MIBs of the routers. The PM is also responsible for deploying the low-level QoS policies to relevant routers in the DiffServ network.
- The QoS monitor (QM) bean is a session bean that is responsible for monitoring the QoS resulted from a policy deployment by retrieving the values of DiffServ MIBs and comparing them to the attribute values of the three low-level QoS policy beans.

Also the EJB-based policy server uses Java Database Connectivity (JDBC), Java Naming and Directory Interface (JNDI) / Lightweight Directory Access Protocol (LDAP), and SNMP connector to support QoS management actions.

2.2 QoS Management Process

To process a policy-based QoS management efficiently and correctly, the following procedures are required: topology discovery, policy definition and validation, policy translation, policy deployment, and QoS monitoring. The QoS management process of SEMA-Q is shown in Fig. 2.

1. Topology discovery: In order to describe the QoS of a DiffServ network, the SEMA-Q should have the knowledge of the routing topology and each

router's role. The TM session bean accomplishes the discovery of the routing topology and router type discovery by using two SNMP MIB-II tables, ipAddrTable and ipRouteTable. The ipAddrTable contains IP addresses of all network interfaces in a router and the ipRouteTable contains an IP routing table that has a next hop host and a network interface for a set of destination IP addresses. The topology and router information discovered from the network are stored in the TD, and are represented as TN entity beans.

2. Policy definition and validation: The SEMA-Q defines HQPs as valid XML documents and validates the XML documents. A Java Servlet on the Web server receives QoS policy data from a Web browser and creates valid XML documents and then validates them. Once the HQPs are validated, a Java Servlet requests a PM session bean to create the instances of LQP entity beans, PCP bean, TCP bean, and QSP bean.
3. Policy translation: The translation of a QoS policy from HQPs to LQPs is done by the PM session bean on the EJB-based policy server. The PM session bean translates HQPs to LQPs by properly setting the attributes of the three LQP entity beans. The attributes of the LQP entity beans are mapped into the device configuration parameters to configure the DiffServ routers for provisioning QoS requirements.
4. Policy deployment: The deployment of LQPs is done by the three LQP entity beans. These three LQP entity beans perform SNMP operations for deploying each LQP. The PCP bean and the TCP bean are deployed to edge routers to control the functions of the edge routers, whereas the QSP bean is deployed to core routers to control the functions of the core routers. The PCP bean classifies packet flows and the TCP bean performs the traffic conditioning such as metering, marking, dropping, and/or shaping packets. The QSP bean performs queuing, scheduling, and/or dropping packets. A set of these actions is accomplished by using the DiffServ MIB. The DiffServ MIB describes a configuration and management aspect of DiffServ routers.
5. QoS monitoring: A deployed QoS policy might not behave as defined in the policy. The QoS monitoring in the SEMA-Q uses the same DiffServ MIB as in the policy deployment. The QM session bean accesses the policy definition in the three LQP beans and compares the observed behavior of a network to the one defined in the policy. If any QoS degradation is observed, the QM session bean notifies an administrator by alerting messages and updates the performance database.

3 Implementation

3.1 Linux-Based DiffServ Routers

Linux-based routers [4] are used for our DiffServ network testbed. Supporting differentiated services are already incorporated in the mainstream Linux kernel source code version 2.4 and later. architecture of a Linux DiffServ router. An SNMP agent with a MIB-II and a DiffServ MIB receives management operations

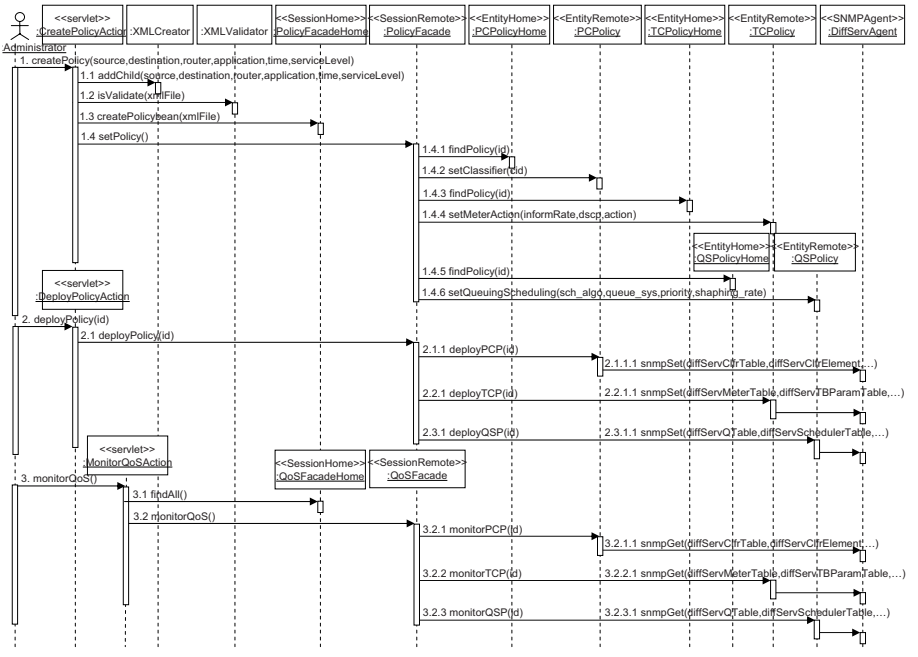


Fig. 3. Sequence diagram for QoS management

from the EJB-based policy server and performs appropriate parameter changes in the Linux traffic control (TC) kernel. Communication between the SNMP agent and the Linux TC kernel is achieved through Netlink socket [6]. An SNMP agent containing a MIB-II and a DiffServ MIB has been implemented by using the UCD-SNMP package 4.2.2 [7] that provides an agent extension capability.

3.2 SEMA-Q Platform

The SEMA-Q is implemented on a Windows 2000 server system. It consists of a Web server and an EJB-based policy server. We use Apache Tomcat 4.0.1 [8] for the Servlet and JSP container.

An EJB-based policy server within the business-tier runs an EJB server to manage EJB components. We use JBoss 2.4.10 [9] for an EJB-based policy server and use EJB 1.1 to implement EJB beans. AdventNet SNMP APIs [10] written in Java are used for handling SNMP operations. The Oracle 8i Enterprise Edition 8.1.6 [11] is used for storing the performance and topology information derived from MIB tables. In the MVC architecture, the view is implemented using the JSP template mechanism and the Composite View pattern. The controller is implemented using the Front Controller pattern (a Action Servlet) and the Session Façade pattern (EJB Session Beans). The model is implemented using EJB Entity Beans and Service Locator pattern.

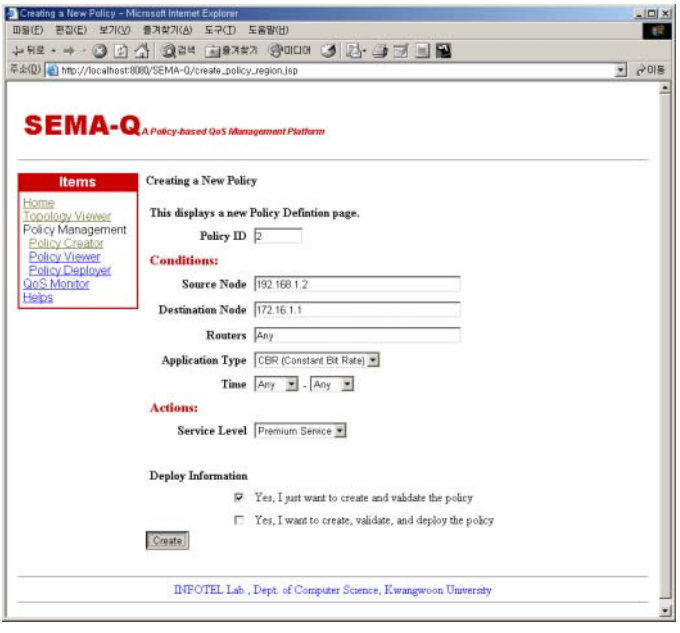


Fig. 4. Snapshot of the SEMA-Q

Fig. 3 shows a UML sequence diagram to implement the main QoS management of the SEMA-Q. In the frist part of the sequence diagram, `CreatePolicyAction` asks `XMLCreator` class and `XMLValidator` class to create and validate a high-level policy, respectively. `CreatePolicyAction` calls `addChild` method of `XMLCreator` class and then calls `isValidate` method of `XMLValidator` class. And then `CreatePolicyAction` asks `PolicyFacadeHome` home interface and `PolicyFacade` session bean for creating the low-level policy beans and setting the attributes of the low-level policy beans. `PolicyFacade` session bean calls `setClassifier`, `setMetterAction`, and `setQueuingScheduling` methods on the appropriate low-level policy beans to set the attributes of the low-level policy beans. In the second part, `DeployPolicyAction` asks `PolicyFacade` session bean to deploy the policies. `PolicyFacade` session bean then calls `deployPCP`, `deployTCP`, and `deployQSP` method of the corresponding low-level policy beans, respectively. Each low-level policy bean calls `snmpSet` method to deploy the policy. In the last part, `MonitorQoSAction` asks `QoSFacade` session bean to monitor QoS. `QoSFacade` session bean then calls `monitorPCP`, `monitorTCP`, and `monitorQSP` methods. Each low-level policy bean also calls `snmpGet` method on the DiffServ MIB to retrieve the QoS data. `XMLCreator` class and `XMLValidator` class are utility classes to create and validate XML policy documents using Apache's Xerces Java DOM-based parser. Fig. 4 shows the input forms for the high-level policy information of the SEMA-Q.

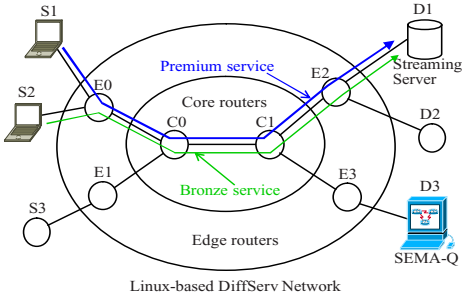


Fig. 5. Experiment Environment

3.3 Experiments

To show the effectiveness of the SEMA-Q, we apply a video streaming server based on Windows Media Streaming Services [12] to our DiffServ network. We configure a testbed shown in Fig. 5.

A VOD server and a policy server are attached to D1 and D3 in the network, respectively. The systems in the testbed are running on the following hardware configurations. The core routers are running on Pentium IV 1.8 GHz with 512 MB main memory, the edge routers on Pentium IV 1.5 GHz with 512 MB main memory, a VOD server on Pentium IV 2.0 GHz with 512 MB main memory, and the other systems on Pentium III 1.0 GHz with 256 MB main memory. All the links in Fig. 5 are connected via FastEthernet NICs.

In the configuration, there are three connections running - two for multimedia connections and the other one for cross traffic. Two connections for multimedia traffics are the connection between S1 and D1, and the one between S2 and D1. Those connections share a link between E0 and E2. To differentiate the services between them, the connection between S1 and D1 is applied by Premium service, while the other multimedia connection between S2 and D1 is applied by Bronze service. To make the sharing link congested, MGEN toolset [13] is used to generate cross traffics on that link, and CBR traffics are used to achieve that goal. Cross traffics are generated at E0 and sinked at E2 router. By doing this, the service levels and the resulted QoS can be explicitly demonstrated.

Fig. 6 shows the results of the experiment. The two figures of Fig. 6(a) and 6(c) are the screenshots of the client windows of each connection, and the two figures of 6(b) and 6(d) are the those of the statistics windows of each connection. Fig. 6(a) and 6(b) are the snapshots of the connection with Premium Service between S1 and D1. Fig. 6(c) and 6(d) are those of the connection with Bronze Service between S2 and D1. As shown in Fig. 6, the difference in the video quality of each connection is explicit. The client S1 with Premium service receives a video stream with bitrate 530.3 kbps and video quality 24.8 fps, while the client S2 with Bronze service receives a video stream with bitrate 245.6 kbps and video quality 14.9 fps.

(a) (b)

(c) (d)

Fig. 6. Snapshots of C1 and C2

From the experiment, we can verify that the SEMA-Q provides differentiated QoS levels to the contending connections using the management platform. Obviously, this work can be extended to a network with more complicated connections.

4 Conclusion

In this paper, we proposed and implemented a policy-based QoS management platform for DiffServ enabled IP networks, called SEMA-Q. The SEMA-Q integrated the functions of policy management and QoS monitoring by extending the original IETF PBM architecture to the policy-based QoS management. We also presented the policy-based QoS provisioning, the QoS management mechanism, and the QoS management procedures of the SEMA-Q.

To show the effectiveness of our SEMA-Q platform, we experimented with video streaming services in our Linux-based DiffServ testbed. In the experiment, we demonstrated that our SEMA-Q is able to manage differentiated QoS provisioning in a DiffServ network. We expect our SEMA-Q to be successfully integrated in the service management systems used by the service providers in order to meet various dynamic QoS requirements from their customers.

References

[1] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, IETF RFC 1633, June 1994. 360

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang W. Weiss, An Architecture for Differentiated Services, IETF RFC 2475, December 1998. 360

[3] R. Yavatkar, D. Pendarakis, R. Guerin, A Framework for Policy-based Admission Control, IETF RFC 2753, January 2000. 361

[4] Differentiated Services on Linux, `http://diffserv.sourceforge.net`. 361, 364

[5] J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, The COPS (Common Open Policy Service) Protocol, IETF RFC 2748, January 2000. 361

[6] G. Dhandapani, A. Sundaresan, Netlink Sockets - Overview, University of Kansas, September 1999, `http://qos.ittc.ukans.edu/netlink`. 365

[7] UCD-SNMP Package, University of California at Davis, `http://www.net-snmp.org/`. 365

[8] Jakarta Tomcat 4.0.1, `http://jakarta.apache.org/tomcat/`. 365

[9] JBoss Application Server, `http://www.jboss.org/`. 365

[10] AdventNet SNMP API, `http://www.adventnet.com/products/`. 365

[11] Oracle 8i Enterprise Edition, `http://otn.oracle.com/software/content.html`. 365

[12] Windows Media Services, `http://www.microsoft.com/korea/windows/windowsme-dia`. 367

[13] The Multi-Generator Toolset, `http://manimac.itd.nrl.navy.mil/MGEN/`. 367

Deterministic Edge-to-Edge Delay Bounds for a Flow in a DiffServ Network Domain

Geunhyung Kim and Cheeha Kim

Department of Computer Science and Engineering
Pohang University of Science and Technology, Pohang, Korea
{geunkim,chkim}@postech.ac.kr

Abstract. It is important to understand delay bounds of an individual flow in the Internet in order to provide real-time applications such as Voice over IP (VoIP). In this paper, we study the deterministic bounds on edge-to-edge delay of a flow in a Differentiated Service (DiffServ) network domain with FIFO aggregation and class-based Guaranteed-Rate (GR) scheduler which provides guaranteed performance with rate reservation for a traffic class. We derive edge-to-edge delay bounds of a flow as a function of service rate allocated for a traffic class and leaky-bucket parameters adopted for flows at the network ingress, and information about joining and leaving flows. The resulting bounds are less than previously derived delay bounds, especially in the case where the burstiness of aggregated traffic is high and the edge-to-edge path is long.

1 Introduction

In the current Internet, only best-effort service is provided. However, with Quality of Service (QoS) requirements raised by a wide range of real-time multimedia applications, best-effort service is no longer sufficient. In addition, with commercialization of the Internet, network service providers are motivated to provide communication services to customers with QoS guarantees, such as end-to-end delay and throughput guarantees. Specially, since real-time applications require very stringent delay guarantees, network delay is an important parameter for these real-time applications.

To provide QoS guarantees on the Internet, the Internet Engineering Task Force (IETF) has considered a number of architecture extensions to the current Internet. Because of its potential scalability in support of QoS guarantees in the Internet, the DiffServ architecture with aggregate packet scheduling [1] has recently attracted much attention in the networking community and is widely accepted as a feasible solution for providing Internet QoS. However, since flows are treated aggregately and resources are allocated for a Per-Hop-Behavior (PHB), supporting per-flow QoS guarantees in a DiffServ network domain requires additional mechanisms such as edge-to-edge signaling or admission control.

Recently, to achieve per-flow bandwidth guarantees without per-flow signaling of core routers in the DiffServ architecture, [2] uses aggregate reservation

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 370–379, 2004.

along the path from ingress router to egress router to reduce the signaling demands placed on core routers. However, since required bandwidth is ensured by explicit aggregate bandwidth reservation, there remains a problem concerning whether it is possible to provide edge-to-edge delay guarantees for a single flow, in the case where only aggregate scheduling is implemented in the core.

Several studies [3, 4, 5, 6, 7] have been conducted to investigate edge-to-edge delay bounds in the DiffServ network domain with FIFO aggregation. The studies in [3, 4, 5] have shown that delay bounds in a network with FIFO aggregation depend on the utilization level and the number of hops and that overall network utilization level must be limited to a small fraction of its link capacities to guarantee an edge-to-edge delay. However, the edge-to-edge delay bounds of these studies are very conservative and rough, since they do not consider real aggregation information. Other research [6, 7] demonstrates that providing good delay bounds may depend on complex global conditions. But this approach [6, 7] requires complex traffic conditioning at the network entry according to the variation in the number of flows joining on any output link along the path.

In this paper, we develop a service-curve [8] based methodology that can be applied to arbitrary topology networks with arrival traffic constraint and routing information. This enables us to obtain edge-to-edge delay bounds which are tighter than any result of other studies [3, 4]. We believe the information about joining and leaving flows and the impact of aggregating flows on the delay of the target flow must be counted for the tighter edge-to-edge delay bounds. These tight edge-to-edge delay bounds are useful to employ admission control for guaranteed service traffic.

The remainder of this paper is organized as follows. In the next section, we review the definitions of deterministic network calculus and represent notations, assumption and network model used in the paper. In Section 3, we derive deterministic edge-to-edge delay bounds in two specific cases and in a general case. In Section 4, a network topology and scenarios for analytical comparison is introduced and analytical results for edge-to-edge delay bounds are presented. Finally, in Section 5, the findings of the paper are summarized.

2 Network Calculus, Notations and Network Model

Network calculus [8] is a framework for analysis and maintenance of deterministic QoS guarantees in packet switched networks. Denote by $A_i(\tau, t)$ the amount of flow i arriving during the interval $[\tau, t]$ and by $S_i(\tau, t)$ the amount of service received by flow i during the same interval. For ease of exposition, we simply use $A_i(t)$ to denote $A_i(0, t)$ and $S_i(t)$ to denote $S_i(0, t)$.

Given a wide-sense increasing function α defined for $t \geq 0$, we say that a flow i is constrained by α if and only if $A_i(t) - A_i(s) \leq \alpha(t - s)$ for all $s \leq t$. Then the flow i is said to have α as an arrival curve or is said to be α-smooth [8]. For example, a flow i is said to be (ρ_i, σ_i)-constrained or leaky-bucket constrained where ρ_i is the sustainable rate and σ_i is the burst parameter of a flow i, if it satisfies $A_i(\tau, \tau + \delta) \leq \rho_i \delta + \sigma_i$ for all $\tau \geq 0$ and $\delta \geq 0$.

Table 1. Notations used in the paper

Notation	Meaning
$\mathcal{P}_i$	the set of nodes on the edge-to-edge path of flow i
$\mathcal{F}_i^k$	the set of flows aggregated with flow i at node k
$\mathcal{P}_{i,j}$	the sub-path, shared by both flow i and flow j, of path $\mathcal{P}_i$
$M_{i,j}$	the first node of sub-path $\mathcal{P}_{i,j}$
$A^k(t)$	the amount of aggregated traffic arriving to the output link of node k
$S^k(t)$	the amount of aggregated traffic serviced on the output link of node k
$A_i^k(t)$	the amount of flow i traffic arriving to the output link of node k
$S_i^k(t)$	the amount of flow i traffic serviced on the output link of node k
R^k	the allocated bandwidth on the output link for traffic class at node k
T^k	the latency caused by scheduler at node k
θ^k	the latency experienced by target flow at node k
$B_i(\mathcal{P}_{i,j})$	the effective service bandwidth of flow i on the sub-path $\mathcal{P}_{i,j}$
$B_i^k(\mathcal{P}_{i,j})$	the effective service bandwidth of flow i at node k on the sub-path $\mathcal{P}_{i,j}$

We have thus far described the bounds on the input flow to a system. We next consider placing lower bounds on the service that the system provides to its input flow. We say that system $\mathcal{R}$ offers to its input flow i a minimum service curve β if and only if for all $t > 0$, there exist some $s \geq 0$, with $s \leq t$, such that $S_i(t) - A_i(s) \geq \beta(t-s)$ [8]. In addition, the scheduler is a Latency-Rate (LR) server [9] for a flow i with rate R and latency T if and only if at every instant t in the busy period, it guarantees that $S_i(\tau, t) \geq R \cdot (t - \tau - T)^+$ where T is the minimum non-negative number (the latency of the server) satisfying the above inequality and $(x)^+ := max\{0, x\}$. Based on these definitions, we derive the edge-to-edge delay bounds of a flow in a DiffServ network domain in the subsequent sections using the notations given in Table 1.

We assume that two flows share only on sub-path segment. Note that we consider one traffic class since most traffic classes are separable.

We define the effective service bandwidth of an aggregated flow to analyze the impact of the aggregation on the delay of target flow. The effective service bandwidth is considered as the service rate transmitting the flow aggregated with target flow along the sub-path. The effective service bandwidth of a flow i at the node k on the sub-path $\mathcal{P}_{i,j}$, $B_i^k(\mathcal{P}_{i,j})$ is defined as $R^k - (\sum_{s \in \{\mathcal{F}_j^k - \mathcal{F}_j^{k-1}\}} \rho_s) \cdot I_{\{k \neq M_{i,j}\}}$, where $I_{\{E\}}$ denotes the indicator function for the event E. The effective service bandwidth of a flow i on the sub-path $\mathcal{P}_{i,j}$, $B_i(\mathcal{P}_{i,j})$ is defined as $min_{k \in \mathcal{P}_{i,j}} B_i^k(\mathcal{P}_{i,j})$.

We consider a single DiffServ network domain with DiffServ routers, which are output buffered and implement aggregate class-based LR scheduling, such as Packet-level Generalized Processor Sharing (PGPS) or Weighted Fair Queueing (WFQ). In addition, we assume that every flow is constrained by a leaky-bucket when it arrives to the ingress router and is transmitted along single path from an ingress router to an egress router. We denote this single path as edge-to-edge path. Because of the aggregation nature of DiffServ network, other flows may

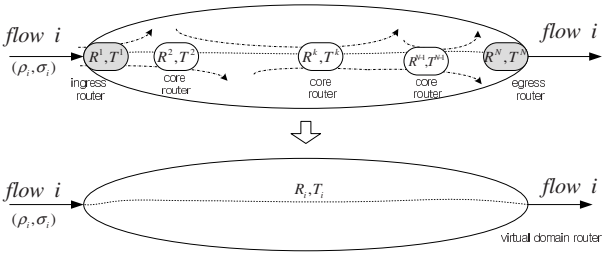


Fig. 1. A per-domain LR server for a flow with flow aggregation

join and leave the edge-to-edge path, which target flow i traverses, as depicted in the upper side of Fig 1.

According to [8, 9, 10, 11], given the arrival curve of an input flow and two parameters of LR server, the worst-case delay bounds experienced by the input flow at the server can be obtained. Hence, We define the virtual domain router which is modelled as a per-domain LR server, representing edge-to-edge service curve, depicted in the lower side of Fig 1. In this paper, we derive a worst-case edge-to-edge delay bounds of target flow i along edge-to-edge path using an edge-to-edge service curve of the per-domain LR server. The service curve of per-domain LR server, represented by edge-to-edge service rate R_i with latency T_i, reflects the impact of burstiness and arrival rate of flows aggregated with flow i along the path $\mathcal{P}_i$.

3 Deterministic Edge-to-Edge Delay in DiffServ Network

In this section, we present the existing result regarding minimum service curve in a general case and obtain a service curve, delay bounds and increased burstiness for a flow in a node where several leaky-bucket constrained flows aggregate. Then, we derive edge-to-edge delay bounds in two specific cases and in a general case.

Theorem 1 (Minimum Service Curves [8]). *Consider a lossless node serving two flows, 1 and 2, aggregated in the FIFO order. Assume that the node guarantees a minimum service curve β to the aggregate of two flows and the arrival curve of flow 2 is α_2-smooth. Then, for any $\theta \geq 0$, the flow 1 is guaranteed the minimum service curve β_1, where*

$$\beta_1(t) = [\beta(t) - \alpha_2(t-\theta)]^+ \cdot I_{\{t>\theta\}}$$

Consider a LR server with rate R and latency T serving two flows aggregated in the FIFO manner and each flow i is (ρ_i, σ_i)-constrained, $i = 1, 2$. Then, the minimum service curve of flow 1, β_1 is $(R-\rho_2)(t - T - \frac{\sigma_2}{R})$ and the delay experienced by each packet of flow 1, D_1 is upper bounded by $T + \frac{\sigma_2}{R} + \frac{\sigma_1}{R-\rho_2}$ [8].

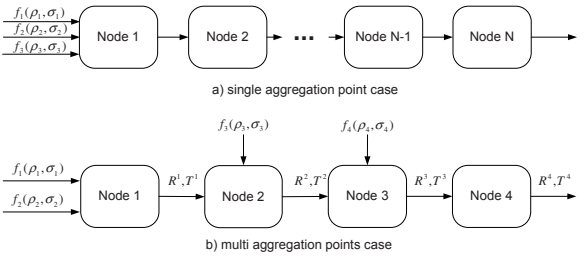


Fig. 2. An example of aggregation in the concatenation of LR Servers

Lemma 1 (Service Curve and Delay Bound in FIFO Aggregation). *Consider a LR server with rate R and latency T serving m flows aggregated in the FIFO manner and each flow i is (ρ_i, σ_i)-constrained. If $\sum_{i=1}^{m} \rho_i < R$, then a minimum service curve of a target flow k is equal to the LR function with rate $R - \sum_{i=1}^{m} \rho_i \cdot I_{\{i \neq k\}}$ and latency $T + \sum_{i=1}^{m} \frac{\sigma_i \cdot I_{\{i \neq k\}}}{R}$. The worst-case increased burstiness of flow k at the output caused by other flows, σ_k^* is $\sigma_k + \rho_k(T + \sum_{i=1}^{m} \frac{\sigma_i \cdot I_{\{i \neq k\}}}{R})$. And the delay experienced by each packet of flow k in the node, D_k is upper bounded by $\frac{\sigma_k}{R - \sum_{i=1}^{m} \rho_i \cdot I_{\{i \neq k\}}} + T + \sum_{i=1}^{m} \frac{\sigma_i \cdot I_{\{i \neq k\}}}{R}$.*

We omit the proof of Lemma 1 on account of space considerations.

Theorem 2 (Composition Theorem [8]). *Assume a flow traverses nodes $\mathcal{N}_1$ and $\mathcal{N}_2$ in sequence. Assume that $\mathcal{N}_i$ offers a service curve of β^i, $i = 1, 2$, to the flow. Then the concatenation of the two nodes offers a service curve of $\beta^1 \otimes \beta^2$ to the flow.*

From Theorem 2, we can represent the concatenation of LR servers into one LR server, whose service rate is equal to the lowest among the allocated rates of the servers on the concatenation, and whose latency is equal to the sum of latencies of the LR servers it replaces. At this point, we derive the edge-to-edge service curve in two cases depicted in Fig. 2 and in a general case.

Proposition 1 (Edge-to-Edge Service Curve with Single Aggregation Point). *Consider N lossless nodes serving three flows, 1,2, and 3 aggregated in the FIFO order, as depicted in a) of Fig. 2. Assume that each node k guarantees minimum LR service curve β_{R^k,T^k} to the aggregate of three flows which are (ρ_i, σ_i)-constrained, $i = 1, 2, 3$. Then the edge-to-edge minimum service curve for flow 1 along the path is given by:*

$$\beta_1(t) = min_{k=1,2,..,N}(R^k - \rho_2 - \rho_3)\left(t - \sum_{j=1}^{N} T^j - \frac{\sigma_2 + \sigma_3}{min(R^1, R^2, .., R^N)}\right)^+$$

Proposition 2 (Edge-to-Edge Service Curve with Multiple Aggregation Point). *Consider nodes serving four flows, 1,2,3 and 4 aggregated in*

the FIFO order, as depicted in b) of Fig. 2. Assume that each node k guarantees a minimum LR service curve β_{R^k,T^k} *and each flow is* (ρ_i, σ_i)*-constrained, i=1,2,3 and 4. Then the edge-to-edge minimum service curve for flow 1 along the path,* $\beta_1(t)$ *is equal to LR function with rate* $min(R^1 - \rho_2, R^2 - \rho_2 - \rho_3, R^3 - \rho_2 - \rho_3 - \rho_4, R^4 - \rho_2 - \rho_3 - \rho_4)$ *and latency* $\sum_{k=1}^{4} T^k + \frac{\sigma_4}{min(R^3,R^4)} + \frac{\sigma_3}{min(R^2,R^3-\rho_4,R^4-\rho_4)} + \frac{\sigma_2}{min(R^1,R^2-\rho_3,R^3-\rho_3-\rho_4,R^4-\rho_3-\rho_4)}$.

We omit the proofs of Proposition 1 and 2 on account of space considerations.

Proposition 3 (Edge-to-Edge Service Curve for a Flow in a Path). *Consider a flow i that traverses the edge-to-edge path* $\mathcal{P}_i$ *consisting of a sequence of N nodes, that service flows aggregated in FIFO order. Assume that the flow f is* (ρ_f, σ_f)*-constrained and each node k guarantees to the aggregate a LR service curve* β_{R^k,T^k}*. If* $\sum_f \rho_f^k < R^k$ *at the output link on every node k along the path* $\mathcal{P}_i$*, then the edge-to-edge minimum service curve of target flow i is given by:*

$$\beta_i(t) = min_{k\in\mathcal{P}_i}\left(R^k - \sum_{f\in\mathcal{F}_i^k}\rho_f\right)\left\{t - \sum_{k\in\mathcal{P}_i}\left(T^k + \sum_{f\in\mathcal{F}_i^k}\frac{\sigma_f \cdot I_{\{k=M_{i,f}\}}}{B_f(\mathcal{P}_{i,f})}\right)\right\}^+.$$

Proof. Let set any t and define $t = t_N(\geq t_{N-1})$. Define t_k for $k < N$ recursively as follows: Given t_k there exist $t_{k-1} \leq t_k$ such that $S_i^{k-1}(t_{k-1}) = A_i^k(t_{k-1})$ and $S^k(t_k) - A^k(t_{k-1}) \geq \beta^k(t_k - t_{k-1})$ since node k guarantees a service curve of $\beta^k(\cdot)$. Hence, for flow i at node k,

$$S_i^k(t_k) - A_i^k(t_{k-1}) \geq \beta^k(t_k - t_{k-1}) - \sum_{f\in\mathcal{F}_i^k}(S_f^k(t_k) - A_f^k(t_{k-1})) \tag{1}$$

Consider node $k-1$ ($k-1$-th node) and node k (k-th node) are successive nodes along the path $\mathcal{P}_i$ and there is no propagation delay on the link. Next, the amount of departure traffic of flow i from node $k-1$ is equal to the arrival traffic amount of flow i at node k at any time t. That is, $A_i^k(t_{k-1})$ is equal to $S_i^{k-1}(t_{k-1})$ for $k = 2, ..., N$. Consequently, we obtain the relation between the arrival traffic amount of flow i on the ingress node on the path $\mathcal{P}_i$ at time t_0 and the departure traffic amount of flow i on the egress node at time t_N as follows.

$$S_i^N(t_N) - A_i^1(t_0) \geq \sum_{k\in\mathcal{P}_i}\beta^k(t_k - t_{k-1}) - \sum_{k\in\mathcal{P}_i}\sum_{f\in\mathcal{F}_i^k}\left(S_f^k(t_k) - A_f^k(t_{k-1})\right) \tag{2}$$

The traffic amount of aggregated flow f serviced during $[t_{k-1}, t_k]$ at node k, $S_f^k(t_k) - A_f^k(t_{k-1})$ is bounded the amount of arrival traffic during $[t_k - t_{k-1} - \theta^k]$ with arbitrary node parameter $\theta^k > 0$. Hence, the equation $S_f^k(t_k) - A_f^k(t_{k-1})$ can be replaced by the arrival curve $\alpha_f(t_k - t_{k-1} - \theta^k)$ and rewrite Eq. 2 as follows.

$$S_i^N(t_N) - A_i^1(t_0) \geq \sum_{k\in\mathcal{P}_i}\beta^k(t_k - t_{k-1}) - \sum_{k\in\mathcal{P}_i}\sum_{f\in\mathcal{F}_i^k}\alpha_f(t_k - t_{k-1} - \theta^k) \tag{3}$$

The traffic amount of (ρ_i, σ_i)-constrained flow i serviced along the path $\mathcal{P}_i$, a concatenation of LR servers with FIFO aggregation offering service curve β_{R^n,T^n}, is defined as follows.

$$S_i^N(t_N) - A_i^1(t_0) \geq \sum_{k\in\mathcal{P}_i} R^k(t_k - t_{k-1} - T^k)^+ - \sum_{k\in\mathcal{P}_i}\sum_{f\in\mathcal{F}_i^k}\{\rho_f(t_k - t_{k-1} - \theta^k) + \sigma_f\}$$

According to Proposition 1 and Lemma 1, the term θ^k consists of the latency of node k's LR server and the duration during which all the burst traffics of aggregated flows are transmitted. If there is no aggregation on the node k, then θ^k is only defined by the the latency of LR server. As shown in Proposition 1, the burstiness of aggregated flows affects on the delay of target flow i only once at the node where the flows aggregate with. Hence, the θ^k is defined as $T^k + \sum_{f\in\mathcal{F}_i^k}(\frac{\sigma_f \cdot I_{\{k=M_{i,f}\}}}{B_f(\mathcal{P}_{i,f})})$. So, we can substitute θ^k with $T^k + \sum_{f\in\mathcal{F}_i^k}(\frac{\sigma_f \cdot I_{\{k=M_{i,f}\}}}{B_f(\mathcal{P}_{i,f})})$. In addition, the equation $\sum_{k\in\mathcal{P}_i}\sum_{f\in\mathcal{F}_i^k}\sigma_f$ denotes the arrival traffic amount caused by the burstiness of flows aggregated with the flow i on the nodes along the path $\mathcal{P}_i$. Since the burstinesss of flows contributes to the arrival traffic amount only once, we can replace $\sum_{k\in\mathcal{P}_i}\sum_{f\in\mathcal{F}_i^k}\sigma_f$ with $\sum_{k\in\mathcal{P}_i}\sum_{f\in\mathcal{F}_i^k}\sigma_f \cdot I_{\{k=M_{i,f}\}}$. Consequently, we can rewrite the above equation as follows.

$$\begin{aligned} S_i^N(t_N) - A_i^1(t_0) \geq & \sum_{k\in\mathcal{P}_i}\big(R^k - \sum_{f\in\mathcal{F}_i^k}\rho_f\big)\big(t_k - t_{k-1} - T^k\big)^+ \\ & - \sum_{k\in\mathcal{P}_i}\left\{\sum_{f\in\mathcal{F}_i^k}\sigma_f \cdot I_{\{k=M_{i,f}\}} - \sum_{f\in\mathcal{F}_i^k}\rho_f \sum_{f'\in\mathcal{F}_i^k}\left(\frac{\sigma_{f'} \cdot I_{\{k=M_{i,f'}\}}}{B_f'(\mathcal{P}_{i,f'})}\right)\right\} \end{aligned} \tag{4}$$

The effective service bandwidth of an aggregating flow f along the sub-path $\mathcal{P}_{i,f}$ is the min-plus convolution of the service bandwidth of flow f at each node k on the sub-path $\mathcal{P}_{i,f}$. The relation $R^k \geq min_{k\in\mathcal{P}_{i,f}} B_f^k(\mathcal{P}_{i,f})$ for every node k on the sub-path $\mathcal{P}_{i,f}$ leads to following equation.

$$\begin{aligned} & S_i^N(t_N) - A_i^1(t_0) \geq \\ & \sum_{k\in\mathcal{P}_i}\left(R^k - \sum_{f\in\mathcal{F}_i^k}\rho_f\right)\left\{t_k - t_{k-1} - T^k - \sum_{f\in\mathcal{F}_i^k}\left(\frac{\sigma_f \cdot I_{\{k=M_{i,f}\}}}{B_f(\mathcal{P}_{i,f})}\right)\right\}^+ \end{aligned} \tag{5}$$

Since the service rate for target flow i at node k is $R^k - \sum_{f\in\mathcal{F}_i^k}\rho_k$, the service rate along the path $\mathcal{P}_i$ for target flow i is $min_{k\in\mathcal{P}_i}(R^k - \sum_{f\in\mathcal{F}_i^k}\rho_f)$. And, the equation $S_i^N(t_N) - A_i^1(t_0)$ is lower bounded by the traffic amount serviced during $[t_0, t_N]$. Hence, we can refer the right-hand side of Eq. 5 as the minimum traffic amount serviced during $[t_0, t_N]$ with the edge-to-edge service curve along the path $\mathcal{P}_i$ for flow i, $\beta_i(t)$. Hence, we can replace $S_i^N(t_N) - A_i^1(t_0)$ of Eq. 5 with $\beta_i(t_N - t_0)$.

Finally, the edge-to-edge minimum service curve of the (ρ_i, σ_i)-constrained flow i along the path $\mathcal{P}_i$ consisting of LR servers is

$$\beta_i(t) = min_{k\in\mathcal{P}_i}\left(R^k - \sum_{f\in\mathcal{F}_i^k}\rho_f\right)\left\{t - \sum_{k\in\mathcal{P}_i}\left(T^k + \sum_{f\in\mathcal{F}_i^k}\frac{\sigma_f \cdot I_{\{k=M_{i,f}\}}}{B_f(\mathcal{P}_{i,f})}\right)\right\}^+ \quad (6)$$

□

Given the arrival curve $\alpha(\cdot)$ and service curve $\beta(\cdot)$ of a flow, the upper bound of delay of the flow is defined as $\sup_{t\geq 0}\inf\{\tau\geq 0 : \alpha(t)\leq\beta(t+\tau)\}$ [8]. Hence, we can obtain upper bounds of edge-to-edge delay of (ρ_i, σ_i)-constrained flow i along the edge-to-edge path $\mathcal{P}_i$ based on the edge-to-edge service curve obtained in Proposition 3. Since the edge-to-edge service curve is a LR function of Eq. 2 and the flow i is leaky-bucket constrained, the upper edge-to-edge delay bounds of flow i are

$$\frac{\sigma_i}{min_{k\in\mathcal{P}_i}\left(R^k - \sum_{f\in\mathcal{F}_i^k}\rho_f\right)} + \sum_{k\in\mathcal{P}_i}\left(T^k + \sum_{f\in\mathcal{F}_i^k}\frac{\sigma_f \cdot I_{\{k=M_{i,f}\}}}{B_f(\mathcal{P}_{i,f})}\right) \quad (7)$$

4 Analytical Results and Observations

This section presents analytical results for the network topology, shown in Fig. 3, adopted in the investigation of edge-to-edge delay bounds. Here, the adopted network has a linear multi-hop topology widely used in previous works, such as in [12, 13].

In Fig. 3, flow f_c is a target flow and the other flows are cross flows of the target flow. The nodes C(0),C(1),..,C(N) denote the core routers, whose scheduler is a class-based WFQ. For simplicity, we assume that links along the path of the target flow have same capacity R^t and that the whole link capacity is allocated to the traffic class to which the target flow belongs. All other edge links shown in Fig. 3 for cross flows have the capacity R^c. In addition, the target flow and cross flows are leaky-bucket constrained.

The configuration of network topology to analyze is as follows. R^c is 10 Mbps. R^t is 2 Mbps. All links are configured to have a link propagation delay

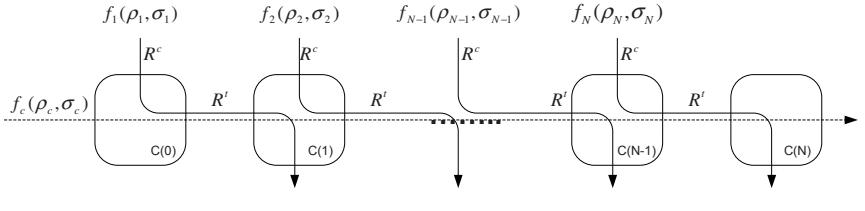
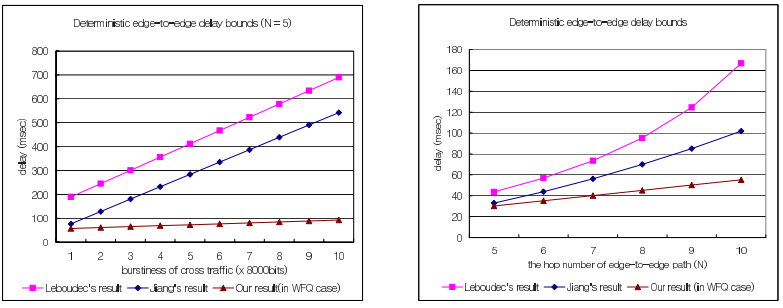


Fig. 3. Generic network topology



(a) Impact of burstiness of cross traffic (N = 5, link utilization = 0.2)

(b) impact of hop number of edge-to-edge path (link utilization = 0.1, core link capacity : 4 Mbps)

Fig. 4. Comparison of deterministic edge-to-edge delay bounds

of 1 ms. All packets have the same size of 1000 bytes. To investigate the impact of burstiness on edge-to-edge delays, we change the burstiness of every cross traffic and obtain the upper edge-to-edge delay experienced by the target flow f_c. Specifically, we set the burstiness (σ_c) of the target flow to 2000 bytes, while for each cross traffic, its burstiness $(\sigma_1, .., \sigma_N)$ changes from 1000 bytes to 10000 bytes. The number (N) of core routers is fixed to 5 to compare the results of [6]. Moreover, we set rate$(\rho_c, \rho_1, .., \rho_N)$ of all traffics to 0.2 Mbps. Next, to investigate the impact of hop number on an edge-to-edge path, we set the core link capacity (R^t) to 4 Mbps to lessen link utilization and set the burstiness of cross traffic to 1000 bytes. The corresponding results are shown in Fig. 4. The analytical results of Yuming and Leboudec are based on [13] and our analytical result is based on Eq. 7 and [9] to consider the WFQ latency.

Fig. 4 (a) shows edge-to-edge delay bounds in terms of cross traffic burstiness and Fig. 4 (b) shows edge-to-edge delay bounds in terms of the hop number of edge-to-edge path. In both cases, the bounds of Jiang and Leboudec are looser than our bounds. This is not surprising, since Leboudec's bounds are obtained without knowledge of the network topology and Jiang's bounds are obtained by the sum of nodal delays. From the result, we conclude that tight edge-to-edge delay bounds are obtained if the network topology, the burstiness of cross traffic, and the route information of cross traffic and target traffic are known. Specifically, the difference between our results and others increases when the burstiness of cross traffic and the hop number of edge-to-edge path grow.

5 Conclusions

In this paper, we analyse edge-to-edge delay bounds of a leaky-bucket constrained flow in a DiffServ network domain consisting of LR schedulers, such as a class-based WFQ. we derive the closed form of delay formulas for the GR service discipline with FIFO aggregation using service-curve based methodology. Our quantitative delay bounds of a flow are based on network calculus concepts of service and arrival curve by way of a deterministic approach and

route interference information of aggregated flows to consider the impact of aggregation. We conclude that if input flows are leaky-bucket constrained and the route information of flows is known in an arbitrary route system with the limit of link utilization, then tight edge-to-edge delay bounds over other bounds can be secured.

To apply derived edge-to-edge delay bounds of a flow to a real DiffServ environment, we consider a centralized resource manager such as a bandwidth broker. Since the centralized resource manager performs per-flow admission and edge-to-edge path selection, the centralized resource manager has edge-to-edge path information and aggregate flow information along the path. In order to guarantee per-flow edge-to-edge delay bounds, per-flow signaling is required at the edge router. However, per-flow signaling of core routers is not required at all. We will extend the applicability of the obtained bounds to the dynamic resource provision with edge-to-edge trunk reservation for EF flows in the DiffServ architecture and compare the derived delay bounds with the result of simulation.

References

[1] S. Blake and et. al., An Archiecture for Differentiated Services, IETF RFC 2475, 1998. 370

[2] Huirong Fu and E. W. Knightly, Aggregation and Scalable QoS: A Performance Study, IWQoS'01, 2001, 307-324. 370

[3] Anna Charny and Jean-Yves Le Boudec, Delay bounds in an Network with Aggregate Scheduling, QOFIS'00, Oct. 2000, 1-13. 371

[4] Yuming Jiang, Delay bounds for a network of guaranteed rate servers with FIFO aggregation, Computer Networks, 40(6), 2002, 684-684. 371

[5] Z. Duan, Z. L. Zhang and Y. T. Hou, Fundamental Trade-offs in Aggregate Packet Scheduling, ICNP'01, 2001. 371

[6] I. Chlamtac, et. al., A deterministic approach to the end-to-end analysis of packet flows in connection oriented networks, IEEE/ACM Trans. on Networking, 1998, 422-431. 371, 378

[7] Jean-Yves Le Boudec and G. Hebuterne, Comment on a deterministic approach to the end-to-end analysis of packet flows in connection oriented network, IEEE/ACM Trans. on Networking, Feb. 2000., 121-124. 371

[8] Jean-Yves Le Boudec and Patrick Thiran(eds.), Network Calculus:A Theory of Deterministic Queuing Systems for the Internet, Springer, Jan. 2002. 371, 372, 373, 374, 377

[9] Dimitrios Stiliadis and Anujan Varma, Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms, IEEE/ACM Trans. on Networking, 6, 1998, no. 5, 611-624. 372, 373, 378

[10] R. L. Cruz, A Calclus for Network Delay, Part I: Network Elements in Isolation, IEEE Trans. Information Theory, 37, no. 1, 114-131. 373

[11] R. L. Cruz, A Calclus for Network Delay, Part II: Network Analysis, IEEE Trans. Information Theory, 37, no. 1, 132-141. 373

[12] R. Guerin and V. Pla, Aggregation of conformance in Differentiated Service network: A case study, ACM CCR, 31(1), Jan. 2001, 21-32. 377

[13] Yuming Jiang and Qi Yao, Impact of FIFO Aggregation on Delay Performance of a Differentiated Services Network, ICOIN'03, Feb. 2003. 377, 378

An Efficient Preemption-Based Service Differentiation Scheme for OBS Networks

Byung-Chul Kim and You-Ze Cho

School of Electrical Engineering and Computer Science
Kyungpook National University, Korea
bckim@eecs.knu.ac.kr
yzcho@knu.ac.kr

Abstract. In this paper, we propose a preemption-based service differentiation scheme for optical burst switching (OBS) networks based on combining a preemption channel selection algorithm and a channel partitioning algorithm. The proposed preemption channel selection algorithm minimizes the length of preempted bursts to improve channel efficiency, while the channel partitioning algorithm controls the degree of service differentiation between service classes. Performance evaluation results showed that the proposed scheme can improve channel efficiency and effectively provide controllable service differentiation.

1 Introduction

The rapid growth of the Internet has resulted in an increasing demand for transmission capacity in core backbones. Accordingly, core networks must evolve to new architectures based upon all optical switching and dense wavelength division multiplexing (DWDM) technologies. Optical burst switching (OBS) has been proposed as the technology basis for such an all-optical Internet [1]-[3].

The future Internet may demand differentiated services for multimedia applications. Conventional QoS supporting mechanisms mainly depend on buffering and scheduling. However, these approaches cannot be directly applied to OBS networks because an efficient optical buffer does not currently exist. Accordingly, several QoS supporting schemes for OBS that do not use buffering at the WDM layer has been proposed [4]-[6].

The representative QoS supporting scheme for OBS is an offset time based service differentiation approach [4]. In the offset time based approach, an additional QoS offset time is assigned for higher priority classes, thereby permitting a higher priority burst to reserve resources in advance. However, since the offset time is reduced as the burst traverses a path, the effective priority of the burst decreases hop-by-hop [7]. And, this approach can only be applied to a delayed reservation based OBS scheme, like just-enough-time (JET) [1].

Accordingly, this paper proposes the preemption-based service differentiation scheme for OBS networks. The proposed scheme combines a preemption channel selection algorithm and channel partitioning algorithm, where the preemption

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 380–389, 2004.

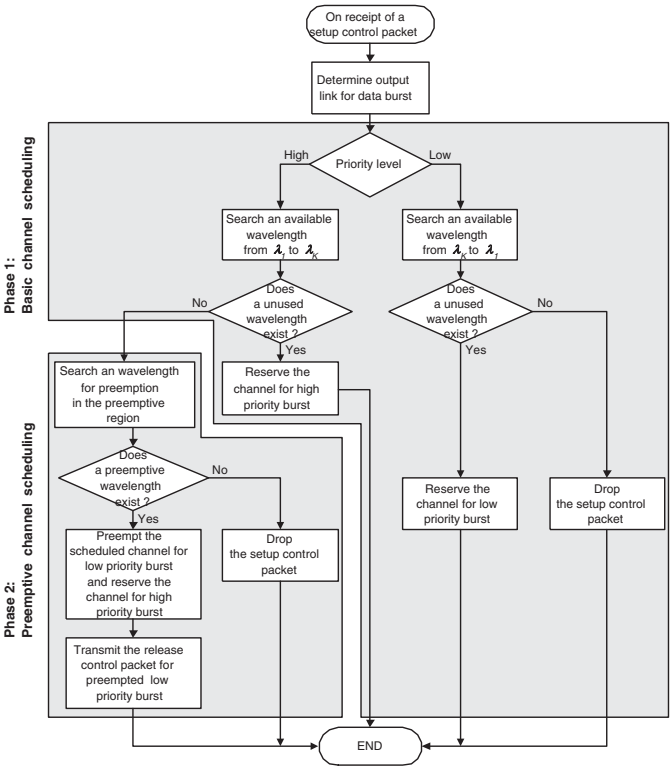


Fig. 1. Operations of the proposed preemption-based service differentiation scheme

channel selection algorithm is used for improving the channel efficiency and the channel partitioning algorithm controls the degree of service differentiation among the service classes.

The remainder of this paper is organized as follows. Section 2 describes the proposed preemption-based service differentiation in OBS networks. Section 3 evaluates the performance of the proposed schemes. Finally, some concluding remarks are presented in Section 4.

2 The Proposed Preemption-Based Service Differentiation Scheme

This section describes the proposed preemption-based service differentiation scheme that combines a preemption channel selection algorithm and channel partitioning algorithm. In this paper, we assume two classes of service: high priority class H and low priority class L. In the proposed preemption-based service differentiation scheme, it is assumed that class H bursts will preempt the scheduled channel for class L bursts, if no available channels are found by the basic

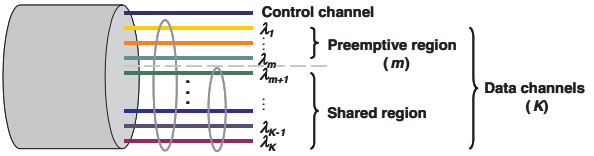


Fig. 2. Concept of the proposed channel partitioning algorithm

data channel scheduling algorithm, so as to guarantee a lower burst blocking probability for class H over class L.

The proposed preemption-based service differentiation scheme uses two phase data channel scheduling algorithm as shown in Fig. 1. First, when a channel reservation request of a new burst arrived at a core node, an unscheduled channel is searched by a basic data channel scheduling algorithm without considering the preemption. If an available channel is found, the burst is scheduled to that channel. If no available channel is found in the first phase and the burst belongs to class H, the burst will preempt a channel scheduled for a class L burst. In this phase, a new preemption channel selection algorithm is applied to improve the channel efficiency. Meanwhile, the proposed channel partitioning algorithm limits the preemptive channels in order to control the degree of service differentiation between service classes and preserve some amount of bandwidth to a low priority class.

2.1 Channel Partitioning Algorithm

This subsection describes the proposed channel partitioning algorithm which controls the degree of service differentiation between the priority classes. Fig. 2 shows the concept of the channel partitioning algorithm, where the data channels are partitioned into two regions: the shared region and preemptive region. In the preemptive region, class H bursts can preempt the channels scheduled for lower priority bursts. Meanwhile, in the shared region, preemptions are not allowed, thereby enabling lower priority bursts to equally contend for data channels with high priority bursts, and somewhat guaranteeing the performance of lower priority bursts in this region. In the proposed channel partitioning algorithm, the number of wavelengths m in the preemptive region can be configured to control the degree of service differentiation.

2.2 Preemption Channel Selection Algorithm

In the OBS network, data channel scheduling is required for efficiently using the data channels. Data channel scheduling algorithms can be classified into two categories: with and without void filling (VF). In with-VF algorithms, the gap/void between two scheduled bursts can be reserved by a new arrival burst. The with-VF algorithms can be applied to a delayed reservation based OBS scheme, like

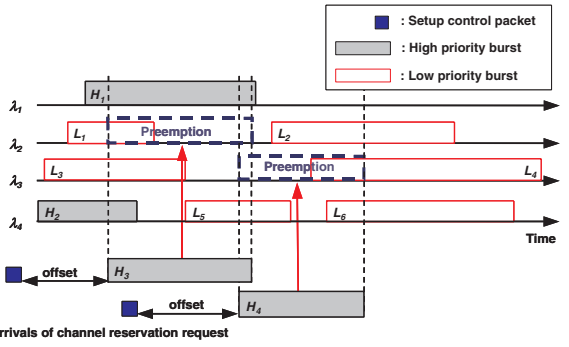


Fig. 3. An example of the proposed LRC algorithm

JET [1]. Whereas, without-VF algorithm can be applied to immediate reservation based OBS scheme, like just-in-time (JIT) and a terabit burst switching mechanism with a lower implementation complexity [2]-[3].

These data channel scheduling algorithms are used as a basic scheduling algorithm in the first phase of the preemption-based service differentiation scheme. However, a new policy is required for selection a channel for preemption in the second phase. Random selection algorithm is the simplest method, in which one channel among data channels scheduled for class L bursts is randomly chosen for preemption.

In this paper, we propose the least reserved channel (LRC) selection algorithm which minimizes the channel waste due to the preemptions. The proposed algorithm can be applied when a with-VF scheduling algorithm, such as first-fit with void filling (FF-VF) and latest available unscheduled channel with void filling (LAUC-VF)[8], is used as a basic data channel scheduling algorithm in the first phase. In the proposed LRC algorithm, the channel with the shortest class L burst reservation is selected for preemption. As such, the LRC algorithm minimizes the duration of the class L preemption, thereby reducing the channel waste due to the preemptions.

Fig. 3 shows an example of the proposed LRC algorithm. It is assumed that all data channels are in the preemptive region, i.e. $K = 4$ and $m = K$, for simplicity. And, high priority bursts H_1 - H_2 and low priority bursts L_1 - L_6 are already scheduled, as shown in this figure. When a channel reservation request for high priority burst H_3 arrives, the scheduler first searches for an available channel. However, no available channel exists in this example, the scheduler then searches among the channels scheduled for class L for preemption. In this figure, bursts L_1 and L_3 can be preempted by burst H_3, however, since the reserved duration of L_1 is shorter than that of L_3, burst L_1 is preempted and burst H_3 is scheduled to λ_2. When a channel reservation request for H_4 arrives, λ_3 and λ_4 can be preempted. In this case, since the total reserved duration of bursts L_5 and L_6 is longer than the reserved duration of burst L_4, burst L_4 is preempted and λ_3 is reserved for burst H_4.

2.3 Signaling Issues

In the proposed preemption-based service differentiation scheme, the occurrence of preemption needs to be signaled to the upstream and downstream nodes to improve the efficiency by releasing and reusing the reserved wavelength on the path for preempted bursts. This can be done by transmitting a channel release control packet, which is similar to the *RELEASE* message defined in [3]. As such, the released channel can be used by other reservation requests and channel utilization is improved.

3 Performance Evaluation

This section investigates the performance of the proposed preemption-based service differentiation scheme in various environments. First, we develop the analytical model for evaluating the performance of the proposed scheme, and compare the analytical results with the simulation results in single-hop model. And, we investigate the performance of the proposed preemption-based service differentiation scheme in multi-hop network environments through simulation.

3.1 Single-Hop Model Analysis

This section analyzes the performance of the proposed preemption-based service differentiation scheme in single-hop model using queueing analysis and simulation. To simplify the analysis, we assume that both burst length and arrival time are exponentially distributed. The offset time between the burst control packet and data burst is assumed to be constant. We consider two classes of services in the system and K wavelengths in a switch.

We model the number of wavelength used by each class in a switch as a continuous time Markov chain. Let λ_h and μ_h be the arrival rate and the service rate of class H, and λ_l and μ_l be the arrival rate and the service rate of class L, respectively. When the total number of class H and class L bursts exceeds the number of wavelengths K, bursts may be blocked or preempted. Namely, an arrival class L burst may be blocked when the wavelength is not remained for the service. However, an arrival class H burst may preempt a class L burst in service if the number of class H bursts in the system is not exceed the boundary m. And, class H burst may be blocked when the number of class H bursts exceed the boundary m and a wavelength is not remained for the service.

Letting the number of class H and class L bursts in the system as i and j respectively, a pair (i, j) forms a two-dimensional Markov chain, and the total number of system states is $(K+1)(K+2)/2$ as shown in Fig. 4.

Denoting $P_{i,j}$ as the steady-state probability of the system being in a state (i, j), we can represent the global balance equations for each state $(i, j), 0 \leq i \leq K, 0 \leq j \leq K, 0 \leq i + j \leq K$, as follows:
1) $i = 0, j = 0$:

$$(\lambda_h + \lambda_l) \cdot P_{0,0} = \mu_h \cdot P_{1,0} + \mu_l \cdot P_{0,1}$$

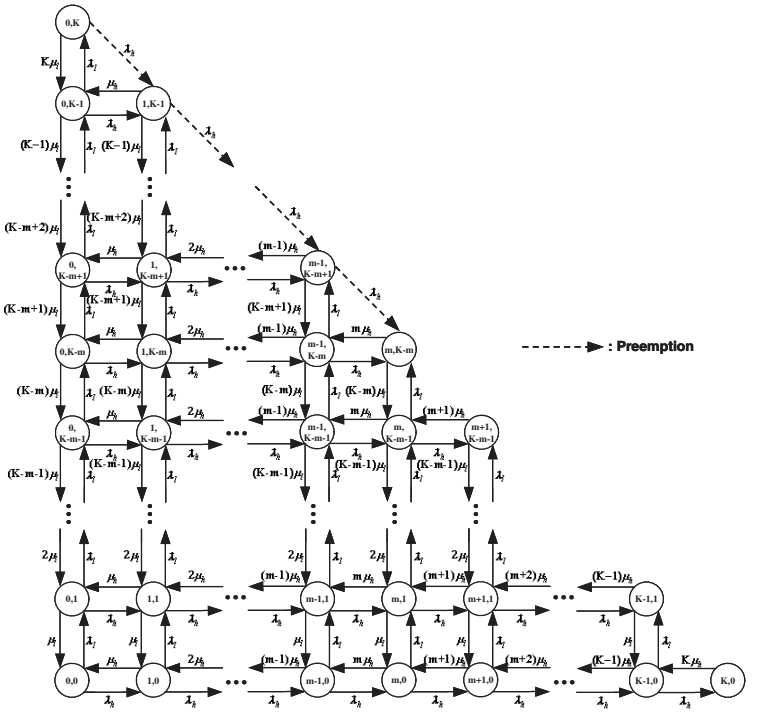


Fig. 4. State transition diagram of the proposed preemption-based service differentiation scheme

2) $i = 0, j = K, m > 0$:

$$(\lambda_h + K\mu_l) \cdot P_{0,K} = \lambda_l \cdot P_{0,K-1}$$

3) $i = 0, j = K, m = 0$:

$$K\mu_l \cdot P_{0,K} = \lambda_l \cdot P_{0,K-1}$$

4) $i = K, j = 0, m < K$:

$$K\mu_h \cdot P_{K,0} = \lambda_h \cdot P_{K-1,0}$$

5) $i = K, j = 0, m = K$:

$$K\mu_h \cdot P_{K,0} = \lambda_h \cdot P_{K-1,0} + \lambda_h \cdot P_{K-1,1}$$

6) $i = 0, 1 \leq j \leq K-1$:

$$(\lambda_h + \lambda_l + j\mu_l) \cdot P_{0,j} = \lambda_l \cdot P_{0,j-1} + (j+1)\mu_l \cdot P_{0,j+1} + \mu_h \cdot P_{1,j}$$

7) $1 \leq i \leq K-1, j = 0$:

$$(\lambda_h + \lambda_l + i\mu_h) \cdot P_{i,0} = \lambda_h \cdot P_{i-1,0} + (i+1)\mu_h \cdot P_{i+1,0} + \mu_l \cdot P_{i,1}$$

8) $i+j=K, i>m$:

$$(i\mu_h + j\mu_l)\cdot P_{i,j} = \lambda_h \cdot P_{i-1,j} + \lambda_l \cdot P_{i,j-1}$$

9) $i+j=K, i<m$:

$$(\lambda_h + i\mu_h + j\mu_l)\cdot P_{i,j} = \lambda_h \cdot P_{i-1,j+1} + \lambda_h \cdot P_{i-1,j} + \lambda_l \cdot P_{i,j-1}$$

10) $i+j=K, i=m$:

$$\begin{aligned}&[m\mu_h + (K-m)\mu_l]\cdot P_{m,K-m}\\ &= \lambda_h \cdot P_{m-1,K-m+1} + \lambda_h \cdot P_{m-1,K-m} + \lambda_l \cdot P_{m,K-m-1}\end{aligned}$$

11) $1 \le i \le K-1, 1 \le j \le K-1, 2 \le i+j \le K-1$:

$$\begin{aligned}&(\lambda_h + \lambda_l + i\mu_h + j\mu_l)\cdot P_{i,j}\\ &= (i+1)\mu_h \cdot P_{i+1,j} + (j+1)\mu_l \cdot P_{i,j+1} + \lambda_h \cdot P_{i-1,j} + \lambda_l \cdot P_{i,j-1}\end{aligned} \tag{1}$$

Representing the steady-state probabilities $\{P_{i,j}\}$ as a row vector $\mathbf{P}$, the global balance equation can be expressed in matrix form as

$$PQ = 0 \tag{2}$$

where $\mathbf{Q}$ is a transition rate matrix. Also, the total sum of $\{P_{i,j}\}$ must be equal to 1, hence it is given by

$$\sum_i \sum_j P_{i,j} = 1 \tag{3}$$

After the solution of row vector $\mathbf{P}$ comes out, the burst blocking probabilities of new arrival class H and class L bursts can be obtained as

$$P_{block,H} = \sum_{i+j=K, i\ge m} P_{i,j} \tag{4}$$

$$P_{block,L} = \sum_{i+j=K} P_{i,j} \tag{5}$$

And, the preempted probability of an accepted burst of class L is given by

$$P_{preemption} = \frac{N_{preempted}}{N_{accepted}} = \frac{1}{\lambda_l(1-P_{block,L})}\left[\sum_{i+j=K, i<m} \lambda_h \cdot P_{i,j}\right] \tag{6}$$

We consider bufferless switches with three data channels in each output link. Each switch is assumed to be capable of full wavelength conversion. We assume that bursts of each class are generated with an exponential inter-arrival time and exponential burst duration. We assume that the fraction of class H bursts

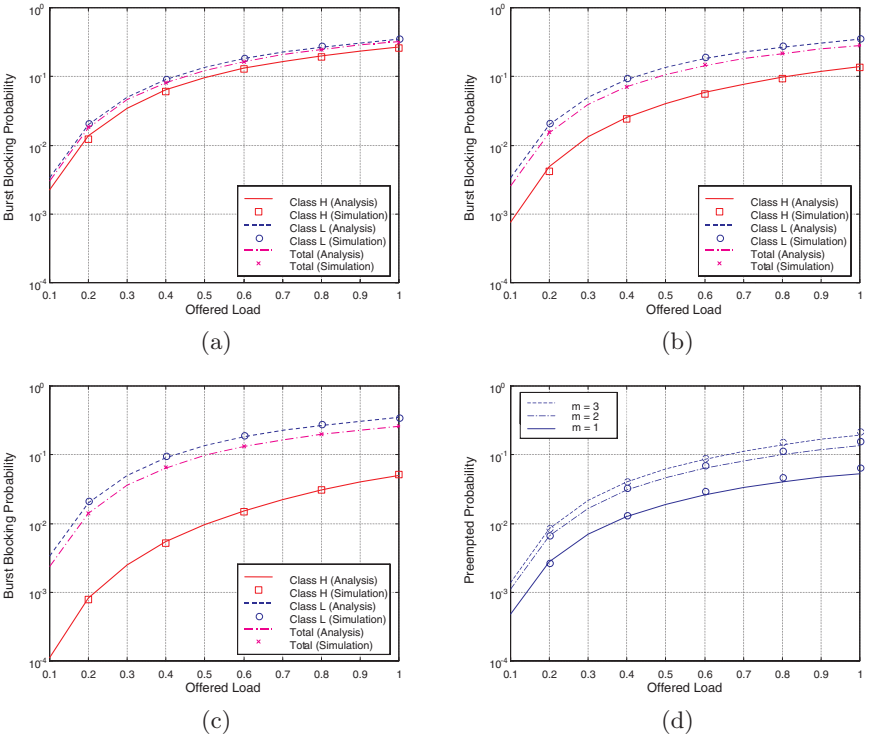


Fig. 5. Performance of the proposed preemption-based service differentiation scheme in single-hop model for $K = 3$. (a) Burst blocking probability for $m = 1$. (b) Burst blocking probability for $m = 2$. (c) Burst blocking probability for $m = 3$. (d) Preempted probability of an accepted class L burst

is 30%, and the fraction of class L bursts is 70%. For simplicity, we assume that the mean burst duration of each class is identical (i.e., $\mu_h = \mu_l = \mu$).

Fig. 5 (a)-(c) show the blocking probabilities of the arrival burst for each class according to the size of preemptive region m and the offered load intensity. Fig. 5 (d) depicts the preempted probabilities of accepted bursts of class L. The offered load is defined as $(\lambda_h + \lambda_l)/K\mu$. These figures show that the difference in the burst blocking probabilities between the class H and class L traffic is reduced when m is decreased. As such, the degree of service differentiation between the classes is effectively controlled by the proposed channel partitioning algorithm. Also, the analytical results match the simulation results very well.

3.2 Multi-hop Model Analysis

This section investigates the performance of the proposed preemption-based service differentiation scheme in multiple hop network environments through simulation. A 14-node NSFNET was used to evaluate the end-to-end performance

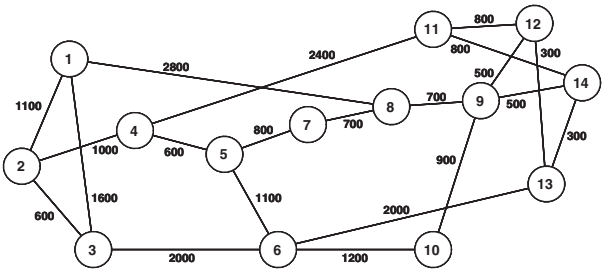


Fig. 6. NSF network with 14 nodes

of the proposed service differentiation scheme, as shown in Fig. 6. The distances indicated are in Km. In the simulation model, it is assumed that the input traffic is uniformly distributed over all the sender-receiver pairs and the shortest path routing is used to find the path between all node pairs. In addition, each link is assumed to consist of eight WDM wavelengths operating at 10Gbps, and one wavelength is used as the control channel. And, full wavelength conversions among the data channels are assumed at the OBS nodes. The burst length is assumed to be exponentially distributed with an average of 100 Kbits, and burst arrivals to the network are Poisson. The fraction of high priority bursts is 30%, and the fraction of low priority bursts is 70%. We also assume that the size of the preemptive region m is 7 for the channel partitioning scheme.

The performance of the proposed LRC algorithm is compared with the random selection algorithm in the multiple hop network environments. Fig. 7(a) shows the blocking probabilities for each class according to the offered load by each node. And, Fig. 7(b) shows the average end-to-end throughput for each class according to the channel selection algorithm. These figures show that the proposed LRC algorithm improves the end-to-end throughput of the class L bursts compared to the random selection algorithm without degrading the throughput of class H even though the burst blocking probabilities are similar. This is due to the fact that the proposed LRC algorithm preempts the shortest class L burst and minimizes the throughput degradation of class L, while the random selection algorithm selects a channel regardless of the burst length.

4 Conclusions

In this paper, we proposed the preemption-based service differentiation scheme for OBS networks based on combining a preemption channel selection algorithm and a channel partitioning algorithm. The proposed preemption channel selection algorithm minimizes the length of preempted bursts to improve the channel efficiency, while the proposed channel partitioning algorithm controls the degree of service differentiation between service classes. Performance evaluation results showed that the proposed preemption-based service differentiation scheme could

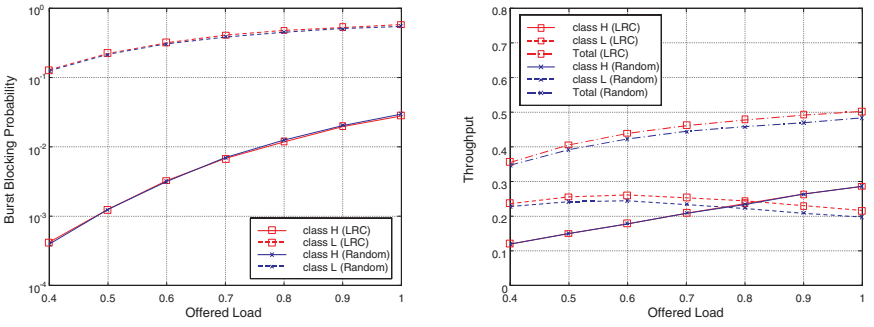


Fig. 7. Comparison of the preemption channel selection algorithms in multi-hop model. (a) Burst blocking probability. (b) End-to-end throughput

effectively differentiate the burst blocking probability based on the channel partitioning algorithm and improve the end-to-end throughput using the LRC algorithm.

Acknowledgement

This work was done as a part of Information & Communication Fundamental Technology Research Program supported by Ministry of Information & Communication in republic of Korea.

References

[1] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69-84, 1999.

[2] J. S. Turner, "Terabit Burst Switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3-16, 1999.

[3] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 82-89, Feb. 2002.

[4] M. Yoo and C. Qiao, "Supporting Multiple Classes of Services in IP over WDM Networks," in *Proc. of IEEE GLOBECOM'99*, pp. 1023-1027, Dec. 1999.

[5] Y. Chen, M. Hamdi, D. Tsang, and C. Qiao, "Proportional QoS over OBS Networks," in *Proc. of IEEE GLOBECOM'01*, pp. 1510-1514, Nov. 2001.

[6] C. Loi, W. Liao, and D. Yang, "Service Differentiation in Optical Burst Switched Networks," in *Proc. of IEEE GLOBECOM'02*, pp. 2313-2317, Nov. 2002.

[7] B. C. Kim, Y. Z. Cho, J. H. Lee, Y. S. Choi, and D. Montgomery, "Performance of Optical Burst Switching Techniques in Multi-Hop Networks," in *Proc. of IEEE GLOBECOM'02*, pp. 2772-2776, Nov. 2002.

[8] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Control Architecture in Optical Burst Switched WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838-1851, Oct. 2000.

Multiple Metric QoS Routing in Differentiated Services Networks Using Preference Functions Measurement Concepts

Wayne Goodridge[1], Bill Robertson[1], Bill Phillips[1], and Shyamala Sivakumar[2]

[1] Department of Enginerring Mathematics, Faculty of Engineering
Dalhousie University Halifax, NS, B3J1Y9
{wgoodrid,phillips}@dal.ca
bill.Robertson@da.ca
[2] St. Mary's University, Department of Commerce
sivas@dal.ca

Abstract. QoS routing protocols involving several combinations of network metrics can be difficult to solve in polynomial time. Our research introduces a novel approach that allows bandwidth brokers to perform QoS management in Diffserv Domains with any number and combination of network metrics with algorithm complexity of O(m(n + 1)) where m is the number of paths between two edge devices in the network and n is the number of metrics involved.

1 Introduction

There are two dimensions to the problem of providing Quality-of-service (QoS) to applications in IP networks. The first is finding the best path to route packets for a given connection, and the second is to reserve network resources on that path. Current Internet routing protocols such as OSPF and RIP use routing that is optimized for a single arbitrary metric (shortest path routing). Path selection within routing is typically formulated as a shortest path optimization problem. However, routing algorithms using one metric to calculate the path cannot satisfy the diverse QoS requirements needed by multimedia applications. Routing algorithms using multiple metric calculations are required, although this cannot be done on a real time basis. The computation becomes more complex as constraints are introduced into the path calculation problem.

A router's QoS support can be divided into the following tasks: (1) Defining packet treatment classes; (2) Specifying the amount of resources for each class; and, (3) Sorting all incoming packets into their corresponding classes. The Differentiated Services [1] effort addresses both tasks 1 and 3 since it specifies traffic classes as well as providing a simple packet classification mechanism. Diffserv Routers also easily sort packets into their corresponding treatment classes by the Type of Service (TOS) value, without having to know to which flows or types of applications the packets belong. The idea of a Bandwidth Broker (BB) [1] was introduced as part of the DS architecture to address the second task. The BB

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 390–399, 2004.

is a program running on a machine that is responsible for allocating preferred services to users as requested, and for configuring the network routers with the correct forwarding behavior for the defined service. The BB plays several roles in administering a differentiated services resource management which include Inter-domain resource management and Intra-domain resource management.

The BB in its role as a global network manager maintains information about all the established connections. The objective of any QoS routing [2] algorithm is to find a path suitable for the flow with minimal operational overheads. In our research we propose that a BB calculates appropriate QoS paths as a by-product of its admission control function. In addition, dynamic link QoS state information, which is required for a QoS path computation, will be implicitly maintained at the BB as it assigns or gets back QoS paths. By maintaining the network state information this way, we may not only eliminate the overhead to exchange network state update messages but also achieve higher routing performance by utilizing accurate network state information in the path computation.

The structure of the paper is as follows. In Section 2 we discuss the difficulties of QoS Routing algorithms. In Section 3 we outline a problem definition and in Section 4 we formulate a solution using preference function modeling. In Section 4.2 we provide an example to illustrate our approach, and in Sections 5 and 6 we evaluate our approach by simulation.

2 QoS Multiple Metric Routing

QoS paths require multiple metrics in the path selection process. This multi metric requirement leads to the following difficulties: (1) Diverse QoS constraints include delay, delay jitter, loss ratio, bandwidth and security. Multiple constraints often make the routing problem intractable. For example, finding a feasible path with two independent path constraints is NP-complete [3]; (2) Links going into and out-of service, and topology changes introduce major convergence issues since QoS routing algorithms depend on up-to-date state information in a dynamic environment; (3). When QoS traffic and Best effort traffic share a given network path performance optimization is complicated since the throughput of the best-effort traffic suffers if the overall traffic distribution is misjudged. Routing QoS algorithms have to maintain global state information (4) at each node in the network resulting in high communication overhead.

Constraints are used in QoS routing algorithms to define what properties the chosen path must have, that is, to limit the search scope for the best route to the possible routes. A path constraint is an end-to-end constraint while a link constraint is local to the link. Common QoS constraint algorithms include: (1) Bandwidth-Delay-Constraint path algorithm [3]; (2) Shortest-Widest Path Routing Algorithm [3]; (3) Delay-Constrained Routing Algorithm [4]; (4) Guerin-Orda algorithm [4]; and, (5) Chen-Nahrstedt algorithm [5]. The main problems with these QoS algorithms are: (1) The widest-shortest path and the shortest-widest path routing algorithms compute paths with more than one metric. However, the resulting path may depend mostly on the first metric. This is because the

second metric is used only when the first metric could not determine the best path; and, (2) Most existing QoS Routing algorithms address bottleneck parameters such as bandwidth, and consider two constraints, typically a combination of bandwidth and delay, or bandwidth and cost at any given time. As a result of more distributed applications users may which to request a wider range of routing constraints and include metrics such as jitter, loss and security. In addition, policy based routing may also introduce additional routing constraints. The above algorithms satisfy or optimize certain kinds of end user flow requests but none of them offer a generic solution to QoS multiple metric routing; and (3) QoS Routing requires periodic exchange of network information to acquire accurate QoS path information. However, frequent update of information causes network overload and any increase of network size causes delay in acquiring accurate information. Hence there is a need for more sophisticated algorithms that will do routing based on imprecise state information.

3 Admission Control in Differentiated Service Networks

We consider a communication network consisting of a set of nodes $N = 1, ..., n$ and a set of unidirectional links $L = 1, ..., l$ where each link l has a set of metrics $w_i(l)$ where i = 1,...,m. There is a set of routes $R = 1, ..., r$ connecting two boundary devices of the network. Each link of a route also has a set $C = 1, ..., c$ of different service classes, where each class c is characterized by its bandwidth requirement $b(c)$ for a given link. When a new flow request of demand $f_1, f_2..., f_m$ of class c arrives at an edge device, the bandwidth broker is called upon to make an admission control decision. The main goals of the of the admission process are: (1) Find a suitable route in polynomial time that meets the demands of the multi metric flow request while at the same time ensuring that network resources are not over allocated; and, (2) If a suitable path is found the flow is admitted with a drop probability that guarantees the level of service specified in the SLA.

To tackle the NP- complete problem of finding QoS paths based on multiple metrics we decided to formulate a solution to this problem as follows: (1) finding all the fixed paths connecting edge devices in an un-weighted graph with the same links; and, (2) feeding the found paths from step 1, along with the flow demand $f_1, f_2..., f_m$ into a decision system which uses a generic multiple metric algorithm to obtain the QoS path in polynomial time.

4 Routing Decision Support System

When a user-application makes a flow request (which typically would include a service type, a target rate, a maximum burst, and the time period when service is required) to an edge router the router in turn uses the Common Open Policy Service (COPS) [6] protocol to gain the admission control services of the bandwidth broker. The request is first combined with network policies that are defined by the network administrator to form a request demand that not only

accounts for the user application needs but also for the needs of the organization that owns the network.

This flow demand is then given to the routing decision support system (RDSS) that then decides the ingress – egress pair to use based on the requirements specified in the Service Level Agreement (SLA) of the user. Once the ingress – egress pair is known all the paths connecting the pair and the known link states are then used by the RDSS to build a preference function (PF) [7, 8, 9]. The preference function is then combined mathematically with the flow request to give a QoS path that is most appropriate for the flow demand and network utilization. If an appropriate path is not found it may be rejected or a negotiation process with the user may be started.

4.1 RDSS Algorithm

The RDSS algorithm for finding QoS-paths is outlined as follows:

1. For a given metric say m_j select all metric values m_{ij} for routes R_i, where $i = 1, 2, .., N$
2. For a given metric m_j the best value from the set of all routes for m_j is assigned to a_{ij}
3. is the route corresponding to the best value for metric m_j.
4. For a given metric m_j the worse value from the set of all routes for m_j is assigned to b_{ij} where i is the route corresponding to the worse value for metric m_j.
5. A preference scale is then derived for each criterion with the following properties:
   (a) $s : A \rightarrow F$ where $F \in \mathbb{R}$ and
   (b) A is bounded such that $s(b) = -1$ and $s(a) = 1$
6. For concave metrics if $x < b$ then $s(x) = -1$ and if $x > a$ and $s(x) = 1$. Additionally, for additive and multiplicative metrics if $x < b$ then $s(x) = 0$ and if $x > a$ and $s(x) = 1$.
7. All other values of x are mapped by s as follows:

$$x = 2\frac{(x-b)}{(a-b)} - 1 \tag{1}$$

8. Let A be a n x m matrix (m = number of criteria (route metrics), n = number of alternative routing paths) containing columns $s_1(x_1), s_2(x_2), \ldots, s_m(x_m)$ where $s_j(x_j)$ represents a scale for each metric m_j. The preference function for routes R_i is therefore given by $A\overline{v} = \overline{o}$ where $\overline{o}$ is an N x 1 vector and v is a M x 1.

When the BB receives a flow demand request the following procedure is followed:

1. Each value in the flow demand is converted to its corresponding scale value as follows: $d = [d_1, d_2, ..., d_m]$, $f(d) = [f(d_1), f(d_2), ..., f(d_m)]$ and $v =$

Table 1. Paths values for network metrics

Path	Delay Jitter
P(A,B,C,F)	5 10
P(A,D,E,F)	6 4
P(A,B,E,F)	9 6
P(A,D,E,B,C,F)	12 12
a	5 4
b	12 12

$[s_1(f(d_1)), s_2(f(d_2)), ..., s_1(f(d_m))]$ where f is a function that returns the closest largest definable point on the scale s. For example, if s is defined for points (2,4,8,20) and the argument of f is 6.5, the closest largest definable point in the domain of s is 8, and therefore s(8) will be evaluated rather than s(6.5).

2. The $A\overline{v} = \overline{o}$ matrix multiplication is then performed which results in N x 1 vector. Vector $\overline{o}$ is examined for the highest value of the vector at position y where $1 \leq y \leq m$. The value of y corresponds to the route that will be considered by the negotiation phase for QoS admission of flow demand d.
3. For the negotiation phase each value of the flow demand is checked against the path metrics for route y. If any value d in the flow demand is not met and the s(d) = -1, then $a \rightarrow d$, and steps 1 and 2 are repeated. If all path metrics for route y are satisfied then the flow is accepted, otherwise the flow is rejected.

Remark. The complexity of the algorithm is O(nm) due to the metric multiplication plus the O(m) due to the search of vector o for the largest value. This makes the algorithm complexity O(m(n+1)). The route j that is selected goes through a QoS negotiation phase that involves checking each value of the flow demand against the route vector. If all the requirements are met the flow is accepted otherwise the negotiation process may offer the applicant an alternative flow demand vector or it may reject the admission requests.

4.2 RDSS Example

To illustrate how the RDSS algorithm works consider the directed network in Figure 1. Suppose network traffic flows use ingress router A and egress router F. Table 1 shows all the possible paths connecting routers A and F and the corresponding delays and jiiters for each path.

$$A = \begin{bmatrix} 1 & -0.5 \\ 0.71 & 1 \\ -0.14 & 0.5 \\ -1 & -1 \end{bmatrix} \tag{2}$$

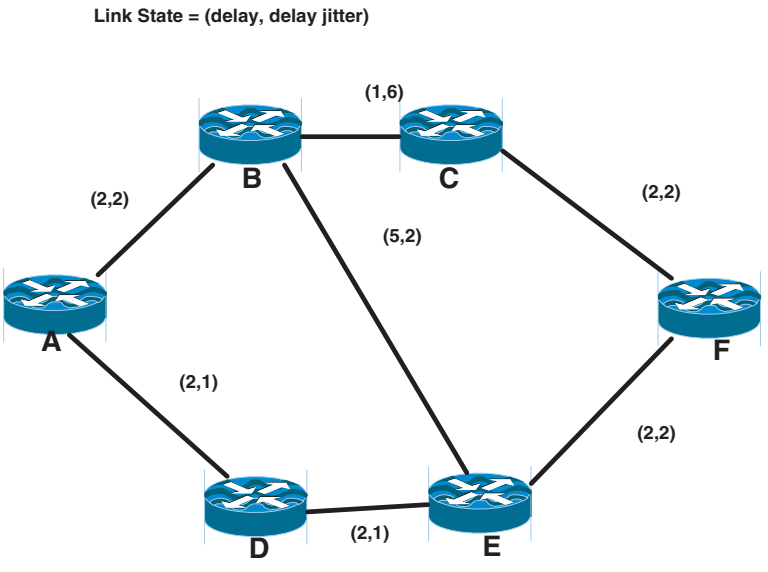


Fig. 1. DS domain of routers

Now suppose a user request the following flow demand $d = [delay = 7, jitter = 4]$ then

$$\begin{bmatrix} 1 & -0.5 \\ 0.71 & 1 \\ -0.14 & 0.5 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 0.43 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.07 \\ 1.31 \\ 0.43 \\ -1.43 \end{bmatrix} \tag{3}$$

Since 1.31 is the largest value the route corresponding to this position is the solution, namely A,D,E,F. Examining Figure 1 confirms that this is the best path to satisfy the given constraints.

5 Simulation

To evaluate our solution to finding a suitable QoS path for a given flow request we implemented a ôrouting serverö in C++ using the preference function technique described in the previous section. Note the routing server is implemented as an additional function of a bandwidth broker and therefore we called the router server a bandwidth broker. The bandwidth broker was incorporated in network simulator (ns). We performed simulations to study the impact of dynamic assignment of QoS paths on network and end-user performance for constant bit rate and connection oriented traffic with varying demands. The traffic specifi-

Table 2. Traffic flow configurations

Traffic Type	Flows	Capacity
CBR UDP	SCR01 to DES01 (flow1)	2M
CBR UDP	SCR02 to DES02 (flow2)	15M
CBR UDP	SCR03 to DES03 (flow3)	5M
TCP	SCR04 to DES04 (flow4)	1M
TCP	SCR05 to DES05 (flow5)	1M
TCP	SCR06 to DES06 (flow6)	1M

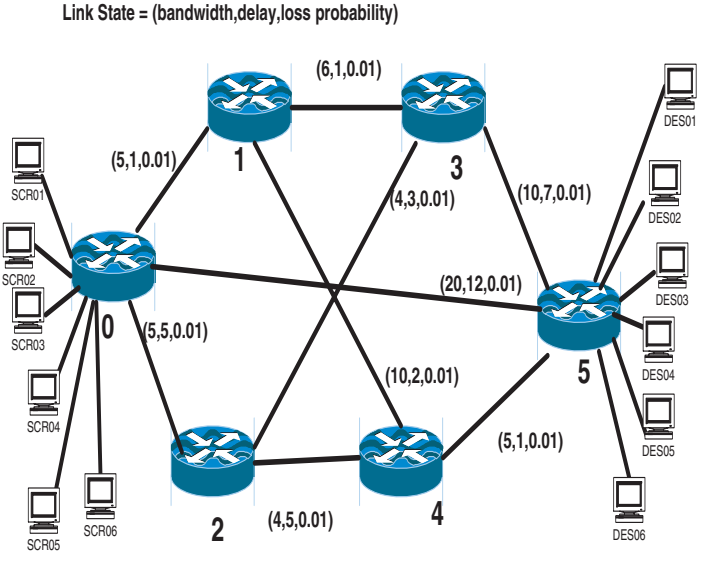


Fig. 2. Network Topology

cations of the flows used in the simulation are summarized in Table 2 (Please relate Table 2 to Figure 2).

A distance vector protocol is setup using the network topology shown in Figure 2 with MPLS protocol running on all the core nodes in the network. For simlicity, the BB is implemented on the edge device LSR0. To illustrate the impact on using QoS routing the above network is operated without any QoS routing in which case all packets travel the shortest path route determined by the distance vector protocol, and then with the BB doing admission control and QoS routing. For both scenarios flows 1-6 are started at 0.21, 0.31,0.41, 0,51,0.61 and 0.71 seconds respectively.

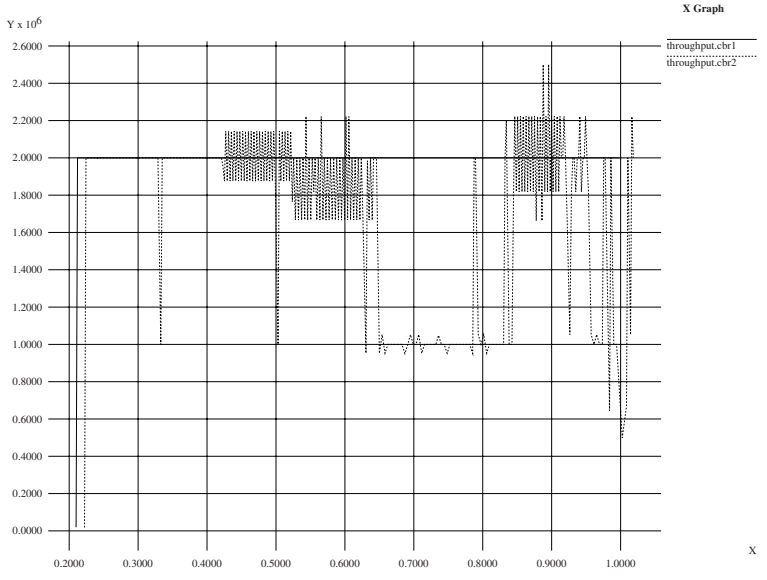


Fig. 3. Throughputs of source and receiver for flows 1 without a Bandwidth Broker

6 Simulation Results / Analysis

Figure 3 shows the throughput graph for flow 1 between SCR01 and DES01 with throughput in bps on the y-axis and time in seconds on the x-axis. When flow 2 started at 0.31s the throughput of the receiver fell sharply but returned to an average throughput of 2M. When flow 3 started at 0.41s the receiver throughput fluctuated around the average throughput of 2M. As flows 4, 5 and 6 are started at 0.51s, 0.61s and 0.71s respectively the throughput of the receiver decreases gradually. In contrast Figure 4, shows the throughput graph for flow 1 when a BB is functioning as a routing server and installing appropriate paths for admitted flows. When flow 2, 3 and 4 are started the throughput of flow 1 is not affected and continues to remain at 2M. As flows 5 and 6 are started at 0.61s, 0.71s respectively the throughput of the receiver fluctuates but still maintains an average throughput of 2M. A possible reason for this fluctuation is that the flows 5 and 6 are TCP flows which are congestion sensitive.

The BB assigned paths 0-2-3-5, 0-5, 0-5, 0-2-3-5 and 0-2-3-5 to flows 1, 2, 3, 4 and 5 respectively. The BB rejected flow 6 since no negotiation policy was specified.

7 Conclusion and Future Work

This paper shows that a bandwidth broker performing admission control with QoS path selection as a by-product provides a significant advantage with regards to traffic engineering. The preference function approach to finding QoS

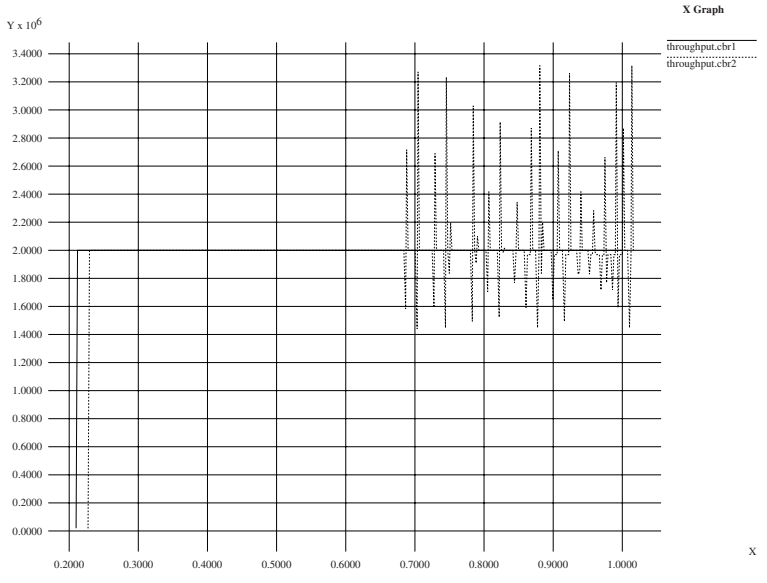


Fig. 4. Throughputs of source and receiver for flows 1 with a Bandwidth Broker

paths for multi metric criteria allowed the bandwidth broker to dynamically find suitable paths for traffic flows in real time. Simulation shows that the preference approach allows for better network resource utilization. An issue that need further investigation when incorporating the BB as a routing server is the maintenance of accurate network topology information so that reliable routing decisions are made. Since the bandwidth broker is the one responsible for allocating bandwidth then the BB should be able to estimate changes in metrics such as link bandwidth, queuing delay and delay jitter. However, we still need to study how reliably the BB can operate with imprecise network information; and links failures and dynamic network topology changes which will introduce major management issues for the BB.

References

[1] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet" Internet Draft, December, 1997. 390

[2] X. Xiao and L. Ni, "Internet QoS: A Big Picture", IEEE Network March/April 1999, **pp8-18** 2000 391

[3] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications", IEEE Journal on Selected Areas in Communications, vol. 14, n. 7, **pp. 1228-1234**, 1996. 391

[4] R. Guerin and A. Orda, D. Williams, QoS Routing Mechanisms and OSPF Extensions, in Proceedings IEEE GLOBECOM, vol. 3, **pp. 1903-8**, 1997 391

[5] S. Chen and K. Nahrstedt, On Finding Multi-Constrained Paths. IEEE ICC'98, **pp 874-879**, June 1998. 391

[6] R. Guerin and A. Orda, D. Williams, "The COPS (Common Open Policy Service) Protocol," RFC 2748, Jan. 2000. 392

[7] J. Barzilai, "A New Methodology for Dealing with Contradicting Engineering Design Criteria," Proceedings of the 18th Annual Meeting of the American Society for Engineering Management, **pp. 73-79**, 1997. 393

[8] J. Barzilai, "Notes on measurement and Decision Theory," Proceedings of the NSF Design and Manufacturing Research Conference, San Juan, Puerto Rico, **pp.1-11**, 2002 393

[9] J. Barzilai, "On the Foundations of Measurement," Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, **pp. 401-406**, 2001. 393

Improvments for Dynamic Sub-mesh Restoration Scheme in Dense WDM Networks

Chen-Shie Ho *, Ing-Yi Chen **, and Sy-Yen Kuo

Department of Electrical Engineering, National Taiwan University
Taipei, Taiwan
hocs@lion.ee.ntu.edu.tw

Abstract. Traditionally link/path protections are widely used means to assure network survivability. Recently sub-path flow-based and sub-mesh topology-based segmentation protection methods are proposed. In this paper, we propose improvements to the sub-mesh-based approach which include potential capacity estimation, adaptive group reconfiguration and corresponding control signaling mechanisms. We consider mainly the dynamic group protection strategies under dynamic traffic demands and examine the relative influence of various network metrics. Simulated performance results for the proposed heuristics are given and discussed.

1 Introduction

Wavelength division multiplexing (WDM) network offers the capability of building large wide-area networks with Tbps order of throughput, and is imperative that these networks are implemented by effective fault tolerance mechanisms, that is, the ability of a network to reconfigure and re-establish original communication upon failure, to minimize the huge avenue loss once if there exists failures[1][2][3]. This paper will mainly consider the dynamic traffic condition on the wide-area wavelength routed mesh topology based backbone environment. The survivability mechanisms against link/node failure in optical layer can be divided to preplanned protection scheme and dynamic restoration scheme. These methods are either link-based or path-based[3][4][5]. In the meanwhile, the primary-backup multiplexing proactive protection method can be employed to further improve resource utilization in expense to gain less than fully restorable guarantee if two primary(working) lightpaths do not fail simultaneously[5]. Recently the sub-path or segment protection strategy was proposed[6][7][8]. The main idea is dividing a given working path into some segments while protecting each segment separately. In [7], a variation of sub-path protection is presented which partition the network to protect the working lightpath segment in each

* Chen-Shie Ho is also with the Department of Computer Science and Information Engineering, Van Nung Institute of Technology, Chung-Li, Tao-Yuan, Taiwan.

** Ing-Yi Chen is with the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 400–409, 2004.

area. It has advantages in multiple failure recovery, ILP scalability and fast recovery time, and suitable for low connection loading condition. When the number of traffic demands increases, the capacity of each area should be adjusted to fit new spare capacity requirements. In [8] a segment partition method under the short leap shared protection (SLSP) framework is proposed. The SLSP provides finer service granularity and higher network throughput, and several algorithms were proposed to perform segment allocation and path selection to realize SLSP. However, all the proposed ideas mainly work on a per-connection basis. In [12], we proposed the sub-mesh restoration scheme to further consider the resource utilization benefit from the viewpoint of multi-flow topology partitioning. The protection group configuration setup is fully dynamic and need not the pre-configuring process in expense of complex distributed information exchange between network nodes by control channel and expensive but efficient resource calculation. The formation of the group is determined by current network resource status and considerations included into the weight function. The goal of group switching is attempting to make efficient management for these hierarchical protection autonomous areas and completely exploit the potential resource within the protection group.

2 Group Protection Switching

We illustrate the group switching concepts in Figure 1. Assuming that there is a connection request between (s, d) = (A,K). After setup process the working path is determined to A-B-C-D-K. If one chooses the protection group A-B-C-D-K-J-I-H-G-F-A as the resource pool for any failure restoration located within this area, A-B-C-G-H-I-J-K will be the spare path if there are more available network capacity existed in C-G than C-H or C-D. As another example, if request (C,G) arrives, the protection path set will be {(C-B-A-F-G),(C-H-G),(C-D-K-J-I-H-G)}. The final choice will depends on which path in the set has the smallest weighting value. Consider the case if request (K,J) arrives, the protection path may be chosen as K-D-C-H-I-J. Namely, the working-protection path multiplexing is allowed. The approach in [12] uses this concept (which is called group protection) with link state protocol which is used to distribute topology information about the network to form a new protection scenario. Given the topology, when a new connection request arrives, we first determine the working path, which is not necessarily a shortest path but a least load path from the source to the destination. That is, the least load first strategy is adopted as the working path selection criteria. Then the wavelength which is determined in the forward reservation period will be assigned to this path.

Once the completion of forward reserveation, we decide a protection path by a weighting criteria that indicates the network capacity metrics, consisting of hop count of the path, available wavelength on the links, node connectivity and lightpath setup cost within the group. The objective of the group finding algorithm is to determine the suitable protection domain compassing by the working path and protection path calculated based on the network status. This

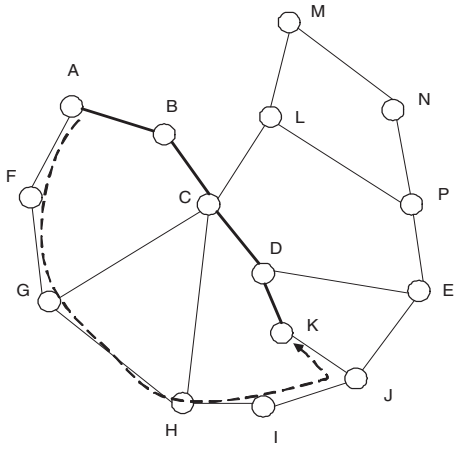


Fig. 1. Illustration of the group-based protection mechanism

protection group will provide necessary backup resource for successive reliable connection requests which source and destination pair located at this group. Each time a new connection arrives maybe affects the protection area size since it changes dynamically depending on the network capacity status. The motivation behind the group protection is that we can use the weight functions to bound the connection blocking probability then to improve network performance[11]. To choose a suitable protection path, the frequent exchange of control signals will be an overhead to each node, and this overhead can be reduced if there exists a group manager that is responsible to perform the task of collecting information sent from other group manager by out-of-band control channel.

The rest of this paper is organized as follows. Section 3 describes the details of the improved dynamic group protection. The results of the simulation experiments are discussed in Section 4. Section 5 presents the control mechanism to adapt and optimize the signaling complexity, and the final section summarizes this paper.

3 Improved Dynamic Grouping Protection

This section presents the dynamic group switching (DGP) scenario and its improved version IDGP. In dynamic traffic condition, the connection requests arrive in a random manner. When a new reliable-demand connection request arrives, we find the working-protection path pair by two node-disjoint paths in two-step approach to protect against single link/node failure. The distributed control signaling mechanism like link-state routing protocol will exchange provisioning status between every adjacent nodes following the forward control packets. We assume that all of the routes between one node to another are stored in each node. So the working-protection paths can be taken just from the routes database. It will

first find a feasible path as the working path, then we use the simple weighting function $w = f(H, W, L, S)$ to find the protection path[11][12]. In this function, H means the hop count which is the link count summing up from source to this intermediate node. W means the available wavelength in a link which is recorded in the resource table in each node for wavelength switching. L indicates the node traffic loads which will be summed for all ports in a node because each node may have higher degree but with unbalanced load distribution. S specifies the setup cost which depends on the type of switching equipment and maybe has different typical values. The scaling factor for each parameter is different for various managing and control strategy. The node-disjoint path finding process is summarized in [11]. The working-protection pair forms a contour as the protection group. The successive connection will overlap the protection path by partially or fully shared with the previous protection path. The connection maybe covers many protection groups so that it will be divided into different segment automatically. In [12], we set the scaling factor for H and W as higher value when performed the simulation. Also we use a simple formula to estimate the resource utilization level, which means the resource capacity in a group. To increase the spare capacity utlization we attempted to push the decision to choose protection path towards the common-used protection ring(as an example, the A-F-G-H-I-J-K path in Fig. 1). All the protection paths corresponding to their working paths will tend to approach to the largest and outside ring so that the nodes along the protection ring should calculate the value of the resource utilization level. This value has to be less than some pre-determined or dynamic-assigned value and it will be regarded as an index of the resource shortage effect which will make the nodes try to find another protection segment having longer route length and more available capacity. From our investigation, the changing process occurs so frequently that much amount of calculation at each node in the common-used protection path are made. This will further increase the signaling complexity and recovery cost, and then decrease the restoration performance. We propose sseveral improvements with respect to the oroginal DGP approach and describe them in the following subsections.

3.1 Potential Capacity Estimation

We found from the simulation that the estimation formula can be modified to let all the potential capacity in one group can be exploited. All the nodes in a group has to perform this calculation according to loading of itself. This calculated value will flood to all other nodes in the same group for sucessive working-protection pair routing decision. Compared to the estimation process in [12], only the nodes along the working path have to make the calculation in the real-time manner in this time. That is, if there is any working path locating at the inner portion or the boundary of the group, the update control signal will be sent form the intersection nodes to other nodes. To maintain the restorability we give a constraint to the traffic across any link by 1-working 1-protection(1W1P) multiplexing limitation. If there is condition which violates this limitation then the group expansion process proposed in [11] will be activated. The experiment

result shows that the signaling complexity in one group is lowered effectively and the occurrence frequency of group expansion process will change according to different source-destination node pair distribution. We will discuss all the results in Section 4.

3.2 Adaptive Group Size Control

In the former section we mentioned that the potential capacity estimation process should be made in the nodes along the newest working path. We can further decrease the control complexity if we assign a pre-determined lower limit to each calculation. Also, the potential capacity will relate to forgoing and current resource status. Assume the potential capacity value for current request be indicated as C_i, then current potential capacity in node j will be:

$$C_i = \alpha * C_{i-1} + (1 - \alpha) * \textit{working bandwidth in node } j \tag{1}$$

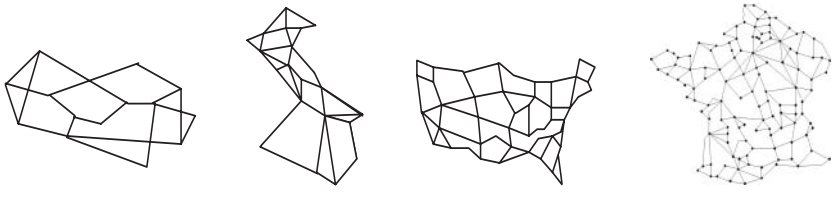
The goal of this calculation is to minimize the amount of exchanging control signal when there is no need to flood excessive capacity information by control packet sent from the nodes which are intersected with the working path. The calculated value will be compared with the pre-determined limit value. In fact, we can avoid extra calculation by assign an intersection factor which means the overlapping degree of protection group and current working path. The update packet can be sent only if the number of overlapped node exceeds a fixed upper limit. The effect of the unnecessary control overhead elimination will be prsented in Section 4.

4 Simulation Results

We evaluate the effectiveness of the proposed algorithms by performing simulations. We list the characteristics of simulated networks in Table 1. Each link in the networks is assumed to be a bi-directional duplex channel consists of 16 wavelength channels. We also list the average nodal degree in Table 1. The connection requests arrive at a node as a Poisson process with exponentially distributed holding time with unit mean. Every node has the same probability to be a source or a destination node for a connection request. The amount of exchanged control packets versus loading factor (in Erlang unit) in IGDP strategy is plotted in Figure 2. We compare the results of DGP and IDGP on 2 dense network samples and additional path-based restoration on network sample 4. The periodic updating control mechanism is used in path-based restoration. The duration of update message delivery is set to be 30 seconds. So the amount of exchanged message of path restoration is relative to the total simualtion period. The event-driven updating control mechanism is chosen for DGP and IDGP. Because of the message flooding of potential capacity and dynamic group size adjustment information, the IDGP outperforms DGP but is rather worse than path-based method. One can find that there exists the advantage in dense networks under

heavy traffic demands than sparse networks since larger group size is often produced in latter networks. We also listed IDGP without and with lower bound of potential capacity respectively in Table 2 under 120 Erlang offered loads. It shows that about 25% control overhead saving in the latter (called IDGP-A) improvement can be achieved in dense networks. In addition, we use 3 metrics, connection blocking probability, resource utilization,and sharability to measure the performance of the proposed mechanisms. We compare the performance of our methods with which of only use path protection. The blocking probability is evaluated by the success ratio that the spare path could be designated during working path establishment phase. The working-protection pair has to be discovered if the current request is belonging to a reliability-required connection request. If the process failed to assign a proper backup route then this request will be blocked.

We demonstrate the blocking probability and resource utlization in IGDP strategy in Figure 3. We note from this figure that the blocking probability value can be adjust dynamically according to the selection strategy of protection group. That is, larger group size makes lower probability value because of higher spare capacity and more degree of sharability. Fixing or reducing the group size will also make more blocks since the heavy contention of the same group area. Also note that even the network 4 has the highest nodal degree and more dense than other network samples, the curve shows that its performance is not the best since there implies some relations between the path length and the group size. When the loading is higher, the traffics which located at one group increase and then decrease the setup successful ratio because of the lower sharability of protection resource. The local wavelength utilization which means the ratio of



utilizing wavelengths and fiber ports within the protection group is higher than the global ratio which means the degree of whole network resource accessing. We only illustrated local utilization in the figure.

The sharability in one group under various traffic loads for network 4 is showed in Figure 4. The average number of connection in a group is recorded and it reveals that only few number of reliable demand can be accomplished within a fixed sized group at some time instant. It can be improved if the larger size group is chosen each time when any connection request arrives.

Table 1. Sample network topology information and statistical simulation results

Target network	Node number	Edge number	Average node degree	Average group size
(a)NSFnet	14	21	3.00	5
(b)Italy	21	35	3.33	4
(c)USAnet	46	76	3.32	9
(d)French	122	214	3.51	7

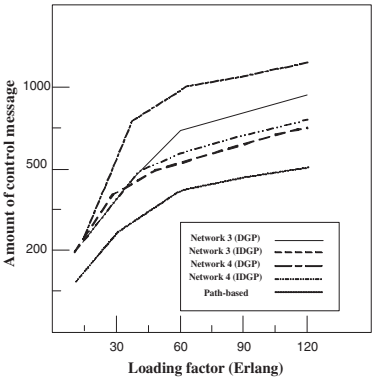


Fig. 2. Performance evaluation in IDGP

Table 2. Control overhead in IDGP and IDGP-A

Target network	# control signal in IDGP	# control signal in IDGP-A	Saving %
(a)NSFnet	483	436	9.73
(b)Italy	475	441	7.16
(c)USAnet	674	546	19.0
(d)French	703	539	23.3

5 Control Mechanism

As mentioned before, the dynamic resource allocation and provisioning is conducted by signaling operations riding over the control channel. The control channel provides connectivity over the links in the network and thereby reflects the latest physical network topology. The amount of control signaling message exchanged and management strategies will affect the restoration time heavily. In this paper, we will adopt the flow-based signaling protocol to establish the message exchange mechanism. Each connection or its segment locates at specific group domain will be designated to a fixed flow identifier, and each node will store the connectivity with other nodes which are in same domain. The flow ID and its protection-related resource will be recorded in the intermediate nodes along the working path when it is provisioned. There is also an additional bit in

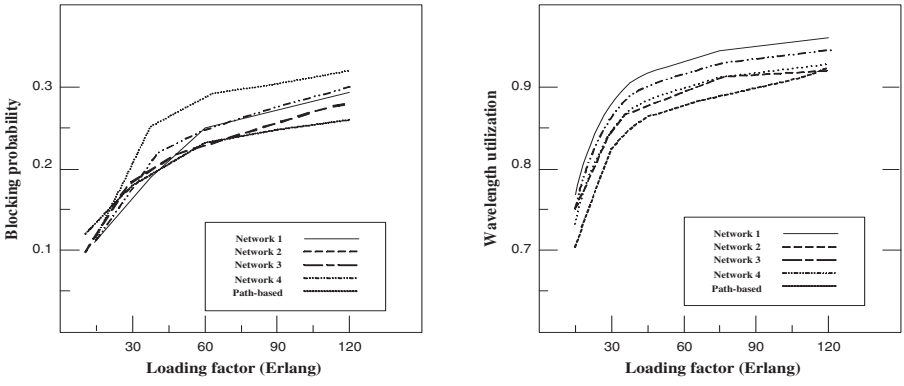


Fig. 3. Various simulation parameters vs. loading factors

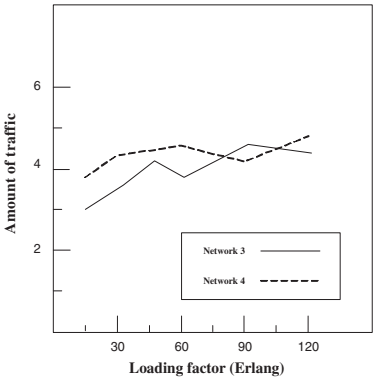


Fig. 4. Sharability in IDGP

control packet to indicate if it is a normal exchange message or failure notification mess-age. These messages can be exchanged periodically or in event-driven manner. The latest resource utilization and the routes information will be saved in the protection table. When the failure occurs, the node which detects this situation will send the failure notification message to the closest node located at the protection path, this process will then continue until the source and destination nodes are aware of the failure condition and perform protection switching. By this way, the operations performed by the OXC nodes in the control plane can be described as follows :

```
Receive(EVENT)
If receive failure notification message then
{
   UPDATE: update the protection table
   PROTECTION PROCESSING:
      1. Find the backup path by protection table
```

```
      2. Send the failure message to the closest node in the
         backup path until it arrives the source/destination node
}
If receive normal control message then
{
   UPDATE: update the protection table
   PROTOCOL PROCESSING:
      1. Extract the control message to handle specific event
      2. Check the resource capacity by scaling function
}
If receive RELEASE message then
{
   Release the resource reserved for this connection
}
If receive SETUP message then
{
   Find the working-protection path pair by cost function, record
   this information in the protection table, send control messages
   to other nodes, perform forward reservation
}
```

The signals are assigned to different priorities. The priority of various control signals from high to low are UPDATE, RELEASE and SETUP, respectively.

6 Conclusion

This paper studied the dynamic group protection based restoration heuristics in optical WDM networks. For the dynamic traffic demands, rather than the traditional link/path form restoration schemes, the adaptive protection group specified by available network capacity is more flexible and properly reflects practical condition due to the variant traffic demands. We extended the sub-mesh restoration with additional potential spare capacity calculation and adaptive group maintainment control mechanism. The simulation results reveal that about 25 percent of control overhead can be eliminated and the blocking level or local resource utilization can be held by the proposed strategies. The sharability anlysis also gives a rough sketch of the traffic load level in the dynamic protection group.

Acknowledgement

This research was supported in part by the Development of Communication Software Core Technology project of Institute for Information Industry and sponsored by MOEA, R.O.C.

References

[1] H. Zang, J. P. Jue, and B. Mukherjee: A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks , Optical Networks Magazine, Vol. 1, No. 1 (2000) 47–60
[2] S.Ramamurthy and B. Mukherjee: Survivable WDM mesh networks, Part II–Restoration, Proc. ICC, Vol. 3 (1999) 2023–2030
[3] S. Ramamurthy and B. Mukherjee: Survivable WDM mesh networks, Part I–protection, Proc. IEEE INFOCOM, (1999) 744–751
[4] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang: Optical network design and restoration, Bell Labs. Tech. J., Jan (1999) 58–84
[5] G. Mohan, Arun K. Somani: Routing Dependable Connections with Specified Failure Restoration Guarantees in WDM Networks, Proc. IEEE INFOCOM, Vol. 3 (2000) 1761–1770
[6] V. Anand, et al.: Sub-path protection: a new framework for optical layer survivability and its quantitative evaluation, UB CSE Tech. Report, Jan (2002)
[7] C. Qu, et al.: Sub-path protection for scalability and fast recovery in WDM mesh networks, Proc. OFC'02', Anaheim, CA, Mar (2002), 495–497
[8] P. H. Ho and T. M. Hussein: A framework for service-guaranteed shared protection in WDM mesh networks, IEEE Communication Magazine, Feb (2002), 97–103
[9] M. Medard, S. G. Finn and R. A. Barry: WDM loop back recovery in mesh networks, IEEE INFOCOM, (1999), 744–751
[10] W. D. Grover: Distributed restoration of the transport networ, Telecommunications Network Management into the 21st Century, , NJ: IEEE Press,(1994) 337–417
[11] C. S. Ho, I. Y. Chen and S. Y. Kuo: An efficient restoration scheme using protection domain for dynamic traffic demands in WDM networks, ICOIN'03, Jeju, Korea, Feb (2003) 108-117
[12] C. S. Ho, I. Y. Chen and S. Y. Kuo: Dynamic sub-mesh protection under dynamic traffic demands in dense WDM networks, AINA'04, Fukuoka, Japan, Mar (2004), to be appeared.

Analysis and Modeling of Traffic from Residential High Speed Internet Subscribers*

Sung-Don Joo[1], Chae-Woo Lee[1], and Yeon Hwa Chung[2]

[1] School of Electrical and Computer Engineering, Ajou University
San 5 Wonchon-dong Paldal-gu, Suwon, Korea
{sungdon,cwlee}@ajou.ac.kr
[2] Technology Laboratory, Korea Telecom Engineering
463-1 Jeonmin-Dong, Yuseong- Gu, Daejeon, Korea
yhj@kt.co.kr

Abstract. Recently we observe wide use of P2P(Peer to Peer) applications of which traffic shows different statistical characteristics compared with traditional applications such as web and FTP(File Transfer Protocol). In this paper, we measured traffic from a subscriber network of KT(Korea Telecom) to analyze P2P traffic characteristics. Analysis results show that P2P traffic is much burstier than web traffic and its upstream and downstream traffic is symmetric which is also very much different from that of conventional applications. To predict QoS related parameters such as packet drop probability and delay, we modeled P2P traffic using two self-similar traffic models and compare their accuracies. With simulation we show that the self-similar traffic models we derive predict the performance of P2P traffic accurately and thus when we design a network or evaluate its performance, we can use the P2P traffic model as reference input traffic.

1 Introduction

With the evolution of Internet, new applications appear continuously. Recently we see that P2P applications such as Napster and E-donkey are widely used. These applications have changed the way we distribute the contents. Before the advent of P2P applications, Internet traffic was asymmetric: there was more downstream traffic than upstream. Most of the information was located in the servers located in the Internet Data Center (IDC). In the past, Internet was usually used to surf the web, listen to the Internet radio, or download videos. Internet users consumed much more information than they generated. After the advent of P2P applications, however, each user is a contents provider as well as its consumer. Thus as P2P applications become popular, traffic distribution in the ISP's network becomes more symmetric.

* This work was supported in part by IT professor invitation program of Institute of Information Technology Assessment in republic of Korea.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 410–419, 2004.

QoS is a very important issue in Internet. In order to provide QoS effectively, we need to understand the characteristics of the traffic to which we want to provide QoS. To understand the characteristics of recent Internet traffic, especially P2P, and its impact to the network, we measured traffic at an access network for a group of high speed home subscribers with KT(Korea Telecom). In this paper, we present the statistics and traffic modeling results for the measured P2P traffic. To do that we analyze measured subscriber network traffic for both upstream and downstream flows and compare the self-similarity of emerging P2P and conventional web traffic. From the analysis, the traffic generated from P2P applications is found to be burstier than that of conventional applications such as FTP and web. The traffic model is an important factor in assessing the network performance and QoS of the given application since it can be used to estimate network performances such as delay and packet drop probability. We investigate packet drop probability and mean delay using SSQ(Single Server Queue) and predict these two parameters using M/Pareto and Fractional Gaussian Noise(FGN) models.

The rest of the paper is organized as follows. In section 2, we summarize self-similarity of traffic. In section 3, we describe two self-similar traffic models, M/Pareto model and FGN model, and the traffic model accuracy assessment method using SSQ. In section 4, we explain where and how we captured the traffic. In section 5, we present important statistics of the traffic from flow analysis. In section 6, we estimate self-similarity of the captured traffic and model it using the FGN and the M/Pareto models. Then we evaluate the accuracy of the models by predicting drop probability and mean delay using SSQ. In section 7, we conclude our paper.

2 Self-Similarity of Traffic

Self-similarity of traffic displays structural similarities across a wide range of time scale [1, 2, 3, 4]. This characteristic was found in Ethernet traffic by Bellcore researchers [3]. Let the time series X be a covariance stationary stochastic process. For each m=0,1,2,..., let $X^{(m)} = \{X_{mk}, k = 0, 1, 2, ...\}$ denote the new covariance stationary process obtained by averaging the original time series X over non-overlapping blocks of size m [1, 2, 3].

$$X^{(m)} = (1/m)(X_{t-m+1} + X_{tm-m+2} + \ldots + X_{tm}) \tag{1}$$

If stochastic process X has the following statistical properties, it is defined as exactly self-similar.

$$\begin{aligned} \mathrm{Var}[X^{(m)}] &= \frac{\mathrm{Var}(X)}{m^{\beta}} \quad \text{Variance} \\ R_{X^{(m)}}(k) &= R_X(k) \quad \text{Autocorrelation} \end{aligned} \tag{2}$$

Traffic models based on Markovian assumption have $\beta = 1$ and the variance of the time average decays in proportional to $1/m$. In self-similar stochastic process, the variance decays at a slower rate (m^{β} , $0 < \beta < 1$) than $1/m$.

In this paper, we use variance-time plot [5] to estimate Hurst parameter of P2P from a traffic trace. Variance-time plot is based on the slowly decaying variance of a self-similar process under aggregation. When the variance is plotted with respected to the aggregation scale(m) as a log-function, the process has self-similarity only when the plot has a slope smaller than -1. We can find the slope using the least square line fitting. Mathematically this can be described as follows.

$$\begin{aligned} &\mathrm{Var}[X^{(m)}] \sim \frac{\mathrm{Var}(X)}{m^{\beta}} \\ &\log[\mathrm{Var}(X^{(m)})] \sim \log[\mathrm{Var}[X] - \beta \log(m), \quad H = 1 - \beta/2 \end{aligned} \tag{3}$$

The self-similar nature of traffic may negatively affect QoS of the application because it can cause longer delay and larger packet drop. Since P2P traffic portion is large, it is interesting to find its degree of burst and to assess its impact to QoS and network performance.

3 Traffic Models

Many kinds of traffic models are suggested to describe the self-similar traffic. In this paper, we use M/Pareto and FGN models which are known to generate self-similar traffic easily. We briefly describe the two traffic models in this section.

3.1 FGN Model

FGN is an increment process of Fractional Brownian Motion (FBM) and is also asymptotically self-similar [6, 7, 8, 9]. FGN model describes the aggregated traffic $A(t)$ in time interal $(0, t]$ by

$$A(t) = mt + \sqrt{ma}Z(t), \tag{4}$$

where the parameter a is the index of dispersion count, m is the average amount of traffic in unit sample time, and $Z(t)$ is Fractional Gaussian Noise. In this model, the Hurst parameter H describes the self-similarity of Fractional Gaussian Noise $Z(t)$. The parameter a and the stochastic process $Z(t)$ determine the type of traffic while the parameter m does the amount of traffic. To summarize, FGN can generate traffic easily with three parameters m, a, H.

3.2 M/Pareto Model

M/Pareto model is a special kind of heavy-tailed ON/OFF model [10], and it is also a particular case of M/G/∞ process [11, 12]. In this model, the session inter-arrival time has exponential distribution and the period of a session has heavy-tailed distribution. Since each session represents a single burst, the arrival of bursts is described as a Poisson process with arrival rate λ. In the

M/Pareto model, burst durations(d) are independent and identically distributed Pareto random variables. This Pareto distribution gives the M/Pareto model self-similarity characteristics. The following parameters are necessary to model or generate traffic using M/Pareto model.

- Poisson arrival rate: λ
- Arrival rate of traffic within a burst: r
- Starting point of Pareto tail: δ
- Decreasing factor of Pareto: γ

We obtain the M/Pareto traffic model parameters using the mean, the variance, and the Hurst parameter of the traffic that we intent to model. This means that we have to find four parameters from the three observed statistical values. Thus we need to search the optimal values of λ and r without changing self-similarity. More detailed explanations on this model can be found in [11, 12] and the references therein.

3.3 Assessing the Accuracy of the Traffic Models Using SSQ

Since traffic models are usually derived from the statistical results of the traffic, it is very difficult to describe the captured traffic perfectly. Thus we need to assess the exactness of traffic models. A simple method to do that is by comparing the performance of SSQ when we feed the captured traffic and the traffic generated from the model into SSQ, since the network can be described as SSQ [13]. We use a discrete time queue model with FIFO (First In First Out). To present queueing process, we let A_n be a continuous random variable that represents the amount of traffic entered during n-th sampling interval, and let C denote the constant service rate of the server. Let Q_n be the unsent traffic at the beginning of the n-th sampling interval. Then the workload of the queue is expressed as

$$Q_{n+1} = \max(0, Q_n + A_n - C),\ n \geq 0. \tag{5}$$

In this paper, we investigate drop probability and mean delay while we change the utilization of the queue.

4 Traffic Measurement

With KT, we captured the traffic of around 400 residential subscribers at an apartment complex for the analysis of end user traffic characteristics. Currently, KT has more than 5 million high speed internet subscribers. Our measurement is performed at two DS3 lines(incoming and outgoing) connected to KT network and captured the traffic for 24 hours during Sep. 15 - 16, 2003.

To capture the traffic we used Tcpdump which can be used freely [14]. When we captured the traffic, to reduce the dumped traffic volume we limited the capture size to 68 bytes per each Ethernet frame. First 68 bytes from an Ethernet packet are enough to extract all the information we need to analyze the traffic such as IP addresses and TCP port numbers and the packet length.

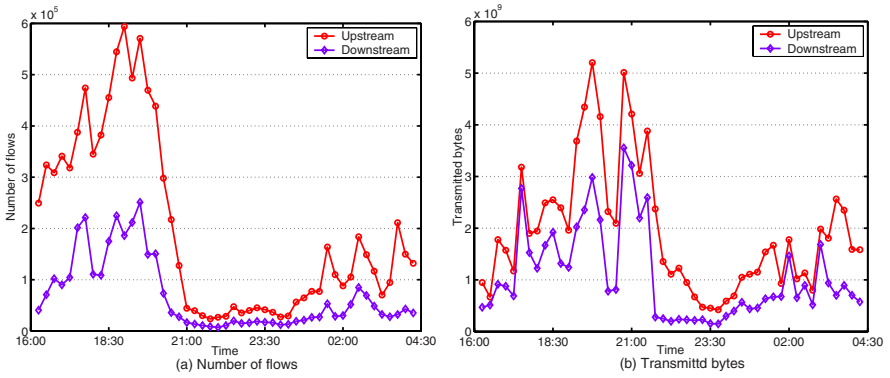


Fig. 1. The number of flows and bytes transmitted during one day

5 Flow Analysis

In this section we present flow and packet level statistics for the captured traffic. We used Coralreef for flow analysis [15]. Generally a flow is defined as a packet stream which has the same ⟨Source IP address, Destination IP address, Source Port, Destination Port⟩ combination [16]. In addition, we consider that a flow has been terminated if packets do not appear more than 64 seconds [16].

Fig. 1 shows the number of flows appeared and the number of bytes transmitted for each 30 minute interval. In Fig. 1, it is surprising to note that the volume of the upstream traffic is more than that of the downstream. This implies that many applications are not traditional client-server type. It may be difficult to observe this phenomenon with ADSL subscribers since its physical speed is asymmetric in nature. However, at a low utilization level in which the upload traffic does not saturate the upward link, we may observe similar phenomenon. In the figure, we can also find that the number of flows and the amounts of traffic transmitted are not proportional. In the midnights and very early in the morning, small number of flows generates traffic with high volume.

To understand why this disparity occurs, we analyzed the traffic with respect to the application type. We classified the traffic into a number of important applications using both protocol and port number. Applications like E-donkey, MSN messenger, V-share, Napster, Soribada, and Genie messenger are classified as P2P. Even though some other applications such as Guruguru are P2P, we did not classify them as P2P. Because such programs can assign port numbers dynamically, it is very difficult to track them only using the dumped result.

Fig. 2 shows the volume of the traffic that each application type generates during 24 hours. In Fig. 2, we can find that, in the downward direction web traffic takes considerable portion, however, in the upward direction, P2P traffic and unclassified P2P traffic which is classified as "others" in the figure take most of the traffic. In the downward direction, we can also find that the number of P2P flows is very high compared with its traffic volume. This is because some

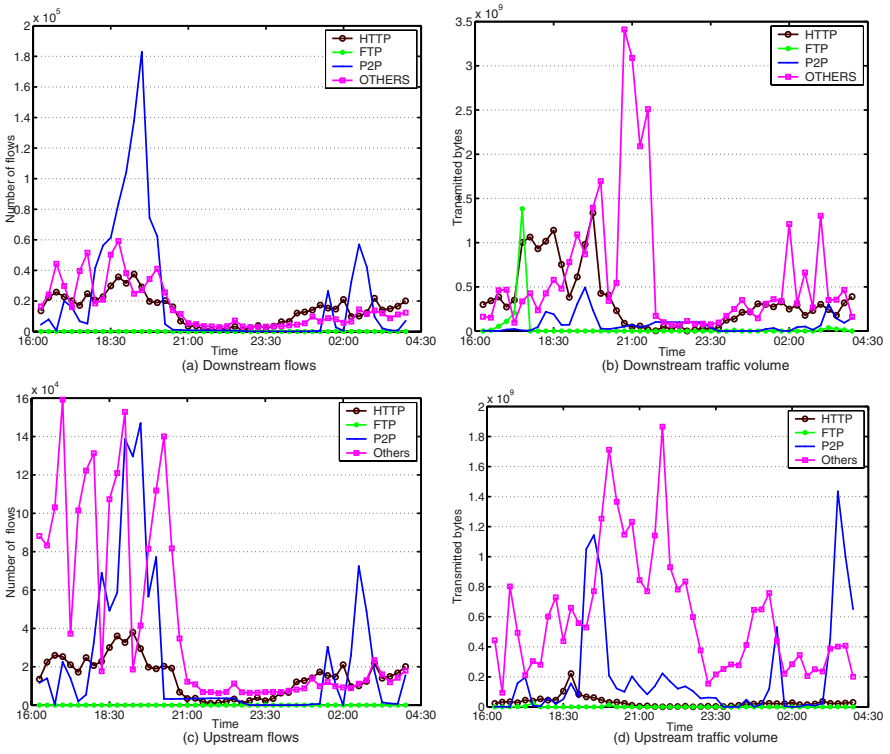
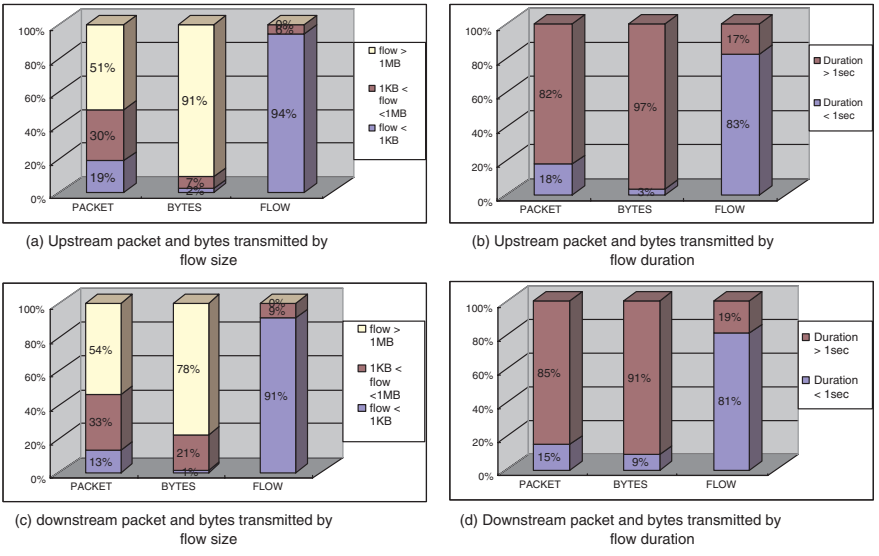


Fig. 2. The number flows and bytes transmitted to the upstream and the downstream

P2P applications such as Soribada exchange Hello Messages frequently using UDP. From Fig. 2, we can guess that the applications like FTP and Telnet are not much used these days and instead of FTP, file transfers are made using P2P programs.

We analyze the measured traffic according to its size and duration. In Fig. 3, we can see that more than 94% of the total flows are smaller than 1 Kbytes long, and around 91% of the traffic volume is generated by large flows, each of which generated more than 1 Mbytes. In the upstream, for flows larger than 1 Mbytes, the applications classified as "others" and P2P traffic respectively generated 66% and 33% of the traffic in volume. The duration of the flow shows a similar distribution. Flows that exist less than one second take more than 83% of the total flows, however, more than 97% of the traffic volume is generated by the flows longer than one second. For the flows longer than one second, the applications classifies as "others" and P2P flows respectively take 64% and 32% of the total flows. We can also observe similar flow statistics for the downward traffic except that the web traffic occupies second biggest portion.



(a) Upstream packet and bytes transmitted by flow size

(b) Upstream packet and bytes transmitted by flow duration

(c) downstream packet and bytes transmitted by flow size

(d) Downstream packet and bytes transmitted by flow duration

Fig. 3. The number of packets and bytes transmitted by the flows

6 Modeling of P2P Traffic

In this section, we compare self-similarity of P2P traffic with web traffic which is well-known for highly self-similar [4]. We model P2P traffic using the FGN and the M/Pareto models described in section 3. We also evaluate the accuracy of both models by predicting drop probability and mean delay in network and by comparing it with that of P2P traffic trace.

6.1 Measurement of Self-Similarity

To find the Hurst parameter, we use the traffic sampled between 11:00 and 12:00 p.m. which was the busy period for the residential subscribers. We estimate the Hurst parameter for the aggregated P2P traffic, instead of measuring the parameter for each P2P application type. To find the Hurst parameter we use the variance-time plot method which is one of the easiest way to find Hurst parameters[1][3][6]. Fig. 4 shows the variance time plots and Hurst parameters for both P2P and web traffic. In Fig. 4, as expected, the web traffic shows strong self-similarity with $H = 0.898$ while P2P traffic shows even stronger self-similarity with $H = 0.974$. This implies that P2P traffic is burstier than web traffic [3].

Since P2P already takes considerable portion in the network traffic and P2P traffic is burstier than other traffic, it is not difficult to imagine that P2P traffic affects the network performance negatively even if the traffic volume is the same. Accordingly QoS of other applications as well as that of P2P will suffer unless the network has effective QoS mechanism.

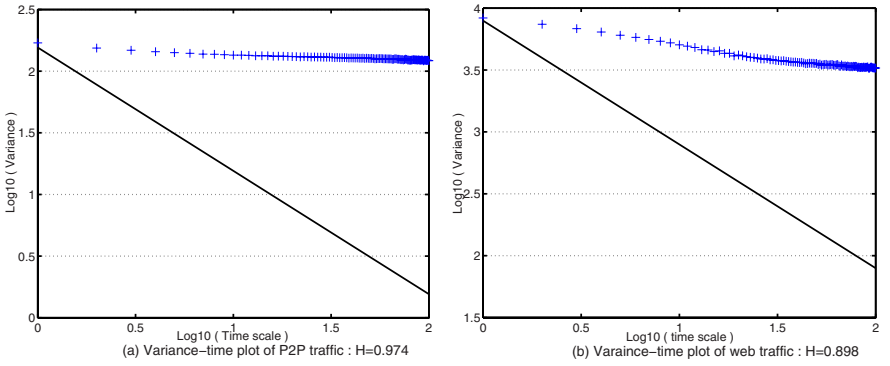


Fig. 4. Variance time plot for P2P and web traffic

Since the network is not designed to accommodate symmetric traffic between users, many ISPs in Korea implicitly limit the rate of traffic that a user can upload so that excessive congestions may not occur. In case of Ajou University, P2P traffic would consume most of the bandwidth if it does not limit the bandwidth allowed for P2P applications. As the number of P2P applications as well as P2P users is expected to increase continuously, soon one of the serious network management problems for ISPs will be how to manage P2P traffic so that QoS of other type of important applications are not suffered.

6.2 Traffic Modeling Results and Their Accuracies

To model busy hour traffic, we sample the traffic with an interval of 0.1s from the same busy hour traffic used in the flow analysis and then model the traffic. From the traffic samples, we get the following statistics: mean = 21.28 Kbytes/0.1sec, variance = 169.6 Kbytes^2, H = 0.974. Using the three values, we model the traffic with FGN and M/Pareto models. The model parameters for both traffic models are summarized in Table 1. To assess the accuracies of the both traffic models, we investigate drop probability and mean delay in SSQ while changing the utilization of the queue. We feed the traffic trace and the synthetic traffic generated from the traffic model into a SSQ and compare both drop probability and mean delay. Fig 5. shows the results. In Fig. 5, we can see that the FGN model models the traffic trace very closely when the utilization of SSQ is higher than 60%. However the M/Pareto model is a little less exact than FGN model in this utilization. At lower utilizations, however, the M/Pareto model can model the traffic trace much more accurately than the FGN model. From Fig. 5, when the entire utilization range is considered, we can determine that the M/Pareto models the traffic trace more accurately.

The above results imply that we can use the P2P traffic models we derived from the traffic trace as reference traffic input when we design a network or evaluate its performance. This will greatly reduce the time to analyze the network or QoS while providing relatively accurate results.

Table 1. Traffic model parameters

M/Pareto model parameter				FGN model parameter		
λ	r	δ	γ	Mean(m)	IDC(a)	Hurst(H)
3.8957	46.766	0.1	1.05	21.28 Kbytes/ 0.1sec	7.896	0.974

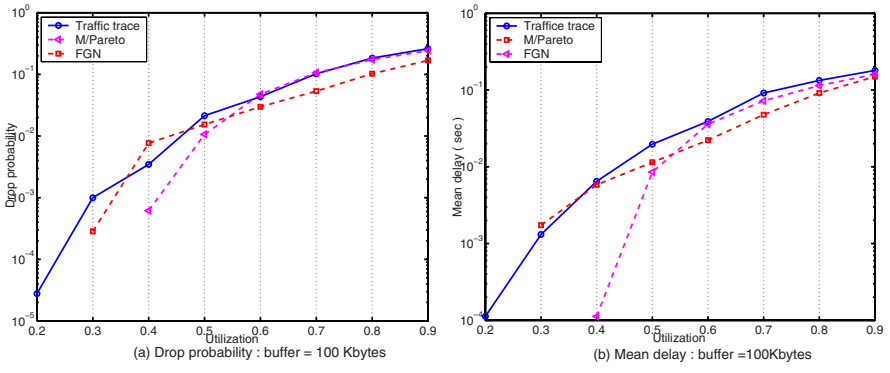


Fig. 5. Drop probability and mean delay of the traffic trace and the traffic models

7 Summary

In this paper, we analyzed the characteristics of emerging P2P traffic and modeled it with self-similar traffic models. The analysis showed that the characteristics of the network traffic are changing because P2P traffic is very bursty and symmetrical in nature, and it is widely used already. The impact of P2P traffic to the network can be summarized as follows. First, as more P2P applications appear we need to improve current asymmetrical network access technologies such as ADSL and Cable modem. They may not be able to meet the increasing upstream bandwidth requirement of the P2P applications. Second, the network topology of ISP needs to be changed. Currently ISP network is designed under the assumption that most of the contents are in its IDC or from other networks, and thus it is not efficient to exchange huge amount of P2P traffic between their subscribers. To accommodate this requirement, ISP's network may need to change to mesh topology. Third, Since P2P traffic is burstier than that of many conventional applications such as web and FTP and it takes major portion in network traffic in the near future, the network will become more easily congested and thus result in lower QoS.

We also modeled the traffic trace with FGN and M/Pareto models. We showed that they can estimate both drop probability and mean delay of traffic trace accurately. Thus the two traffic models can be used to predict end to end QoS of P2P applications as well as its impact to that of other applications. We expect that more P2P applications will emerge and P2P traffic will take most of the network traffic in the near future. In order to provide efficient network ser-

vice and QoS to applications, it is very important to understand network traffic and more studies are necessary.

References

[1] William, S.: High-Speed Networks: TCP/IP and ATM Design Principle. Prentice Hall, (1998) 182–207 411

[2] Park, K., Willinger, W.: Self-Similar Network Traffic and Performance Evaluation. JOHN WILEY & SONS, (2000) 1–38 411

[3] Leland, W. E., Taqqu, M. S., Willinger, W., Wilson, D. V.: On the self similar nature of Ethernet traffic (extension version). IEEE/ACM Transactions on Networking, Vol.2. No.1. (1994) 1–15 411

[4] Crovella, M. E., Bestavros, A.: Self-Similarity in World Wide Web Traffic Evidence and Possible Causes. IEEE ACM Transactions on Networking, December (1997) 835–846 411, 416

[5] Rose, O.: Estimation of the Hurst Parameter of Long-Range Dependent Time Series. Research Report, (1996) 412

[6] Norros, I.: A storage model with self-similar input. Queueing systems, Vol.16. (1994) 387–396 412

[7] Norros, I.: Studies for a Model for Connectionless traffic, based on Fractional Brownian Motion. Conference on Applied Probability in Engineering, Computer and Communication Science Paris, June (1993) 412

[8] Giordano, S., Pagano, M., Pannocchia, R., Russo, F.: A new call admission control scheme based on the self similar nature of multimedia traffic. IEEE ICC'96, Vol.3. (1996) 23–27 412

[9] Lau, W. C., Erramilli, A, Wang, J. L., Willinger, W.: Self-similar Traffic Generation : The Random Midpoint Displacement Algorithm and Its Properties. ICC'95, Vol.1. (1995) 466–472 412

[10] Pruthi P., Erramilli A.: Heavy-tailed ON/OFF Source Behaviour and Self-similar Traffic. ICC'95, (1995) 445–150 412

[11] Neame, T. D., Zukerman, M., Addie, R. G.: Application of the M/Pareto Process to Modeling Broadband Traffic Streams. ICON'99, (1999) 53–58 412, 413

[12] Addie, R.D, Neame, T. D., Zukerman, M.: Modeling Superposition of Many Sources Generating Self Similar Traffic. ICC'99, (1999) 387–391 412, 413

[13] Addie, R.D, Zukerman, M., Neame, T. D.: Fractal Traffic : Measurements, Modeling and Performance Evaluation. ICC'95, Vol.3. April (1998) 471–476 413

[14] `http://www.tcpdump.org/` 413

[15] `http://www.caida.org/tools/measurement/coralreef/` 414

[16] Thompson, K., Miller, G. J., Wilder, R.: Wide-Area Internet Traffic Patterns and Characteristics. IEEE Network, Vol.11. No.6. Nov-Dec. (1997) 10–23 414

Multi-constrained End-to-End Admission Control in Core-Stateless Networks*

Yong Cui, Ke Xu, and Jianping Wu

Department of Computer Science, Tsinghua University
Beijing, P.R.China, 100084
{cy,xuke}@csnet1.cs.tsinghua.edu.cn
jianping@cernet.edu.cn

Abstract. As one of the most important mechanisms in next-generation networks with QoS support, admission control that allots network resources should be scalable and efficient. Keeping the connectionless nature of the Internet without resource reservation, we propose a novel multi-constrained end-to-end admission control mechanism to provide statistical QoS guarantees. We divide the Internet into the core and edge networks hierarchically. The core only maintains its own QoS state and uses connectionless hop-by-hop QoS routing. The ingress router performs the admission control by coordinating with all of the routers along the end-to-end path and maintains the per-flow state. The four trips of a request in the admission control process reduce the impact of stale routing information. Simulations show the effectiveness of the proposed mechanism without resource reservation.

1 Introduction

It is a desirable feature of the future Internet to provide the end-to-end quality-of-service (QoS), especially to control the different QoS parameters (e.g. bandwidth, delay, cost, loss rate and etc.). However, strict QoS guarantees are too difficult to achieve in connectionless Internet. In the paper, we investigate the statistical QoS guarantees with connectionless hop-by-hop routing.

As an essential mechanism to allot the network resources [1], admission control is mostly studied based on three basic techniques: (A) Integrated Service (IntServ) [2], (B) bandwidth broker [3, 4], or (C) end-to-end measurement [1, 5, 6, 7, 8]. Most of the current admission control mechanisms have some of the following limitations: (1) Some are opposite to the essence of the current Internet, i.e. connectionless hop-by-hop routing without resource reservation (e.g. class A and B). (2) Some are not scalable to the number of the flows and/or the network scale (e.g. class A and B). (3) Some have a long delay in the process of admission control (e.g. class C). (4) Some cannot control different QoS parameters respectively including delay, cost, loss rate, bandwidth and

* Supported by: (1) the National High Technology Research and Development Plan of China (No. 2002AA103067); (2) the National Natural Science Foundation of China (No. 90104002; No. 69725003); (3) Chinese 973 project (No. 2003CB314801).

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 420–429, 2004.

so on (e.g. class B and C). Therefore, it is difficult to deploy them in the real Internet.

As the essence of Internet, the connectionless data flow with hop-by-hop routing brings the great success of the Internet. Hence, the proposed admission control mechanism follows the essence with connectionless QoS routing rather than resource reservation. It has five advantages: (1) Independent routing of different packets in a flow can be deployed as the essence of connectionless Internet. (2) It divides the Internet into two levels, the core network and the edge network, to enhance its scalability while support the end-to-end QoS. (3) The stateless core further improves the scalability. (4) The utilization of the network resources can be improved without resource reservation. (5) It can support different QoS parameters based on QoS routing independently.

The rest of this paper is organized as follows. We give the routing model as a background in Section II. The problem is formulated in Section III. In Section IV we present the proposed admission control mechanism. In Section V simulations are given to evaluate the proposal. Finally, conclusions appear in Section VI.

2 Network Routing Model

The next generation Internet needs to provide QoS support rather than the traditional best effort service. The Integrated Service (IntServ), providing the perfect QoS guarantees, cannot be deployed due to its limitation on scalability [2]. Although DiffServ is scalable, it cannot distinguish different flows to control multiple end-to-end QoS parameters (e.g. cost, delay, bandwidth, loss rate) separately. Although packet scheduling can provide QoS to a certain extent [9, 10], it is not a global optimal mechanism. For example, scheduling may lead that some links are congested while others are free. Therefore, routing is an important mechanism for QoS support. Even in some special cases, routing is the only way to support global QoS in the Internet. For example, there are two paths between two host in the network. One path is via satellite links, and the other is made up of optical fibers passing multiple Autonomous Systems (AS). Because the path via the satellite has fewer hops or costs in the configuration, it will be selected by traditional routing. However, its long delay and low bandwidth cannot satisfy the real-time high-bandwidth multimedia applications.

QoS routing is mainly studied based on IntServ with connection-oriented resource reservation to support QoS guarantees [11]. Because IntServ is not scalable, it cannot be deployed in the real Internet. The essence of Internet is simplicity, e.g. connectionless, hop-by-hop routing, no resource reservation. However, it is difficult to support QoS guarantees without connection-oriented resource reservation. Therefore, settling for the lesser to improve the practicability, connectionless QoS routing may achieve statistical QoS guarantees rather than the strict QoS guarantees.

Although it is an NP-complete problem to find a multi-constrained path, some proposed heuristics achieve a high performance with a good scalability [12,

13, 14]. Hence, the routing algorithm itself can be regard as a solved problem. QoS routing is usually under the assumption that all packets in each flow are pinned onto a specific path. However, a recent study shows that hop-by-hop QoS routing can still provide a statistical QoS guarantees at a high probability [15]. Additionally, the stability of the core network [16] further enhances the feasibility of connectionless QoS routing.

In the paper we assume that each router collects the local network state, e.g. available bandwidth, delay, loss rate and so on. The local state is flooded by hierarchical QoS routing protocols so that each router maintains the aggregated global network state. Pre-computation routing algorithms [12, 13] can pre-compute the QoS routing table based on the maintained network state. Because the QoS routing table performs well when it is only several times the traditional best-effort routing table, it is a promising routing mechanism in the future Internet.

Each packet of a QoS flow carries the QoS constraints. For example, each packet may use four bytes in the IP option field to contain the required bandwidth, delay, cost and loss rate. When the packet arrives at a router, the router looks up the next hop address in the routing table. Just like the Time-To-Live (TTL) field in the IP header, the constraints carried in the packet header are changed by the router according to the actual cost of the forward process on the router before the packet is forwarded to the next hop. For example, the new delay constraint is the old one minus the delay on this hop.

Since admission control can refuse the flows that networks do not have enough resources to support, the useless traffic can be decreased efficiently. Hence, admission control is essential to satisfy the QoS requirements of the flows that have already been accepted by the networks. Because flows are connectionless and there is no resource reservation, QoS routing can only support statistical QoS guarantees. Therefore, in order to improve the scalability and practicability, the object of the proposed admission control is consistent with that of QoS routing to provide the statistical QoS guarantees, i.e. to provide the QoS guarantees in a high probability [5].

3 Problem Formulation

A directed graph $G(V, E)$ presents the network. V is the node set and the element $v \in V$ is called a node representing a router in the network. E is the set of edges representing links that connect the routers. The element $e_{ij} \in E$ represents the edge $e = v_i \rightarrow v_j$ in G. Each link has a group of independent QoS weights $w = (w_0(e), w_1(e), \cdots, w_{k-1}(e))$, where integer $k \geq 2$. In the paper, the weight elements include additive weights (e.g. cost, delay, jitter), multipliable weights (e.g. loss rate) and minimum weights (e.g. bandwidth) (See definitions in [17]). Multipliable weights can be converted to additive weights by logarithm. Minimum constraint can be easily dealt with in a preprocessing step by pruning all links that do not satisfy the constraint and computing a path from the rest sub-graph.

Definition 1. *QoS Flow*
In an end-to-end QoS data communication, all of the packets in a single direction with same QoS constraints are made up of a QoS flow.

A QoS flow has the same source address and port number, destination address and port number, protocol type and QoS constraints generally. A QoS flow is also called as data flow or flow.

Definition 2. *Feasible Path*
For a given graph G, source node s, destination node t, $k \geq 2$ and a constraint vector $c = (c_0, c_1, \cdots, c_{k-1})$, the path p from s to t is called a feasible path, if $w_l(p) \leq c_l$ for each $0 \leq l \leq k-1$ (we write $w(p) \leq c$ in brief).

For a given QoS request and its constraints c, QoS routing seeks to find a feasible path p satisfying $w(p) \leq c$ based on the current network state.

Definition 3. *Remainder Constraint*
For the constraint c on the path $p = e_0 \rightarrow e_1 \rightarrow \cdots, \rightarrow e_m$, $c' = c - w(e_0) - \cdots - w(e_n)$ is called the remainder constraint for the sub-path $p' = e_(n+1) \rightarrow \cdots \rightarrow e_m$, where $0 < n \leq m$.

In the forwarding process of a packet, the remainder constraint represents that the efficient constraint for the later sub-path will decrease with the increase of the forward sub-path.

Definition 4. *Constraint Update*
For the QoS packet with constraint c on link $e = v_i \rightarrow e_j$, after node v_j receives the packet, it changes the constraint c maintained by the packet to $c' = c - w(e)$. This procedure is called constraint update.

Constraint update in the hop-by-hop forwarding process of a packet makes it possible that only remainder constraint is carried in the packet header. Thus, the following routers along the path can perform hop-by-hop QoS routing to find a feasible/optimal next hop based on the remainder constraint.

4 Connectionless Hierarchical Admission Control

The Internet measurement shows that the QoS state of a link in the core is comparably stable because the number of users is large enough [16]. On the contrary, the state of edge networks changes quickly because the relative small number of users changes quickly and the state may depend on a particular user. Therefore, in order to decrease the overload of QoS routing, we divide Internet into two hierarchies: the core network and the edge networks. The edge networks use the traditional best-effort routing. The core uses connectionless QoS routing based on the aggregated global network state of the core, while it does not maintain the QoS information of the edge networks to alleviate the burden of the core.

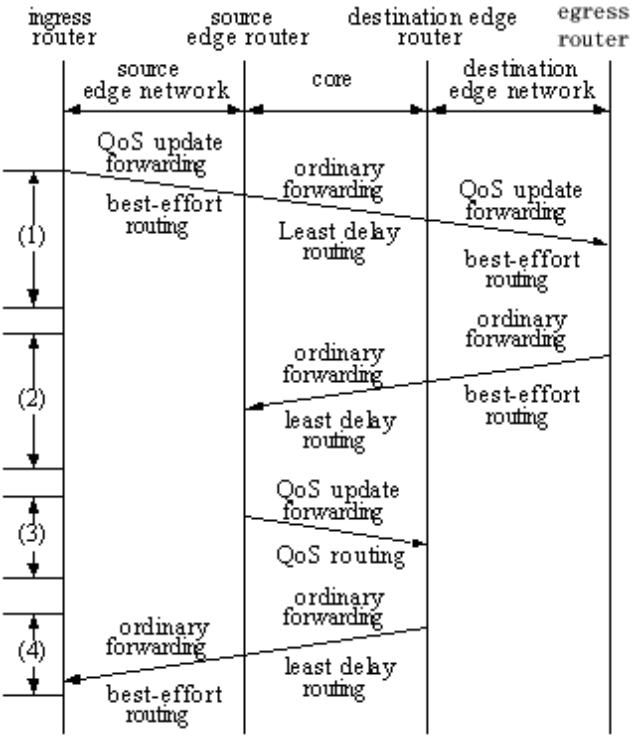


Fig. 1. Four trips of a request for admission control

Because the ingress router does not know the QoS state of the egress edge network and the core, it cannot provide accurate admission control. Additionally, in order to achieve QoS routing accurately in the core network, the core needs to know the remainder constraint that only affords the forwarding process in the core. Hence, we propose an end-to-end distributed admission control mechanism. Rather than being performed on the ingress router independently, admission control is achieved by all of the routers along the end-to-end path. The mechanism includes four forwarding trips of the QoS request as shown in Fig. 1.

The 1st Trip: Before the source host sends a QoS flow, it sends the destination host a QoS request including the concrete QoS constraints. When the request is forwarded from the source to the destination, the ingress and egress edge networks use best-effort routing with the constraint update. Thus, when the destination receives the request, the request only maintains the remainder constraints that the core network can use.

The 2nd Trip: QoS weights on a link are asymmetrical, and connectionless QoS routing can improve the asymmetry. A path satisfying the remainder constraints needs to be found from the source edge to the destination edge in the core, rather than from the destination edge to the source edge. Therefore, the destination sends the request maintaining the remainder constraints back to source edge router that connects the source edge network with the core. This trip may use the least delay path in the core.

The 3rd Trip: The source edge starts to find a feasible sub-path in the core that satisfies the remainder constraints. Because the staleness of the network state information and the dynamics of the hop-by-hop routing, even if the source edge router finds a feasible path, the packets may fail to be forwarded to the destination along the path selected by the ingress router. Consequently, the successful arrival of the request to the destination edge router means that the network state satisfies the QoS request.

The 4th Trip: After the destination edge router receives the request with non-negative constraints, it sends back the request to the ingress router to indicate that the request can be satisfied. In summary, the 1st trip gathers the QoS cost used in both the source and destination edge networks. The 2nd trip sends the remainder constraint to the source edge router. The 3rd trip seeks a feasible sub-path in the core and the 4th trip confirms the feasibility of the request to the ingress router.

In order to monitor the ingress flow, the ingress router takes charge of the response to the QoS request. In case of an acceptance of a flow, the ingress router sends the acceptance packet to the source host as the response after it finds that the networks have enough resource for the flow (i.e. the 4th trip succeeds). On the contrary, if any router along the end-to-end path finds the unfeasibility of the request, the router sends the refusal response to the ingress router. Then the ingress router forwards the response to the source host to refuse the request. Thus, the ingress router maintains an accepted flow table, according to which it can monitor each ingress flow to prevent illegal flows.

If the QoS request is accepted, the source host can send the QoS flow, where each packet should contain the concrete QoS constraints. In the forwarding process of the data packets, each router first uses a pre-computation or on-line computation QoS routing algorithm to seek a feasible path according to the remainder constraints carried by the packet. Then a feasible path is found for the remainder constraints, the router forwards the packet to the next hop with constraint update. Both the admission control and the QoS routing mechanisms support statistical QoS guarantees. Therefore, if a feasible path cannot be found in the forwarding process, packets can still be forwarded along a nearly feasible path.

5 Experimental Results

The efficiency of the network resources and success ratio are studied to demonstrate the necessity and the performance of the proposed admission control mechanism.

Definition 5. *Success Ratio*

The satisfaction percentage is defined as the number of the flows whose QoS constraints are satisfied divided by the number of the flows that admission control accepts.

5.1 Simulation Method

Because it is easy to avoid congestion in edge networks (e.g. recording per-flow state and high-speed flooding of network state), each edge network can be simplified as a node. Thus, we only simulate the core network and regard an edge network as an edge router for simplicity.

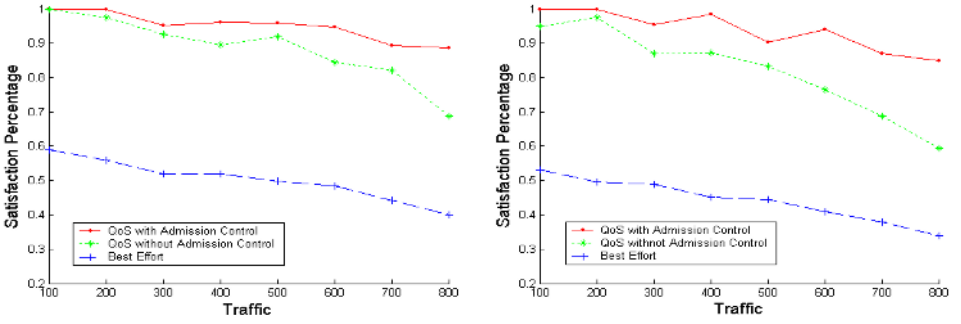
NS-2 is used in the simulation and we change its routing protocol to support QoS information. Each node uses the multi-constrained pre-computation algorithm (MEFPA) to pre-compute the QoS routing table [12]. We use GT-ITM [18] to generate two pure random network graphs: one with 50 nodes and 109 links, and the other one with 100 nodes and 235 links. For each link, we then generate a minimum QoS weight (e.g. bandwidth), which changes according to the accepted flows during the simulation, and three kinds of constant additive weights (may represent delay, jitter and loss rate). All of the weights obey uniform [1,1000].

We adopt the method of the threshold to update QoS state information [18]. The ratio of the available bandwidth is defined as $Bv = |Bw(new) - Bw(old)|/Bw(old)$, where $Bw(new)$ is the available bandwidth at present and $Bw(old)$ is the one at last update. We take the threshold 50%. Thus, when $Bv \geq 50\%$, the node will update the QoS state information to others and set $Bw(old) = Bw(new)$.

The flows are generated as follows: We first select two nodes randomly in the network, where the minimum number of hops between them is three. We then use MEFPA to find a least-energy path with a random energy coefficient a, where the energy is defined as $\sum_{l=0}^{k-1} a_l w_l$ [12]. Thus, we take the additive QoS weights of the path as the constraints of the generated flow. For the bandwidth of the flow, we take 10% bandwidth of the least-energy path in the initial network as the required bandwidth in 50 nodes graph, and 20% in 100 nodes graph. The flows obey the Poisson arrival.

5.2 QoS Performance Improved by Admission Control

Fig. 2 shows the satisfaction percentage of the flows with different flow intensities compared in three networks: QoS routing with admission control, QoS routing without admission control and best effort routing. The x-axis is the number of



(a) 50 nodes (b) 100 nodes

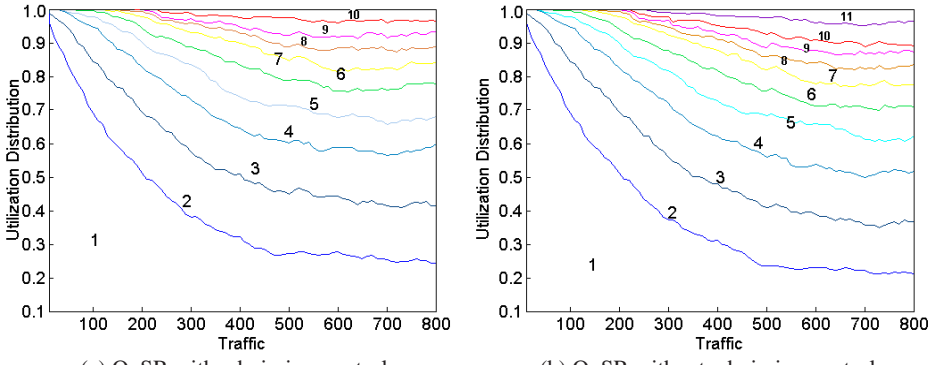
Fig. 2. Satisfaction percentage of flows

flows. We obtain three points here: (1) Satisfaction percentages in QoS routing are higher than that in best effort. The reason is that QoS routing can adjust the end-to-end path according to the requirement in the dynamic networks. (2) With the increase of the flow numbers, satisfaction percentages with QoS routing decrease, because of the staleness of the network state maintained by each node and simultaneity of the flow arrivals. (3) When the flow number is large (e.g. 500 or larger), the decline of the satisfaction percentage with admission control is much less than that without admission control. This aspect shows the necessity and performance of admission control to QoS routing.

In order to show the performance improved by admission control, the update threshold is set to zero. Fig. 3 shows the utilization distribution of the network resources in the case of Fig. 2 with/without admission control. Once 10 flows arrive, the utilization distribution of all links is computed. The y-axis is the percentage of utilization distributions. From the bottom to the top, each block presents the percentage of links whose utilizations are between $0\% - 10\%, 10\% - 20\%, \cdots, 90\% - 100\%$. The topmost block in Fig. 3(b) represents the percentage of congested links. Because there is no staleness, in Fig. 3(a) there are no congested links. Additionally, the utilization percentage between 90%-100% is much higher in Fig. 3(b) than in Fig. 3(a). Therefore, the absence of admission control increases the traffic burden on some links heavily. The congestion induced by the burden decreases the satisfaction percentage in Fig. 2 when the flow number is large.

6 Conclusion

We propose a novel approach, the connectionless distributed end-to-end admission control mechanism, to the statistical QoS guarantees for future Internet. Based on the connectionless hop-by-hop routing, we divide the Internet into two hierarchies: the core and the edge networks. The core uses hop-by-hop QoS routing while it only maintains the QoS state of itself. Per-flow state is maintained



(a) QoSR with admission control (b) QoSR without admission control

Fig. 3. Distribution of link utilizations

on the ingress router. The proposed admission control mechanism is accomplished in four trips of a QoS request with the coordination of the routers along the end-to-end path. Extensive simulations show that the mechanism can provide statistical QoS guarantees with a high probability. The hierarchical network model used in the mechanism enhances the scalability of QoS routing. Additionally, the connectionless hop-by-hop routing without resource reservation is still kept. Therefore, the scalability and practicality of the mechanism is obvious.

The paper is our first step into QoS control mechanisms in connectionless hop-by-hop routing networks. We are now developing such router prototypes to validate the practical feasibility of these mechanisms.

References

[1] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica and H. Zhang, Endpoint admission control: architectural issues and performance, ACM SIGCOMM, 2000 420

[2] R. Braden, D. Clark and S. Shenker, Integrated services in the Internet architecture: an overview, Internet RFC 1633, June 1994 420, 421

[3] S. Bhatnagar and B. Nath, Distributed admission control to support guaranteed services in core-stateless networks, IEEE INFOCOM, April 2003 420

[4] Z. Zhang, Z. Duan, L. Gao and Y. Hou, Decoupling QoS control from core routers: a novel bandwidth broker architecture for scalable support of guaranteed services, ACM SIGCOMM, pp. 71-83, October 2000 420

[5] E. Knightly and N. Shroff, Admission control for statistical QoS: theory and practice, IEEE Network, vol. 13, no. 2, pp. 20-29, 1999 420, 422

[6] D. Tse and M. Grossglauser, Measurement-based call admission control: analysis and simulation, IEEE INFOCOM, April 1997 420

[7] L. Breslau, S. Jamin and S. Shenker, Comments on the performance of measurement-based admission control algorithms, IEEE INFOCOM, April 2000 420

[8] Peter Key and L. Massoulie, Probing strategies for distributed admission control in large and small scale systems, IEEE INFOCOM, April 2003 420
[9] J. Bennett and H. Zhang, Hierarchical packet fair queuing algorithms, ACM SIGCOMM'96, Aug. 1996 421
[10] L. Zhang, Virtual Clock: A new traffic control algorithm for packet switching networks, ACM SIGCOMM'90, pp. 19-29, Sep. 1990 421
[11] S. Shenker, C. Partridge and R. Guerin, Specification of guaranteed quality of service, IETF RFC 2212, September 1997 421
[12] Y. Cui, K. Xu and J. P. Wu, Precomputation for multi-constrained QoS routing in high-speed networks, IEEE INFOCOM, April 2003 422, 426
[13] A. Orda and A. Sprintson, QoS routing: the precomputation perspective, IEEE INFOCOM, vol. 1, pp. 128-136, 2000 422
[14] Y. Cui, K. Xu and J. P. Wu, Adjustable multi-constrained routing with a novel evaluation method, IEEE IPCCC 2003, Phoenix, AZ, April 2003 422
[15] P. Van Mieghem, H. D. Neve and F. Kuipers, Hop-by-hop quality of service routing, Computer Netwroks, vol. 37, pp. 407-423, 2001 422
[16] C. Barakat, P. Thiran, G. Iannaccone, C. Diot and P. Owezarski, A flow-based model for Internet backbone traffic, ACM SIGCOMM'02 Internet Measurement Workshop (IMW), 2002 422, 423
[17] S. Chen and K. Nahrstedt, An overview of quality-of-service routing for next-generation high-speed networks: Problems and solutions, IEEE Network, vol. 12, no. 6, pp. 64-79, Nov. 1998 422
[18] E. W. Zegura, K. L. Calvert and S. Bhattacharjee, How to model an internetwork, IEEE INFOCOM'96, vol.2, pp. 594-602, Apr. 1996 426

Quality of Service for the Zone on the Internet

Ashraf Uddin Ahmed[1], S. Sakashita[2], Z. Cheng[3], and S. Saito[4]

[1] Computer Communication and Networking Laboratory,
[2] Department of Computer Systems,
[3] Graduate School of Computer Science and Engineering,
[4] The University of Aizu, Aizuwakamatsu City,
Fukushima 965-8580, Japan
d8042203@u-aizu.ac.jp

Abstract. It is possible to enjoy the quality of service related informations of a single node or a link between two nodes, which is understandable for the technical people only. Yet, there is no zone-based quality of service in the Global Internet which is understandable for both the technical and non-technical people. Zone is used as an autonomous systems (AS) in this document. To fulfill this requirement, this paper approached a new model called Ryuki Model to produce quality of services for the zone on the Internet.

Keywords: Ryuki, Quality of Service (QoS), Autonomous Systems (AS), Traffic Monitoring, Packet Flow, Flow Modeling.

1 Introduction

Nowadays, the Internet has become a strong part of the social media. But, there is a big gap between the technical and non-technical users. First, QoS related information (throughput, delay, etc.) are understandable for the technical people (network engineer) only, but not for non-technical people. In order to alleviate this problem, we have defined three new metrics, which are easily understandable for both the technical and non-technical users. Non-technical people are the people who have a little or no knowledge on QoS related information. Second, in the global Internet, QoS related information is available for a single node or a link between two nodes. Yet, there is no QoS related information for the 'Zone' on the Internet. Zone is used as an autonomous systems in this paper where multi-nodes and multi-links are connected. The Internet applications are coming with new and friendly interfaces, made it possible to enjoy different Internet based network services easily. The network users (human and applications) need to know more about the underlying network to expect a better service or performance of the network. But the isolation of the network layers made it difficult to provide the network related information to the top most application layer where the users and applications are directly involved. Many research efforts have been made on collecting network information and analyze them to produce performance related information. QoS related information is important for the

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 430–439, 2004.

Internet administrator, but may not satisfy the non-technical users. To satisfy the non-technical users, the most important requirement is to make visualize the network information as simple as possible. Our motivation of this work is to answer to this requirement: introduce some new metrics, which will better represent the network and will be visible from the application layer. So far, network management related information (SNMP-MIB [3], IRR-MIB [1] [6] , WHOIS [2]) and their representation have been limited and useful to the technical people. From this information, a network engineer or a technical person can understand the static information such as which device is connected to where and their connectivity status. Drawing a network map [8] was a nice attempt to visualize the connectivity information. A. Ashir et. al. [7] worked to put dynamic information (throughput, congestion) on the map. T. Saito et. al. [9] has succeeded to capture remote congestion monitoring.

However, all these works did not consider any geographical location. In this work, we supplement their work by mapping the network connectivity into the respective geographical locations. We are introducing a new model "Ryuki Model," which takes care of representing the network information in a network weather map. Ryuki is a Japanese word, introduced by S. Saito [5] [4]: the character "Ryu" means flow and "ki" means atmosphere. The reason to choose this word is, because our research also characterizes the atmospheric flow (network flow) coming in or going out from a geographical zone. We focused on the term "geographical zone." To classify the network information on geographical zones, one needs to know the network resources (network devices, their connectivity information) and other metrics on that zone. Our contribution in this work is to map all these information into a specified area. In practice, the proposed Ryuki model has its link with different network planes. It collects network traffic information from the lower planes. The collected information is analyzed to produce QoS related information i.e. delay, throughput etc. But while looking at the geographical zone, first we identify the network devices available on that zone. Then, measure the outgoing flow, incoming flow and also consumption flow (which is generated inside the zone and also dissolves inside the same zone). This information is important for the next plane of the Ryuki Model. We have introduced three new indexes: Ryuki temperature, Ryuki pressure and Ryuki density. These were named in the analogy with the observation item of the atmosphere. The temperature is an index to the amenity of the climate, and Ryuki temperature shows the amenity of the communication services. Ryuki pressure contrasted with the property of anticyclone and cyclone expresses the distributability of the flow. By simulating thing in which buoyancy of the wing and propagation velocity of the sound increase with that air density is high, Ryuki density shows the expected value of the usable bandwidth. Then, in the second plane of the Ryuki model offered the service status of a zone. The results of Estimation plane is then calculated to produce the service status of zone. These different type of services is mapped on the top of the Ryuki Model, stays the Ryuki Map. This map is transparent to the network users. In short, this work introduces zone-based quality of services(QoS).

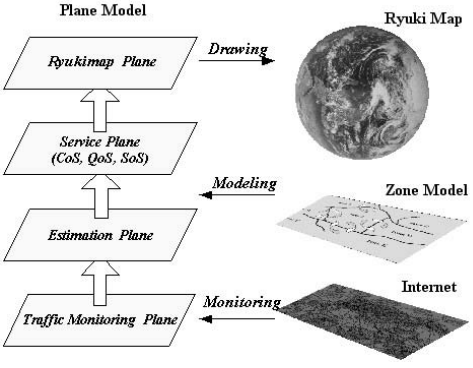


Fig. 1. Ryuki Model Architecture

2 Ryuki Model Architecture

The proposed Ryuki Model has four planes. In Fig. 1, we have shown the hierarchical architecture of Ryuki model. Each plane has its own responsibility to process the network information. The bottom plane collects raw network information from the network devices, SNMP-MIBs and IRR-MIBs and passes the information to the second plane. At the second plane, the traffic is classified into different flows. From a traffic flow, we can easily understand the IP end-points. The IP addresses are mapped into their respective AS addresses. From AS addresses, we can easily get the geographical locations they belong to. Service-related information is derived at the third plane. The Ryuki Map at the top plane contains the network weather related information. In the following subsections, detail descriptions of the different planes are presented.

2.1 Traffic Monitoring Plane

Network traffic is collected in this plane. Available traffic sources we are considering are: the sniffers (e.g. tcpdump), MIBs, network logs (server and application logs). The collected traffic will be classified into different flows. By a flow, we mean a traffic traversing between the same source and destination pair using the same application. Therefore, to sort out the flows, one needs to find out the source-destination pair on an application. Flow based network characteristics can be derived from this flow information.

2.2 Estimation Plane

In this plane, the flow based traffic is analyzed to see the flow based performance index: throughput, delay etc. Therefore we can estimate the network performance of a link. A link here is a virtual connection between the source

and destination points of a flow. These source and destination node points have their IP addresses. Once we know the IP addresses, we can located their corresponding AS addresses by using whois [2]. The flow performances are then mapped as the Inter-AS performance indexes. An AS has its geographical location. This information will give us Zone based performance. In short, we can obtain zone-based traffic performances. We define the performance indexes as (i) Throughput: This is a standard network transmission capability which indicates how much traffic has been transmitted in a unit time.(ii) Delay: Delay is a metric which is measured in time. As the network delay is very small, the unit is presented in milliseconds (msec). In this case, by delay we indicate the delay between borders routers of the zone. (iii) Utilization: Utilization is the metric which tells how much of the resource has been used. By using these standard metrics, we define the following three Ryuki Indexes: Ryuki Temperature, Ryuki Pressure and Ryuki Density.

Ryuki Temperature: Ryuki Temperature is an index which shows the network health status of a zone. Network health status is defined as how much information flow is in the zone at time t. To estimate the network health status of zone we have considered the following items:

$$U_{z,n} \equiv \frac{1}{n}\sum_{r=1}^{n} U_r \equiv \frac{1}{n}\sum_{r=1}^{n}(\frac{1}{l}\sum_{t=1}^{l}\frac{c_t}{max_t}) \tag{1}$$

U_r indicates utilization of a border router; c_t – current throughput of a link; max_t – maximum throughput of a link; l – number of total links in a border router; $U_{z,n}$ indicates utilization of a zone; n – number of total border routers in a zone. z – indicates the zone. Expected delay is measured in time. A zone has many border routers. The end-to-end delay between two border routers are measured in two steps. $Step-1$: we took the bandwidth BWs of a zone from the saturation point of throughput. $Step-2$: we took the delay $d_{i,j}$ from the fifty percent of BWs as $BWs/2$. This delay is called expected delay of a zone. Current delay $dc_{z,t}$ indicates the end-to-end delay between two border routers of a zone. Ryuki Temperature is defined as follows,

$$T_{z,t} \equiv k_q \cdot U_{z,n} \cdot \frac{de_{z,n}}{dc_{z,t}} \equiv k_q \cdot U_{z,n} \cdot \frac{\sum_{i,j=1}^{n} de_{i,j}}{\sum_{i,j=1}^{n} dc_{i,j}} \tag{2}$$

where, $T_{z,t}$ represents the Ryuki Temperature of a Zone at time t; $U_{z,n}$ – utilization of zone; k_q is an arbitrary constant; $de_{z,n}$ – expected delay; $dc_{z,t}$ – current delay. n – total number of border routers.

Equation (2), tells that the temperature is dependent on utilization and current delay, since expected delay of zone are fixed. When the current delay is high, the delay coefficient is then small, therefore, temperature is low and when the current delay is less than expected delay, the delay coefficient is then big which means that the temperature is high.

Ryuki Pressure: Ryuki Pressure is an index to show the traffic smoothness of a zone. How the traffic is flowing in the zone. Is the traffic goes smoothly through the zone. For the estimation of zone pressure we have considered (i) Consumption Flow of Zone which shows that how much flow consume a zone at time t. A zone can consume flow from two sources. Outside sources and inside sources. As we considered that the Ryuki zone is a black box. So it is possible to measure the consumption flow by border routers using the source address (sa) and destination address (da). The definition of consumption flow is as below:

$$F_z^{cf} \equiv \sum_{j=1}^{n} (F_j^{sa} + F_j^{da}) \tag{3}$$

where, F_z^{cf} – consumption flow of a zone [bps]; j – routers belongs to the zone; F_j^{sa} – flow with source address equal routers j, where routers j belongs to the zone; F_j^{da} – flow with destination address equal router j, where routers j is belongs to the zone.(ii) Incomming flow is equal to how much flow receives the zone at time t. F_z^{in} – total incoming flow of a zone; IF_j^{in} – incoming flow at border router j with destination address is not inside Zone. (iii) Outgoing flow equal how much flow sends the zone at time t. (iv) Transit Flow and Flow activity of the zone are defined by the following expressions:

$$TransFlow \equiv \sum_{j=1}^{n} (IF_j^{in} + OF_j^{out}) \tag{4}$$

$$F_z^{act} \equiv \frac{\sum_{j=1}^{n} (F_j^{sa} + F_j^{da})}{\sum_{j=1}^{n} (IF_j^{in} + OF_j^{out})} \tag{5}$$

We have discussed the current delay and expected delay in subsection 2.2. Using the above items we have proposed the Ryuki Pressure in the following equation:

$$P_{z,t} \equiv k_s \cdot F_z^{act} \cdot \frac{dc_{z,t}}{de_{z,n}} \tag{6}$$

where, k_s is an arbitrary constant; $P_{z,t}$ – the Ryuki Pressure of a Zone at time t; F_z^{act} – flow activity of zone; $de_{z,n}$ – expected delay; $dc_{z,t}$ – current delay. we have proposed the equation (6) to check the smoothness of zone. In equation (6), pressure is high or low dependent on flow activity and delay coeficient. Example: If expected delay is greater than current delay, the value of pressure is low which means that the smoothness of zone is good otherwise bad.

Ryuki Density: Ryuki Density is an index to show the expected bandwidth availability of a zone. i.e, how much bandwidth is available in zone. To represent the Ryuki density we took the (i) Maximum Bandwidth: If a zone has three border routers and physical bandwidths are 1.5 Mbps, 2.5 Mbps and 3.5 Mbps respectively then we will call maximum bandwidth maxB is 3.5 Mbps of the zone (ii) Peak Bandwidth: peakB indicates the Peak Bandwidth of the zone which is

logical. If the physical bandwidth is 1.5 Mbps and the link used maximum 1.0 Mbps throughput at time t then we will say, peak bandwidth is 1.0 Mbps. note that, peakB is less than or equal to maxB and (iii) Average Bandwidth: If the delay between two borders is 3 milliseconds. Bandwidth is utilized 10%, 20% and 30% at time t_1, t_2 and t_3 milliseconds. Then we will call average bandwidth utilization is $\frac{10\%+20\%+30\%}{3} = 20\%$ at time 3 milliseconds.

$$D_{z,t} \equiv k_p \cdot \frac{max(peakBW_r | r = 1, n)}{\frac{1}{n}\sum_{r=1}^{n} \frac{1}{r}\sum_{l=1}^{r} avgBW_l} \cdot \frac{de_{z,n}}{dc_{z,t}} \tag{7}$$

where, $D_{z,n}$ shows the Ryuki Density of a zone at time t; k_p is an arbitrary constant; $avgBW_l$ – average bandwidth of links l; $peakBW_(r)$ – the peak bandwidth of border routers r [Mbps]; $avgBW_r$ – average bandwidth of routers r; Max – maximum bandwidth of border router [Mbps]; $de_{z,n}$ – expected delay; $dc_{z,t}$ – indicates the current delay; n – total number of border routers of a zone. In equation (7), the density of zone depends on current delay, since bandwidth availability and expected delay of zone are fixed. When the current delay is greater than expected delay then density is low and when the current delay is less than expected delay density is high means zone has available bandwidth.

2.3 Service Plane

From the estimation plane, we obtained zone based Ryuki indexes (Temperature, Pressure and Density). Based on the Ryuki indexes, we find out the service expectation in this plane. The proposed services are:

- Quality of Service: Expected quality at a certain time from a certain Zone
- Cost of Service: Indicates the cost to enjoy any network services

Quality of Service (QoS) of the Zone: QoS is still vague term in any environment of the earth. Quality can encompass many properties in networking, but people generally use quality to describe the process of delivering data in a reliable manner or even some how in a manner better than normal [10]. Yet, there are only QoS parameters (like: throughput, delay, jitter, bandwidth etc.) which are understable for technical users. In this research, we have proposed zone-based QoS definition which is understandable for both the technical and non-technical users. To represents the QoS definition, we have considered temperature and density as follows:

$$QoS_z \equiv F(T_{z,t}, D_{z,t}) \equiv T_{z,t} * D_{z,t} \tag{8}$$

Cost of Service (CoS) of the Zone: CoS is quite important for any environment. Still, there is no zone-based service cost offers for the Internet users in the world, so far. In this document lower cost and better service is estimated for ryuki information service users. When determining CoS_z, division of pressure $P_{z,t}$ by density $D_{z,t}$ of zone has been used as belows:

$$CoS_z \equiv F(P_{z,t}, D_{z,t}) \equiv \frac{P_{z,t}}{D_{z,t}} \tag{9}$$

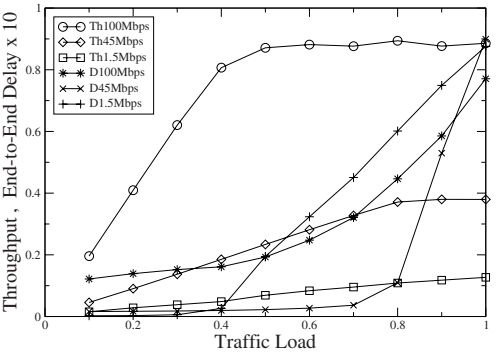


Fig. 2. Traffic Load Vs. Throughput, Delay

2.4 Ryuki Map Plane

This is the top plane of the Ryuki Model. Ryuki map is drawn in this plane. As discussed earlier that internet become social impact and for the non-technical user visual map is needed. To present all Ryuki information in the Ryuki map, we have considered:

- Ryuki Status
- Quality of information communication service and
- Cost of information communication service.

Ryuki configuration information, referred to as Ryuki map in the following, comprises of information about zone objects. The map information will cover the interconnection between the various servers of zones. This map will show the Ryuki services and functions. The Ryuki map covered both geometric and geographical information. The geometric information describes the Ryuki properties to the end user. The geographical information describes the geographical location of zones.

3 Simulation: Results and Analysis

The proposed Ryuki Model is under simulation by using a popular network simulator. The network traffic has been collected and analyzed to produce QoS related information and mapped to display in the Ryuki Map.

3.1 Results

In this section, we have explained the obtained results from the simulation. Based on the results we have analyzed the service parameters (Quality of Service and Cost of Service) in the next section. We have obtained the throughput (Th) and delays (D) of a zone from the simulation results. In Fig.2, when the traffic load

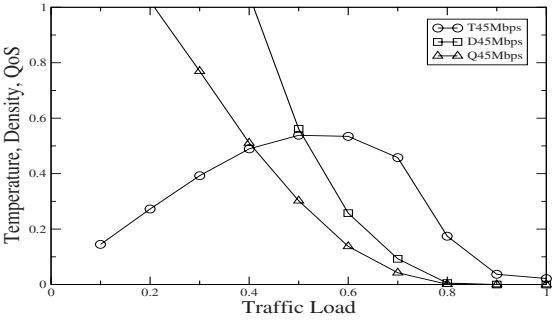


Fig. 3. Traffic Load Vs. Temperature, Density and QoS($k_q = k_s = k_p = 1$)

are sequentially increasing dealys are also increasing this is because the zone is congested with traffic injecting by application. In $100Base-T$ (100Mbps) and $T3$ (45Mbps) the maximum throughput is when the traffic load are 0.5 and 0.9 respectively where the delay of $100Base-T$ link is smallar than the delay of $T3$. So, the zone which connected with $100Base-T$ is good for packet transmission for the users.

3.2 Analysis

Ryuki service parameters refer the service status of the zone. Service parameters are useful for Internet users, subnet managers etc.

Quality of Service (QoS): In Fig.3, we show the results of the zone quality by comparing with the different value of temperature, density and percentage of traffic load. In real weather forecast, when the density of atmospheric air is increased then temperature is also increased, and when the density of atmospheric air is decreased then temperature is also decreased. Ryuki temperature and density have changed due to concentration of information flow in a zone. Therefore, If a zone has low temperature and high density, we will call the quality of zone is good. Again, if a zone has high temperature and low density then we will call the quality of zone is bad. As a whole, users demand is good quality zone. Comperatively, high quality is when the zone is less busy.

Cost of Service (CoS): The results in Fig.4 is shown the achieved cost of service of the zone. In general, users desire is lower cost. CoS characteristics provide the quality / congestion in the zone. A relatively better cost is when the zone is less busy. In Fig.4, we show the results of zone cost by comparing with the different value of pressure, density and percentage of traffic load. If a zone has high pressure and high density, we will call the cost of zone is high. Again, if a zone has low pressure with low density then we will call the cost of zone is low.

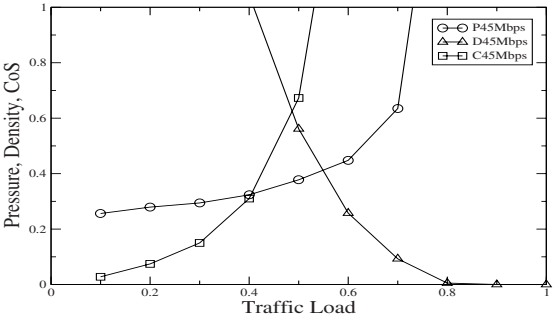


Fig. 4. Traffic Load Vs. Pressure, Density and CoS ($k_q = k_s = k_p = 1$)

General user like less congested and bandwidth availability of zone because of low cost, on the other hand, network manager like congested zone so that they can charge more to the user.

3.3 Evaluation of Service Parameters:

The following table shows the overall performances of the zone from the customers (General user and the Network Manager) point of view. Here, QoS^+ indicates that the users need more quality and QoS^- indicates that the users need less quality i.e., more flow needed. CoS^+ indicates that zone need more pressure. 'U' indicates general user and 'M' indicates Network Manager. In Table:1, we have shown the different pairs of service status. Example: If both the quality and cost of the zone is low what will be the service for the general user and network manager? – the second column and second row shows the service for user is normal but network manager needs high quality. For the low cost and high quality service status shows that the service is excellent for user but network manager expect low quality.

Table 1. Service status of the Zone

Quality: → **Cost: ↓**	Low	Average	High
Low	U: normal M: QoS^+	U: Good M: CoS^+	U: excellent M: QoS^-
Average	U: Poor M: QoS^+	U: Normal M: CoS^+	U: Good M: Normal
High	U: Very Poor M: QoS^+	U: Poor M: QoS^+	U: Normal M: Normal

4 Conclusion

In this work, we coined a new field of representing Internet traffic related information in a network weather map where users of a broad range can benefit from it. QoS related information is understandable to the technical people. We have focused on the non-technical users demand and came up with a new idea of Ryuki Model, which displays the network information in a network weather map by which people can have the basic understanding the present status of a network and also will be able to predict a near future performance. We have defined two types of services (i.e. QoS and CoS) which are visualize in the Ryuki Map. In this research we have proposed the zone based network performances. The important point is that using geographical zone based Ryuki model people will be able to compare the zones according to the strength of Internet activities. An ISP may charge more at a time when the Ryuki service status is higher than other time. A web administrator may duplicate a server at a mirror site where traffic is less congested. A user can choose a time depending on the application QoS requirement. These are important application examples that can be easily implemented from the proposed base technology, the Ryuki model. Some future works are summarized as (i) Multi-zone based information flow measurements (ii) Flow measurements using real data (iii) Develop a Ryuki Map by Java and/or any suitable programming language to visualize Ryuki Indexes and Services.

References

[1] http://www.irr.net 431
[2] http://www.whois.com 431, 433
[3] SNMPv3, RFC2574, April 1999 431
[4] Ashraf Uddin Ahmed, *Ryuki Model for the Estimation of Information Flow*, Masters Thesis September, 2002, Graduate Dept. of Computer Systems, The University of Aizu, Japan. 431
[5] S. Saito, *A study of the basic technologies for the Observation of Network Information Flow in the Gigabit-Class Internet*, proc. of TAO reserach Conference 2002, pp.299-304, May 28-29, 2002, Tokyo. 431
[6] K. Tomomoto, *IRR Application for the Observation of Internet Information Flow*, Master Thesis of the Univ. of Aizu, 2001. 431
[7] Ahmed Ashir, Glenn Mansfield and Norio Shiratori, *Estimation of Network Characteristics and Its Use in Improving Performance of Network Applications*, IEICE TRANS.INF. and SYS., Vol.E82-D No.4 April 1999. 431
[8] Glenn Mansfield, K. Jayanthi, A. Ashir and N. Shiratori, *Network Maps: Synthetics and Applications*,International Conference, APSIT99, Mongolia, August 1999. 431
[9] T. Saito, G. Mansfield, N. Shiratori, *Network Congestion Monitoring and Detection using the IMI infrastructure*, Proceeding of the 1999 International Conference on Parallel Processing, pp.442-469, 1999. 431
[10] Paul Ferguson, Geoff Huston; *Quality of Service: delivering QoS on the internet and in corporate networks* ISBN 0-471-24358-2. 435

Efficient Algorithm for Reducing Delay Variation on Bounded Multicast Trees*

Moonseong Kim[1], Young-Cheol Bang[2], and Hyunseung Choo[1]

[1] School of Information and Communication Engineering, Sungkyunkwan University
440-746, Suwon, Korea
+82-31-290-7145
{moonseong,choo}@ece.skku.ac.kr
[2] Department of Computer Engineering, Korea Polytechnic University
429-793, Gyeonggi-Do, Korea
+82-31-496-8292
ybang@kpu.ac.kr

Abstract. With the proliferation of multimedia group applications, the construction of multicast trees satisfying QoS requirements is becoming a problem of prime importance. In this paper, we study the delay- and delay variation-bounded multicast tree (DVBMT) problem which is NP-complete. The problem is to construct a spanning tree for destination node, which has the minimized multicast delay variation, and the delay on the path from the source to each destination is bounded. A solution to this problem is required to provide decent real-time communication services such as on-line games, shopping, and teleconferencing. Performance comparison shows that the proposed scheme outperforms DDVCA which is known to be effective so far in any network topology. The enhancement is up to about 3.6%∼11.1% in terms of normalized surcharge for DDVCA. The time complexity of our algorithm is $O(mn^2)$.

1 Introduction

New communication services involving multicast communications and real time multimedia applications are becoming prevalent. In multicast communications, messages are sent to multiple destinations that belong to the same multicast group. These group applications demand a certain amount of reserved bandwidth to satisfy their quality of service (QoS) requirements. An efficient solution for multicast communications includes the construction of a multicast tree that is rooted at a source and spanning to all the group destinations. For higher bandwidth applications, the multicast tree should be designed to minimize the network resources in terms of the tree cost, where the tree cost is defined by summing costs of all links in the tree. The minimum cost multicast tree is known as the Steiner tree [3], and finding such tree is a famous NP-complete problem [4]. Many heuristics for this problem have been proposed in the literature [2, 5, 6, 7, 8, 14].

* This paper was supported in part by Brain Korea 21 and University ITRC project. Dr. H. Choo is the corresponding author.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 440–450, 2004.

While total tree cost as a measurement of bandwidth efficiency is certainly an important metric, it is not sufficient to characterize the quality of the tree as perceived by interactive multimedia and real-time applications. In real-time communications, messages must be transmitted to their destination nodes within a limited amount of time, otherwise the messages will be nullified. Computer networks have to guarantee an upper bound on the end-to-end delay from the source to each destination. This is known as the multicast end-to-end delay problem [10, 11].

During a teleconference, it is important that the current speaker must be heard by all participants at the same time, but otherwise the communication may lose the feeling of an interactive face-to-face discussion. Another similar dispute can be easily found in on-line video gaming. These are all related to the multicast delay variation problem [12]. In this paper, we improves the Delay and delay variation constraint algorithm (DDVCA) that is known as the best algorithm [13]. The shortcoming of [13] is that the selection of a core node over several candidates (possible core nodes) is overlooked which means a core node is randomly selected among candidates. Meanwhile we investigate candidate nodes without increasing the time complexity for the better selection. Computer simulations show that our algorithm is efficient in terms of the multicast delay variation. The enhancement is up to about 3.6%∼11.1% in terms of normalized surcharge for DDVCA. The time complexity of our algorithm is $O(mn^2)$.

The rest of the paper is organized as follows. In Section 2, we state the network model for multicasting, and section 3 presents details of the proposed algorithm. Then, in section 4, we evaluate the proposed algorithm by the simulation model. Section 5 concludes this paper.

2 Preliminaries

2.1 Network Model for Multicasting

We consider a computer network represented by a directed graph $G = (V, E)$ with n nodes and l links or arcs, where V is a set of nodes and E is a set of links (arcs), respectively. Each link $(i, j) \in E$ is associated with delay $d_{(i,j)}$. The delay of a link is the sum of the perceived queueing delay, transmission delay, and propagation delay over that link. We assume that the available delay on each arc is asymmetric in general.

Given a network G, we define a path as sequence of nodes $u,\ i,\ j,\ \ldots,\ k,\ v$, such that $(u, i),\ (i, j),\ \ldots, (k, v)$, belongs to E. Let $P(u, v) = \{(u, i),\ (i, j),\ \ldots,\ (k, v)\}$ denote the path from node u to node v. If all nodes of the path are distinct, then we say that it is a simple path. We define the length of the path $P(u, v)$, denoted by $n(P(u, v))$, as a number of links in $P(u, v)$. Let $\preceq$ be a binary relation on $P(u, v)$ defined by $(a, b) \preceq (c, d) \leftrightarrow n(P(u, b)) \leq n(P(u, d))$, $^{\forall}(a, b)$, $(c, d) \in P(u, v)$. $(P(u, v), \preceq)$ is a totally ordered set. For a given source node $s \in V$ and a destination node $d \in V$, $(2^{s \Rightarrow d}, \infty)$ is the set of all possible paths from s to d.

$$(2^{s \Rightarrow d}, \infty) = \{\ P_k(s, d) \mid \text{all possible paths from } s \text{ to } d,\ ^{\forall}s,\ d \in V,\ ^{\forall}k \in \Lambda\ \}$$

where Λ is a index set. The delay of arbitrary path P_k are assumed to be injective function from $(2^{s \Rightarrow d}, \infty)$ to positive real number $\mathcal{R}^+$. Since $(P_k, \preceq)$ is a totally ordered set, if there exists a bijective function f_k then P_k is isomorphic to $\mathcal{N}_{n(P_k)}$.

$$P_k = \{(u,i),(i,j),\ldots,(m,v)\} \xrightarrow{f_k} \mathcal{N}_{n(P_k)} = \{1,2,\ldots,n(P_k)\}$$

We define,

$$\text{function of delay along the path } \phi_D(P_k) = \sum_{r=1}^{n(P_k)} d_{f_k^{-1}(r)} \ , \ {}^{\forall}P_k \in (2^{s \Rightarrow d}, \infty) \ .$$

$(2^{s \Rightarrow d}, supD)$ is the set of paths from s to d for which the end-to-end delay is bounded by $supD$. Therefore $(2^{s \Rightarrow d}, supD) \subseteq (2^{s \Rightarrow d}, \infty)$.

For multicast communications, messages need to be delivered to all receivers in the set $M \subseteq V \setminus \{s\}$ which is called multicast group, where $|M| = m$. The path traversed by messages from the source s to a multicast receiver, m_i, is given by $P(s, m_i)$. Thus multicast routing tree can be defined as $T(s, M) = \bigcup_{m_i \in M} P(s, m_i)$, and messages are sent from s to destination of M using $T(s, M)$.

2.2 The DVBMT Problem

In the following we now introduce two important qualities of service metrics in multicast communications [12]. The multicast end-to-end delay constraint, $supD$, represents an upper bound on the acceptable end-to-end delay along any path from the source to a destination node. This metric reflects the fact that the information carried by the multicast messages becomes stale $supD$ time units after its transmission at the source.

The multicast delay variation, δ, is the maximum difference between the end-to-end delays along the paths from the source to any two destination nodes.

$$\delta \ = \ max\{ \ |\phi_D(P(s,m_i)) - \phi_D(P(s,m_j))|, \ {}^{\forall}m_i, m_j \in M, \ i \neq j \ \}$$

The issue first defined and discussed in [12] is of minimizing multicast delay variation under multicast end-to-end delay constraint. The authors referred to this problem as Delay- and Delay Variation-Bounded Multicast Tree (DVBMT) problem. The DVBMT problem is to find the tree that satisfies

$$min\{ \ \delta_\alpha \ | \ {}^{\forall}m_i \in M, \ {}^{\forall}P(s,m_i) \in (2^{s \Rightarrow m_i}, supD), \ {}^{\forall}P(s,m_i) \subseteq T_\alpha, \ {}^{\forall}\alpha \in \Lambda \ \}$$

where T_α denotes any multicast tree spanning $M \cup \{s\}$, and is known to be NP-complete [12].

2.3 Previous Works

There are two well known approaches to construct multicast tree for the DVBMT problem. One is DVMA (Delay Variation Multicast Algorithm) [12] and the other DDVCA. The algorithm DDVCA [13] proposed by Pi-Rong Sheu and Shan-Tai Chen is based on the Core Baded Tree (CBT) [9].

The DDVCA first calculates the minimum delay for each (m_i, v) pair with $m_i \in M$ and $v \in V$. Next for each node DDVCA computes the associated delay variation, δ_α, between the node and each receiver. Then it selects the node with the minimum delay variation, δ, as the core node. Finally, each recever is connected to this core node through the minimum delay path. The source node is also connected to the core node through the minimum delay path. If the core node in quest does not conform to the requirement of $supD$, the DDVCA will go on to pick the node with the next minimum delay variation as the next possible core node and the same process is repeated until we find a core node which fulfills the requirement of $supD$.

It has been shown that DDVCA outperforms DVMA in terms of the delay variation of the constructed tree. Moreover, the time complexities of algorithms are $O(mn^2)$ and $O(klmn^4)$ for DDVCA and DVMA, respectively, where n is the number of receivers.

3 The Proposed Algorithm

In this section, we present our proposed novel algorithm to construct multicast trees that is superior to DDVCA. In order to define a multicast tree, the basic idea of the proposed algorithm is based on CBT [9]. The method used in CBT for the establishment of a multicast tree is first to choose some core routers which compose the backbone. We also select a core router addressed as a core node.

3.1 Description of the Proposed Algorithm

The goal of this paper is to propose an algorithm which produces multicast trees with low multicast delay variation. In this subsection, we describe our proposed algorithm with time complexity analysis. We show an example with explanation of the algorithm in the following subsection.

The proposed algorithm consists of a core node selection part and the multicast tree construction part. Hence we take an interest in a core node selection. When candidate of core node is several nodes, the DDVCA randomly choose a core node among candidates but the proposed algorithm presents lucid solution. The proposed algorithm is described in detail as follows.

Proposed Algorithm $(G(V,E),\ M,\ s,\ supD)$
Input: A directed graph $G(V,E)$, M is the multicast group with $m = |M|$, a source node s, a end-to-end delay bound $supD$.
Output: The multicast tree T such that $\phi_D(P(s,m_i)) \leq supD$, $^{\forall}P(s,m_i) \subseteq T$, $^{\forall}m_i \in M$, and has a small multicast delay variation.

01. **Begin**
02. $pass = Null;\ diff_{min} = \infty;\ candidate = \emptyset;$
    $c = Null;\ compare = Null;\ T = \emptyset$
/* *candidate* : the candidates of core node */
/* *compare* : the max difference between core nodes and visited destinations */
03. **For** $^{\forall}m_k \in M \cup \{s\}$ **Do**
04. $Dij(m_k, v_i)$ = Calculate the minimum delay between m_k and v_i, $^{\forall}v_i \in V$
05. **For** $^{\forall}v_i \in V$ **Do**
06. $max_i = max\{Dij(v_k, v_i) \mid {}^{\forall}v_k \in M\}$
07. $min_i = min\{Dij(v_k, v_i) \mid {}^{\forall}v_k \in M\}$
08. $diff_i = max_i - min_i$
09. **For** $^{\forall}l \in$ {the minimum delay path from s to v_i} **Do**
/* l : the nodes in minimum delay path from s to v_i */
10. **If** $l = m_k$, $^{\forall}m_k \in M$
11. **then** $pass(s, v_i, m_k) = Dij(s, m_k)$
12. **else** $pass(s, v_i, m_k) = 0$
13. **If** $diff_i < diff_{min}$ and $Dij(s, v_i) + max_i \leq supD$
14. **then** $diff_{min} = diff_i\ ;\ c = i$
15. **For** $^{\forall}v_i \in V$ **Do**
16. **If** $diff_i = diff_c$ and $Dij(s, v_i) + max_i \leq supD$
17. **then** $candidate = candidate \cup v_i$
18. **If** $candidate = \emptyset$ **then** print " Tree construct fail ! "
19. **For** $^{\forall}c_i \in candidate$ **Do**
20. **If** $pass(s, c_i, m_k) = 0$, for every $m_k \in M$
21. **then** $compare_i = 0$
22. **else** $compare_i = Dij(s, c_i) - min\{pass(s, c_i, m_j)|$ positive and$^{\forall}m_j \in M\}$
23. $c = min\{\ i \mid compare_i\ \}$
24. **For** $^{\forall}m_k \in M$ **Do**
25. $T = T \cup \{\ l \mid l \in$ the minimum delay path from m_k to $v_c\}$
26. $T = T \cup \{\ l \mid l \in$ the minimum delay path from s to $v_c\}$
27. Return T
28. **End Algorithm.**

In selecting such a core node, we use the minimum delay path algorithm. In steps 3-4, the proposed algorithm calculates the minimum delay from each destination node and source node to each other node in the network. The results are presented in Table 1. In steps 5-23, the proposed algorithm looks for a core node. During the process, for each node, step 8 calculates the associated delay variation between the node and each destination node. Steps 9-12 check whether any destination node is visited in the path from source node to each other node. If any destination node is visited, then the proposed algorithm records in '*pass*' data structure. Steps 13-14 and 15-17 conform *supD* and select nodes with the minimum delay variation as the candidates of core node. Next, the proposed algorithm chooses the core node with $min\{\ \phi_D(P(s, c_i)) - min\{\ pass(s, c_i, m_j)\ \}\ \}$ in steps 19-23.

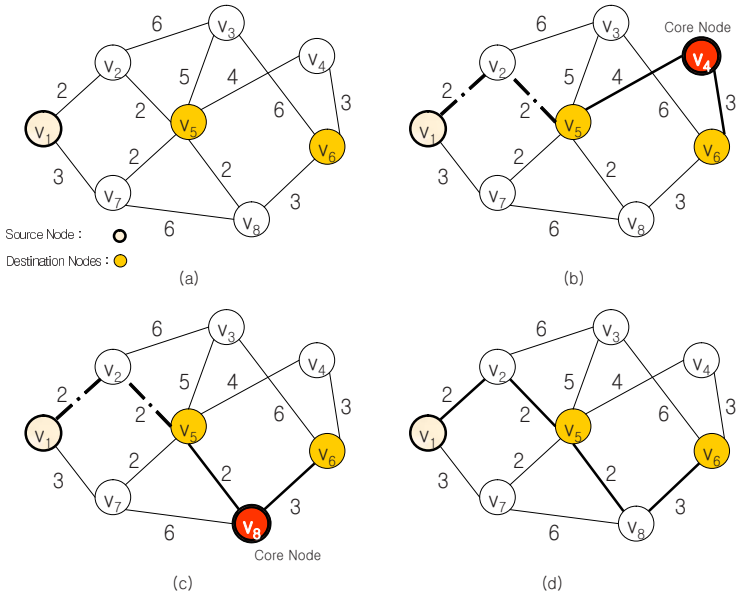


Fig. 1. (a) A Given network $G(V, E)$ and link delays are shown to each link, (b) DDVCA and $\delta_{DDVCA} = 7$, (c) Proposed Algorithm and $\delta_{Proposed} = 5$, (d) Optimal tree and $\delta_{Optimum} = 5$

In construction of a multicast tree, each destination node is connected to this core node through the minimum delay path in steps 24-25. The source node is also connected to the core node through the minimum delay path in step 26. Finally, step 27 produces the resulted multicast tree T.

The time complexity of the proposed algorithm is evaluated as follows. Steps 3-4 can be completed in $O(mn^2)$ time using Dijkstra's Algorithm [1], where $n = |V|$. Steps 5-14 take at most $O(n^2)$ time, since step 10 can be computed in $O(1)$ using appropriate data structure. Steps 15-17 can be completed in $O(n)$. Steps 19-22 take $O(m^2|candidate|)$ time which is at most $O(m^2n)$. Because step 25 has a time $O(n)$, the overall time of steps 24-25 is $O(mn)$. Likewise, the time of step 26 is $O(n)$. As a result, the total time complexity of the proposed algorithm is $O(mn^2)$, which matches the complexity of the DDVCA. As mentioned earlier, the time complexity of the DDVCA [13] is $O(mn^2)$. The time complexity of our algorithm is the same as that of the DDVCA.

3.2 A Case Study

In the following, we illustrate the operational mechanism of proposed algorithm with examples. Fig. 1 (a) shows a given network topology with link delays specified on each link. Suppose that the multicast end-to-end delay constraint *supD*

Table 1. The method by which proposed algorithm selects a core node

		v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
source	v_1	0	2	8	**8**	4	9	3	**6**
pass	v_5	0	0	0	**4**	4	4	0	**4**
	v_6	0	0	0	**0**	0	9	0	**0**
destination	v_5	4	2	5	4	0	5	2	2
	v_6	9	7	6	3	5	0	7	3
max_i		9	7	6	4	5	5	7	3
min_i		4	2	5	3	0	0	2	2
$diff_i$		5	5	**1**	**1**	5	5	5	**1**

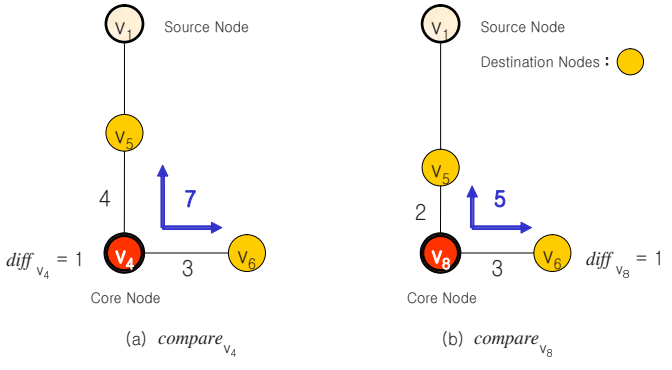


Fig. 2. The structure of $compare_{v_4}$ and $compare_{v_8}$

is 11. Fig. 1 (b) represents the ultimate multicast tree obtained by the DDVCA. Fig. 1 (c) shows the path constructed by the proposed algorithm.

From Table 1, we know that nodes with the smallest multicast delay variation are v_3, v_4, and v_8. However, since we must consider the delay bound $supD$, the node v_3 is ignored. The DDVCA randomly selects the node v_4, but the proposed algorithm selects the node v_8 as a core node. Because the proposed algorithm calculates the minimum among $compare_{v_4} = 8-4 = 4$ and $compare_{v_8} = 6-4 = 2$ in steps 19-23, we take the node v_8 as core node.

Finally, the DDVCA's multicast delay variation is 7, but the proposed algorithm's multicast delay variation is 5.

Fig. 2 shows that proposed algorithm chooses v_8 in Fig. 2 (b) in case of the same $diff_i$ value when it selects a core node.

4 Performance Evaluation

We compare our proposed algorithm with the DDVCA in terms of multicast delay variation. We describe the generation of random network topologies for the evaluation and the simulation results based on the network topology generated.

4.1 Random Network Topology for the Simulation

The details of the generation for random network topologies are as follows. The method uses parameters n - the number of nodes in networks, and P_e - the probability of edge existence between any node pair [15]. Let us remark that if a random graph models a random network then this graph should be connected. Hence, the graph should contain at least a spanning tree. So, firstly a random spanning tree is generated. As we know, we consider cases for $n \geq 3$. A tree with 3 nodes is unique, and thus we use this as an initial tree. And we expand to a spanning tree with n nodes. After adjusting the probability P_e, we generate other non-tree edges at random for the graph based network topology. Let us calculate the adjusted probability P_e^a. By $Prob\{event\}$ denote a probability of the event. Suppose e is a possible edge between a couple of nodes, then we have

$$
\begin{aligned}
P_e &= Prob\{\ e \in \textit{spanning tree}\ \} + Prob\{\ e \notin \textit{spanning tree}\ \} \cdot P_e^a \\
P_e &= \frac{n-1}{n(n-1)/2} + (1 - \frac{n-1}{n(n-1)/2}) \cdot P_e^a \\
\therefore\ P_e^a &= \frac{nP_e - 2}{n-2}\ .
\end{aligned}
$$

Let us describe a pseudo code for random network topologies. Here A is an incident matrix, r is a simple variable, and $random()$ is a function producing uniformly distributed random values between 0 and 1.

Graph Generation Algorithm
Begin
$A_{1,2} = A_{2,1} = A_{2,3} = A_{3,2} = 1$
For $i = 4$ to n **Do**
  $r = (i-1) \times random() + 1$
  $A_{r,i} = A_{i,r} = 1$
For $i = 1$ to $(n-1)$ **Do**
  For $j = (i+1)$ to n **Do**
    If $P_e > random()$ **Then** $A_{i,j} = A_{j,i} = 1$
End Algorithm.

4.2 Simulation Results

We now describe some numerical results with which we compare the performance of the proposed scheme. The proposed algorithm is implemented in C++. We randomly selected a source node. We generate 10 different networks for each size

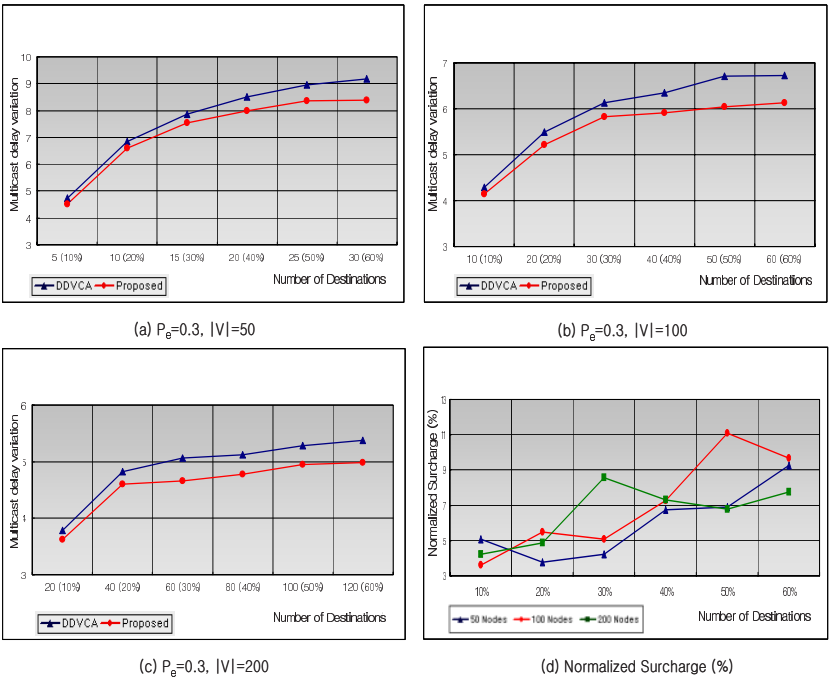


Fig. 3. The multicast delay variations of the three different networks and Normalized Surcharges versus number of nodes in networks

of given 50, 100, and 200. The destination nodes are picked uniformly from the set of nodes in the network topology (excluding the nodes already selected for the destination). Moreover, the destination nodes in the multicast group will occupy 10, 20, 30, 40, 50, and 60% of the overall nodes on the network, respectively. We randomly choose $supD$. We simulate 1000 times ($10 \times 100 = 1000$) for each $|V|$ and $P_e = 0.3$.

For the performance comparison, we implement the DDVCA in the same simulation environment. We use the normalized surcharge, introduced in [10], of the algorithm with respect to our method defined as follows:

$$\bar{\delta} = \frac{\delta_{DDVCA} - \delta_{Proposed}}{\delta_{Proposed}}$$

In our plotting, we express this as a percentage, *i.e.*, $\bar{\delta}$ is multiplied by 100. Fig. 3 (a), (b), and (c) show the simulation results of multicast delay variations. As indicated in Fig. 3 (d), it is easily noticed that the proposed algorithm is always better than the DDVCA. The enhancement is up to about 3.6%∼11.1% in terms of normalized surcharge for the DDVCA. Also, an interesting result is that the normalized surcharge $\bar{\delta}$ increases as the number of destinations increases for each $|V|$.

5 Conclusion

In this paper, we consider the transmission of a message that guarantees certain bounds on the end-to-end delays from a source to a set of destinations as well as on the multicast delay variations among these delays over a computer network. There are two well known approaches for constructing a multicast tree with the DVBMT problem, which is known to be NP-complete. The one is the DVMA [12]. Although it provides smart performance in terms of the multicast delay variation, its time complexity is as high as $O(klmn^4)$. As we all know, a high time complexity dose not fit in large scale high speed networks. The other is the DDVCA [13]. It has been shown that the DDVCA outperforms the DVMA slightly in terms of the multicast delay variation for the constructed tree. Moreover, the time complexity of the DDVCA is $O(mn^2)$.

In the meantime, the time complexity of the proposed algorithm is $O(mn^2)$, which is the same as that of the DDVCA. Furthermore, the comprehensive computer simulation results show that the proposed scheme obtains the better minimum multicast delay variation than the DDVCA.

References

[1] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, pp. 269-271, 1959. 445

[2] E. N. Gilbert and H. O. Pollak, "Steiner minimal tree," SIAM J. Appl. Math., vol. 16, 1968. 440

[3] S. L. Hakimi, "Steiner's problem in graphs and its implication," Networks, vol. 1, pp. 113-133, 1971. 440

[4] M. R. Garey, R. L. Graham, and D. S. Johnson, "The complexity of computing steiner minimal trees," SIAM J. Appl. Math., vol. 32, no. 4, pp. 835-859, June 1977. 440

[5] H. Takahashi and A. Matsuyame, "An approximate solution for the steiner problem in graphs," Mathematica Japonica, vol. 24, no. 6, pp. 573-577, 1980. 440

[6] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," Acta Informatica, vol. 15, pp. 141-145, 1981. 440

[7] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," IEEE Trans. Commun., vol. COMM-31, no. 3, pp. 343-351, March 1983. 440

[8] B. W. Waxman, "Routing of multipoint connections," IEEE JSAC, vol. 6, no. 9, pp. 1617-1622, December 1988 440

[9] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT) : An architecture for scalable inter-domain multicast routing," Computer Commun. Rev., vol. 23, no. 4, pp. 85-95, 1993. 443

[10] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," IEEE/ACM Trans. Networking, vol. 1, no. 3, pp. 286-292, June 1993. 441, 448

[11] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for near-optimum delay-constrained multicasting," Proc. IEEE INFOCOM'95, pp. 377-385, March 1995. 441

[12] G. N. Rouskas and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," IEEE JSAC, vol. 15, no. 3, pp. 346-356, April 1997. 441, 442, 443, 449

[13] P.-R. Sheu and S.-T. Chen, "A fast and efficient heuristic algorithm for the delay- and delay variation bound multicast tree problem," Information Networking, Proc. ICOIN-15, pp. 611-618, January 2001. 441, 443, 445, 449

[14] Y.-C. Bang and H. Choo, "On multicasting with minimum costs for the Internet topology," Springer-Verlag Lecture Notes in Computer Science, vol. 2400, pp.736-744, August 2002. 440

[15] A. S. Rodionov and H. Choo, "On generating random network structures: Trees," Springer-Verlag Lecture Notes in Computer Science, vol. 2658, pp. 879-887, June 2003. 447

Virtual Routing and Management Algorithm for QoS and Security in Internet*

Ilyoung Chong[1], Seong Ho Jeong[1], and Hyun Kook Kahng[2]

[1] Dept. of Info. and Comms. Eng., Hankuk Univ. of FS
Seoul , Korea
{iychong,shjeong}@hufs.ac.kr
[2] Dept. of Electronics Information Eng., Korea Univ.
Seoul, Korea
kahng@korea.ac.kr

Abstract. The motivation for simultaneous providing QoS and security in the network will accelerate an introduction of new services, and is able to provide safe, secure access to the network resources, while allowing the resource is securely managed by end users needing with required QoS. This paper describes one approach to provide quality of service (QoS) guarantees in a network using secure networking mechanism. For provisioning of secured QoS in internet, additional routing and management mechanism are required through the logical architecture of overlay networking. It also suggests a novel algorithm to provide secure overlay networking, routing and forwarding mechanism, and admission and resource management algorithm to compute resource consumption amount and to evaluate its acceptability in the secured overlay network providing its QoS level is based on the discrete-time fluid flow process. And its results is applied to manage the virtual resource for the required security and QoS level and to make decision of a new flow admission in overlay network.

1 Introduction

In information networking, we are experiencing a significant paradigm shift resulting in a new information technologies and architectures. The motivatin behind this shift is a elusive goal of secure and high quality assured service provision, networking and management resulting from new customer and application requirement. Research works in this area has mainly focused on the QoS guarantee in networking and application, but currently many kinds of applications are necessary to have special features in secure delivery and secured networking. So, next generation networks has to be capable of supporting multitude of service with user demanded quality assurance and secure networking efficiently. And the networking features in the next generation network will be required to provide very versatile and flexible usability.

* This work was supported by grant No. R01-2003-000-10562-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 451–461, 2004.

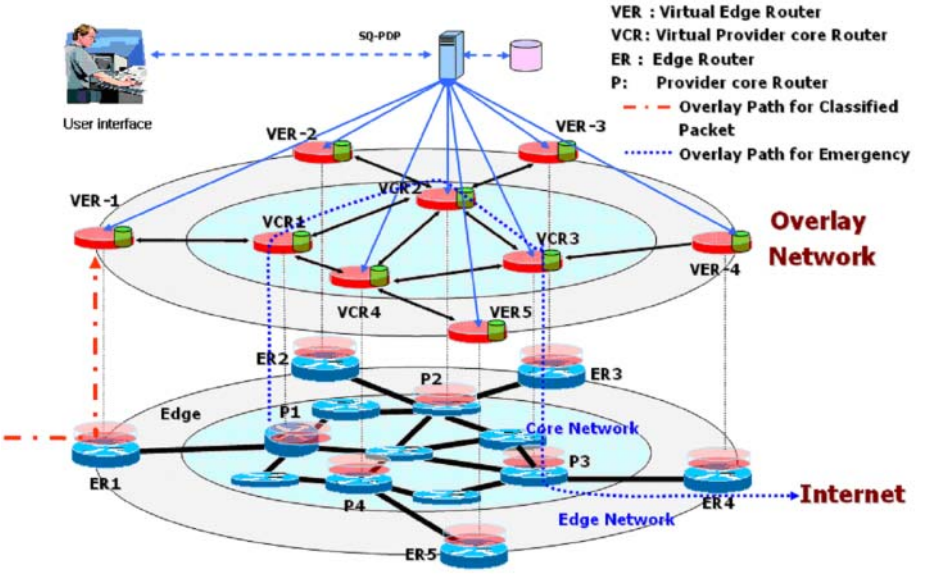


Fig. 1. Policy based Overlay Networking Architecture Providing QoS and Security

The goal of secure and QoS routing is to protect against two types of threat to QoS provision, admission and policing. Balancing performance, flexibility, and security considerations suggests that we make common operation (e.g., those used to classify packets) cheap, and make less common operations more expensive if this contributes to reducing the cost of common operations. An example of this approach is to provide heavyweight authentication mechanism at the level of aggregates of packets such as channels or flows so that these checks need not be done on individual packets. This suggests that a architecture where authentication and other resource management decisions are "front-loaded" to reduce the cost of subsequent decisions. We view the scheme as one where expensive static checks are traded for cheaper dynamic checks. Thus, this paper suggests the overlay networking architecture to provide QoS and security procedures as shown in Fig. 1. Policy server and agent mechanism with modified protocols is applied to provide secure and QoS routing mechanisms in internet. The virtual routing is applied to the model based on overlay configuration, which incoming packets are classified at edge level and the packets for secured QoS service are forwarded to policy-based overlay network. Thus it provides secured routing and QoS managed functions through admission control and other control mechanisms.

On the basic functional architecture, a advanced routing protocol is necessary to adopt QoS and security features in internet efficiently, and the paper describes routing algorithm to provide security and QoS capability on overlay network based on policy-based control in section 2. As mentioned before ad-

mission control provides important role to guarantee required QoS level in the overlay network, and section 3 suggests the novel resource management algorithm to compute required bandwidth in the overlay network. And section 4 indicates a part of numerical results on admission control based on the esource computation algorithm proposed.

2 Virtual Routing Mechanism and Management for QoS and Security in Overlay Network

2.1 Virtual Routing Mechanisms for Secure and QoS

The ER (Edge Router) with virtual networking function (VNF) will be one of the two kinds of multiple routing functions, which is performed through multiple routing tables or routing processors. The VER (Virtual Edge Router) is the one of multiple routing functions in the ER, and its routing will be performed by routing table or the allocated processor. Incoming packets are classified into few levels according to required QoS and security each application se rvices. The classification is performed through DSCP (Differentiated Service Code Point) of incoming IP packet. Another important factor is the information from PDP (Policy Decision Point) in overlay network , and the agent in PEP (Policy Execution Point) at VER receives policies or dispatches them to the appropriate VER.

As shown in Fig.2, routing information from incoming packets and PDP will make its own routing table in VER (see (c), (d) and (e) in Fig. 2). And the multiple virtual routing tables constructed according to security level and required QoS are applied to perform routing function over overlay network. In this paper, the details on overlay networking and VNF will not be handled due to a insufficient space of the paper. Thus, it is advised to refer the procedure of Fig. 2 for the details on VNF routing table construction.

In order to construct a forwarding information base from the information on routing and policies, the modified routing management algorithm is necessary. This paper proposes the novel routing management algorithms as shown in Fig. 3. The primary goal of VNF provisioning on overlay network is to provide multiple levels of secured QoS paths. The VNF routing function is performed independently with other VNF routing in the network. If a service user wants to get a specific level of secured QoS path in internet, the DSCP is assigned to the incoming packets and the packets are classified and allocated to its VNF table to have security and QoS-based routing. The principle of the VNF routing algorithm are shown in Fig. 3.

In order to adopt a practical approach to consider real network situations, policy server with bandwidth managing function takes a important role to control the overlay network resource efficiently, and its admission control for a call request will be managed by the resource managing function of the policy server. The (a) of Fig. 3 describes the procedures to make virtual forwarding table using QSA (QoS and Security Advertisement) message. The (b) shows the updating procedure virtual forwarding table on virtual overlay network.

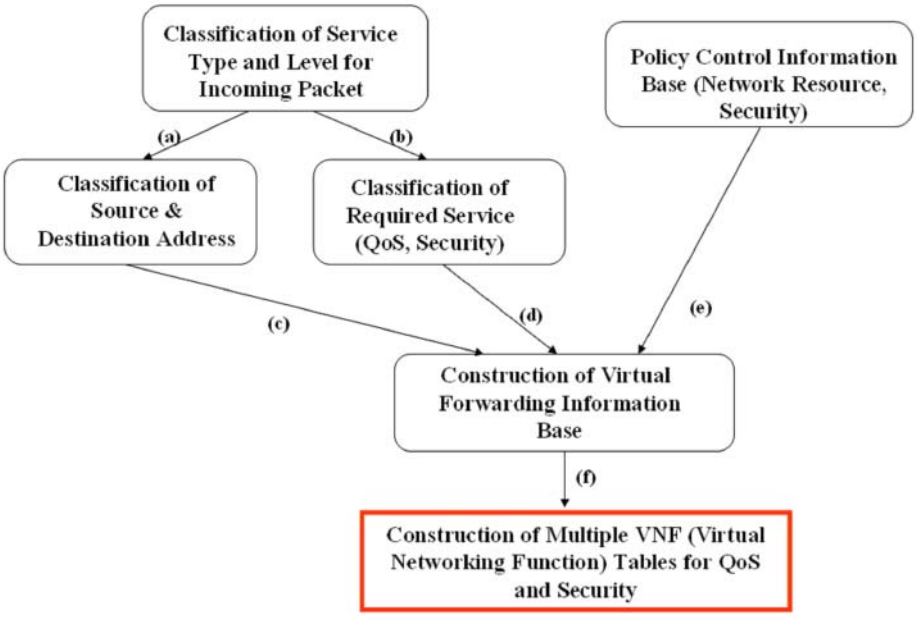


Fig. 2. Registration Procedure for Virtual Networking in Overlay Network

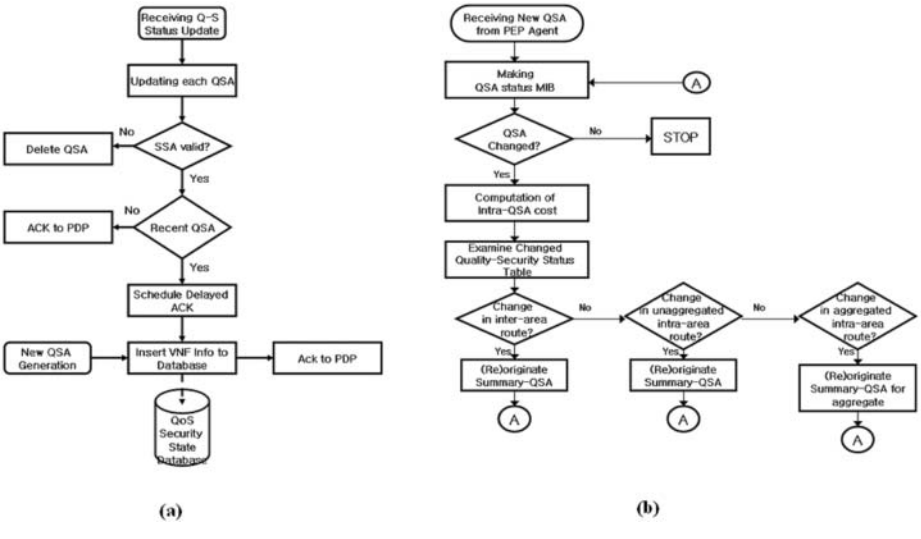


Fig. 3. Routing Mechanisms for Policy-based Overlay Networking with QoS-Security

2.2 Proposed Policy-Based Resource Management Protocols in Overlay Network

The policy agent/bandwidth manager implements all the allocation policies, and uses that information to manage the set of agents implemented in the edge node. The policy agent/bandwidth manager initializes the agents with specific source, destination, and bandwidth restrictions; once initialized the bandwidth associated with the agents remains reserved until the policy agent/bandwidth manager specifically authorizes the media stream at which point the edge node allows the media to pass through the agents.

If we define the required protocols on the policy agent and policy server, it is defined as the followings. It indicates summarized protocol procedures on the interaction between policy agent and server with specific. Messages initiated by the policy agent/bandwidth manager include some specific control functions, e.g., Agent-Alloc, Agent-Set, Agent-Info, Agent-Open, and Agent-Delete, while the edge node may initiate the Agent-Close message.

The policy agent/bandwidth manager initiated messages are sent using client specific objects within the decision object of COPS DECISION messages. The responses to the policy agent/bandwidth manager initiated messages are sent as a REPORT-STATE message with client specific objects in the Client object by the edge node. For the ACK messages the COPS Report-Type value MUST be 1 and for the ERR messages the Report-Type MUST be 2. In this study the protocol message format of revised COPS protocol between policy server and agent has bee defined as the Fig. 4. In the example, it shows that Direction is either 0 for a downstream agent, or 1 for an upstream agent. ProtocolID is the value to match in the IP header, or zero for no match.

3 Virtual Resource Management Algorithm in Policy-Based Overlay Network

3.1 Virtual Resource Computation Algorithm

For the computation algorithm of expected occupied resource amount of connections, this paper uses a discrete-time Markov BD process model and proposes a novel concept derived from a conventional first passage time [3], which is a time duration that state i reaches to state j (i j) in Markov BD process.

The proposed passage time in the paper is conditioned by initial state, and its transition path is diversified rather than a conventional first passage time. The novel concept, which is a special case of the first passage time, the first up-passage time (FUT) and the first down-passage time(FDT), is introduced in [6]. Using the concepts (FUT and FDT), as shown in Fig. 5, the novel concepts to find the computation algorithm of packet loss probability in the network.

The Loss period to the virtual capacity of overlay network is computed from the results, FUT and FDT and U_i, obtained at previous sections. The expected offered load period ($\overline{W}$)is computed by using the heap occurrence probability

Length = 60		S-Num = 5	S-Type = 1
Direction	ProtocolID	Flags	Session Class
Source IP Address (32-bits)			
Destination IP Address (32-bits)			
Source Port (16-bits)		Destination Port (16-bits)	
DiffServ Code Point			
Timer T1 value		Reserved	
Timer-T7 value		Timer-T8 value	
Token Bucket Rate [r] (32-bit IEEE floating point number)			
Token Bucket Size [b] (32-bit IEEE floating point number)			
Peak Data Rate (p) (32-bit IEEE floating point number)			
Minimum Policed Unit [m] (32-bit integer)			
Maximum Packet Size [M] (32-bit integer)			
Rate [R] (32-bit IEEE floating point number)			
Slack Term [S] (32-bit integer)			

Fig. 4. A Frame Format Proposed COPS between Policy Agent/Server

(U_i) as shown in Appendix and the property of probabilistic similarity in expected path length computation.

$$E[W] = \sum_{i=1}^{N} \bar{W}_i \cdot U_i = \sum_{i=1}^{N} \left(\bar{W}_{i,up} + \bar{W}_{i,down}\right) \cdot U_i \tag{1}$$

Define E[W_{up}] and E[W_{down}] an expected upward-path length and an expected downward-path length for all heaps, respectively. The upward-path length indicates the total path length of when the number of active sources takes journey from state 0 to the top state k. There may be fluctuations going down and up, and the total length of upward-path is a summation of those paths. The downward-path length is also the total path length of while the number of active sources reach to top state from state 0. First E[W_{up}] is rewritten by using the concept and some formula in [6] as:

$$E[W_{up}] = \sum_{i=1}^{N-1} \left(\sum_{k=1}^{i} X_{k,(1,i)} \right) \cdot U_{i+1} = \sum_{i=1}^{N-1} \left[\sum_{k=1}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{q_j} \right) \cdot q_k \cdot \bar{S}_k \right] \cdot U_{i+1} \tag{2}$$

From (2), we can see that E[W_{up}] is a function of S_k(k=1,2,...,N-1). So, we can rewrite (2) by $A_k(S_k)$ as:

$$E[W_{up}] = A_1(\bar{S}_1) + A_2(\bar{S}_2) + A_3(\bar{S}_3) + ... + A_{N-1}(\bar{S}_{N-1}) \tag{3}$$

where,

$$A_k(\bar{S}_k) = \sum_{i=1}^{N-1} X_{k,(1,i)} \cdot U_{k+1}, X_{k,(1,i)} = \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{p_l} \right) \cdot \frac{1}{q_k} \cdot \bar{S}_k$$

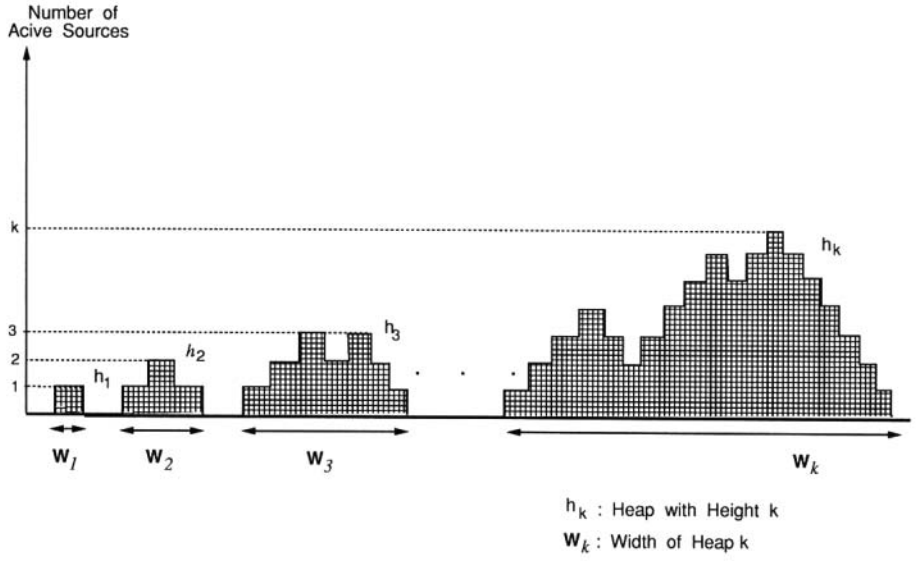


Fig. 5. An Example of Heap Generation in Virtual Capacity

In the computation of E[W$_{down}$],the similar procedure with E[W$_{up}$] will be applied.

$$E[W_{down}] = \sum_{i=1}^{N} \bar{W}_{i,down} \cdot U_i = \sum_{i=1}^{N} \left[\sum_{k=1}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{p_l}{q_l} \right) \cdot p_j \cdot \bar{S}_j \right] \cdot U_i \qquad (4)$$

As we see(4), E[W$_{down}$] is also a function of $\bar{S}_k$ (k=1,2,...,N-1), (4) is rewritten by A$_k(\bar{S}_k)$, which is the summation of Y$_{k,(i,1)}$.

$$E[W_{down}] = B_1(\bar{S}_1) + B_2(\bar{S}_2) + B_3(\bar{S}_3) + ... + B_N(\bar{S}_N) \qquad (5)$$

where,

$$B_1(\bar{S}_1) = \sum_{i=1}^{N} Y_{1,(i,1)} \cdot U_i, , B_2(\bar{S}_2) = \sum_{i=2}^{N} Y_{2,(i,1)} \cdot U_i, B_k(\bar{S}_k) = \sum_{i=k}^{N} Y_{k,(i,1)} \cdot U_k$$

$$B_N(\bar{S}_N) = Y_{N,(N,1)} \cdot U_N \quad and\ Y_{k,(i,1)} = \sum_{j=1}^{k} \left(\prod_{l=j}^{k} \frac{p_l}{q_l} \right) \cdot \frac{1}{p_k} \cdot \bar{S}_k \qquad (6)$$

As shown so far, the expected offered load period ($\bar{W}$) is expressed by the expected path length ($\bar{S}_i$) of each state. The expected offered packets during load period can be computed by product of arriving packets at each state during one frame period (D). And It will be noted that the expected *Loss Period* ($E[T_{loss}]$)

at this model is simply calculated without any further computation labor. That is, from (6), the overflow period is counted. The expected *Loss Period* ($E[T_{losss}]$) is as follows:

$$\begin{aligned} E[T_{loss}] &= \sum_{i=D+1}^{N-1} A_i(\bar{S}_i) + \sum_{i=D+1}^{N} B_i(\bar{S}_i) \\ &= \sum_{i=D+1}^{N-1} \left(\sum_{k=1}^{i} X_{k,(1,i)} \right) \cdot U_{i+1} + \sum_{i=1}^{N} \left(\sum_{k=1}^{i} X_{k,(1,i)} \right) \cdot U_i \\ &= \sum_{i=D+1}^{N-1} \left[\sum_{k=1}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{p_l} \right) \cdot q_k \cdot \bar{S}_k \right] \cdot U_{i+1} \\ &\quad + \prod_{i=D+1}^{N} \left[\sum_{k=l}^{i} \sum_{j=k}^{i} \left(\prod_{l=k}^{j} \frac{q_l}{p_l} \right) \cdot q_j \cdot \bar{S}_j \right] \cdot U_i \end{aligned} \tag{7}$$

And the expected offered packets during load period can be computed by product of arriving packets at each state during one frame period (D). Let $\bar{N}_{offered}$be the expected offered packets during offered load period.

$$\bar{N}_{offered} = \sum_{i=1}^{N-1} A_i(\bar{S}_i) \cdot i + \sum_{i=1}^{N} B_i(\bar{S}_i) \cdot i \tag{8}$$

The mean number of lost packets during loss period ($E[T_{loss}]$) is computed from (7). Let $\bar{N}_{lost}$ the expected lost packets during loss period ($E[T_{loss}]$).

$$\bar{N}_{lost} = \sum_{i=D+1}^{N-1} A_i(\bar{S}_i) \cdot (i - D) + \sum_{i=D+1}^{N} B_i(\bar{S}_i) \cdot (i - D) \tag{9}$$

Finally, we can obtain the PLR (Packet Loss Ratio) from the results of (8) and (9).

$$\begin{aligned} PLR &= \frac{lost\_packets}{offered\_packets} = \frac{\bar{N}_{lost}}{\bar{N}_{offered}} \\ &= \frac{\sum_{i=D+1}^{N-1} A_i(\bar{S}_i) \cdot (i - D) + \sum_{i=D+1}^{N} B_i(\bar{S}_i) \cdot (i - D)}{\sum_{i=1}^{N-1} A_i(\bar{S}_i) \cdot i + \sum_{i=1}^{N} B_i(\bar{S}_i) \cdot i} \end{aligned} \tag{10}$$

3.2 Virtual Resource Management Algorithm and Its Feasibility

As anticipated QoS traffic demand and network availability estimates are forecasts, they should br treated as such by the traffic management and service layer

functions. Actual offered traffic will fluctuate around the forecast values in the long term. The algorithm introduced in the paper is *fluid-flow approximation algorithm* to make a decision of incoming new call request on virtual overlay network. Many approximation algorithms for bursty traffic sources have been proposed, but they take limited applications in real world due to complexity in computation or due to too much approximation. As shown in the paper, the proposed *fluid-flow approximation algorithm* provides a less computational cost for real-time application than the fluid flow approximation approach. In our approximation algorithm,$A_{N-1}(\bar{S}_{N-1})$ and $B_k(\bar{S}_k)$are used as function of λ, μ and N. The values of $A_{N-1}(\bar{S}_{N-1})$ and $B_k(\bar{S}_k)$ can be computed and tabulated according to input traffic parameters in advance. The tabulation is constructed as a function of a number of sources. The approximation algorithm can make a table considering few adjacent possible situations in advance, and it reduces the computation cost (e.g., $O(n)$) to estimate and reserve system resource for newly incoming traffic. It should be more useful approximate for network resource computation with fixed-sized packets. The numerical results on the study has been analyzed, but it is regret not to cover it due to the shortage of available space of the paper.

If we consider the numerical results, the algorithm shows that results is very likely to fluid-flow approach in [6] and [7].

4 Conclusion

Main strength of our approach to provide virtual secured QoS routing algorithm in overlay networking can be summarized as follows:

- The proposed architecture on virtual overlay routing algorithm makes a clear separation to the physical internet through VNF mechanism. And its feature will show efficient virtual routing algorithm to guarantee QoS and security on overlay network.
- The proposed architecture on policy-based overlay network resource management makes the measurement of overlay network resources likely real-time, and resource control will be performed more practically.
- We have proposed the approximated algorithm to compute resource capacity and to control admission for incoming calls. The algorithm compute a dynamic virtual capacity for each VNF tables, and its algorithm is shown to have scalability in resource control.
- The proposed architecture will also provide diverse service features in QoS and securities as user requests to network service providers.

References

[1] Christos Tsarouchis, Spyros Denazis, et al, "A policy-Based Management Architecture for Active and Programmable Networks:, IEEE Network, May/June 2003
[2] Man Li, "Policy-Based IPsec Management", IEEE Network, November/December 2003

[3] Gisli Hjalmtysson, "Control-on-Demand: A Efficient Approach to Router Programmability", IEEE JSAC, Vol. 9, Sept., 1999
[4] Danel P. Heyman and Matthew J. Sobel, "Stochastic Models in Operations Research - Volume I," McGraw Hill Book Company pp 38-104, 1988
[5] Ilyoung Chong, "Cost Minimization Allocation (CMA) Algorithm", Technical Report, Univ. of Massachusetts, 1992.
[6] Ilyoung Chong, "Traffic Control at Burst Level", Ph.D. Thesis, Univ. of Massachusetts, 1992
[7] Thomas M. Chen, "Evolution to the Programmable Internet", IEEE Communications Magazine, March 2000
[8] D. Scott Alexander, William A. Arbaught, Angelos D. Keromytis, Steve Muir, and Jonathan M. Smith, "Secure Qulaity of Service Handling: SQoSH", IEEE Communications Magazine 2000
[9] Roch Guerin, Hamid Ahmadi and Mahmoud Naghshineh, "Equivalent Capacity and its Aplication to Bandwidth Allocation in High Speed Networks," IBM Research Report RC 16317, 1990
[10] Panos Trimintzios, Geoge Pavou et al, "Service-Driven Traffic Engineering for Intradomain Qulaity of Service management," IEEE Network, May/June 2003

Appendix 1: Computation of Occurrence Probability of Heap Height i (U_i)

Suppose that the system enters state k and returns to state 0. In order to find the heap occurrence probability U_i that a heap with maximum height k occurs in the system, we consider the probability to obtain a connected digraph with maximum state k. Fig. 6 shows a connected digraph constructed by the available paths in Markov BD chain with maximum N states and an example for the maximum height of 3. Let U denote the probability that a graph with maximum state i is created in the system during state transition. The graph doesn't contain state 0 at intermediate transition nodes. In the computation of U, it will be necessary to check whether the transition time at each state affects the transition probability or not. This property is based on the property of Markov BD process. A time period is required to reach the next state. The required time period will be computed by using the algorithms of FUT and FDT. However, in the discrete-time Markov BD process, given the current state, the identity of the next state visited is independent of the time required to get there. Here, the procedure to compute probabilities for all possible paths in Markov BD chains in shown. And let $\alpha_i = \mathrm{p}_i \, \mathrm{q}_{i+1}$.

1. At first, consider a digraph with maximum state 1. Its transition path is 0 $\rightarrow$ 1 $\rightarrow$ 0, and the probability to obtain this graph is,

$$U_1 = p_0 q_1 = \alpha_{\,0}. \tag{11}$$

2. For maximum state = 3,

$$U_3 = \alpha_0 \alpha_1 \alpha_2 \left[1 + \frac{1}{\alpha_1 - \alpha_2} \left(\frac{\alpha_1^2}{1 - \alpha_1} - \frac{\alpha_2^2}{1 - \alpha_2} \right) \right]$$

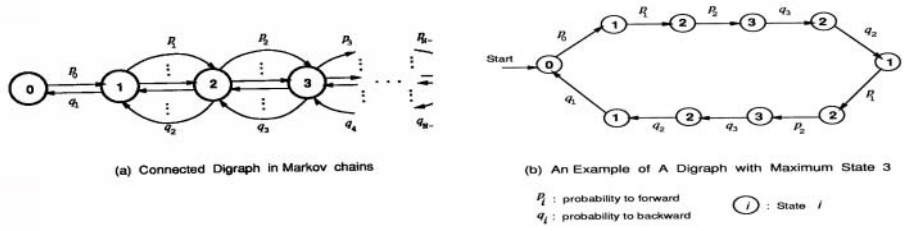


Fig. 6. Construction of Diraph for h3 heap in Marcov Chain

3. For maximum state = 4, the sum of probabilities of possible paths in digraph chains is computed similarly to the above. In order to express U_i simply, let define the *Combination Function*, $H_i^{(m)}(\alpha_1, \alpha_2, ..., \alpha_{m-1})$,, which denotes the combinations of α_i's, $1 < i < m-1$ with i different α variables. The superscript m indicates the maximum state and subscript i does the number of a variables of *Combination Function*. That is,

$$\begin{aligned} H_i^{(m)}(\alpha_1, \alpha_2, ..., \alpha_{m-1}) &= \alpha_1\alpha_2\alpha_3...\alpha_{i-1}\alpha_i \\ &+ \alpha_1\alpha_2\alpha_3...\alpha_{i-1}\alpha_{i+1} \\ &+ ... + \alpha_{m-i-1}\alpha_{m-i}...\alpha_{m-2} \\ &+ \alpha_{m-i}\alpha_{m-i}\alpha_{m-i+1}...\alpha_{m-1} \end{aligned} \tag{12}$$

From (12), we obtain the general formula of U_i ($i \geq 4$), we approximate the *heap occurrence probability*, U_i. The state fluctuation in the heap with maximum height i is described as same with the digraph with maximum state I as follows:

$$U_i \cong \frac{(\prod_{k=0}^{i-1})\left[1 + \sum_{k=1}^{i-2}\sum_{j=k+1}^{i-1}\frac{1}{\alpha_k - \alpha_j}(\frac{\alpha_k^2}{1-\alpha_k} - \frac{\alpha_j^2}{1-\alpha_j}) - \sum_{k=1}^{i-1}\frac{\alpha_k}{1-\alpha_k}\right]}{1 - H_3^{(i)}(\alpha_1, ..., \alpha_{i-1}) + \sum_{k=2}^{i-2}\left[(-1)^k(1 + \sum_{j=1}^{k-1}2^{2j-1})H_{k+2}^{(i)}(\alpha_1, ..., \alpha_{i-1})\right]} \tag{13}$$

and for i= 1, 2 and 3, U_i is as follows:

$$U_3 = \alpha_0\alpha_1\alpha_2\left[1 + \frac{1}{\alpha_1 - \alpha_2}(\frac{\alpha_1^2}{1-\alpha_1} - \frac{\alpha_2^2}{1-\alpha_2})\right], U_2 = \frac{\alpha_0\alpha_1}{1-\alpha_1}, U_1 = \alpha_0$$

HDR Forward Link Scheduler Supporting Service Differentiation with Fairness Bound*

Jaesung Choi[1], Myungwhan Choi[1], and C.M. Krishna[2]

[1] Dept. of Computer Science and Eng., Sogang Univ.
Seoul 121-742, Korea
jschoi@netdelta.sogang.ac.kr
mchoi@ccs.sogang.ac.kr
[2] Dept. of Electrical and Computer Eng., Univ. of Massachusetts
Amherst, MA 01003, USA
krishna@ecs.umass.edu

Abstract. We propose a high data rate (HDR) forward link scheduling algorithm maintaining the prescribed throughput ratios among users with fairness bound. The main idea is to increase the overall system throughput by delaying transmission chances of users whose feasible transmission rates are low until their transmission rates are high enough, within a specified time limit. By doing so, the short-term fairness in the sense of how well the prescribed throughput ratios are maintained gets damaged, but still with the long-term fairness maintained. The amount of short-term fairness damage is bounded by the specified limit. For this, we develop a mechanism to approximate the operation of the generalized processor sharing (GPS) model of HDR for packet-by-packet operation. The expected features of the proposed scheme are verified and their throughput performances are studied through simulation.

1 Introduction

The HDR is specified as the service in the CDMA2000 1xEV-DO (IS-856) to provide for the high-speed downstream Internet access to the relatively small number of mobile users using the current CDMA physical layer technology [1], [2], [3].

In the HDR, the access point (AP) (or base station, BS) can transmit packets to only one access terminal (AT) (or user) at one time and the transmission rate from the AP to the AT is determined by the signal to interference ratio (SIR) at the AT. Therefore, the scheduling scheme at the AP determines the forward link performance to each AT and accordingly the overall system performance such as total system throughput and overall customer satisfaction level.

There are several scheduling schemes proposed in the literature. Proportional fairness scheduling (PFS) [4] is proposed by the QUALCOMM and is being used

* This work was supported by the Institute for Applied Science and Technology, Sogang University.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 462–472, 2004.

for the CDMA2000 1xEV-DO service. In the PFS scheme, the priorities of the packets for the connections over the wireless forward link are assigned by the ratio of the data rate (supportable during the current time slot) to the moving average (MA) of the throughput of the corresponding connection over a certain predetermined time interval. Data is transmitted to the AT for which the priority is the highest. The PFS tends to give high priority to the ATs which move into the region where the received SIRs are greatly improved and accordingly the supportable data rates are high, especially when the MAs are small. Accordingly the AP tends to transmit packets over the forward links which can transmit data at high rate, which yields high overall system throughput. It has, however, no feature for service differentiation such as keeping prescribed throughput ratios among backlogged users.

The authors in [5] proposed a time-fraction assignment scheme to provide differential service, that provides "long-term" fairness among users. In other words, the *expectation* of the number of time slots (corresponding to a predetermined fraction of time) assigned to a user is guaranteed. However the proposed scheme does not allow the "fairness bound" to be specified, which gives a measure of fairness. That is, fairness damage in short-term cannot be bounded by some specified limit.

The authors in [6] formulated the optimization problem to obtain optimal long-run throughputs for given target throughput ratios among users and proposed an adaptive algorithm for this. This scheme, however, has the similar limitation as for the scheme proposed in [5].

In this paper, we propose a *new* HDR forward link scheduling scheme which provides the differential service such as keeping the prescribed throughput ratios, with fairness bound.

This paper is organized as follows. In Sect. 2, our system model is described. In Sect. 3, we describe the proposed algorithms and their properties. In Sect. 4, we show simulation results to illustrate the performance of the proposed algorithms. Conclusion follows in Sect. 5.

2 System Model

The forward link of the HDR consists of slots of length 1.67 ms and the AP can transmit data to only one AT during each time slot . The data rate of the forward link is not fixed. Instead, it changes as the received SIR at the corresponding AT changes. The SIR (or corresponding E_c/N_t) guaranteeing 1% packet loss ratio, packet length and the number of time slots needed to transmit the packet for a given data rate are shown in Table 1

The scheduling scheme in the AP does the following: at each decision epoch, the scheduler in the AP decides which user should be assigned the time slot based on the objective of the system operation. The objective of our scheduling system is to provide the differential service among users with fairness bound.

The well known GPS scheme can be used to provide different grade of service to different users in the *wireline* network [7]. However, the GPS scheme which

Table 1. SNR for 1% packet error rate [1]

Data rate (kbps)	E_c/N_t (dB)	Packet length	
		(bits)	(slots)
38.4	-12.5	1024	16
76.8	-9.5	1024	8
153.6	-6.5	1024	4
307.2	-4.0	1024	2
614.4	-1.0	1024	1
921.6	1.3	3072	2
1228.8	3.0	2048	1
1843.2	7.2	3072	1
2457.6	9.5	4096	1

allocates the *fixed* bandwidth among users according to the prescribed performance (e.g., throughput) ratios can not be directly used to achieve our objective in the wireless networks such as HDR network. In the next section, we extend the GPS to make this possible in the HDR network.

3 Scheduling Policy

3.1 Extension of GPS to HDR Service

GPS is defined as follows [7]. Let N denote the number of sessions serviced by a server with the total capacity C_{GPS} bits per second. Also assume that ϕ_i is assigned to session i, $i = 1, \ldots, N$, where the positive real number ϕ_i represents the relative service weight among sessions. Let $W_i(t_1, t_2)$ and $W(t_1, t_2)$ denote the amount of session i traffic serviced and the total traffic serviced by the server during the time interval $[t_1, t_2)$, respectively. Then, (1) is satisfied for the GPS server:

$$\frac{W_i(t_1, t_2)}{\phi_i} = \frac{W(t_1, t_2)}{\sum_{j \in B(t_1)} \phi_j} \qquad \forall i \in B(t_1) \ . \tag{1}$$

Here, $B(t)$ represents the set of sessions backlogged at time t and it is assumed that the set $B(t)$ does not change during the time interval $[t_1, t_2)$.

From (1), it can be shown that (2) is satisfied if the sessions i and j are continuously backlogged during the interval $[t_1, t_2)$:

$$\frac{W_i(t_1,t_2)}{\phi_i} = \frac{W_j(t_1,t_2)}{\phi_j} \quad . \tag{2}$$

This says that the GPS server services the backlogged sessions simultaneously and the amount of session i service received is proportional to its weight ϕ_i.

GPS can be modeled by the fluid system in which infinitesimally small amount of data each from multiple traffic streams can be transmitted simultaneously. Notice however that the system capacity at time t for the HDR is determined by the transmission rate to session i selected for transmission at time t and the HDR cannot be modeled by the fluid system.

The authors in [8] introduced the HDR-GPS that extends the GPS to have the features of the HDR that only one session can be serviced at one point of time and the transmission rate for each session may differ for different sessions.

3.2 Packetized HDR-GPS

For the HDR-GPS model, it was assumed that the packets can be arbitrarily small, which is unrealistic. Thus we need to devise a mechanism to approximate the operation of the HDR-GPS for the packet-by-packet transmission. For this we follow the approach used to approximate the operation of the GPS for the packet-by-packet transmission: WFQ (weighted fair queuing) [7] and WF^{2}Q (worst-case fair weighted fair queuing) [9] which closely approximate the operation of GPS.

Before we describe the proposed schemes it is helpful to mention the specific features of the operation of the HDR. First of all, at any time t, the transmission rate is determined by the channel condition over the link between the AP and the AT selected for transmission. On the contrary, for the WFQ and the WF^{2}Q, the transmission rate is fixed. Secondly, the size of the packet transmitted and the number of slots used to transmit that packet vary according to the channel state as shown in Table 1. These features should be taken into account in the design of the packetized HDR-GPS algorithms.

Now we extend the WFQ and the WF^{2}Q algorithms which are proven to approximate the GPS well to have the features mentioned above. We assume that the arriving packet size is 1024 bits long. Up to 4 packets are packed into one transmission *frame* depending on the transmission rate which varies as the channel condition changes. When multiple packets are packed into one frame (multi-packet transmission), one or more null packets may be included if needed. For instance, when the transmission rate is 2457.6 kbps, the transmission frame size is 4096 bits long and accordingly the transmission frame can accommodate up to 4 packets. If enough packets are not available when the corresponding session is scheduled for transmission, AP needs null packets to form a 4096-bit transmission frame. Such null packets do not contribute to the amount of service since they are not the actual data.

Algorithm 1. Let s_n be the time instant of the nth decision epoch. Since multiple slots may be used to transmit a frame, the scheduling is not performed

every time slot. Let t_m be the start time of the mth slot. If $s_n = t_m$ and session i is selected for transmission at decision epoch s_n, the next decision epoch s_{n+1} is $t_{m+N_S(r_i(t_m))}$, where $N_S(r)$ is the number of slots used to transmit a transmission frame when the transmission rate is r.

Now we define the virtual time function $V(t)$ as follows to make the proposed algorithm to be suitable to HDR:

$$V(t) = V(s_n) + \int_{s_n}^{t} \frac{r(u)\alpha(s_n)}{\sum_{i \in B_{IDEAL}(u)} \phi_i} du \quad \text{for } s_n \le t < s_{n+1} \ , \tag{3}$$

where $B_{IDEAL}(u)$ is the set of backlogged sessions at time u under the condition that all sessions are fairly served, $r(t)$ is the actual transmission rate at time t, and $\alpha(s_n)$ is the ratio of the number of data packets to the number of data and null packets transmitted during $[s_n, s_{n+1})$.

Let p_i^k, a_i^k, and L_i^k represent the kth packet for the session i, its actual arrival time, and its length, respectively. Since $L_i^k = 1024$, the virtual start time and the finish time of a packet p_i^k are given as follows, using the virtual time function $V(t)$ defined in (3):

$$\begin{aligned} S_i^k &= \max\left\{F_i^{k-1}, V(a_i^k)\right\} \ , \\ F_i^k &= S_i^k + \tfrac{1024}{\phi_i} \ . \end{aligned} \tag{4}$$

Let the HOQ_i and the EOQ_i be the indexes of the packets located in the head and the tail of the queue of session i, respectively. Let the EOT_i be the index for the last data packet which is packed into the transmission frame if session i is selected for the transmission at s_n. Then, EOT_i is determined by

$$EOT_i = \max_k \arg \left\{ \begin{array}{l} k \le \min\{EOQ_i, HOQ_i + N_P(r_i(s_n)) - 1\} \text{ and} \\ F_i^k \le \left(S_i^{HOQ_i} + \frac{1024 \times N_P(r_i(s_n))}{\phi_i} + A_i\right) \end{array} \right\} \ , \tag{5}$$

where A_i is the parameter to limit the loss in fairness caused by the multi-packet transmission, $r_i(t)$ is the transmission rate for session i at time t, and $N_P(r)$ is the size of the transmission frame in number of packets when the transmission rate is r.

At decision epoch s_n, session i's virtual start time S_i and virtual finish time F_i are defined as follow:

$$\begin{aligned} S_i &= S_i^{HOQ_i} \ , \\ F_i &= F_i^{EOT_i} \ . \end{aligned} \tag{6}$$

Then, at decision epoch s_n, our algorithm selects the session i whose virtual finish time F_i is the smallest among candidates which satisfy the condition $S_i \le V(s_n)$ for the transmission.

The set $B_{IDEAL}(u)$ necessary to define the virtual time function $V(t)$ of (3) is obtained as follows:

$$B_{IDEAL}(u) = \left\{ i \;\middle|\; \begin{array}{c} Q_i(u) = 0 \text{ and } F_i^{LAST_i} > V(u) \\ \text{or} \\ Q_i(u) \neq 0 \text{ and } F_i^{EOQ_i} > V(u) \end{array} \right\} \ , \tag{7}$$

where $Q_i(u)$ is the queue length of session i at time u and $LAST_i$ is the index for the packet last transmitted for session i.

Algorithm 2. Notice that the channel state changes rapidly. Considering this, if we delay transmissions to users whose feasible transmission rates are low until their transmission rates are high enough, within a specified time limit, it would be possible to improve the system throughput. The goal of the Algorithm 2 is to achieve this with the bounded loss of short-term fairness.

Let DF_i denote the session i's deferred virtual finish time. Then, at decision epoch s_n, our algorithm selects the session whose DF_i is the smallest among the backlogged sessions for the transmission. DF_i is defined as follows:

$$DF_i = F_i + D_i(r_i(s_n)) \ , \tag{8}$$

where $D_i(r_i(s_n))$ is a non-increasing function of $r_i(s_n)$. $D_i(r_i(s_n)) \geq 0$ and $D_i(r_{MAX}) = 0$, where r_{MAX} is the highest transmission rate available, 2547.6 kbps. When the channel state of session i gets worse, $D_i(r_i(s_n))$ increases and accordingly the deferred virtual finish time increases. Consequently, the selection of that session for transmission will be delayed. The extent of deferment is determined by $D_i(r_i(s_n))$. If the channel gets better, $D_i(r_i(s_n))$ decreases and accordingly the deferred virtual finish time decreases. In the decision epoch, if the newly computed DF_i is the smallest among the deferred virtual finish times of the sessions, the session i will be selected for transmission. In this way, we can improve system throughput by provoking packet transmission for the user whose channel state is good enough.

Fairness analysis. Proportional fairness index is defined as the weighted difference in the received amount of service for any sessions i and j that are continuously backlogged for some duration $[t_m, t_n)$ [10]. Let $D_i(r_i(t_m)) \leq D_{MAX}$ and $L_{MAX} = 4096$ bits which is the maximum transmission frame size in HDR. Then the upper bound for the proportional fairness index given by the theorem below shows that the proposed algorithm provides *hard* fairness bound which is determined by D_{MAX}, L_{MAX}, ϕ_k, and A_k.

Theorem 1. *Let $i, j \in B_{IDEAL}(u)$ at time u, $u \in [t_m, t_n)$. Then the following inequality holds.*

$$\left| \frac{W_i(t_m, t_n)}{\phi_i} - \frac{W_i(t_m, t_n)}{\phi_j} \right| \leq 2 \cdot \left\{ D_{MAX} + \max_k \left\{ \frac{L_{MAX}}{\phi_k} + A_k \right\} \right\} \ . \tag{9}$$

The proof is omitted here due to space limitation and presented in [11].

Design of the function $D_i(r_i(s_n))$**.** Typically the ATs residing near the AP are under good channel condition and the ATs residing far from the AP experience bad channel condition in the wireless environment although the situation may be quite opposite due to the fading effect. As an example, consider the situation where the session 1 is currently under good channel condition and its transmission rate is in the range of from 1228.8 to 2457.6 kbps. Also assume that the session 2 is currently in the bad channel state and its transmission rate is in the range of from 38.4 to 921.6 kbps. Assume that at decision epoch s_n, $F_1 = F_2$, the session 1's transmission rate is 1228.8 kbps and the session 2's transmission rate is 921.6 kbps, and accordingly session 1 is selected for transmission by the Algorithm 2. Notice that $DF_1 < DF_2$ assuming that $D_i(r_i(s_n))$ is a decreasing function of $r_i(s_n)$ only. If this situation lasts, more chances will be given to session 1 for transmission and it may make it difficult to meet the fairness requirement. Therefore it would be desirable to select the session 2 for transmission even though its transmission rate is lower than that of the session 1 if the session 1 is in the relatively poor channel state among those states where it might be and the session 2 is in the relatively good channel state among those states where it might be, to improve the fairness among users.

In this paper, we propose the following function for $D_i(r_i(s_n))$ to have the above mentioned feature:

$$D_i(r_i(s_n)) = \begin{cases} 0 & \text{if } r_i(s_n) \geq \overline{r}_i(s_n) \\ \left(\frac{\overline{r}_i(s_n) - r_i(s_n)}{\overline{r}_i(s_n) - r_{MIN}}\right)^{\beta} \times D_{MAX} & \text{otherwise} \end{cases} \tag{10}$$

Here, β is any real number greater than 0, r_{MIN} is the transmission rate feasible in the worst channel state (38.4 kbps), and $\overline{r}_i(s_n)$ is the average transmission rate, computed at s_n, for the slots during which packets are transmitted for session i.

Determining D_{MAX} ***in number of time slots.*** When a session is in bad channel condition and it is determined by the proposed algorithm to defer the transmission of packets for that session, it may be sometimes more convenient to set the bound on the fairness index by delaying the packet transmission within a certain time limit expressed in number of time slots.

Let DS_{MAX} denote the actual time limit expressed in number of time slots. To make it possible for the proposed algorithm to delay the packet transmission within DS_{MAX}, we propose the following scheme to be used to determine D_{MAX}, which allows to compute the virtual time bound D_{MAX} adaptively warranted by the corresponding DS_{MAX}:

$$D_{MAX} = I(t_m) \times DS_{MAX} \ , \tag{11}$$

$$I(t_m) = \left(1 - \frac{1}{k_c}\right) \times I(t_{m-1}) + \frac{1}{k_c} \times \{V(t_m) - V(t_{m-1})\} \ , \tag{12}$$

Table 2. Average long-term throughput of each session

i	ϕ_i	Mean of $r_i(t_m)$ (kbps)	Throughput (kbps)		
			A1	A2, $DS_{MAX} = 500$ $\beta = 1.5$	A2, $DS_{MAX} = 1000$ $\beta = 1.5$
1	0.2	1051.8	43.6	145.0	160.8
2	0.2	732.5	43.6	145.0	160.8
3	0.2	1398.7	43.6	145.0	160.8
4	0.1	2120.5	21.8	72.5	80.4
5	0.1	1408.9	21.8	72.4	80.4
6	0.1	1389.8	21.8	72.5	80.4
7	0.1	1235.6	21.8	72.5	80.4

where $I(t_m)$ is the moving average of the virtual time increments per time slot using the averaging filter time constant k_c.

4 Simulation Results

We present the performance of the proposed scheduling schemes, Algorithm 1 (A1) and Algorithm 2 (A2). In the simulation, it is assumed that the traffic source of each session is persistent and every time slot is completely filled by the traffic source. To observe the throughput performance experienced by users when the channel conditions change, we conducted the simulation under the time-varying channel conditions along with $k_c = 1000$ and $A_i = 0$.

In the experiment, we assume that seven users are classified into 2 groups of users, each group with the identical throughput requirements: $\phi_1 = \phi_2 = \phi_3 = 0.2$ and $\phi_4 = \phi_5 = \phi_6 = \phi_7 = 0.1$. That is, the prescribed throughput ratios among different groups are 2:1. Table 2 shows that for both Algorithms 1 and 2 the long-term throughput performances of the users are well maintained as specified by the weight ϕ_i and the throughput of Algorithm 2 is well over three times as high as that of Algorithm 1. For Algorithm 2, the throughput obtained using $DS_{MAX} = 1000$ is higher about by 10 % than using $DS_{MAX} = 500$. But using large DS_{MAX} yields large fairness bound.

Figures 1 and 2 show the average feasible transmission rate and the average achieved throughput, respectively over the moving time window (MWin) of 1000 time slots (MWin = 1000), plotted every 200-time-slot interval, which would give some feeling about the short-term fairness. Notice that the 1000 time slots correspond to the 1.67 sec.

Although not shown here, it is observed that the Algoirthm 1 maintains the prescribed throughput ratios among users even during the short-term period.

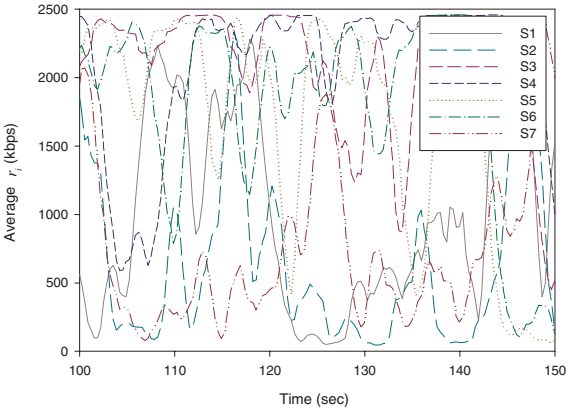


Fig. 1. Change of average feasible transmission rate of each session (MWin = 1000)

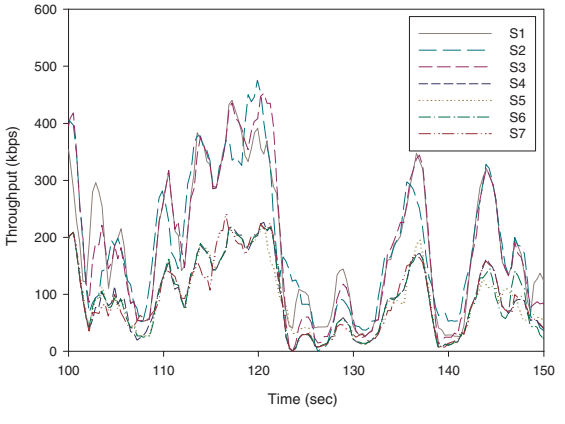


Fig. 2. Throughput of each session (A2, $DS_{MAX} = 500$, $\beta = 1.5$, MWin = 1000)

On the other hand, the Algorithm 2 cannot perfectly maintain the prescribed throughput ratios among users in short-term period as shown in Fig. 2. However, the degree to which the prescribed throughput ratios among users are maintained improves as the DS_{MAX} decreases as expected. Also notice that the long-term throughput performance ratios are well maintained by Algorithm 2 with much higher throughput performance compared with that using Algorithm 1.

As shown in Fig. 2, the throughput degrades severely over some intervals even using Algorithm 2. This happens because some sessions experience very poor channel condition for quite long period of time. For example, over the time interval [122 sec, 128 sec], the feasible transmission rates of sessions 1 and

2 are very low (see Fig. 1) and this situation lasts for about 6 seconds. By using 500 or 1000 for DS_{MAX}, the Algorithm 2 can at the very most delay the transmission chances up to 500 or 1000 slots which corresponds to 0.835 seconds or 1.67 seconds. Since the channels for sessions 1 and 2 are in poor condition well over the time interval determined by DS_{MAX}, the sessions in poor channel conditions are selected for transmission resulting in degraded system throughput. To overcome this kind of problem, it is necessary to increase the DS_{MAX} value but with poorer achieved fairness.

The effects of β on the system throughput were also studied through simulation and we found that the system throughput performance is very insensitive to β and β between 1.0 and 1.5 gives the best performance.

5 Conclusion

It is verified by both simulation and analysis that our scheme provides a good tradeoff mechanism with hard fairness bound between maximizing the system throughput and meeting the fairness performance. There is , however, a potential to underutilize the available system throughput especially when some sessions are in deep fading for a considerably long period of time. Therefore, it would be desirable to devise a mechanism to overcome this limit and we believe that the proposed algorithm sheds light on achieving this target.

References

[1] Bender, P., Black, P., Grob, M., Padovani, R, Sindhushayana, N., Viterbi, A.: CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users. IEEE Communications Magazine, Vol. 38, No. 7 (2000) 70–77 462, 464

[2] Sindhushayana, N. T., Black, P. J.: Forward Link Coding and Modulation for CDMA2000 1XEV-DO (IS-856). Proc. PIMRC (2002) 1839–1846 462

[3] Black, P. J., Gurelli, M. I.: Capacity Simulation of cdma2000 1xEV Wireless Internet Access System. Proc. IEEE MWCN Conference (2001) 90–95 462

[4] Jalali, A., Padovani, R., Pankaj, R.: Data Throughput of CDMA-HDR a High Efficiency-High Data Rate Personal Communication Wireless System. Proc. IEEE VTC (2000) 1854–1858 462

[5] Liu, X., Chong, Edwin K. P., Shroff, N. B.: Opportunistic Transmission Scheduling With Resource-Sharing Constraints in Wireless Networks. IEEE J. on Selected Areas in Communications, Vol. 19, No. 10 (2001) 2053–2064 463

[6] Borst, S., Whiting, P.: Dynamic Rate Control Algorithms for HDR Throughput Optimization. Proc. IEEE INFOCOM (2001) 976–985 463

[7] Parekh, A. K., Gallager, R. G.: A Genralized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. IEEE/ACM Transactions on Networking, Vol. 1, No. 3 (1993) 344–357 463, 464, 465

[8] Choi, J., Choi, M.: A Bandwidth-efficient HDR Forward Link Scheduler Supporting Service Differentiation. Proc. CIC (2003) 465

[9] Bennett, Jon C. R., Zhang, Hui: WF^2Q: Worst-case Fair Weighted Fair Queueing. Proc. IEEE INFOCOM (1996) 120–128 465

[10] Bensaou, B., Tsang, Danny H. K., Chan, K. T.: Credit-Based Fair Queueing (CBFQ): A Simple Service-Scheduling Algorithm for Packet-Switched Networks. IEEE/ACM Transactions on Networking, Vol. 9, No. 5 (2001) 591–604 467

[11] Choi, J., Choi, M., Krishna, C. M.: HDR Forward Link Scheduler Supporting Service Differentiation with Fairness Bound. Tech. Rep. TR-04-01, Dept. of Computer Sci. and Eng., Sogang Univ. (2004) http://netbeta.sogang.ac.kr 467

A Fast Method to Estimate Loss Rate

Weiping Zhu[1] and Zhi Geng[2]

[1] School of Computer Science, The University of New South Wales, Australia
[2] Institute of Mathematical Science, Peking University, China

Abstract. Loss tomography, as a key component of network tomography, aims to obtain the loss rate of each link in a network by end-to-end measurement. If knowing the loss model of a link, we, in fact, deal with a parametric estimate problem with incomplete data. Maximum likelihood estimates are often used in this situation to identify the unknown parameters in the loss model. The estimation methods either rely on iterative approximation to identify the parameters or solve some high order simultaneous equations. Both require a long execution time, and the former also needs to consider how to avoid trap into a local maximum. In this paper, we propose an estimate that is based on the correlation between a link and its sibling brothers to identify the loss rate of the link. It, instead of using an iterative approach to approximate the maximum, employs a bottom-up approach to identify the loss rates of the links of a network.

1 Introduction

Network characteristics, such as loss rate, average delay, available bandwidth, are important to network design and performance evaluation. However, commercial interests prohibit ISPs to exchange this type of information with their competitors. Network tomography tends to obtain the characteristics by end-to-end measurement. The acquired characteristics can help us to identify network problems and find solutions for future networks.

Network tomography relies on a trigger-response scheme to discover network characteristics. It sends probe packets (called probes later) from a node or a number of nodes to an interested network with ongoing traffic, the probes head to a number of destinated receivers, via the network. When probes arrive at the receivers, they carry the information about the network. To determine the characteristics from observations, we normally select a probability model to describe the corresponding characteristics of a link with some or all parameters undetermined. Network tomography in this circumstance investigates the methods and methodologies to determine those parameters. Sending probes to a network is a method to have the parametric information available for identification. A number of methods, including EM and maximum likelihood estimators, have been proposed to carry out the statistical inference. All those methods either use the iterative approximating approach to estimate the characteristics or are involved in solving a set of high order polynomials [1], [2]. No matter which method is

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 473–482, 2004.

used, the time spent on the estimation increases sharply as the size of the network being estimated, which may restrict the technique to be used in practice. To overcome the problem, we need to search for other alternatives to speed up the inference [2]. In this paper, we propose a bottom-up approach to estimate loss rates. As other methods, the proposed method depends on the correlation between receivers to identify parameters. It is different to the previous ones in its inference method, it relies on the differences observed between some receivers connected to a link being estimated and the sibling brothers of those receivers to estimate the loss rate of the link. It starts from leaf links since the loss rate of a leaf link can be estimated directly from the observations of the receiver attached to the link and its sibling brothers. Once the loss rates of leaf links are determined, the proposed method moves one level up along the multicast tree to estimate the loss rates of those links that connect to leaf links, this process is continued until it reaches the source. The proposed method only needs simple arithmetic calculation to determine loss rates, which is equivalent to an analytical solution. The method is not a MLE, its estimation is slightly lower than that of a MLE, which can be remedied by a top-down approach.

The rest of the paper is organized as follows. In Section 2, we present the related work. We then introduce loss tomography and the principle used in statistical inference in Section 3. In Section 4, we detail the bottom up approach presented in this paper with some examples. Section 5 presents the results of the inference algorithm based on the data collected from a simulation platform built on *ns-2* [3]. The last section is devoted to concluding remark, it also contain our current and future work in line of measuring network performance.

2 Related Work

Network tomography has a number of components for loss, delay, and bandwidth, respectively. Each component has its unique name to distinguish itself from others. Loss tomography, as named, aims to find loss rates of links. It depends on sending probes to the receivers attached to the end-nodes and apply the correlation observed by the receivers to identify the loss rates of those links that form the multicast tree [4], [5], [6], [7] [2], [1]. Two methods are widely used to create correlation, i.e., multicast probes or unicast probes.

Statistical inference views each probe sent to receivers as a trial and what receivers observed as a sample of the trial. While, the sample obtained from receivers is incomplete with regards to the internal nodes because their states are not visible, statistical inference aims to uncover the loss rates of all links, including those that cannot be observed directly. So far, the methods proposed to discover the loss rates can be divided into two classes: classic statistics and Bayesian statistics, each class has its advantages and disadvantages.

Cáceres *et al* are the pioneer to use the multicast-based approach to create correlation and subsequently find loss rates [1], [8], [9]. Both simulation and experiment study on the Mbone show the feasibility and potential of this approach. The group also attempted to use multiple sources to cover a more general net-

work later [10]. This time, instead of the polynomial method, they used two methods, EM and MVWA, to infer the loss rates from observations obtained at destinated receivers and find the EM algorithm produces more accurate result than the other [10]. Harfoush *et al* and Coates *et al* independently proposed the use of the unicast-based approach to discover link-level characteristics [11], [12]. Their simulations confirm the feasibility of this method. Coates and Nowak also suggested to use EM algorithm to estimate the correlation between packet pairs for loss rates.

3 Loss Inference

The multicast tree used to send probes to receivers can be abstracted by a three-element tuple (V, E, Θ). The first two elements represent the nodes and links that have the same definitions as that in graph theory. While, $\Theta = \{\theta_1, ..., \theta_m\}$ is an m-element vector, each for a link that is the parameters to be determined by statistical inference. For instance, if assuming the losses occurred on a link are independent, the Bernoulli model is adopted and Θ denotes the loss rates directly. While a Gaussian model is used, Θ denotes both means and variances.

When a probe is multicasted from the source to its receivers, the probe must first reach the root of the multicast tree before it is delivered to the receivers. Taking the extra leg into account, a multicast tree is a bit different from a regular tree at its root that has only a single child. However, as a regular tree, a multicast tree can be defined recursively, i.e., each sub-multicast tree has a root that has only one child that connects a normal tree. As a regular tree, we assign a unique number to each link $(1, 2, 3, .., n)$ and a unique number to each node $(0, 1, 2, ..., n)$, the two sets of numbers map each other in the same way as a normal tree, e.g., link 1 connects node 1's parent (node 0) to node 1, link 2 connects node 2's parent to node 2, and so on. Figure 1 shows an example, apart from node 0 that is the root of the multcast tree, every node has only one input link that has the same number as the node.

Statistical inference is used here to estimate loss rates from observations, in particular for those links that cannot be observed directly. Each observation corresponds to a set of joint probabilities that lead to the observation. Given an observation and the multicast tree structure, one is able to construct the

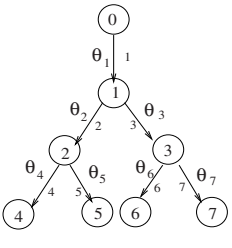


Fig. 1. Network structure

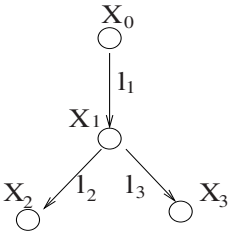


Fig. 2. A Simple Multicast Tree

corresponded joint probabilities. For a probe sent to a multicast tree, each node in the multicast tree has only two possible outcomes, observed or missed, let 1 denote the observed outcome, and 0 denote the other. In addition, let X denote a node, F_x denote the parent of X, the model used to describe the loss rates of the link connecting F_x to X is a conditional probability, i.e. $P(X = 0|F_x = 1)$.

With a large number of observations, statistical inference aims to identify the parameters occurred in the joint probabilities, if those parameters are identifiable. Thus, loss tomography is a parametric estimation with incomplete data in statistics. If the maximum likelihood estimate (MLE) is applied to determine the loss rates, it can be written in a log-likelihood format:

$$\arg\max_{\Theta} L(\Theta) = \arg\max_{\Theta} \sum_{r \in \Omega_R} n(r) \log P(r; \Theta) \tag{1}$$

where $n(r)$ is a function that counts the number of occurrences of observation r in the trials. A number of methods, e.g. neural net, Monte-Carlo, Expectation-Maximization (EM), have been proposed to solve Equation (1). Unfortunately, those methods all require a long execution time to find a feasible solution for a large network.

4 Bottom Up Approach

What we are concerned here is whether there are other alternatives to conduct the estimation, which are simple, efficient and accurate, in particular if we want to use it for network controls.

If assuming the losses occurred on two serially connected links are independent, i.e. spatial independent, and identical distributed (iid), when the multicast approach is used to create correlation among receivers, we can have a simple approach to estimate the loss rates of a network that takes a bottom-up approach to conduct its estimation. For the bottom approach, the loss rates of all leaf links can be estimated directly since their correlations are observable . Once, the loss rates of all leaf links have been identified, the proposed method moves one level up to estimate the loss rates of the links that are parents of the leaf links. In this level, each link plus the subtree connected to the link is regarded as a virtual link, then the loss rate of the virtual link can be estimated by the same method.

By knowing the loss rates of all leaf links, we are able to obtain the loss rate of the parent link from the loss rate of the virtual link. This process is continued until it reaches the source. The following three subsections are used to detail the proposed algorithm for leaf links, internal links and top link, respectively.

4.1 Leaf Link

After a series of trials, the proposed method starts to estimate the loss rates of those leaf links that have all their sibling brothers' observations available. Let X be one of the links, and let B_x denote the observations of its sibling brothers, which is a binary set. Each element in the set represents the state of a receiver for a trial, 1 means the receiver observed the probe, 0 means otherwise. If at least one of elements in B_x is not 0, we say $B_x \neq 0$ that implies the parent of X observed the probe. Since B_x is independent from X, the loss rate of link X can be derived:

$$P(X = 0|F_x = 1) = P(X = 0|F_x = 1, B_x \neq 0) = P(X = 0|B_x \neq 0) \quad (2)$$

Recall that $n(y)$ is a count function that records the number of y appeared in the trials. We can use $n(\cdot)$ to estimate $P(X = 0|B_x \neq 0)$,

$$P(X = 0|B_x \neq 0) = \frac{\sum_{B_x} n(X = 0, B_x \neq 0)}{\sum_{B_x} n(B_x \neq 0)} \quad (3)$$

Note that $n(B_x \neq 0) = n(X = 0, B_x \neq 0) + n(X = 1, B_x \neq 0)$. For example, to estimate the loss rate of link 4 of Figure 1, we have

$$P(X_4 = 0|X_2 = 1) = \frac{n(X_4 = 0, X_5 = 1)}{n(X_4 = 0, X_5 = 1) + n(X_4 = 1, X_5 = 1)}$$

which is identical to the formula derived by Cáceres *et al* from a high order polynomial [1], [8].

4.2 Internal Link

For an internal link, X, once the loss rates of all its children, which can be a set of leaf links, or a set of subtrees, or a combination of the previous two, have been estimated, the loss rate of the subtree rooted at node X can be estimated, which is equal to the sum of the products of the loss rates of those links that form *cuts* of the subtree. A cut is a group of links that can separate the subtree horizontally into two parts. For instance, for the subtree rooted at node 1 of Figure 1, there are four cuts, i.e. 1) link 2 and link 3; 2) link 3, link 4 and link 5; 3) link 2, link 6 and link 7; and 4) link 4, link 5, link 6 and link 7.

To identify all cuts of a multi-level subtree is a difficult task. However, as a recursive structure, the cuts of a tree can be identified from the links that connect its subtrees and the cuts identified from its subtrees. In the above example, there are 4 cuts, the first one consists of the links that connect to the

two subtrees rooted at node 2 and 3, respectively; for the other three cuts, one consists of link 3 and the cut of the subtree rooted at node 2, one consists of link 2 and the cut of the subtree rooted at node 3, the last consists of the cuts of the subtrees rooted at node 2 and 3. Nevertheless, identifying the cuts of a subtree is not important here, the important thing is how to obtain the combined loss rate created by the cuts, which in fact is the loss rate of the subtree. To avoid repeatly calculating the loss rates of the same subtrees. A general formula is developed. Let C_x denote the set of links that connect node X to its children, let $f_i(0)$ denote the loss rate of link i and let $f_i(1)$ be the loss rate of a subtree rooted at node i. Then, the loss rate rate of a subtree rooted at node X is equal to the sum of the follows:

- the product of the loss rates of those links that connect X to its children, $\prod_{i \in C_x} f_i(0)$
- the sum of the products obtained from the combination of the loss rates of some links connecting X to its children, denoted by SC_x and the loss rates of those subtrees in $C_x \setminus SC_x$, where SC_x can be empty.

If there are n subtrees connected to node X, there are $1 + \sum_{i=1}^{n} C_n^i = 2^n$ terms in the formula. Let g_x represent the loss rate of a subtree rooted at node X, then,

$$g_x = \sum_{i_1=0}^{1} .. \sum_{i_n=0}^{1} f_{s1}(i_1) f_{s2}(i_2) ... f_{sn}(i_n) \tag{4}$$

where n is the number of X's children, from $s1$ to sn. $f_x(\cdot)$ is determined by the following rules:

- For a leaf link, X, after its loss rate is estimated, we set $f_x(0) = \hat{P}(X = 0|Pa_x = 1)$ and $f_x(1) = 0$ since a leaf link does not connect to any subtree. Then, we can move one level up.
- For a non-leaf link, Y, after estimating $P(Y = 0|F_Y = 1)$, we set $f_y(0) = P(Y = 0|F_Y = 1)$ that is the loss probability of link y and $f_y(1) = g_y[1 - f_y(0)]$ that is the product of the pass rate of link Y and the loss rate of the subtree rooted at node Y. When the loss rates of all links on this level have been estimated, we move one level up to estimate the loss rates of those links in that level. The process is continued until all links have been estimated.

For example, the loss rate of the subtree rooted at node 1 of Figure 1 can be estimated by

$$g_1 = f_2(0)f_3(0) + f_2(1)f_3(0) + f_2(0)f_3(1) + f_2(1)f_3(1). \tag{5}$$

The four terms of RHS correspond to the four cuts previously listed. The bottom up approach ensures when it is estimating the loss rate of an internal link, X, the loss rates of of those links in the subtree rooted at X are available. Then, link X and the subtree rooted at node X can be regarded as a virtual link,

denoted as V_x. From the viewpoint of F_x, the parent of X, V_x is strongly related to B_{V_x}, the sibling brothers of V_x. The view of V_x for a probe is defined as:

$$V_x = \begin{cases} 1, \exists i,\ i \in R(X),\ y_i = 1 \\ 0, \forall i,\ i \in R(X),\ y_i = 0 \end{cases} \tag{6}$$

where $R(X)$ denotes those receivers attached to the subtree rooted at node X. Applying the same method used in the previous subsection, we have,

$$\hat{P}(V_x = 0 | F_{V_x} = 1) = \frac{\sum_{B_{V_x}} n(V_x = 0, B_{V_x} \neq 0)}{\sum_{B_{V_x}} n(B_{V_x} \neq 0)} \tag{7}$$

Since

$$P(V_x = 0 | F_{V_x} = 1) = g_x + (1 - g_x) P(X = 0 | F_x = 1)$$

where g_x is the loss rate of the subtree rooted at node X. Then, we have

$$\hat{P}(X = 0 | F_x = 1) = \frac{1}{1 - g_x} [P(V_x = 0 | F_x = 1) - g_x] \tag{8}$$

For example, if the multicast tree used to send probes to receivers is as shown in Figure 1, the loss rates for all leaf links, i.e. links 4 to 7 can be obtained by formula (2). Once the loss rates of all leaf links have been obtained, we can estimate the loss rates of link 2 and link 3. Using the equation (8), we have the estimate of the loss rate of link 2, which is

$$\begin{aligned} P(X_2 = 0 | X_1 = 1) &= \frac{P(V_2 = 0 | X_1 = 1) - g_2}{1 - g_2} \\ &= \frac{P(X_4 = 0 \wedge X_5 = 0 | X_6 = 1 \vee X_7 = 1) - P(X_4 = 0 | X_2 = 1) P(X_5 = 0 | X_2 = 1)}{1 - P(X_4 = 0 | X_2 = 1) P(X_5 = 0 | X_2 = 1)}. \end{aligned} \tag{9}$$

Since the terms of the right hand side (RHS) are either known or observable, the LHS is estimable. This principle is repeated applied from bottom-up to obtain the loss rates of all links.

In addition, before estimating the loss rates of a network after a series of trials, we need to have a pre-processing to eliminate those links whose loss rates cannot be estimated from the observations. We first delete all the leaf links that have not received any probes since their loss rates are inestimable. If all leaf links connected to the same parent node are deleted, the link connecting the grandparent to the parent node is also deleted since there is no observation that can assist us to estimate the loss rate of the link. In addition, if an internal node X has only one child C, we delete the node X and connect the parent of X, F_x, to C directly since $P(C = 0 | X = 1)$ and $P(X = 0 | F_x = 1)$ are not identifiable. The pre-processing aims to eliminate those subtrees, including end nodes, that have not observe any probe sent to them. The loss rates of those links connecting the subtree cannot be estimated because of this. This also ensures the above formulae are properly defined. For example, if receiver 4 in Figure 1 misses all probes, the loss rate of link 5, $\theta_5 = P(X_5 = 0 | X_2 = 1) = P(X_5 = 0 | X_4 = 1)$

is undefined. In the pre-processing, we replace the subtree with a new node. The new node's observation is equal to the combined observations of the sibling brothers of the trouble node, i.e. node 4. Then, node 2, 4 and 5 will be merged as a new one, X_{25}, and its observation is equal to the observation received by X_5. Then, $P(X_{25} = 0|X_1 = 1)$ can be estimated, but remember $P(X_{25} = 0|X_1 = 1) \neq P(X_2 = 0|X_1 = 1)$ and $P(X_{25} = 0|X_1 = 1) \neq P(X_5 = 0|X_2 = 1)$. The RHSs are inestimable in this situation.

5 Simulation Study

To demonstrate its effectiveness of the proposed method, we conducted a series tests on a simulation environment built on *ns2* that has 8 nodes connected by 7 links, named 1 to 7, into a tree structure, as shown in Figure 1. Link 1 had 3Mbps of bandwidth, 2ms of propagation delay; link 2 and 3 also had 3Mbps of bandwidth, but 10ms of propagation delay; the other 4 links had 1.5Mbps of bandwidth and 10ms propagation delay. All nodes has a FIFO queue and except node 1 has a queue with a limit of 20 packets, all other nodes can at most queue 10 packets at a time. The droptail policy is employed by all nodes to handle congestion, i.e. when a queue is full, newly arrived packets were dropped. Probe packets, 40 bytes each, were periodically multicasted from the root to the receivers attached to the leaf nodes. The background traffic consists of:

1. two TCP streams with window size=50 and packet size=1KB flow from node 0 to node 4 and 5, respectively;
2. four exponential distributed on-off UDP streams
   - one burst stream with burst period=400ms, idle period=300ms, bit_rate=1000k, and packet size=200B flows from node 0 to node 4;
   - one burst stream with burst period=300ms, idle period=300ms, bit_rate=800k, and packet size=200B flows from node 0 to node 5;
   - one burst stream with burst period=300ms, idle period=200ms, bit_rate=400k, and packet size=500B flows from node 1 to node 6;
   - one burst stream with burst period=200ms, idle period=200ms, bit_rate=400k, and packet size=500B flows from node 1 to node 7;
3. one FTP stream flows from node 0 to node 4 with window size=60 and packet size=600; and
4. three FTP streams flow from node 0 to node 5, 6 and 7, respectively, with window size=60, and packet size=800.

where the burst periods and idle periods yield exponential distribution, and the numbers provided above are the means of the corresponding exponential distribution. Except 2. that started at 0s and suspended at 50s, and resumed at 80s, all other streams started flow from 0s to 95s.

What we were interested in this study is to find out the packets loss rate at each link by end-to-end measurement. θ_i, $i \in \{1, .., 7\}$ in Figure 1 represents the loss rate of link i. A multicast agent is added on the root node (0) to multicast probe packets on a regular basis to the 4 leaf nodes. A sequence number is

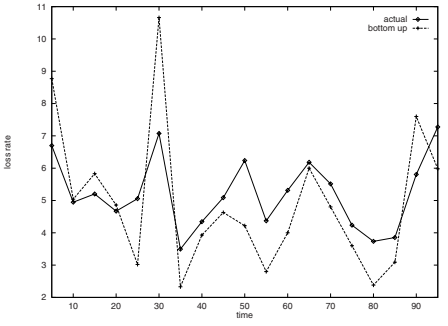


Fig. 3. Loss rate on link 1 with probe interval= 0.02s

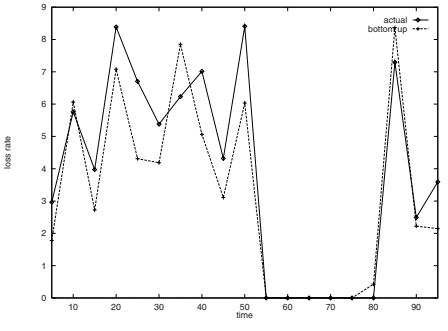


Fig. 4. Loss rate on link 4 with probe interval= 0.02s

attached to each probe packet, then based on the sequence number, a receiver can identify whether a probe packets is lost, if so, its position in the probe stream. During the experiments, we conducted an inference every 5 seconds based on the data collected at the four receivers, the simulator uses the same interval to collect the actual link-level data, packet sent and dropped, at every node. We call the data collected by the simulator actual result.

In the experiment, the inferred results on link 2, 3, 6 and 7 match the true results perfectly since these links were lightly loaded. Figures 3 and 4 shows the difference between inferred result and true result on link 1 and link 4, respectively. Although there are some differences between the inferred and actual results, the inferred results correctly show the loss trend of the background traffic, in particular, when stream 2 was suspended.

6 Conclusion

Network tomography depends on statistical inference to identify the information which cannot be observed directly. Maximum likelihood estimation is one of the most popular strategy used in inference. For a large network with hundred of links, using MLE to do inference processing can take considerable amount of

time since most available algorithms use the iterative approximation to search for a feasible solution in a complex solution space. Apart from that, the solution identified by a method may not be the global maximum since it may trap into a local optimum. To overcome the first problem, we in this paper present a simple bottom-up approach to estimate the loss rates of a network, that in principle applies the observed correlation between a link and its sibling brother to identify the loss rate of the link. The advantage of this approach relies on its simplicity, efficiency and consistency. In fact, the proposed approach is an analytical solution. Comparing with the MLE in simulations set up on ns-2, we find the proposed method achieves identical results as the MLE. The form or shape of the multicast tree used to send probes to receivers is another issue that requires further study, which is related to the network topology and identifiability that determines the number of receivers and their locations.

References

[1] Cáceres, R., Duffield, N., Horowitz, J., Towsley, D.: Multicast-based inference of network-internal loss characteristics. IEEE Trans. on Information Theory **45** (1999) 473, 474, 477
[2] Coates, M., Hero, A., Nowak, R., Yu, B.: Internet tomography. IEEE Signal Processing Magazine **19** (2002) 473, 474
[3] The network simulator 2. Technical report, (www.isi.edu/nsnam/ns2) 474
[4] Felix: Independent monitoring for network survivality. Technical report, (ftp://ftp.bellcore.com/pub/mwg/felix/index.html) 474
[5] Ipma: Internet performance measurement and analysis. Technical report, (http://www.merit.edu/ipma) 474
[6] Mahdavi, J., Paxson, V., Adams, A., Mathis, M.: Creating a scalable architecture for internet measurement. (In: INET'98) 474
[7] Surveyor. Technical report, (http://io.advanced.org/surveyor) 474
[8] Cáceres, R., Duffield, N., Moon, S., Towsley, D.: Inference of Internal Loss Rates in the MBone . In: IEEE/ISOC Global Internet'99. (1999) 474, 477
[9] Cáceres, R., Duffield, N., Moon, S., Towsley, D.: Inferring link-level performance from end-to-end multicast measurements. Technical report, University of Massachusetts (1999) 474
[10] Bu, T., Duffield, N., Presti, F., Towsley, D.: Network tomography on General Topologies. In: SIGCOMM 2002. (2002) 475
[11] Harfoush, K., Bestavros, A., Byers, J.: Robust identification of shared losses using end-to-end unicast probes. In: Technical Report BUCS-2000-013. (Boston University, 2000) 475
[12] Coates, M., Nowak, R.: Unicast network tomography using EM algorthms. Technical Report TR-0004, Rice University (2000) 475

On Generating Random Network Structures: Connected Graphs*

Alexey S. Rodionov[1] and Hyunseung Choo[2]

[1] Institute of Computational Mathematics and Mathematical Geophysics
Siberian Division of the Russian Academy of Science
Novosibirsk, Russia
+383-2-396211
alrod@rav.sscc.ru

[2] School of Information and Communication, Sungkyunkwan University
440-742, Suwon, Korea
+82-31-290-7145
choo@ece.skku.ac.kr

Abstract. In this paper we present the set of base algorithms for generating connected random graphs (RG). RG can be used for testing different algorithms on networks. The fast algorithms with proved properties are presented for random generation of connected graphs, sugraphs (subgraphs on the complete set of nodes) and others in conditions of given restrictions, such as limited node degree, given node degrees, different probabilities of edges existence etc. Special attention is given to generating graphs "similar to real networks." The algorithms are presented in the Pascal-like pseudo code.

1 Introduction

In this paper we present the set of base algorithms for generating connected Random Graphs (RGs). It is the sequel of our paper [1] in which we have discussed the generation of Random Trees (RT). RGs are widely used for testing different algorithms on networks [2, 3, 4, 5]. In fact, we can say that it is the only good model for the task as using different real network structures for algorithms testing is usually impossible. So the random graphs are widely used as a most appropriate model. In [6] some parameters of real networks are presented while in [7] some estimations are done based on the RG of a special kind. While RGs are widely used as a model, the task of their generation is almost unexplored. It seems that most researchers that use RGs as a model consider the task as an obvious job. Yet there is a lot of problems in the task of RGs generation, especially when complicated limitations are put on the graph structure. The usual requirements are to guarantee *uniformity* on the given space of graphs (to the renumbering) and *attainability* of all graphs from the given space. Uniformity

* This paper was partially supported by Brain Korea 21, University ITRC project and RFBR. Dr. H. Choo is the corresponding author.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 483–491, 2004.

is needed for simple estimation of mean values for the parameters and indexes under estimation. If we cannot guarantee the uniformity, then we must know a distribution and take it into account at estimation. Attainability is also important: if some of theoretically possible structures cannot be obtained by an algorithm then we cannot rely on the simulation results. Our approach always guarantees this property. In [1] we present base approach to RGs generation and a set of algorithms for random trees generation. Some of the generating algorithms were programmed for the application package Graph-ES, see [8, 9, 10]. Hereafter we discuss some basic approaches to the generating of connected RGs and present several effective algorithms. The rest of the paper is organized as follows. Section 2 is devoted to the base approaches, notations and concepts. In Section 3 the algorithm for generating connected RGs with some limitations. Section 4 is devoted to the generation of RGs "similar to real webs." Section 5 is a brief conclusion.

2 Base Notations and Approaches

Let us denote arbitrary non-oriented graph with N nodes and M edges and without multiple edges as $G(N, M)$. Several approaches can be used for the random graphs generation. First and most inefficient approach (but still widely used) is the *trial method*. A random graph with given numbers of nodes (and, possibly, edges) is generated and then its properties are checked to answer the given limitations. In our case it means at least that a graph ought to be connected one. Obviously the number of trials grows with the number of nodes quickly. Indeed, the total number of different graphs $G(N, N)$ (with all possible re-numbering of nodes) is

$$S = \left(\begin{array}{c} \frac{N(N-1)}{2} \\ N \end{array} \right) \frac{N(N-1)}{2}! = \frac{\left[\frac{N(N-1)}{2}! \right]^2}{N! \left[\frac{N(N-1)}{2} - N \right]!} , \qquad (1)$$

while the total number of all N-node cycles (with all possible re-numbering of nodes also) is only $N!^2$. But even this simplest approach can be realized with different efficiency.

Let us consider the example of generating a $G(N, M)$. The simplest way is to use the adjacency matrix (VV). Let VV be initially zeroed. In the cycle from 1 to M the pair of random numbers i and j is chosen from $[1, \ldots, N]$, $\quad i \neq j$ and if $VV_{ij} = 0$ then let $VV_{ij} = 1$ and $VV_{ji} = 1$, otherwise the choice is repeated. This simplest algorithm requires the following average number of choices:

$$N_c = 1 + n \left(\frac{1}{n-1} + \frac{1}{n-2} + \ldots + \frac{1}{n-M+1} \right), \qquad (2)$$

where $n = N(N-1)/2$.

Note that to choose a pair from the set {1,2,...,n} by the trial method requires $n/(n-1)$ attempts in average, thus the whole number of calls for random number generator in our case is in average

$$N_r = \frac{n}{n-1}\left[1 + n\left(\frac{1}{n-1} + \frac{1}{n-2} + \ldots + \frac{1}{n-M+1}\right)\right], \tag{3}$$

For $N = 10$, $M = 20$, for example, $N_r = 26.65$ and for $M = 25$ it is 36.69 already. Thus the sampling without repetition is preferable (see [1]).

Next base approach is the "sequential growth": each new element is added to a generated graph without violation the limitation and guaranteeing its properties. Once more there are two possibilities: to use trial or "smart" method on each step.

As "smart" method we will use the *method of admissible choice (MAC)*, that is the method, in which on each step the choice is done only from the set of elements that 1) keeps the graph in a given class and 2) do not allow to broke any of limitations. Before the examples will be discussed, let us denote:

X_i – set of generated graph's nodes on the i-th step;
E_i – set of generated graph's edges on the i-th step;
A_i – set of edges that are allowed to add to the generated graph on the i-th step;
In_i – set of edges that are to be added to A_i before the $i+1$-th step;
Ex_i – set of edges that are to be excluded from A_i before the $i+1$-th step;
v_i – i-th node (vertex) of a graph;
e_{ij} – edge that connects v_i and v_j;
$X(G)$ – set of nodes of the graph G;
$E(G)$ – set of edges of the graph G;
$C(N)$ – the complete graph with N nodes.

Thus, $A_{i+1} = A_i \setminus Ex_i \cup In_i$. Some limitations can lead to the empty A_{i+1}. If it is not due to impossibility of obtaining a graph with given properties at all (this ought to be checked before generating), then the rollback is heeded by one step. If e_{st} was the edge last chosen before deadlock then it transfers from A_i to Ex_i.

3 Generating Connected Graphs

This task is a good example of algorithm that needs to change the rules of In_i and Ex_i definition during the generating process.

Each connected graph has at least one spanning tree (one is if the graph is a tree itself), so the generating process for a connected graph consists from two parts: generating of spanning tree and allocation of remaining edges. The first task is analyzed in [1] in details. Let T be the spanning tree generated on the first stage. If we want no multi-edges and an edge e_{st} is chosen on the i-th step then on the second stage ($i \geq N$) we have the following rules for In_i and Ex_i (note that $A_{N-1} = E(C(N)) \setminus E(T)$):

$$Ex_i = \{e_{st}\} \tag{4}$$
$$In_i = \emptyset \tag{5}$$

If we need to generate connected graphs with some additional limitations, then a spanning tree ought to satisfy them, and these limitations ought to be taken into account at Ex_i definition. Let L be the set of limitations designated to a given graph space and that are of the kind $f_j(G) \leq B_j$ and $R(G, L)$ be the indicator function:

$$R(G,L) = \begin{cases} -1 \text{ if } G \text{ violates the limitations, } \exists j f_j(G) > B_j \\ 0 \quad \text{if } G \text{ is just on the border, } \exists j f_j(G) = B_j \\ 1 \quad \text{if } G \text{ is under the violation of the limitations.} \end{cases} \tag{6}$$

Then the general rule for MAC is:

1. If $R(G(X_i, E_i)) = -1$ and e_{st} is the edge selected on last step, then take this step back and remove e_{st} from A_i.
2. If $R(G(X_i, E_i)) = 0$, then edges which addition to a current graph obviously can violate the limitations are added to the Ex_i.
3. If $R(G(X_i, E_i)) = 0$ and e_{st} is the edge selected on last step, then $Ex_i = e_{st}$.

The efficiency of a generating algorithm highly depends on the violation checking and prediction. If the state of been on the limitation border is detected easy, then it is possible not to allow any violations at all. Thus, if node degree has achieved its upper limit then it is enough not to consider free edges connected to it any more. But, for example, checking the violation of limitations on a graph diameter beforehand for all edges from A_i will take more operations then checking it after the edge selection and, possibly, make rollback.

3.1 Generating Random Connected Graphs with Given Node Degrees

If generating RTs with prescribed node degrees is possible to make without deadlocks (see [1]), it is impossible for RGs with $M > N$ if multi-edges are impossible. For example, let us consider the case of $N = 4$, $M = 5$, $Degrees = (3, 3, 2, 2)$. On the first stage the random tree is generated in which node degrees are *limited* by given values and on the second stage additional edges are selected in such a way that to make strict correspondence with given degrees. In the Fig 1 the possible deadlock situation is presented: a) is the spanning tree generated on first stage, b) is the only right variant for the given degrees while c) shows the result of basic algorithm in the case of the multi-edges being allowed. It is clear that after selection of first $(1, 2)$-edge the rollback is needed.

Nevertheless, it is experimentally shown that the following scheme gives less rollback in average than general one. Let us obtain the random spanning tree on the first stage of the algorithm. We have $\Delta Deg_i = Degrees[i] - deg(v_i) \geq 0$. Let L be the number of nodes for which the inequality is strict. Then if

$$\sum_{j=4}^{N} \Delta Deg_i > L(L-1) \tag{7}$$

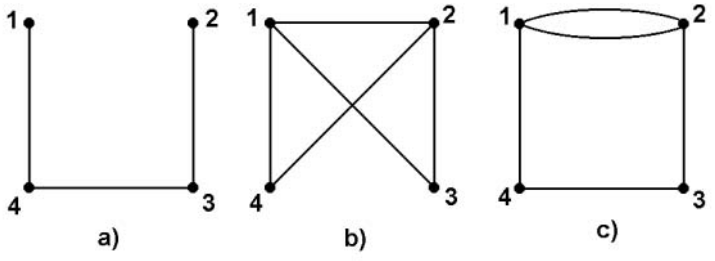


Fig. 1. Example of a situation that brings to the rollback

then to satisfy the given degrees the multi-edges are needed and our spanning tree is not suitable, repetition of spanning tree generation is needed.

Obviously it is better to check the condition 7 during the spanning tree generation.

3.2 Generating Connected k-Partite Graphs

k-partite graphs are widely used in different areas, for example they are used for solving the problem of channel assignment in radio communication networks.

Let us define the sets for MAC in this task. Let $B_i,\ i = 1, \ldots, k,\ \forall i \neq j,\ B_i \bigcap B_j = \emptyset$ be the sets of nodes that corresponds to different partitions of a k-partite graph. Then

$$A_1 = \{e_{ij} | i \in B_v,\ j \in B_u,\ v \neq u\} \tag{8}$$

and if on some i-th step the edge e_{st} was chosen then

$$In_i = \emptyset,\ Ex_i = \{e_{st}\}. \tag{9}$$

If the connected k-partite graph is needed then we first generate a random N-vertex tree using the following (f is a root, let $f \in B_u$):

$$A_1 = \{e_{nu},\ u \notin B_u\} \tag{10}$$

If on the i-th step the edge e_{st} is selected (t-th node is new) then

$$Ex_i = \{e_{ct} | v_c \in X_i\}, \tag{11}$$

$$In_i = \{e_{td} | v_d \in \bar{X}_i\} \backslash \{e_{uq} | \exists r (u \in B_r \& q \in B_r)\}. \tag{12}$$

On the second stage

$$A_{N-1} = E(C(N)) \setminus E(T) \backslash \{e_{uq} | \exists r (u \in B_r \& q \in B_r)\}, \tag{13}$$

and for $N - 1 < i \leq M$ we have again

$$In_i = \emptyset,\ Ex_i = \{e_{st}\}. \tag{14}$$

The limitations on node degrees can be taken into account similar to the previous case of connected $G(N, M)$.

4 Generating Graphs Similar to Real Nets

As it was stated in the Introduction, real networks have a lot of special features. In [2, 3, 4, 5, 6, 7] authors discuss different aspects of Internet, Intranet and other Webs structures. Based on their results and own experience in modeling different kind of nets (electric communication and wireless) we present the following most common web's properties.

1) connectivity;
2) "planarity" or "almost planarity" (if there are intersections of edges, their number is a little one);
3) nodes of a web are disposed in a coordinate plane;
4) non-uniformity of distribution of node's coordinates on a plane: nodes are grouped around of one or several centers with reduction of density with distance;
5) limitation on a minimum distance between nodes: it is obvious that in the communication networks too short distances are as improbable as too long one: it is hard to imaging the distance of several meters between two base stations in cellular network, for example;
6) limitation on maximum degree for the majority of nodes, or, more precisely, probability of availability of nodes with large number of incident edges is small. At the same time in large webs the existence of one or several nodes with a large degree is quite possible.

For particular classes of webs the additional properties and can be added and the quantitative values for indicated are defined.

4.1 Probability of Edge Existence

It is commonly appreciated that the probability of edge existence depends on its length (distance between nodes) but the kind of dependency is a matter for discussion. In [5], for example, the probability of the existence of an edge between nodes i and j it was chosen as proportional to the negative exponent of a distance ($e^{-d_{ij}}$). In the Fig. 2 we can see random trees obtained by the fast algorithm, proposed in [1] in case of random coordinates of the nodes. In the Fig. 2a) the probabilities were calculated with this proportion while in In the Fig. 2b) and c) the probabilities of edge existence are proportional to ($e^{-d_{ij}^{\alpha}}$), where α is 7.5 and 2 respectively. It is clear that no graph of real communication network can be similar to the first tree as it has too many intersections and long edges. It is clear that the second and third trees are more similar to the real communication network. In Fig. 3 in the next section the probabilities are proportional to this exponent with α equal 1.5 and 2 respectively. It is clear that $\alpha = 2$ is better choice.

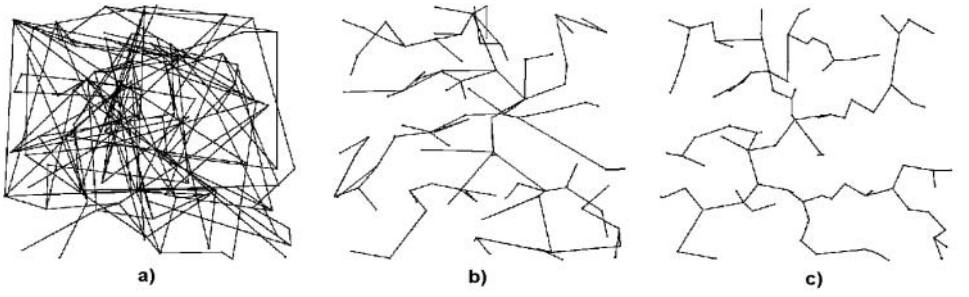


Fig. 2. Random trees in case of the edge existence probabilities proportional to $e^{-d_{ij}^{\alpha}}$, $\alpha = 1, 1.5, 2$

4.2 General Generation Scheme

In light of the considered properties the general arrangement of generation of a random web with given number of nodes and edges looks as follows.

1. For each local subnet the coordinates of a central node are defined. At that the limitation on a minimum distance between these nodes is superimposed. Coordinates are uniformly defined in the limits of given area (rectangle, circle, ellipse etc. in dependencies on the configuration of real area. It is possible to use the real shape of area, but it is more difficult for realization).
2. According to given (may be random) distribution of the common numbers of nodes between local subnets the coordinates of these nodes are defined. At this the use of two-dimensional normal probability distribution with independent coordinates and equal variances on axes is recommended . A degree of a spread (variance) for different subnets can be different. The re-testing is required at violation of the limitation on minimum distance between nodes or exit outside the boundaries of common area.
3. According to a given decreasing function from distance the matrix of probabilities of availability of edges (connections) between nodes is defined.
4. The random spanning tree is created by the described above algorithm for generating random trees with different probabilities of edges existence. The limitations on a node degree are taken into account.
5. The remaining edges of a web are chosen randomly according to given probabilities of existence and taking into account all the limitations.

The version is possible, at which the total number of edges is beforehand distributed on belonging to local subnets and connecting lines. In this case at first each subnet is created independently according to algorithm described above, then the random tree aggregating them is created with usage of free edges connecting nodes of different subnets and, at last the remaining connecting lines are selected from this set with taking all limitations put on a web into account, also.

In Fig. 3 and 4 the results of one-level and two-level graphs generation with the proposed scheme are presented. The nodes coordinates are the same, $N = 100$, $M = 156$, $Deg = 6$.

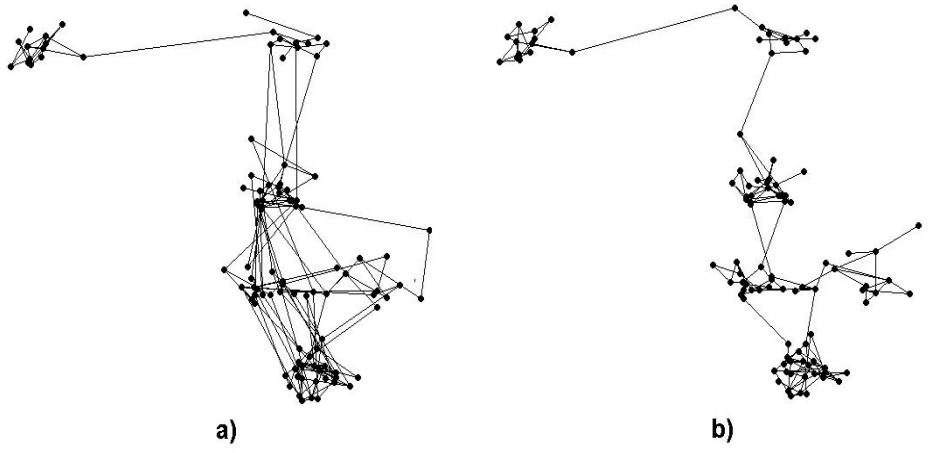


Fig. 3. One-level RG obtained by the proposed scheme. Probabilities are proportional to: a) $e^{-d^{3/2}}$, and b) e^{-d^2}

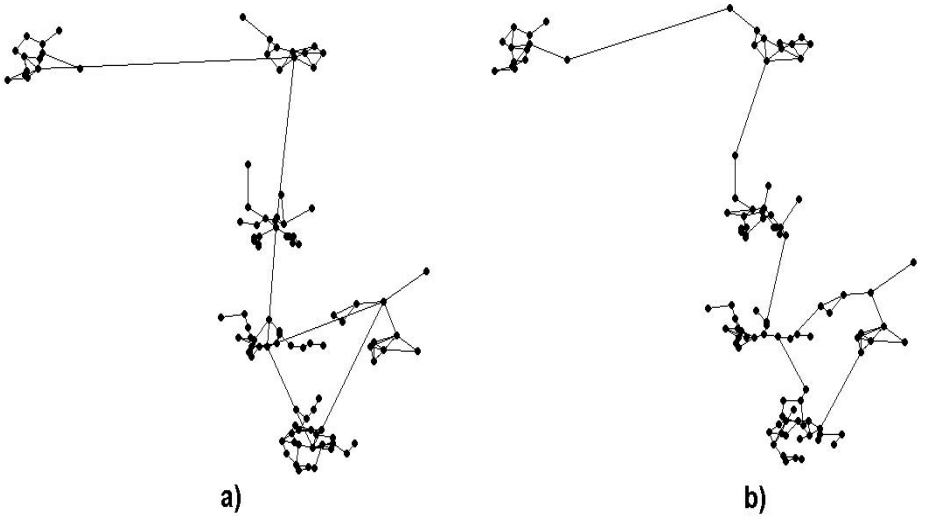


Fig. 4. Two-level RG obtained by the proposed scheme: a) with central nodes, and b) without central nodes

Note that the generation scheme can be easily updated to the case when subnets are of some specific kind, circular for example. In this case we use different limitations for different kinds of subnets.

5 Conclusion

We have presented general method algorithms for the generating random connected graphs with different properties. It is shown that the use of trial method is mostly ineffective. Meanwhile the proposed algorithms based on the method of admissible choice are proved to be effective in the terms of number of operations, and memory use. The task of generating graphs "similar to real nets" has been discussed and the list of most common properties of such graphs has been presented. Based on these properties One- and multi-level generating schemes are developed that proved to be quit appropriate for the task. The problem of an edge existence probabilities as function if their length have been discussed also and correspondent proposition is made. The presented examples show adaptability of our algorithms.

References

[1] A. S. Rodionov, and H. Choo "On Generating Random Network Structures: Trees," *Lecture Notes in Computer Science, Vol. 2658*, pp. 879-887, 2003. 483, 484, 485, 486, 488

[2] B. M. Waxman, "Routing of Multipoint Connections," *IEEE JSAC, Vol. 9*, pp. 1617-1622, 1993. 483, 488

[3] M. Doar, Multicast in the ATM environment. PhD thesis, Cambridge Univ., Computer Lab., September 1993. 483, 488

[4] M. Doar, "A Better Mode for Generating Test Retworks," *Proc. Global Telecommunication Conf. GLOBECOM'96*, pp. 86-95, 1996. 483, 488

[5] Chai-Keong Toh, "Performance Evaluation of Crossover Switch Discovery Algorithms for Wireless ATM LANs," *Proc. of INFOCOM'96*, pp. 1386-1387, 1993. 483, 488

[6] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an Internetwork," *Proc. of the INFOVCOM'96 Conf.*, pp. 594-602, 1996. 483, 488

[7] R. Kumar, P. Raghavan, S. Rajagopalan, D Sivakumar, A. Tomkins, and E Upfal, "Stochastic models for the Web graph," *Proc. 41st Annual Symposium on Foundations of Computer Science, 2000*, pp. 57-65, 2000. 483, 488

[8] A. S. Rodionov, L. N. Postnikova, and O. K. Rodionova, "Program Complex for Graph Generating in the GRAPH-ES Package," *Materials of the Conf. "Computers and System Analysis", Novosibirsk, 1288*, pp. 61-18, 1982. (in Russian) 484

[9] A. S. Rodionov, "Graph Generating in the GRAPH-ES Package", *Materials of the Conf. "Methods asd Programs for Solving Optimization Problens on Graphs and Nets," Novosibirsk, 1982, Part 1*, pp. 174-771, 1982. (in Russian) 484

[10] M. I. Netchepurenko, V. K. Pqpkov, S. M. Mainagashev, etc, *Algorithms and Programs for Solving Optimization Problems on Graphs and Networks*. Novosibirsk, Nauka P. A., 1990, 515 p. (in Russian) 484

Structures of Human Relations and User-Dynamics Revealed by Traffic Data

Masaki Aida[1], Keisuke Ishibashi[1], Hiroyoshi Miwa[2], and Chisa Takano[3]

[1] NTT Information Sharing Platform Laboratories, NTT Corporation,
3–9–11 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
aida.masaki@lab.ntt.co.jp
[2] Department of Informatics, School of Science and Technology
Kwansei Gakuin University
[3] Traffic Engineering Division, NTT Advanced Technology Corporation

Abstract. The number of customers of a service for Internet access from cellular phones in Japan has been explosively increasing for some time. We analyze the relation between the number of customers and the volume of traffic, with a view to finding clues to the structure of human relations among the very large set of potential customers of the service. The traffic data reveals that this structure is a scale-free network, and we calculate the exponent that governs the distribution of node degree in this network. The data also indicates that people who have many friends tend to subscribe to the service at an earlier stage. These results are useful for investigating various fields, including marketing strategies, the propagation of rumors, the spread of computer viruses, and so on.

1 Introduction

The number of customers of NTT DoCoMo's 'i-mode service' in Japan (for Internet access from cellular phones) has been explosively increasing and recently reached about 40,000,000, although the service was only introduced five years ago [1]. Statistics on i-mode's growth thus provide an interesting body of information on the behavior of large numbers of users. Complex networks such as structures of social relations do not have an engineered architecture; rather, they are self-organized by the actions of large numbers of individuals. The local interactions can lead to the nontrivial global phenomenon of a scale-free distribution of node degree [2], which in turn leads to a small-world property [3, 4]. In this paper, we analyze the relation between the number of i-mode customers and the volume of traffic, with a view to finding clues to the structure of human relations among the very large set of potential i-mode customers. The traffic data reveals that this structure is a scale-free network, and we calculate the exponent that governs the distribution of node degree in this network. The data also indicates that people who have more friends tend to subscribe to the i-mode service at an earlier stage.

If we consider each person as a node and each relation between two people as a link, we have a graphical model of human relations. If we can systematically

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 492–502, 2004.

characterize the structure of graphs thus derived, the characterization should be applicable to marketing strategies, the propagation of rumors and epidemic, and demand forecasting for telecommunications services, among others.

Networks of hyperlinks among web pages on the Internet and certain social networks have been reported to show small-world properties and act as scale-free networks. A scale-free network has a small number of 'hub' nodes, each of which has quite a lot of links. This feature acts to suppress increases in network diameter when the number of nodes increases. As a result, the average numbers of hops in the routes between all pairs of nodes are extremely small and information spreads with remarkable speed. The defining feature of a scale-free network is that the distribution of node degree obeys a power law, i.e. $n(k) \propto k^{-\gamma}$, where k and $n(k)$ denote the node degree (number of links) and the number of nodes of degree k, respectively, and γ is a positive constant. A wide variety of scale-free networks has been found, in both technological and social realms. In most cases, the relation $2.0 \leq \gamma \leq 3.4$ applies [5]. Ebel et al. [6] analyzed the logs of the e-mail server at Kiel University and produced a graph that represents the relationships among the e-mail accounts of the students. A link in the graph indicates the passage of at least one e-mail message between the corresponding pair of accounts. The graph in this case was a scale-free network with the slightly atypical γ value of 1.81. This result reflects human relations within a small community, in this case the set of people who use the university's e-mail server. The result is thus not applicable to people in general. Furthermore, since an e-mail log will almost certainly include records of multicast messages, i.e. messages sent to accounts on a mailing list, the passage of an e-mail message does not necessarily indicate a relationship between the owners of the corresponding pair of accounts, so the result does not solely reflect human relations. Abello et al. [7] and Aiello et al. [8] analyzed telephone calls on a certain day and produced a graph that represents the relationships among phone numbers. In this case, a link represents the setting up of a connection between the corresponding pair of numbers. This graph is a scale-free network with $\gamma = 2.1$. The data in this case is on a large number of non-specified people so the result should be generally applicable; however, a phone number often corresponds to a company or family rather than an individual, so the result does not reliably reflect human relations.

This paper is on our investigation of the relation between the number of i-mode customers and the volume of i-mode traffic. Simple analysis of this relation reveals some fundamental features of human relations for a large number of non-specified persons (the population within reach of the service). In addition, we clarify this population's dynamic behavior as a set of potential service subscribers.

The rest of this paper is organized as follows. In Sec. 2, we introduce our traffic data and point out the characteristics which make them desirable as a basis for analyzing human relations. In Sec. 3, we explain our assumptions and the framework of our investigation, and present the analytical and general form of the relation that characterizes i-mode e-mail usage. In sections 4–6, we assume various rules for the selection of new i-mode subscribers, and investigate

the patterns of human relations they reflect and the user dynamics we would expect if the rule were correct. The rules are random distribution, identical and independent distributions, and a deterministic rule. Section 7 reveals that the structure of human relations forms a scale-free network. Finally, we conclude our discussion with Sec. 8.

2 Traffic Data

Data on i-mode service traffic is of particular interest for the following reasons.

(a) The explosive growth of the service minimizes the effect on traffic of external factors such as changes in economic circumstances, family structure, etc.
(b) Since almost all cellular phone terminals are for personal use, the transfer of an e-mail message between two terminals unambiguously represents communication between the corresponding pair of customers.
(c) Almost all e-mail traffic in the service is one to one, so we can assume proportionality between the volume of e-mail traffic and the number of customer pairs exchanging messages.
(d) Sending an e-mail message is much cheaper than a voice communication, so external factors, e.g. the income of users, only have negligible effects on the traffic patterns.
(e) In the early stages of popularization of the i-mode service, the combination of few e-mail advertisements and little sensationalism to attract nuisance users meant that very little of the traffic was independent of relationships among people.
(f) The service was heavily advertised in the mass media. Information about the i-mode service was thus widely propagated within a short period and the intensity of the public campaign meant that propagation was independent of the topology of human relations.

The number of customers grew about three-fold, from 1,290,000 to 3,740,000, over the six months from Aug. 1999 to Jan. 2000 [1]. The relationship between the number of customers and the volume of i-mode web-service traffic in this period is shown in Fig. 1. Let the number of i-mode customers be m; the relationship is then written as

$$\text{(i-mode web traffic)} \propto m \ . \tag{1}$$

The most reasonable explanation for this is a stable frequency of web access per user. The reason for this is as follows: if users who have subscribed to the i-mode service at an earlier stage are heavier users than more recent subscribers, the volume of web traffic will not be proportional to m. The result thus implies that the usage characteristics of the i-mode service for the average customer were stable over this period. The number of i-mode customers and the number of i-mode messages in the same period is given in Fig. 1. The data follows this power law:

$$\text{(i-mode e-mail traffic)} \propto m^{1.55} \ . \tag{2}$$

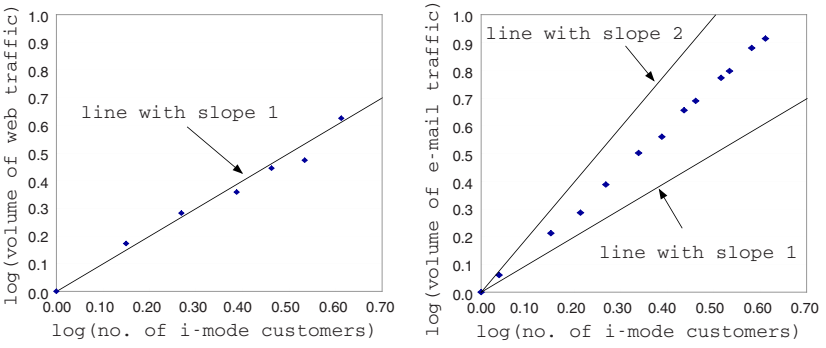


Fig. 1. A log-log plots of the number of i-mode customers versus the volume of web traffic, as the number of site-access operations, (left panel) and versus the volume of e-mail traffic, as the number of messages sent, (right panel). Traffic data is for August 1, 1999 to January 31, 2000

Therefore, the number of e-mail messages increases more quickly than the volume of web traffic. If the number of e-mail messages for an average customer is independent of the number of i-mode customers and is stable, it should be proportional to m. A value greater than m reflects growth over time in the number of partners with whom the average customer might want to communicate. On the other hand, if the average customer knows a certain constant proportion of customers (even if this proportion is small, e.g. 0.0001%), the volume of e-mail traffic should be proportional to m^2. The fact that the volume of e-mail traffic is proportional to $m^{1+\alpha}$, where $0 < \alpha < 1$, means that while the number of possible communication partners increases, the ratio of this number to the number of all customers falls. The parameter $\alpha = 0.55$ characterizes the rate of growth for e-mail traffic. It also tells us something about the strength of human relations. Hereafter, we investigate the characteristics of human relations that satisfy (2).

3 Notation and Assumptions

Let the set of people in Japan (i.e. the set of all potential customers of the i-mode service) be V, and the set of pairs of people who exchange information with each other be E. The number of elements in V is $|V| = n$. We define human relations as a graph $G(V, E)$. We assume that $G(V, E)$ is stationary. Next, we use a rule to select m elements from V and let the subset of these selected elements be V_m $(m \leq n)$. Let the subgraph induced by V_m from $G(V, E)$ be $G_m(V_m, E_m)$. That is, a node pair is connected by a link in $G_m(V_m, E_m)$ if and only if the corresponding node pair in $G(V, E)$ is connected by a link. Each element of V_m is an i-mode customer and human relations among all i-mode customers are represented by $G_m(V_m, E_m)$ (see Fig. 2). We assume that the number of links,

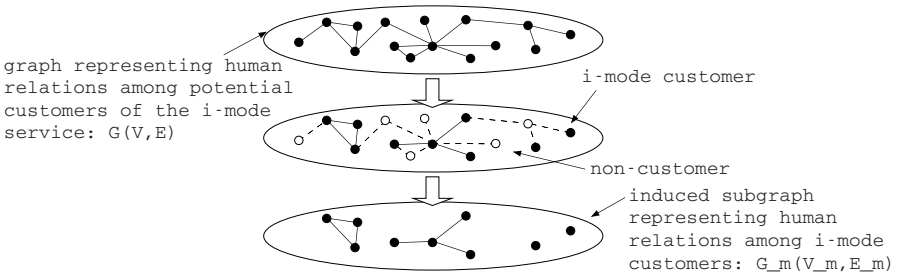


Fig. 2. The graph representing human relations, $G(V, E)$, and the subgraph $G_m(V_m, E_m)$ induced by i-mode customers

$|E_m|$, in the induced subgraph $G_m(V_m, E_m)$ is proportional to the volume of e-mail traffic (as the number of messages) flowing in the i-mode service. Thus, to clarify the origin of the behavior that leads to (2), i.e. to $0 < \alpha < 1$, we need to find the condition of human relations $G(V, E)$, and not just of the subset of relations $G_m(V_m, E_m)$, that satisfies

$$|E_m| \propto m^{1+\alpha} \ . \tag{3}$$

In the following sections, we assume various possible rules for selection of subscribers to the i-mode service, identify the rule which corresponds with our data, that is, the rule which characterizes user-participation dynamics, and show the structure of human relations $G(V, E)$ thus implied.

4 Random Selection of New i-mode Customers

We sort all elements of V into descending order of degree (number of links connected to the element) with respect to the graph $G(V, E)$, and let the degree of the i-th element be D_i $(i = 1, 2, \ldots, n)$. In cases where multiple nodes have the same degree, i is arbitrarily assigned. Similarly, all elements of V_m are sorted into descending order of degree with respect to the subgraph $G_m(V_m, E_m)$. We let the degree of the j-th element be d_j $(j = 1, 2, \ldots, m)$.

Next, let us consider continuous versions of the degree distributions D_i and d_j, denoted by $D(x)$ and $d(y)$, respectively, where $0 \leq x \leq n$ and $0 \leq y \leq m$. The distribution $D(x)$ of degree is a monotonically decreasing function of x, and we choose $D(x)$ that satisfies $\sum_{i=a}^{b} D_i = \int_{a-1}^{b} D(x)\,\mathrm{d}x$, where arbitrary parameters a and b are integers which satisfy $1 \leq a \leq b \leq n$. We choose the distribution $d(y)$ in a similar way.

Since the elements of V_m are chosen from V, a node, which has the j-th largest degree, d_j $(j \in V_m)$, in $G_m(V_m, E_m)$, will on average correspond to a node in the set with the i-th $(i = (n/m)\, j)$ largest degree in $G(V, E)$. In addition, since

the probability that the nodes connected to the node $i \in V$ in $G(V, E)$ are in V_m is $(m-1)/(n-1)$,

$$d_j \simeq \frac{m-1}{n-1} D_i \ , \tag{4}$$

on average. So, the expectation of the number of links in $G_m(V_m, E_m)$, that is, $F(m) := \mathrm{E}\left[|E_m|\right]$ are expressed as

$$\begin{aligned} F(m) &= \frac{1}{2} \sum_{j=1}^{m} d_j = \frac{1}{2} \int_0^m d(y)\, \mathrm{d}y \\ &\simeq \frac{1}{2} \frac{m-1}{n-1} \int_0^n D(x) \frac{\mathrm{d}y}{\mathrm{d}x}\, \mathrm{d}x = \frac{1}{2} \frac{m\,(m-1)}{n\,(n-1)} \int_0^n D(x)\, \mathrm{d}x \qquad (5) \\ &= O(m^2) \ . \qquad (6) \end{aligned}$$

From the law of large numbers, we have $F(m) = |E_m|$ for $m \gg 1$. We therefore obtain $|E_m| = O(m^2)$. Consequently, if we assume that new customers of i-mode are selected at random, the volume of e-mail traffic is independent of the structure of human relations and increases by $O(m^2)$. This does not agree with (3).

5 Selection of New i-mode Customers According to Identical and Independent Distributions

In a similar way to the previous section, we sort all elements of V into descending order of degree, and let the degree of the i-th element be D_i $(i = 1, 2, \ldots, n)$. We assign the probability p_i $(i = 1, 2, \ldots, n)$ to all nodes $i \in V$ and select m nodes from V according to the probability p_i. We assume that node selection is according to an identical and independent distribution (i.i.d.), that is, it negligibly affects the probability distribution p_i. This assumption indicates that $m \ll n$ and requires $p_i \ll 1$ for all $i \in V$.

Let the set of m nodes selected from V be V_m, and the subgraph of $G(V, E)$ induced by V_m be $G_m(V_m, E_m)$. All elements of V_m are again sorted into descending order of degree in the subgraph $G_m(V_m, E_m)$; and let the degree of the j-th element be d_j $(j = 1, 2, \ldots, m)$.

As we did with the random-selection case, let us consider the continuous versions of the distributions of degree D_i and d_j as $D(x)$ and $d(y)$, and of probability p_i as $p(x)$, where $0 \le x \le n$ and $0 \le y \le m$. If $p_i > 0$ for all i $(i = 1, 2, \ldots, n)$, we can choose the density $p(x)$, which satisfies $p(x) > 0$ for arbitrary x $(0 \le x \le n)$ and $\sum_{i=a}^{b} p_i = \int_{a-1}^{b} p(x)\, \mathrm{d}x$, where a and b are arbitrary integers that satisfy $1 \le a \ \le b \le n$.

Let us consider the distribution function that corresponds to the density function $p(x)$, $P(x) := \int_0^x p(s)\, \mathrm{d}s$. Since $p(x) > 0$, there exists an inverse of the distribution function, $P^{-1}(u)$, for all $u \in [0, 1]$. Consider $\{x_1, x_2, \ldots, x_m\}$, which is a sequence of the points selected by probability density function $p(x)$ from $[0, 1]$ and sorted into ascending order. If we apply the distribution function P to

transform the points $\{x_1, x_2, \ldots, x_m\}$, the points in $\{P(x_1), P(x_2), \ldots, P(x_m)\}$ are uniformly distributed on $[0, 1]$. So, the node corresponding to a node $j \in V_m$ is, on average, $x = P^{-1}(j/m)$. Therefore, we can express the relationship between $D(x)$ and $d(y)$ as

$$\int_0^m d(y)\,\mathrm{d}y \simeq c(m) \int_0^m D(P^{-1}(y/m))\,\mathrm{d}y \ , \tag{7}$$

where $c(m)$ is a function of m. Incidentally, in the case in the previous section, $c(m) = (m-1)(n-1)$, and $p(x)$ is a constant.

Then, the expectation of the number of links in $G_m(V_m, E_m)$, $F(m) := \mathrm{E}\,[|E_m|]$ is expressed by

$$F(m) = \frac{1}{2} \sum_{j=1}^{m} d_j \simeq \frac{1}{2}\, c(m) \int_0^n D(x)\, \frac{\mathrm{d}y}{\mathrm{d}x}\, \mathrm{d}x \ . \tag{8}$$

Since $x = P^{-1}(y/m)$, we can obtain $\mathrm{d}y/\mathrm{d}x = m\,p(x)$. Thus,

$$F(m) = \frac{m}{2}\, c(m) \int_0^n D(x)\, p(x)\, \mathrm{d}x \ . \tag{9}$$

If the probability density $p(x)$ is independent of the topology of the original graph $G(V, E)$, we can determine $c(m) = (m-1)/(n-1)$, and get

$$F(m) = \frac{1}{2}\, \frac{m\,(m-1)}{(n-1)} \sum_{i=1}^{n} D_i\, p_i \ . \tag{10}$$

Comparison with (5) shows us that (10) is equivalent to selecting nodes at random for the graph with degree distribution $D'_i = n\,D_i\,p_i$. In particular, if nodes are selected at random, that is, $p(x) = 1/n$, (10) becomes identical with (5).

In popularization of the i-mode service, advertising in the mass media had far more power than did word of mouth. So, we can assume that the popularization process was independent of the topology of human relations. The number of subscribers among the average user's friends and acquaintances is thus proportional to the number of i-mode subscribers, so we can state that

$$c(m) \propto m \ . \tag{11}$$

Thus, we have

$$F(m) \simeq O(m^2) \times \int_0^n D(x)\, p(x)\, \mathrm{d}x \ . \tag{12}$$

Since, in order to realize $|V_m| = O(m^{1+\alpha})$, $F(m)$ has to satisfy $F(m) = O(m^{1+\alpha})$. So, we have

$$\sum_{i=1}^{n} D_i\, p_i = O(m^{\alpha-1}) \ . \tag{13}$$

Since D_i is independent of m, (13) means that p_i depends on m. This result contradicts the assumption of i.i.d.

6 The Structure of Human Relations and the Rule for Subscription to the i-mode Service

The previous section demonstrated that the probability of selecting a given node changes after each node selection. Since setting a new probability for every node selection is complex, we adopt a deterministic approach.

We sort all elements of V into descending order of degree (number of links connected to the element), and let the degree of the i-th element be D_i $(i = 1, 2, \ldots, n)$. In cases where multiple nodes have the same degree, i is arbitrarily assigned. Similarly, all elements of V_m are sorted into descending order of degree in the subgraph $G_m(V_m, E_m)$. We let the degree of the j-th element be d_j $(j = 1, 2, \ldots, m)$.

On the other hand, we sort those elements of V that have been selected for V_m into their order of selection, and let the degree within V of the k-th element be $D_k^{(s)}$ $(k = 1, 2, \ldots, n)$. All elements of V_m are sorted into the same order. Let the degree within V_m of the h-th element be $d_h^{(s)}$ $(h = 1, 2, \ldots, m)$. While the g-th element, $g \in V_m$, corresponds to $g \in V$, $d_g^{(s)}$ is very rarely the same as $D_g^{(s)}$.

If we select m elements from V, the degree within $G_m(V_m, E_m)$ of the selected elements can be written as

$$\sum_{k=1}^{m} d_k^{(s)} \simeq c(m) \sum_{k=1}^{m} D_k^{(s)} \quad , \tag{14}$$

where $c(m)$ is a function of m. The number of links in $G_m(V_m, E_m)$ is then written as

$$|E_m| = \frac{1}{2} \sum_{k=1}^{m} d_k^{(s)} \simeq \frac{1}{2} c(m) \sum_{k=1}^{m} D_k^{(s)} \quad . \tag{15}$$

As was earlier stated, advertising in the mass media had far more power as a popularizer of the i-mode service than did word of mouth, from which we obtained (11). Then, from (3), we have

$$\sum_{k=1}^{m} D_k^{(s)} \propto m^{\alpha} \quad . \tag{16}$$

Since (16) is valid for all m, we have $D_k^{(s)} \propto k^{\alpha - 1}$. The above results are rephrased below.

- (Result A) The elements of V_m tend to be selected from V in descending order of degree in $G(V, E)$. In other words, a person who has many friends tends to subscribe to the i-mode service at an earlier stage.
- (Result B) The distribution of degree among the elements of V obeys Zipf's law [9]. That is, for the degree D_i of the i-th element of V (the indices are

determined in descending order of degree)

$$D_i \propto i^{-\beta} \tag{17}$$

holds, where $\beta > 0$ is a constant and $\beta = 1 - \alpha$.

Conversely, Results A and B along with $c(m) \propto m$ lead us to (3).

7 The Structure of Human Relations as a Scale-Free Network

We consider the distribution of degree, $n(k)$, for a $G(V, E)$ the elements of which satisfy Result B. Let the slope of Zipf's law in the log-log plane be $-\beta$. We consider the degree of the x-th element, $Z(x)$. For constants $a > 0$ and $c_1 > 0$, the degrees of the x-th and ax-th elements are written as $Z(x) = c_1/x^{\beta}$, and $Z(ax) = c_1/(ax)^{\beta}$. Since the slopes at both $(x, Z(x))$ and $(ax, Z(ax))$ are $-\beta$ on a log-log scale, let us consider two triangles which are congruent in the log-log plane (see Fig. 3). The lengths of the bases of the two triangles are related to $n(Z(x))$ and $n(Z(ax))$. More specifically, the ratio of the two lengths in the linear plane is equal to $n(Z(x)) : n(Z(ax))$. Thus, we have $n(Z(x)) : n(Z(ax)) = 1 : a$. Consequently, for a constant $c_2 > 0$, $n(Z(x)) = c_2$ and $n(Z(ax)) = c_2\, a$. We can plot the points $(Z(x), n(Z(x)))$ and $(Z(ax), n(Z(ax)))$ and then derive the slope of the line connecting the two points on the log-log plane as

$$\gamma = \frac{\log n(Z(ax)) - \log n(Z(x))}{\log Z(x) - \log Z(ax)} = \frac{1}{\beta} \ . \tag{18}$$

The above equation, (18), is independent of a. Thus, the distribution of degree is obtained as $n(k) \propto n^{-\gamma}$, where $\gamma = 1/\beta$. This relation is called Lotka's law [9];

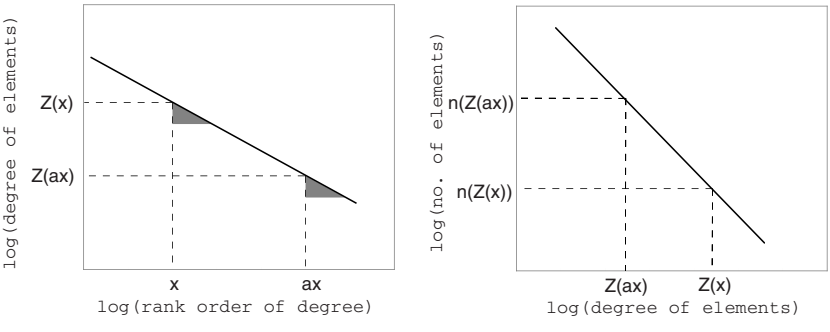


Fig. 3. Two points extracted from data that satisfies Zipf's law (left panel) and plotted on a Lotka-type graph (right panel)

since it is a consequence of Zipf's law, the two laws accompany each other (e.g. in Internet access [10]). From the above discussion and Result B, we get the following result:

– (Result C) The graph that represents human relations is a scale-free network, with distribution of degree described by $n(k) \propto n^{-\gamma}$ and

$$\gamma = \frac{1}{1-\alpha} \quad . \tag{19}$$

By using the experimental value $\alpha = 0.55$, we get $n(k) \propto n^{-2.22}$.

8 Concluding Remarks

Results A–C were derived on the assumption that $c(m) \propto m$, where m is the number of subscribers and $c(m)$ is, for the average user, the ratio of the number of friends who are subscribers to the i-mode service to his/her total number of friends. This is based on the assumption that the process of popularization is independent of the topology of human relations, which is in turn based on the fact that advertisements for i-mode in the mass media have been very much more powerful as popularizers of the service than the diffusion of information by word of mouth. However, in cases where the diffusion of information by word of mouth plays a non-negligible role, $c(m) \neq O(m)$. Our purpose in this paper has been to describe the characteristics of human relations revealed by data on the i-mode service. Results B and C are valid as descriptions of human relations and both will be useful for investigating other fields including marketing strategies, the propagation of rumors, and the spread of epidemics.

In addition to information on the structure of human relations, we also obtained Result A, which concerns the dynamic behavior of customers. This result is applicable as a description of how a money spinner emerges. We expect that combining knowledge of the three results will lead to efficient marketing strategies.

References

[1] DoCoMo Net, Usage of i-mode (in Japanese), http://www.nttdocomo.co.jp/p_s/imode/. 492, 494
[2] A.-L. Barabási and R. Albert, Emergence of scaling in random networks, Science, vol. 286, pp. 509–512, 1999. 492
[3] D. J. Watts and S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature (London), vol. 393, pp. 440–442, 1998. 492
[4] D. J. Watts, Small-Worlds, Princeton University Press, Princeton, New Jersey, 1999. 492
[5] R. Albert and A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys., vol. 74, no. 47, 2002. 493
[6] H. Ebel, L.-I. Mielsch, and S. Bornholdt, Scale-free topology of e-mail networks, Phys. Rev. E, vol. E66, 035103(R), 2002. 493

[7] J. Abello, P. M. Pardalos, and M. G. C. Resende, On maximum clique problems in very large graphs, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 50, pp. 119–130. 493
[8] W. Aiello, F. Chung, and L. Lu, A random graph model for massive graphs, 32nd ACM Symposium on the Theory of Computing, pp. 171–180. 493
[9] R. Rousseau and S. Rousseau, Informetric distributions: A tutorial review, Canadian Journal of Information and Library Science, vol. 18, no. 2, pp.51–63, 1993. 499, 500
[10] M. Aida, N. Takahashi, and T. Abe, A proposal of dual Zipfian model describing http access trends and its application to address cache design, IEICE Transactions on Communications, vol. E81-B, no. 7, pp. 1475–1485, 1998. 501

Parallel Fair Round Robin Scheduling in WDM Packet Switching Networks

Deming Liu[1], Yann-Hang Lee[1], and Yoonmee Doh[2]

[1] Department of Computer Science and Engineering, Arizona State University
Tempe, AZ 85287, USA
{dmliu,yhlee}@asu.edu
[2] Information and Communications University
Munji-dong, Yuseong-gu, Daejeon, 305-714, South Korea
ydoh@icu.ac.kr

Abstract. Assuming data traffic of fixed-length cells, we propose a frame-oriented scheduling discipline, PFRR (parallel fair round robin), for WDM optical switches by applying pfair (proportionate fairness) scheduling so that deterministic communication performance is guaranteed. Bandwidth reservation for an active session is performed by holding a number of cell slots for the session in a two-dimension frame, which is transferred iteratively over the multiple channels in an optical fiber. To determine the transmission order of cells in a frame, pfair scheduling is used so that the cells belonging to a session are distributed over the frame as uniformly as possible. Through the analysis by network calculus and by network simulator, it is shown that PFRR possesses tight delay bounds and lenient buffer requirements. Also, with a minor modification to PFRR, a new service discipline called MPFRR is proposed that can be used as traffic regulator to support sessions with jitter requirements.

1 Introduction

Bringing WDM (wavelength division multiplexing)to packet switching networks results in new constraints on packet scheduling algorithms, especially in the cases of real-time applications in that, at a switch, multiple sessions compete for multiple physical channels working parallel through an optical fiber. Thus, not only must a scheduling algorithm determine the order of packet transmission, but also select the channels that packets should be routed. The lack of packet scheduling algorithms supporting hard real-time services over WDM optical networks probably results from the intractability of most multi-resource scheduling problems since the two classes of problems are almost equivalent.

Even in switching networks without WDM, adding parallel links is more cost-efficient than using a high-bandwidth single link to increase the bandwidth of an existing link between two switch nodes [3]. We use *session constraint* to refer to the requirement that packets from a session cannot be transferred on multiple physical channels concurrently. Anderson et al. suggested two reasons for which session constraint is preferred [3]. (i) Session constraint avoids the necessity to

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 503–513, 2004.

include a sequence number in each of the parallel transferred packets and sort these packets at the receiving switch. (ii) Session constraint is compatible with currently defined standards such as ATM. We give the third advantage of session constraint from the viewpoint of switch implementation. If parallel transfer of packets of a session is allowed, high rate buffer, difficult for implementation, is required for the session so that parallel arriving packets from the session can be put into the corresponding queue. Since we can regard a wavelength channel inside a WDM optical fiber as a processor, and a session as a task, multiprocessor scheduling algorithms are applicable for packet scheduling in WDM optical networks [4].

To provide a fair access to resources in multiprocessor system and QoS service guarantee, the concept of pfair (proportionate fairness) scheduling can be applied. Basing on slotted time and assuming that all input parameters, including task period, execution time, and release time, are integers, Baruah et al. [5][6] proposed pfair scheduling which can not only guarantee task deadlines to be respected, but also make tasks execute at steady progressive rates. Making use of pfair scheduling, we will propose a frame-based service discipline, PFRR (parallel fair round robin), which extends the result of single channel switching in [2] to the parallel switching scenario. Similar to EDF-RR (earliest deadline first round robin) algorithm in [2], PFRR takes the advantage of low on-line computational complexity by assuming that setting up a frame is a infrequent event, and possesses tight delay bounds and lenient buffer requirements. The rest of this paper is organized as follows. Section 2 describes the PFRR service discipline. Section 3 gives frame algorithms and their computational complexity. In Section 4, delay bound and buffer requirement are analyzed by applying network calculus. Section 5 includes some simulation results. In Section 6, some application-related issues are addressed. Finally conclusions are given in Section 7.

2 Parallel Fair Round Robin Scheduling

To describe PFRR, we need some definitions of pfair scheduling in the parallel packet switching domain, similar to what Baruah et al. used in the multiprocessor scheduling domain [5].

- Scheduling decisions occur at integral values of time, numbered from 1. The real interval between time t-1 and t (including t-1 and excluding t) will be referred as slot t, $t \in \mathbf{N}$, where $\mathbf{N}$ denotes {1, 2, ...}. Time instant 0 is the beginning of a backlog period during which some sessions are backlogged.
- We consider an instance Φ of the fair sharing problem with m channels of identical bandwidth and n backlogged sessions over a frame over L time slots. Without losing generality, we assume $n \geq m$. Specific sessions will be denoted by identifiers x and y, which range over T, the set of all sessions. A time slot is referred to a time interval in the general meaning. Without confusion, we also use time slot (or slot) of a frame to refer to a column of the frame and cell slot to an element in the frame. Thus, a time slot contains

m cell slots. Since there are m parallel channels, a frame has two dimensions, $m \times L$, and contains mL cell slots.

- Data traffic consists of fixed-length cells. The amount of traffic that a cell carries is 1 unit. The amount of traffic that is transferred in a time slot for a channel is a cell.
- A session x has a reservation demand e_x defined as the number of cell slots in the frame that are held for transferring cells from session x. We assume that $0 < e_x \leq L$ and define the weight (or utilization) w_x of session x as $w_x = e_x/L$. Note that $0 < w_x \leq 1$ and w_x is a rational number. Without losing generality, we confine our investigation to the case that $\sum_{x \in T} w_x = m$. If $\sum_{x \in T} w_x < m$, we can add one or several dummy sessions to make $\sum_{x \in T} w_x = m$. Dummy sessions are always idle.

With respect to instance Φ of the resource sharing problem, let $earliest(x, i)$ ($latest(x, i)$) denote the earliest (latest) slot during which the ith cell of session x gets serviced.

$$earliest(x, i) = \lfloor (i-1)/w_x \rfloor + 1. \tag{1}$$

$$latest(x, i) = \lceil i/w_x \rceil \tag{2}$$

A schedule is pfair if and only if the ith cell of session x gets serviced in $[earliest(x, i), latest(x, i)]$ for all $i \in \mathbf{N}$. Note that, for a pfair schedule, $earliest(x, i) < latest(x, i)$, $x \in T$, $i \in \mathbf{N}$. Furthermore, $earliest(x, i+1)$ - $latest(x, i)$ is either 0 or 1. In other words, there is at most one slot in which both the ith cell and the $(i+1)$th cell of x can get serviced. It is obvious that if we can guarantee pfair over a frame, then pfair over infinite time can be guaranteed by repeating the pfair frame every L slots.

PFRR Discipline

a. Forming a pfair $m \times L$ frame F respecting both session constraints and allocation constraints by using the algorithms in Section 3 so that each row of F corresponds to a channel and each element of F represents a cell slot assigned to a session.
b. If all sessions are backlogged, the cells are transferred frame by frame through the m channels such that each frame is identical to frame F.
c. If not all sessions are backlogged in slot t, two cases are addressed as follows.
   c.1. If the m sessions in a time slot t of F that are receiving service are all idle, then time slot t is skipped such that the sessions in time slot $t+1$ of F get serviced immediately.
   c.2. Let $R_{idle}(t)$ denote the set of idle channels in a time slot t of F and $T^*_{busy}(t)$ the set of backlogged sessions that do not get serviced in t. Let $q = min\{|R_{idle}(t)|, |T^*_{busy}(t)|\}$. If some of the m sessions in the time slot t of F are idle, select q sessions from $T^*_{busy}(t)$ to fill q idle channels in $R_{idle}(t)$. Then the cells in time slot t are transferred except for the ones that violate allocation constraints.

With the pfair scheduling in a frame, if all sessions are backlogged, the number of cells scheduled for a session x of instance Φ between time 0 and an integral time instant t is either $\lfloor w_x t \rfloor$ or $\lceil w_x t \rceil$ by the definition of pfair [5]. Thus step b of PFRR guarantees deterministic performance of the scheduling. Step c of PFRR tries to make the scheduler work-conserving. Let $S_x(a, b)$ be the number of cells transferred in the interval $[a, b]$ for session x under PFRR. The following theorem gives the minimal communication capacity guaranteed for each session.

Theorem 1. *In any busy interval of session* x *scheduled with PFRR, we always have*

$$S_x(0, t_2) - S_x(0, t_1) > w_x(t_2 - t_1) - 2 \tag{3}$$

where the busy interval of session x *begins at time 0 and,* t_1 *and* t_2 *are any two time instants in the busy interval with* $t_2 \geq t_1$.

3 Pfair Frame Scheduling

Baruah et al. gave an on-line pfair scheduling algorithm PD (pseudo-deadline) with running time of $O(\min\{m \log n,\ n\})$ per time slot for periodic tasks over identical processors [7] in which deadlines are compared and ties are broken in constant time by inspecting 4 different parameters. Basing on Baruah et al's work, Anderson et al. reduced the number of tie-breaking inspections to 2 in the algorithm PD^2 [8][11]. Applying Anderson's multiprocessor algorithm, we can obtain a pfair frame for instance Φ of cell scheduling problem. Related definitions are given as follows.

- A session with weight less than 1/2 is called a light session; a session with weight at least 1/2 is called a heavy session.
- Each cell to be scheduled in the frame is attached a pseudo-release and a pseudo-deadline. Let $c(x, i)$ denotes the ith cell of session x, $i \in N$ and $r(c(x, i))$ ($d(c(x, i))$) the pseudo-release (pseudo-deadline), then,

$$r(c(x, i)) = \lfloor (i - 1)/w_x \rfloor + 1. \tag{4}$$

$$d(c(x, i)) = \lceil i/w_x \rceil. \tag{5}$$

  The interval $[r(c(x, i)),\ d(c(x, i))]$ is called the *window* of $c(x, i)$ denoted by win(c(x, i)). The size of win($c(x, i)$) (the number of time slots over which win($c(x, i)$) spans) is denoted by $|\text{win}(c(x, i))|$.
- According to the definition of pfair, $r(c(x, i + 1))$ is either $d(c(x, i))$ or $d(c(x, i)) + 1$. Then $b(c(x, i))$ is defined to distinguish these two possibilities. $b(c(x, i)) = 1$ if $r(c(x, i + 1)) = d(c(x, i))$, $b(c(x, i)) = 0$, otherwise.
- Consider a sequence $c(x, i), \ldots, c(x, j)$ of cells of a heavy session x such that $|\text{win}(c(x, k))| = 2 \cap b(c(x, k)) = 1$ for all $i < k \leq j$ and either $|\text{win}(c(x, j+1))| = 3$ or $|\text{win}(c(x, j+1))| = 2 \cap b(c(x, j+1)) = 0$. Then we define $d(c(x, j))$ to be the group deadline for the group of cells $c(x, i), \ldots, c(x, j)$. We also define

that cells belonging to light session have group deadline 0. Accordingly, cells of heavy sessions always have larger group deadline than the cells of light sessions. Every cell $c(x, l)$ has a group deadline that is denoted by $D(c(x, l))$.

The pfair frame algorithm is a priority-based algorithm. Session x's priority at time slot t of the frame is defined to be $(d(c(x, ix)), b(c(x, ix)), D(c(x, ix)))$, where i_x is the number such that i_x-1 cells already are scheduled for session x before time slot t. Session priorities are ordered according to the principle that $(d', b', D') \leq (d, b, D)$ if and only if

$$(d < d') \cup ((d = d') \cap (b > b')) \cup ((d = d') \cap (b = b') \cap (D \geq D')) \qquad (6)$$

At each time slot t, the m sessions that have the highest priorities are selected (ties are broken arbitrarily) so that exactly one cell from each of the m sessions is assigned to time slot t.

Anderson et al [8] proved that the aforementioned algorithm gives a pfair schedule. Since there are L time slots in the frame and the algorithm has the time complexity of $O(\min\{m\log n,\ n\})$ per time slot, then the time complexity of the frame algorithm is $O(\min\{mL\log n,\ nL\})$ per frame. Even though getting a fair frame takes time as much as $O(\min\{mL\log n,\ nL\})$, a fair frame can be used repeatedly once it is determined. Therefore from the round robin property of PFRR, the online operation is just a table lookup if we ignore the overhead of checking whether sessions are idle. Determining a pfair frame is only needed when there are sessions that start up or terminate. These operations are regarded as infrequent events in practice.

4 Delay Bound and Buffer Requirement Analysis

In this section, we will use network calculus [9][10], a mathematic analysis tool for networks, to obtain delay bounds and buffer requirements of PFRR for both single-node and multiple-node cases. Also an intuitive explanation is given for the analysis results.

To analyze delay bounds and buffer requirements, traffic models must be established to specify session traffic characteristics such as average rate and burstiness. A bursty traffic model of (σ, ρ) is one of them [1]. A session traffic flow is said to satisfy (σ, ρ) model if there are at most $\sigma + \rho t$ units of traffic during any time interval t. σ and ρ denote the burstiness and average rate of the traffic respectively. For example, traffic flow coming from the traffic regulator of leaky bucket satisfies (σ, ρ) model. According to the definition of *arrival curve*, the statement that a traffic flow satisfies (σ, ρ) model is equivalent to that the traffic flow is constrained by arrival curve $\sigma + \rho t$. In this paper we assume session traffic has arrival curve $\sigma + \rho t$ and traffic unit is fixed-length cell. In the following text we do not distinguish server and switch by the convention of network calculus literature. The results in this section are obtained by applying Theorem 1 and network calculus. Please refer to [9][10] for the concepts and results of network calculus.

Lemma 1. *If session x passes through a PFRR server, the PFRR server offers the service curve* $w_x t - 2$ *for session x.*

In the single-node case in which a session traffic flow goes through a PFRR server, we have Theorems 2 and 3 addressing the delay bound and the buffer requirement respectively.

Theorem 2. *If the session x traffic flow is constrained by arrival curve* $\sigma_x + w_x t$*, the delay it experiences passing through a PFRR server is not more than* $(\sigma_x + 2)/w_x$ *time slots.*

Theorem 3. *If the session x traffic flow constrained by arrival curve* $\sigma_x + w_x t$ *passes through a PFRR server without buffer overflow, the buffer size that the server needs for session x is not more than* σ_x *+2 cells.*

Now we consider the multiple-node case in which a session traffic flow traverses multiple PFRR servers. We define the *minimum weight server* for session x is the PFRR server which has the minimum weight among the sequence of PFRR servers that session x traverses. Lemma 2, Theorems 4 and 5 as follows are based on the assumption that session x passes through a number of PFRR servers without any buffer overflow and the session x traffic flow is constrained by arrival curve $w_x^* t + \sigma_x$, where $w_x^*$is x's weight at its minimum weight server.

Lemma 2. *The output flow from the kth sever for session x is constrained by arrival curve* $w_x^* t + \rho_x + 2k$*.*

Theorem 4. *The delay that the traffic flow of session x experiences from the source to the kth server is not more than* $(\sigma_x + 2k)/w_x^*$ *time slots.*

Theorem 5. *The buffer size needed by the kth server for session x is not more than* $(\sigma_x + 2k)$ *cells.*

Apparently if we ignore the computation for judging whether a session buffer is empty or not, PFRR has the computational complexity of O(1). On the other hand, it is necessary to have buffer empty checking taken into account from the point of view of implementation. The simple way to avoid this overhead is to prevent the empty cell slots of the idle sessions from being occupied by other backlogged sessions. But this strategy would waste the expensive resource, bandwidth. In the more efficient method, those empty cell slots can be used to transfer traffic from backlogged best-effort sessions that do not reserve any cell slot in the current time slot. If there do not exist best-effort sessions, a set of sessions with no reservation on the current time slot can be assumed to be nonempty by some heuristic approaches. Then if some of the found sessions are really nonempty, they will use those empty cell slots. Otherwise these empty slots will be discarded. No mater the empty slots are employed or not, the worst-case performance of real-time sessions will not be impaired.

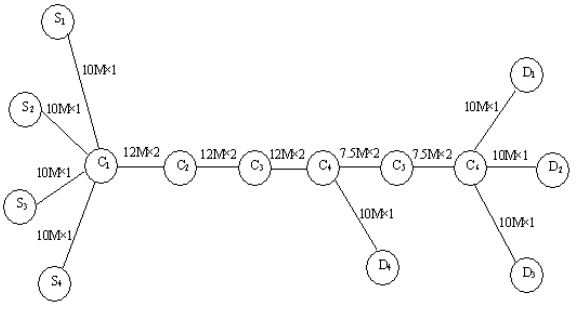


Fig. 1. The network topology for the simulation

5 Simulation Results

In this section, we will study PFRR's performance by simulation. We will compare PFRR with another frame-based service discipline for multichannel switching that we call PWRR (parallel weighted round robin).

Given the cell scheduling instance Φ, an $m \times L$ frame is constructed in the following way according to PWRR. The n sessions are scheduled one by one such that the frame is traversed row by row in the top-down direction. $w_x L$ cell slots for a session x is assigned to the current available row from left to right starting at the first available cell slot if the current row contains enough empty cell slots to accommodate $w_x L$ cells. Otherwise, the current row will be filled to full by session x and the rest cells of session x will be scheduled to the next row starting form the leftmost cell slot. Since a session can occupy at most L cell slots in the frame, two cells from the same session cannot be assigned to one column. This guarantees that no session can use more than one channel at one time slot. Except that PWRR is different from PFRR in forming the frame, the scheduling strategy of PWRR is the same as PFRR, i.e., steps b and c of PFRR.

The network topology for the simulation is shown in Fig. 1, where there are 2 channels in some links. The number of wavelength channels in each link connecting two core nodes is 2; the number of channels in each link connecting a core node and an end node is 1. In the following simulation, there are 4 sessions established in the network, denoted by source-destination pairs, S_1-D_1, S_2-D_2, S_3-D_3 and S_4-D_4. From Fig. 1, links C_iC_{i+1} ($i = 1, 2, \ldots, 5$) are multiplexed by more than one session. The reservation demands allocated to multiple sessions through the multiplex links and frame sizes of the multiplex links are given in Table 1. These multiplex links can be scheduled according to PFRR and PWRR. In the simulation as follows, exponential distributed traffic consisting of a sequence of cells is generated at the source node and destined to the destination node for each of the 4 sessions. End-to-end delay from source to destination and queue length at link C_1C_2 for session S_1-D_1 are observed and compared for PFRR and PWRR.

Table 1. Reservation demands of sessions and frame sizes for multiplex links

	C_1C_2	C_2C_3	C_3C_4	C_4C_5	C_5C_6
S_1-D_1	25	25	25	25	25
S_2-D_2	34	34	34	34	34
S_3-D_3	15	15	15	15	15
S_4-D_4	46	46	46	46	46
Frame size	2×60	2×60	2×60	2×60	2×60

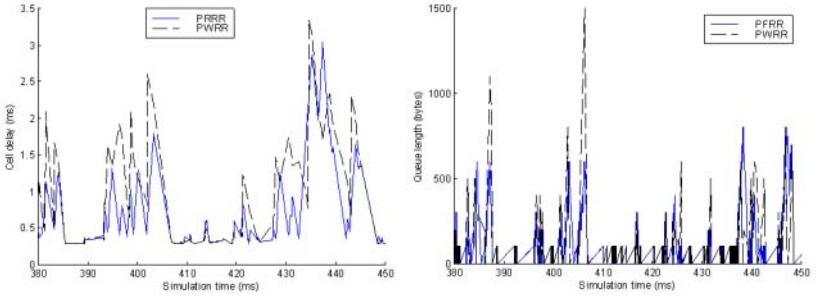


Fig. 2. Cell delay curves and queue length curves

In the simulation, we assume a cell has the length of 100 bytes. The same simulation loading with the same traffic is performed twice so that each time one of PFRR and PWRR is applied to all multiplex links. The observed cell delay and queue length of session S_1-D_1 for PFRR and PWFQ are shown in Fig. 2. In the figure, we see that PWRR exhibits larger fluctuation of packet latency and queue length than PFRR, thus inferrior to PFRR in terms of fairness.

6 Application of PFRR

In applying PFRR, rescheduling a frame is needed only when there are sessions to be established, cancelled or updated, which happen infrequently from the perspective of users. The associated overhead can be ignored since a new transmission frame can be computed in parallel to the current transmission, and is swapped at the next frame boundary.

It is worthy to consider the application of PFRR in the message communications of a distributed real-time system. Consecutive messages are sent from a source application to a corresponding destination application. Assume that each message consisting of P cells is with a deadline requirement, and will be routed through k switches to reach its destination. To utilize the proposed PFRR scheme, a session of proper reservation demands and the required buffers must be established to facilitate the message communication. Let us keep the same

assumptions as in Theorems 4 and 5. We define the session delay for a packet as the interval between the packet head entering the source node and the packet completely leaving the destination node. If a packet of session x can be broken into P cells, the packet's delay bound along the k PFRR servers that session x travels through can be expressed as $(\sigma_x + P - 1 + 2k)/w_x^*$.

In other words, bandwidth reservation for session x in a server is performed by reserving e_x cell slots in the corresponding frame. The criteria of how to determine the frame length L can be based on the facts that (i) L should not be too smaller, otherwise we cannot guarantee that sufficient granularity for allocating bandwidth, and (ii) the computation of rescheduling a frame increases as L increases. Thus large frame length may introduce long session setting-up or updating time.

Besides meeting deadline requirements, the other concern in real-time systems is message jitter. A session may require that the message delivery jitter be minimized. This can be accomplished if the message cells only use the scheduled cell slots regardless any empty cell slots in a frame. We call a session with this requirement a *nonwork-conserving session* and other sessions *work-conserving sessions*. Now let's consider a scenario of existing mixed types of sessions, in which some sessions only need to meet their deadline requirements, i.e., work-conserving sessions, and others are nonwork-conserving sessions with additional jitter constraints. Then the question comes up whether we can let a set of backlogged sessions share the empty cell slots in the frame and the other set of backlogged sessions keep their original scheduled cell slots such that all sessions' delay bound and buffer requirement cannot be deteriorated. We introduce an MPFRR (modified PFRR) discipline intended to address this requirement.

MPFRR Discipline

a. Same as step a of PFRR.
b. Same as step b of PFRR.
c. If not all sessions are backlogged in time slot t, we let $R_{idle}(t)$ denote the set of idle channels in time slot t of F and $T^*_{busy}(t)$ be the set of backlogged sessions that do not get serviced in time slot t excluding the nonwork-conserving sessions. Select $\min\{|R_{idle(t)}|, |T^*_{busy}(t)|\}$ sessions from to fill $\mathrm{mmin}\{|R_{idle(t)}|, |T^*_{busy}(t)|\}$ idle channels in $R_{idle}(t)$. Then the cells in time slot t are transferred except for the cells that violate their allocation constraints.

Lemma 3. *In any busy interval of a nonwork-conserving session x with MPFRR, we always have*

$$w_x(t_2 - t_1) - 2 < S_x(0, t_2) - S_x(0, t_1) < w_x(t_2 - t_1) + 2 \tag{7}$$

where the busy interval of session i begins at time 0 and, t_1 and t_2 are any two time instants in the busy interval with $t_1 \leq t_2$.

MPFRR is a hybrid of work-conserving and non-work-conserving policies. Since the cells of nonwork-conserving sessions are always transferred relatively

at the same positions, the performances of these sessions are kept unchanged no matter how many idle slots there are in the frame. On the other hand, delay bound and buffer requirement of work-conserving sessions are still guaranteed when some sessions are idle. This is due to the fact that the cell slots allocated to backlogged work-conserving sessions in the frame are definitely used by themselves no matter there are any idle sessions or not. If there are some empty cell slots in the frame in the case of existing idle sessions, more cells may be transferred for work-conserving sessions in these cell slots

7 Conclusions

In this paper, we proposed an applicable packet-scheduling algorithm for real-time WDM parallel switching networks. The algorithm is a round-robin based discipline with fixed-length cell being the basic unit for transmission and scheduling. Bandwidth reservation for a session at a switch is carried out through reserving a number of cell slots in a two-dimension frame. The pfair algorithms are used to determine cell slot assignment in the frame with or without allocation constraints considered so that cell slots of a session are distributed as uniformly as possible in the frame. Through the analysis by network calculus and by network simulator, it is shown that PFRR takes the advantage of low on-line computational complexity, and possesses tight delay bounds and lenient buffer requirements. In addition, a modified version called MPFRR can be used as traffic regulator to support sessions with jitter requirements.

References

[1] Hui Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proceeding of the IEEE*, Vol. 83, No. 10, Oct. 1995, pp. 1374-1396.

[2] Deming Liu and Yann-Hang Lee, "An efficient scheduling discipline for packet switching networks using earliest deadline first round robin," *The 12th International Conference on Computer Communicaitons and Networks*, Oct. 2003.

[3] J. Anderson et al., "Parallel switching in connection-oriented networks," *Proceedings of the 20th IEEE Real-Time Systems Symposium*, 1999, pp. 200-209.

[4] Laura E. Jackson and George N. Rouskas, "Deterministic preemptive scheduling of real-time tasks," *Computer*, vol. 35, issue 5, May 2002, pp. 72-79.

[5] S. Baruah, N. Cohen, C. G. Plaxton, and D. Varvel, "Proportionate progress: A notation of fairness in resource allocation," *Algorithmica*, vol. 15, no. 6, 1996, pp. 600-625.

[6] S. Baruah et al., "Fairness in periodic real-time scheduling," *Proceedings of 16th Annual IEEE Real-Time Systems Symposium*, December, 1995, pp. 200-209.

[7] S. Baruah, J. Gehrke, C. G. Plaxton, and I. Stoica, "Fast scheduling of periodic tasks on multiple resources," *Proceedings of the 9th International Parallel Processing Symposium*, April 1996, pp. 280-288.

[8] J. Anderson and A. Srinivasan, "A new look at pfair priorities," Technical report, Dept. of Computer Science, Univ. of North Carolina, 1999.

[9] Jean-Yves Le Boudec, "Application of network calculus to guaranteed service networks," *IEEE Transactions on Information Theory*, Vol. 44, No. 3, May 1998, pp. 1087-1096.

[10] Jean-Yves Le Boudec and Patrick Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, Springer, 2001.

[11] J. Anderson and A. Srinivasan, "Mixed pfair/Erfair scheduling of asynchronous periodic tasks," *Proceedings of the 13th Euromicro Conference on Real-Time Systems*, June 2001, pp. 76-85.

Post-dialing Delay of Multimedia Sessions in 3G Mobile Networks

Sung J. Yi[1], Krzysztof Pawlikowski[1], Harsha Sirisena[2], and Prasan De Silva[3]

[1] Department of Computer Science, University of Canterbury
Christchurch, New Zealand
{sjy17,krys}@cosc.canterbury.ac.nz
[2] Department of Electrical and Computer Engineering, University of Canterbury
Christchurch, New Zealand
h.sirisena@elec.canterbury.ac.nz
[3] Mobile Network Planning, Telecom New Zealand
Wellington, New Zealand
prasan.desilva@telecom.co.nz

Abstract. Session Initiation Protocol (SIP) is an application layer protocol that controls multimedia sessions of Third Generation Mobile Networks. A major concern with SIP in 3G networks is ensuring the efficient use of the bandwidth constraints inherent in the wireless medium. Because of the significant size of SIP messages, 3G terminals can suffer from substantial set-up delays. IETF has proposed a general solution to this problem, named SigComp. This paper investigates the performance of SigComp, in particular how much time saving the scheme is capable of offering. A novel scheme that can replace SIP for increasing time savings and reducing the load of the SigComp layer is proposed. The scheme, referred to as Binary Hybrid SIP (BHSIP), is based on binary sequences and plain text. This paper evaluates the performance of the scheme, and compares it with normal SIP. The paper provides experimental results obtained when comparing the performance of SigComp/SIP versus SigComp/BHSIP by using an emulated network environment, and discusses the benefits of using BHSIP.

1 Introduction

Numerous protocols have been proposed for carrying various forms of real-time multimedia sessions for such data as voice, video and text messages. The Session Initiation Protocol (SIP) works in concert with these protocols by enabling Internet endpoints to discover one another and to agree on a characterization of a session they would like to share. SIP enables the creation of an infrastructure of network hosts to which user agents can send registrations, invitations to sessions, and other requests. SIP is an agile, general-purpose tool for creating, modifying, and terminating sessions. It works independently of underlying transport protocols and without dependency on the type of session that is being established [1].

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 514–523, 2004.

The third generation (3G) of mobile network systems is seen as the technology that will bring full scope of the Internet services to the cellular worlds. It is claimed that 3G will provide ubiquitous access to all the successful services provided by the Internet. In a significant move to merge Internet technologies, Third Generation Partnership Project (3GPP) and Third Generation Partnership Project 2 (3GPP2) adopted the Internet based session control protocol SIP as their multimedia session control protocol [7][8]. SIP and SDP (Session Description Protocol [2] carried within SIP) were designed for broadband links. However, the amount of wireless bandwidth that 3G networks are capable to provide is still several orders of magnitude behind. Consequently, the managing of post-dialing delays becomes a significant issue. In order to reduce the substantial delay due to the size of SIP messages, Internet Engineering Task Force (IETF) has proposed novel scheme based on dynamic compression. IETF named the scheme SigComp. It is a solution for reducing the session set up time by compressing the messages that need to be exchanged via narrow bandwidth channels. This paper investigates the performance of SIP using the compression scheme proposed by IETF, SigComp [3], for reducing the session set-up delays, and proposes a novel mechanism to further reduce these delays.

The rest of this paper is organized as follows. Firstly, we briefly explain the 3G specific SIP call flow between User Equipment (UE) and the Proxy-CSCF as defined in [6]. Secondly, SigComp proposed by IETF is discussed. We also introduce a novel scheme that redefines the structure of SIP messages in order to reduce the size of the message and the load to the SigComp layer. Next, we discuss the assumed experiment environment and present our results. This paper concludes with a summary of the main results.

2 SIP Signaling in IP Multimedia Subsystem

3GPP defines three types of Call/Session Control Functions (CSCF) residing within the IP Multimedia Subsystem (IMS), which control all the multimedia sessions of 3G network [5][14]. CSCFs are regarded as SIP proxies and/or redirect servers, according to their roles in the particular circumstances. Proxy CSCF (P-CSCF) is the point of contact of the terminal to the network and it is regarded as an outbound proxy. Serving CSCF (S-CSCF) provides service to the user. When a terminal REGISTERs, it is associated with an S-CSCF, which provides the terminal with services that the user is subscribed to. Both incoming and outgoing sessions will traverse the S-CSCF associated with the terminal. This way, assigned S-CSCF can provide services on both types of sessions. The role of Interrogating CSCF (I-CSCF) is to find the proper S-CSCF for the particular user. For incoming sessions, the I-CSCF is the point of contact within the provider's network; it receives requests for a user within its domain and routes them to the proper S-CSCF. Most of the security measures for incoming data toward the network are to be dealt with this node. For outgoing sessions, the I-CSCF receives requests from the user and routes them to the associated S-CSCF. I-CSCF is also regarded as a stateful SIP proxy server. The roles of

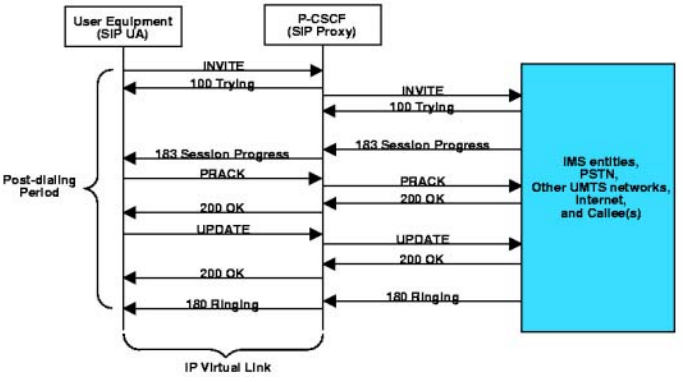


Fig. 1. Typical SIP flow during a session initiation

CSCFs in session setup phase are discussed in more detail in [5], [11] and [12]. During message exchanges between CSCFs, the network must follow appropriate resource reservation mechanism to ensure the session meets Quality of Service (QoS) parameter specified by users. The IETF's Common Open Policy Service (COPS) is a key part of this process.

Figure 1 shows how SIP messages traverse from UE to 3G network. The figure is deliberately abbreviated to emphasize the post-dialing period. The duration between the first INVITE message and the *180 Ringing* response is defined as post-dialing period, and the time it takes is called post-dialing delay [10] or, most commonly, dial-to-ring delay. It is the time elapsed between the time instant when user finishes dialing and the time instant when caller hears ringing response from the terminal. ITU-T Recommendation Q.543 specifies sets of values for the tolerable delay in this period [15]. The recommendations are originally intended for ISDN based networks. Though mobile networks involve complex signaling related to mobility, 2.5G networks have matured to the point where operators can target sub- 1–2 second post-dialing delays. Therefore, significant post-dialing delay of 3G sessions will not be tolerated by users. As in [6][12], multimedia session establishment requires a lot of message exchanges among several inter-networked entities.

3 SigComp Signaling

Many application protocols used for multimedia communications are text-based and engineered for broadband links. SIP and SDP are the typical examples. The sizes of SIP messages including SDP vary from a few hundred bytes up to few thousand. To avoid potentially long transmission delay, 3GPP and 3GPP2 had to adopt SigComp scheme, see e.g. [3], [7] and [8]. SigComp is a solution for compressing messages generated by application protocols, such as SIP and SDP. SigComp, in logical terms, resides between the application layer and the

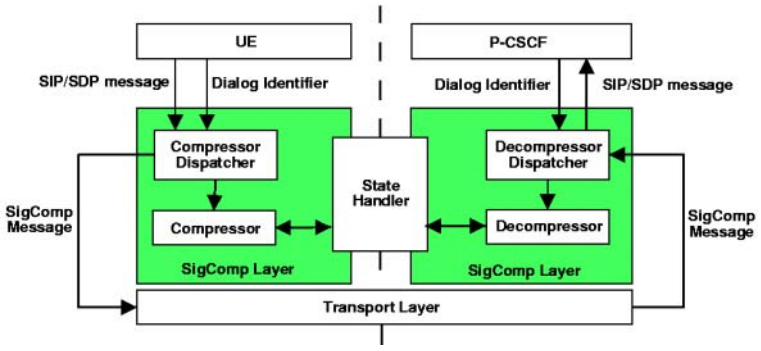


Fig. 2. Structure of SigComp layer in 3G

transport layer (see Fig 2). That is, application layer messages get compressed by SigComp layer and then passed to the transport layer.

In mobile networks, the scarce bandwidth of the radio link has always been the problematic. Hence, the SIP message exchange between UE and the P-CSCF must utilize all the possible bandwidth assigned. Network entities on the wire-line portion do not need to employ the SigComp layer, because they have no bandwidth shortages.

The results from [13] confirm that the compression ratio reaches its maximum, if compressor and decompressor perform dynamic updates on the predefined dictionaries. As shown in Fig 2, SigComp layer uses state handler and SIP dialog identifier to uniquely compress and decompress the messages. The state handler reassembles dictionary for given dialog value, hence achieving maximum compression ratio after a few message exchanges. The compression dictionary defined in [4] corresponds to the initial state of SigComp layer.

4 Binary Hybrid SIP

The size of SIP messages has always been problematic when they are carried over narrow bandwidth links. SIP can adopt compact forms of header field names, such as *i* for *Call-ID* header, etc. However, it only applies to limited number of headers, and does not reduce the overall message size significantly. Binary Hybrid SIP is an approach that redefines SIP in order to reduce post-dialing delay by reducing the size of messages without sacrificing the performance and also reducing the load of SigComp layer.

Table 1 shows simple examples of BHSIP message lines and corresponding SIP lines. The numbers in hexadecimal form represent the binary values of the message. BHSIP can be viewed as partial translation of SIP messages with binary sequences instead of plain text.

As for SIP, BHSIP messages always start with a request or response line. Instead of including the phrase SIP/2.0 to identify SIP messages, it reserves the first byte to represent the type of message and the next byte to be padded with

Table 1. BHSIP examples

Request line	
SIP	*INVITE tel:+1-212-555-2222 SIP/2.0*
BHSIP	*0x01 oxff tel:+1-212-555-2222*

Response line	
SIP	*SIP/2.0 200 OK*
BHSIP	*0xc8 oxff OK*

Header field	
SIP	*Cseq: 127 INVITE*
BHSIP	*0x10 0x00fe 0x01*
SIP	*Call-ID: cb03a0s09a2sdfglkj490333*
BHSIP	*0x08 cb03a0s09a2sdfglkj490333*

ones. The start-line, each message-header line, and the empty line are terminated by a carriage-return line-feed sequence (CRLF), such as in SIP [1], BHSIP follows the same notation. Every header field starts with binary identifier of 1 byte, that replaces text based name. In terms of size, it is equivalent to assigning the one letter field name for every header of SIP. In particular, headers, such as *Cseq* have fixed format. BHSIP defines these header fields completely with binary sequences. Let us look at *Cseq* header field example in Table 1. It starts with field name followed by sequence number and request type. The field name is substituted by binary sequence that uniquely identify this field. According to [1], the sequence number can not exceed 2^{31}, hence BHSIP defines it as 32 bit or 4 byte binary sequence. As for request line, every request name has its own binary representation. For example, the phrase INVITE is defined as binary representation of *0x01*.

The experimental results that reveals the benefits of using BHSIP over SIP are discussed in the later section.

5 Experimental Setup

Figure 3 illustrates the experimental setup. Two Linux machines were setup to act as UE and P-CSCF. These nodes were programmed to generate the sequences of SIP messages as shown in Fig 1. SigComp layer was implemented with standard Linux compression library, *zlib 1.1.4*, and DEFLATE [9] was used to form compressed messages. Other algorithms may compress differently. However, since the experiment was designed to evaluate the post-dialing delay in general

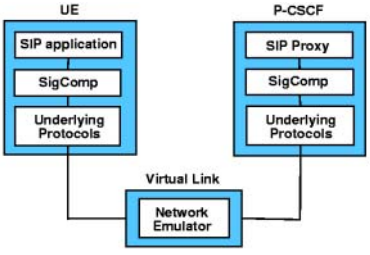


Fig. 3. Experiment Setup

and the compression ratio is more dependent on the contents of the dictionary than the capabilities of the compression algorithms, only one type of algorithm was used.

The contents of SIP and SDP within SIP message exactly followed the example given in chapter 7 of [6]. That is, SDP portion of SIP message carries the descriptions of 2 video codecs and 2 audio codecs. These nodes are not fully functional SIP entities. They are specially implemented to imitate the behavior of UE and P-CSCF during post-dialing period. The link between UE and P-CSCF, in practice, traverses several *relay* points, such as NodeB, RNC, SGSN and GGSN. However, since they have no effects on IP level connections and higher layer protocols, for the purposes of this experiment, we assumed that there existed a virtual IP level connection between UE and P-CSCF. The virtual link was described by three factors: the delays caused by the *relay* points, the bandwidth limited by air interface, and the rate of packet losses. Radio Access Network (RAN) and P-CSCF are connected with broadband link, whereas UE is connected to RAN by a narrow band link. Hence, we assumed the virtual link had the bandwidth equal to that of a wireless radio link. In our set of experiments, we ignored the effects of delay and packet losses for the simplicity. The network emulator from National Institute of Science and Technology, NISTnet [16] was installed and utilized to emulate the characteristics of the virtual link. The current version of NISTnet does not support IPv6 which has been mandated for IMS by 3GPP. In order to prepare the experiment that resembles a 3GPP network more closely, the emulator was modified to handle IPv6. We had also ignored the need for any negotiation with codecs, beyond of what is proposed in the initial INVITE message (SDP context) sent by the caller to the callee.

Each emulated simulation was conducted to measure the round trip time of SIP messages between UE and P-CSCF during the post-dialing period. It was repeated at least 50 times, and the experiment was terminated when certain statistical requirements were met. The measurements average values with with statistical errors below 5% at the confidence level of 95% were estimated. Since the error of results was so small they are not depicted in the graph, in this paper.

Table 2. SIP with SDP based message sizes (in bytes)

Type	Original Size	Static Compression	Dynamic Compression
INVITE	1696	568	568
Trying	243	159	40
Session Progress	1176	527	276
PRACK	982	462	90
OK (PRACK)	693	324	45
UPDATE	1068	504	61
OK (UPDATE)	785	350	73
Ringing	466	238	37
Total Size	7109	3132	1190

6 Experimental Results

6.1 Size of Controlling Message

Table 2 shows the size of SIP messages and their compressed forms. The first SIP message, INVITE, is compressed with initial state of SigComp. That is, INVITE message is only compressed with the initial value of dictionary that is defined in [4]. Hence, the size of the message is the same as if it is compressed without dynamic dictionary updates. Overall, with static compression approximately 56% of size reduction is achieved, and about 83% reduction is observed with dynamic compression.

As in Table 2 and 3, the size of messages dramatically reduces once we introduce the adaptation based on dynamic dictionary updating. BHSIP messages, which have exactly the same meaning as previous SIP messages have slightly smaller sizes than in normal SIP-based schemes, because these already have header names and request and response lines in binary format, in the way described in previous section. Again in this case, reduction in the size of controlling messages is noticeably large. Dynamically compressed BHSIP messages are 16% smaller than their corresponding SIP messages.

The main advantage of the BHSIP scheme is that it uses dictionary that is relatively smaller than that of SIP. Since all the header fields are already in binary forms, the dictionary does not need to contain the phrases for the header field names. That implies smaller storage requirements for the memory constrained device or UE. In addition, since BHSIP uses shorter messages than SIP, updating information for the state handler of SigComp is consequently smaller, hence the updating can be done slightly faster than in the SIP-based schemes, especially P-CSCF which is responsible for large number of UEs in the network.

Table 3. BHSIP based message sizes (in bytes)

Type	BHSIP Size	Static Compression	Dynamic Compression
INVITE	1502	530	530
Trying	191	160	38
Session Progress	1052	514	272
PRACK	859	439	86
OK (PRACK)	627	331	46
UPDATE	913	479	72
OK (UPDATE)	717	357	71
Ringing	385	242	36
Total Size	6246	3052	1151

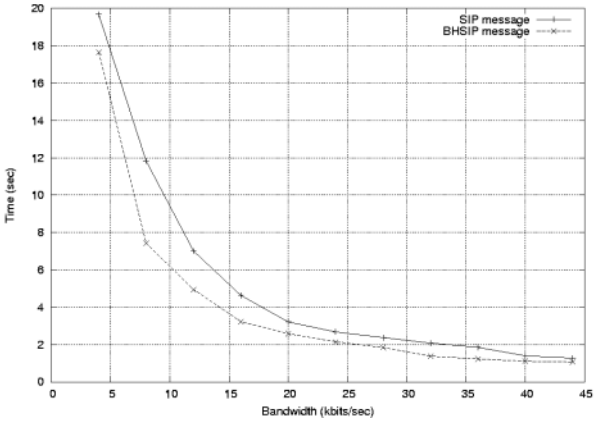


Fig. 4. Mean post-dialing delay of SIP and BHSIP messages

6.2 Post-dialing Delay

Figure 4 and Fig 5 show mean round trip time delay when using normal uncompressed control messages and SigComp schemes during post-dialing period respectively. In our experiment we omitted the QoS reservation time delay, as we are only interested in reducing the round trip time of SIP messages between UE and P-CSCF link. Despite this omission, it is clear that a normal SIP message requires too much bandwidth to satisfy the recommendation and it is noticeable that employing SigComp significantly reduces total round trip time.

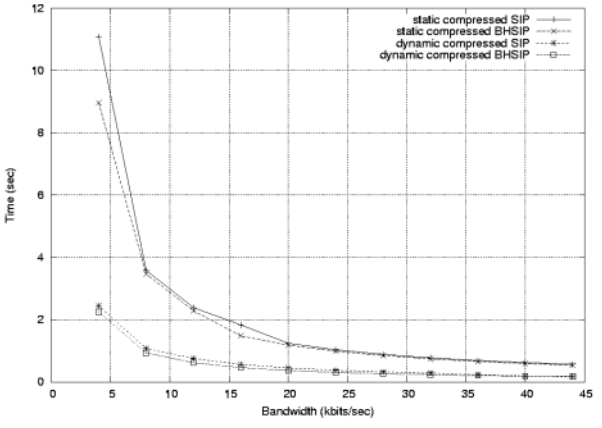


Fig. 5. Mean post-dialing delay of SIP and BHSIP with SigComp

The schemes investigated in the experiments have both advantages and disadvantages. SigComp introduces the complexity to P-CSCFs and UEs, which could exacerbate power consumption issues, which plague current 3G devices. The static compressor version gets rid of the need for state managements at the SigComp layer, which implies less intensive information transfer from application to SigComp layer. Hence, it results in a simpler model than SigComp, although it will require more bandwidth to exchange the messages.

7 Conclusion

The experiment confirms that SigComp does reduce the signalling message size significantly and consequently reduces the post-dialing delay of SIP sessions. It has also revealed that the static compression scheme can provide relatively good performance, without introducing too much complexity to the system. However, in order to fully estimate the delay, the time for reserving the resources during that period, the effects of *relay* entities and of lower layer protocol overhead must be taken into account as well.

We have briefly introduced a variant of SIP, named BHSIP. BHSIP provides the same flexibility and functionality as normal SIP does, yet it results in smaller size messages at the SIP application layer, which, in turn, results in a smaller dictionary and less SigComp updating information, hence reducing the load of the SigComp layer.

References

[1] Rosenberg, J., et al.: SIP: Session Initiation Protocol. RFC3261, IETF. June (2002) 514, 518

[2] Handley, M., Jacoson, V.: SDP: Session Description Protocol. RFC2327, IETF. April (1998) 515

[3] Proce, R., et al.: Signaling Compression (SigComp). RFC3320, IETF. January (2003) 515, 516

[4] Garcia-Martin, M., et al.: The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp). RFC3485, IETF. February (2003) 517, 520

[5] 3GPP: 3rd Generation Partnership Project, Technical Specification Group Service and System Aspects; IP Multimedia Subsystem (IMS); Stage2, Release 5. 3G TS 23.228 515, 516

[6] 3GPP: 3rd Generation Partnership Project, Technical Specification Group Core Network Signaling flows for the IP multimedia call control based on SIP and SDP (Release 5). 3G TS 24.228 515, 516, 519

[7] 3GPP: 3GPP IETF Dependencies and Priorities. April (2003) 515, 516

[8] 3GPP2: 3GPP2-IETF Draft Dependencies. March (2003) 515, 516

[9] Deutsche, P.: DEFLATE Compressed Data Format Specification version 1.3. RFC1951, IETF. May (1996) 518

[10] Curcio, I., Lundan, M.: SIP Call Setup Delay in 3G network. Proceedings of the 7th International Symposium on Computer and Communications (ISCC '02). IEEE July(2002) 835–840 516

[11] Salsano, S., Veltri L., Papalilo, D.: SIP Security Issues: The SIP Authentication Procedure and its Processing Load. IEEE Network, **16** November/December(2002) 38–44 516

[12] Grech, M., Torabi, M., Unmehopa, M.: Service Control Architecture in the UMTS IP Multimedia Core Network Subsystem. 3G Mobile Communication Technologies, IEE Conference Publication **489** May(2002) 22–26 516

[13] West, M., Conroy, L., Hancock, R., Price, R., Surtees, A.: IP Header and Signaling Compression for 3G System. 3G Mobile Communication Technologies, IEE Conference Publication **489** May(2002) 102–106 517

[14] Wisely, D., Eardley, P., Burness, L.: IP for 3G, Networking Technologies for Mobile Communications. ISBN 0-471-48697-3, Wiley, (2002) 263–264 515

[15] ITU-T: Digital Exchange Performance, Design Objectives. Recommendation Q.543. March (1993) 516

[16] NISTnet: http://snad.ncsl.nist.gov/itg/nistnet 519

Decision Point of AAL2 Multiplexing for Voice and Data Services in 3G WCDMA Network

Hyun-Jin Lee[1], Jae-Hyun Kim, and Bong-Ho Kim[2]

[1] Ajou University, San 5 Wonchon-Dong
Paldal-Gu, Suwon 442-749, Korea
{l33hyun,jkim}@ajou.ac.kr
[2] Bell Labs Advanced Technologies, Lucent Technologies Holmdel
NJ 07733 USA
bhkim@lucent.com

Abstract. To predict AAL2 multiplexing performance, we derived bandwidth gain analytically using discrete Markov chain model for voice service and validated these results with a simulation. And we also performed detailed simulation for voice and data services in a concentrator. Based on the analysis, we find that there is no major benefit of AAL2 multiplexing in a concentrator. The benefit of AAL2 multiplexing in Iub for data services is much less than that for voice service. The results also indicate that end-to-end application-level performance must be incorporated in all development phases and the benefit of AAL2 multiplexing depends heavily on the traffic load.

1 Introduction

Asynchronous Transfer Mode (ATM) is selected for the UMTS Terrestrial Radio Access Network (UTRAN) transport due to its ubiquitous nature for heterogeneous traffic types, Quality of Service (QoS) guarantee. However, applying ATM to low bit rate mobile voice stream is inefficient due to generate the useless traffic in filling out the payload of an ATM cell [1]. Recognizing the problems, ITU-T standardized a ATM Adaptation Layer type2 (AAL2) for the bandwidth efficient transmission of low bit rate delay sensitive application.
Since the ATM/AAL2 protocol suite is mandatory in the first and second release of UMTS and may be applied to both the access and the core network. Hence, the traffic performance of AAL2 is one of the most important topics in UTRAN Engineering. And many papers analyzed it by computer simulation or by simple experiment [2,4,5,3]. some studies show that the gain obtained by AAL2 is significant in terms of bandwidth and point out the importance of selecting Timer_CU value since it significantly affects the linkefficiency [3,5]. However, most of these papers are not UMTS network focused and not considering UMTS specific protocol behaviors to evaluate the bandwidth efficiency in AAL2 multiplexing. The lately published paper [5] includes some of protocols which impact throughput in Iub such as Radio Link Control (RLC) and Framing Protocol (FP) but focuses on comparison between different scheduling mechanisms in order to choose the

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 524–533, 2004.

suitable algorithm for the Common Part Sublayer (CPS) multiplexer so briefly mentioned the importance of selecting AAL2 multiplexing within the Iub without quantitative analysis. [5] still uses simplified models such as approximated protocol overhead, no RLC retransmission because Block Error Rate (BLER) is not explicitly modeled, and no Source Controlled Rate (SCR) Operation [6], which changes voice traffic behavior and may result in reducing the bandwidth usage and increasing the ATM cell packing density, the ratio of the average user bytes (excluded ATM and CPS headers) in a cell onto the ATM cell length.

2 UMTS Model Description

We models all protocol layers from the physical to the application layer and models details of the packet handling characteristics of each network element along the path. The reference architecture and connections are based on 3GPP UMTS Release 99 standards and the UMTS application models are based on a combination of standards and published traffic characteristics [7], [8]. For the voice traffic model a mobile-to-mobile reference connection is assumed and for the web browsing, client-server models are used.

2.1 UMTS Network Architecture

The network architecture is a simplified from the original UMTS Simulator described in [9]. Fig. 1 shows the network model used in this paper. Voice traffic and web browsing data traffic are offered to the UMTS network, which consist of ten Node-Bs, one ATM/AAL2 multiplexing, one RNC in the UTRAN and other core network element to support the voice and data application traffic includes web server and voice called parties. The Node-B to concentrator link has a capacity of one E1 (2.048 Mbps) and the concentrator to RNC link capacity is STM-1 (155.520 Mbps).

2.2 Protocol Stack Models for CS and PS Service

Fig. 2 shows the protocol stack modeled in the simulator for Packet Switched Service (PSS) and Circuit Switched Service (CSS) within UTRAN with AAL2 multiplexing along the Iub interface. The left column in the shaded box is for PSS and the right column is for the CSS. The non-shaded boxes are common for PSS and CSS. AAL2, Adaptive Multi Rate (AMR), RLC and Dedicated channel Frame Protocol (DchFP) models are described in detail in the following sections. The air interface is not modeled explicitly as that would slow the simulation down too much. Instead a separate model was used to generate trace files of air interface performance as given by BLER under various conditions. This trace file of BLER was fed into the RLC layer.

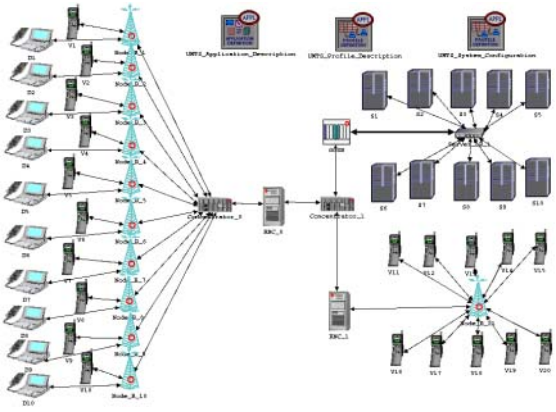


Fig. 1. UMTS network which support the voice and data traffic

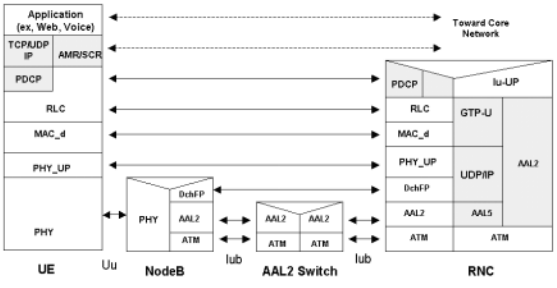


Fig. 2. Protocol Stack Model for PSS and CSS

AAL2 Multiplexing consists of two sublayers, which are called CPS and Service Specific Conversion Sublayer (SSCS). The CPS provides multiplexing and demultiplexing of CPS packets over a single ATM Virtual Channel Connection (VCC). i.e each CPS generates CPS packets with a 3 byte packet header and a variable length payload. AAL2 uses the 8 bit Channel ID (CID) in the CPS packet header to multiplex multiple AAL2 users onto a single VCC. Because of a limited CID size and some reserved values, only 248 individual connections can be differentiated within a single VCC. The CPS collects CPS packets from the AAL2 users multiplexed onto the same VCC over a specified interval of time (Timer_CU). If the cell is not completely packed within the time period determined by this Timer_CU value, the timer expires and the partially packed cell will be sent. The CPS-Protocol Data Unit (PDU) employs a one octet Start Field (STF) followed by a 47 byte payload.

The AMR Codec is the most important vocoder which is a mandatory speech processing function in UMTS and Source Controlled Rate (SCR), called "Discontinuous transmission" in GSM, functionality is also part of the standard [11], [6].

The AMR codec uses eight source codecs with bit-rates of 12.2, 10.2, 7.95, 7.40, 6.70, 5.90, 5.15 and 4.75 kbps and the coder operates on speech frames of 20 ms corresponding to 160 samples at the sampling frequency of 8000 sample/s. In this paper we consider not only one of the specified rates (for 12.2 kbps) during the ON state (talkspurt) but also the comfort noise during the OFF state(silence) of the AMR codec.

RLC Protocol provides segmentation and retransmission services for both user and control data. Each RLC instance is configured by RRC to operate in one of three modes: Transparent mode (Tr), Unacknowledged Mode (UM) or Acknowledged Mode (AM). Tr and AM are used for the user plane and UM is not used in this study because the usage of this mode is mainly for RRC signaling and VoIP. Therefore these two modes are described in this section as modeled in the simulator. Tr entities are defined to be unidirectional, whereas AM entities are described bi-directional. In Tr no protocol overhead is added to higher layer data and the transmission of the streaming type in which higher layer data is not segmented. The Tr mode can be used for circuit service such as voice call. In AM, an Automatic Repeat reQuest (ARQ) mechanism is used for error correction. The AM is the normal RLC mode for packet-type services such as web browsing and email downloading. In this paper RLC retransmission attempts to recover the corrupted blocks within the air interface according to the BLER model before a recovery mechanism from an upper layer protocol such as TCP is in action. The RLC retransmission model has an important role of changing the traffic model within the Iub.

DchFP allows for multiplexing of coordinated dedicated transport channels, with the same Transmission Time Interval (TTI), onto one transport bearer. The transport blocks of all the coordinated Dedicated CHannels (DCH) for one TTI are included in one frame [13]. There are two types of DchFP frames (indicated by the Frame type field): DCH data frame and DCH control frame. We modeled DCH data frame protocol since the user plain is the interest in this paper.

2.3 Service Traffic Model and Transport over Iub

In this paper it is assumed that all radio bearers are carried with Dedicated Physical Data CHannels (DPDCH) within DchFP over Iub.

Voice Traffic Model It has been found that length of talkspurt and silence are exponentially distributed [7]. In the voice traffic model 3 sec for both talkspurt and silence period is used followed by [14].

Web Browsing Traffic Model The distributions of the parameters for the web browsing traffic model is determined in [8] and an application session is divided in ON/OFF periods representing web page downloads and the intermediate reading

times. The web traffic will depend on the version of HTTP used by the web browsers and servers. HTTP 1.1, one of the widely used protocols, is used in the web browsing simulation. It is assumed that maximum data rate is 64kbps for UpLink (UL) and 144 kbps for DownLink (DL).

2.4 Assumption and Parameters

We assumed that only voice or web traffic and consists of multiple voice traffic generators or data traffic generators in UEs. Each traffic generator can generate multiple sessions of voice or web browsing traffic. In the evaluation, only uplink traffic was monitored for voice traffic since UL and DL traffic would be similar, and for the web traffic both UL and DL are monitored because of the asymmetric traffic behavior. AAL2 multiplexing in a Node-B and a concentrator is used with Timer_CU value from 1 ms to 4 ms. The maximum number of application sessions generated to a Node-B from UEs attached in the Node-B is the "Max {number of sessions generate traffic = E1 link, 248 sessions}". TTI is set to 20 ms for voice session and 40 ms for web browsing session. We used 1% BLER for voice and 4% BLER for 64 kbps UL and 5% BLER for 144 kbps DL for data traffic. The maximum allowable retransmission to recover the block error between RLC in the UTRAN is limited to three, and those packets could not recovered by the RLC are rely on the recovery mechanism in TCP protocol.

3 Packing Density and Bandwidth Gain Analysis

To understand how many AAL2 packet bits are packed into an ATM cell, we derived the payload packing density in a cell. Payload packing density in cell is calculated by Eq. (1)

$$\psi\,(\%) = \frac{B}{47} \times 100 \quad , (1 \le B \le 47) \tag{1}$$

where, B is the average payload size in a CPS-PDU except STF. AMR codec generates the packet with 244 bits during the talkspurt and the packet with 39 bits during the silence but the lengths of a packet in each period are 42 bytes and 15 bytes including DchFP packet overhead. P_{talk} and $P_{silence}$ be the probabilities of packet arrivals in Timer_CU (τ) for an User Equipment (UE) in talkspurt or silence period, respectively. The probabilities of a packet arrival from a user are given by

$$P_{talk} = \frac{E\left[L_{talk}\right]\tau}{6000} = \frac{\left(\bar{T}_{talk} + H\right)\tau}{6000 \times TTI} \tag{2}$$

$$P_{silence} = \frac{E\left[L_{silence}\right]\tau}{6000} = \frac{\left(\bar{T}_{silence} - H\right)\tau}{6000 \times 8 \times TTI} \tag{3}$$

L_{talk} and $L_{silence}$ are the number of packet generated in each time interval. $\bar{T}_{talk}$ and $\bar{T}_{silence}$ are the average duration of each period. H means hangover used to maintenance connection (140 ms). Voice sources are assumed to be i.i.d. After

the receipt of the first packet in a cell, the probability of n packet arrivals from all other sources within (τ) can be derived in Eq (4).

$$P[(N-1)=n] = \binom{N-1}{n}(1-P_{talk}-P_{silence})^{N-n-1} \times (P_{talk}+P_{silence})^{n} \tag{4}$$

where, N is the number of UEs. And given n packet arrivals, the probability of i talkspurt packets and j silence packets among n is given by

$$\begin{aligned} p_{ij} &= P\left[N_{talk}=i, N_{silence}=j\right] \\ &= \binom{N-1}{n}\binom{n}{i}(1-P_{talk}-P_{silence})^{N-n-1} \\ &\quad \times (P_{talk})^{i}(P_{silence})^{j}, \qquad i+j=n \end{aligned} \tag{5}$$

where, $N_{silence}$ and N_{talk} are the number of concurrent users for talk state and silence state. Using the probabilities p_{ij}, we can calculate the payload packing density. The average number of bytes in an ATM cell depends on whether there is a remainder from the previous cell. Since the maximum packet size is 42 byte (talkspurt packet), the reminder varies from 0 to 41. If r_n is the reminder bytes in n^{th} cell. r_n can be modeled the state of Markov chain since rn only depends on r_{n-1} $(1 \le n \le 41)$. Consider the stationary state where all $\{r_n\}$ have the same probability distribution, Let r denote the random variable for the reminder length, and which is given

$$\pi_i = P[r=i], \qquad i=0,\cdots,41 \tag{6}$$

And the probabilities that AAL2 packet is expired denote Eq (7).

$$\begin{aligned} Q_i &= P[Timer\_CU \text{expires} | r=i] \\ &= \begin{cases} p_{00}+p_{01}+p_{02}+p_{03}+p_{10} & ,0 \le i \le 2 \\ p_{00}+p_{01}+p_{02}+p_{10} & ,3 \le i \le 5 \\ p_{00}+p_{01}+p_{02} & ,6 \le i \le 17 \\ p_{00}+p_{01} & ,18 \le i \le 32 \\ p_{00} & ,33 \le i \le 41 \end{cases} \end{aligned} \tag{7}$$

Transition probability matrix $\mathbb{P}$ is given by Eq (8).

$$\mathbb{P} = \begin{pmatrix} Q_0 & 0 & 0 & 0 & 0 & \cdots \\ Q_1 & 0 & 0 & 0 & 0 & \cdots \\ Q_2+\frac{(1-Q_2)p_{02}}{p_{02}+p_{03}+p_{11}} & 0 & 0 & 0 & 0 & \cdots \\ Q_3 & \frac{(1-Q_3)p_{03}}{p_{03}+p_{11}+p_{20}} & 0 & 0 & 0 & \cdots \\ Q_4 & 0 & \frac{(1-Q_4)p_{03}}{p_{03}+p_{11}+p_{20}} & 0 & 0 & \cdots \\ Q_5+\frac{(1-Q_5)p_{11}}{p_{11}+p_{03}} & 0 & 0 & \frac{(1-Q_5)p_{03}}{p_{03}+p_{11}} & 0 & \cdots \\ Q_6 & \frac{(1-Q_6)p_{10}}{p_{03}+p_{10}} & 0 & 0 & \frac{(1-Q_6)p_{03}}{p_{03}+p_{10}} & \cdots \\ Q_7 & 0 & \frac{(1-Q_7)p_{10}}{p_{03}+p_{10}} & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{8}$$

In the stationary state, we have $\mathbf{\Pi} = \mathbf{\Pi}\mathbb{P}$. And using these probabilities, the average of bytes in an ATM cell is given Eq. (9)

$$\begin{aligned} B = p_{00}\sum_{i=0}^{41} i\pi_i + p_{01}\sum_{i=0}^{32}(i+15)\pi_i + p_{02}\sum_{i=0}^{17}(i+30)\pi_i \\ + p_{03}\sum_{i=0}^{2}(i+45)\pi_i + p_{10}\sum_{i=0}^{5}(i+42)\pi_i \\ + 47(1 - p_{00} - p_{01} - p_{02} - p_{03} - p_{10}) \end{aligned} \tag{9}$$

The bandwidth gain ($\xi(\tau, n)$) of the AAL2 in $\mathrm{I_{ub}}$ is derived by

$$\xi(\tau, n) = \frac{E\left[B(\tau, n)\right] - E\left[B(0, n)\right]}{E\left[B(0, n)\right] + 5} \times 100 \tag{10}$$

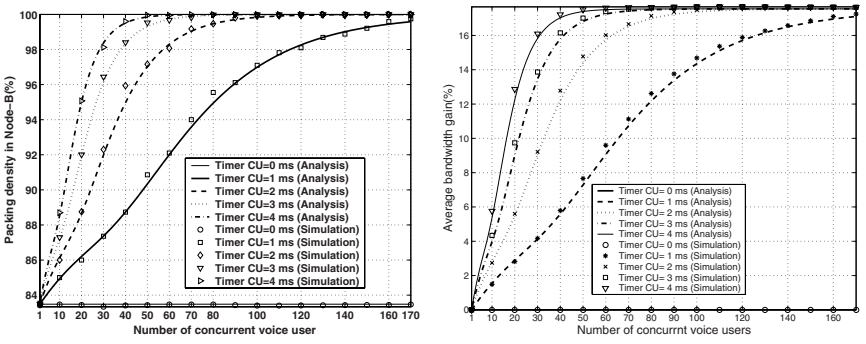
where, $B(\tau, n)$ means the user payload in a CPS-PDU during Timer_CU with AAL2 multiplexing and $B(0, n)$ means the user payload of ATM cell without AAL2 multiplexing.

4 Numerical and Simulation Results

To understand the bandwidth gain in Node-B and in a concentrator, a set of simulations has been performed with various Timer_CU values and number of concurrent users for the voice and web browsing traffic.

4.1 Voice Traffic Scenarios

The ATM cell packing density and the bandwidth gain results with various Timer_CU values and number of concurrent users for the voice traffic are summarized in Fig.3(a) through Fig.4. In Fig.3(a) and Fig.3(b), we measured the bandwidth gain and packing density in Node-B using analysis and simulation. we set up one Node-B and vary the number of concurrent voice users in a voice UE from 1 to 170 users. Timer_CU in Node-B is set to 0 through 4 ms. Fig.3(a) presents the packing density versus the number of concurrent voice users and Timer_CU. The symbols present simulation results and the curves mean analytical result. Based on the analytical results, the minimum packing density, which obtained without AAL2 multiplexing (Timer_CU=0 ms), is about 83.5%. AAL2 packing density reaches 96% when the number of concurrent voice users is 80 with Timer_CU=1 ms while it has same value for 40 concurrent voice users and Timer_CU=2 ms. The analytic result closed to simulation one. Fig.3(b) shows average bandwidth gain of AAL2 multiplexing in Node-B. The result indicates that maximum bandwidth gain with AAL2 is about 18% higher than the bandwidth gain without AAL2. If link type between Node-B and concentrator is E1, then 170 concurrent voice users can be connected simultaneously in a Node-B with a Timer_CU=0 ms. By setting Timer_CU=1 ms in Node-B, however, more



(a) ATM cell packing density (b) Bandwidth gain

Fig. 3. ATM cell packing density and Bandwidth gain with various Timer_CU and number of concurrent voice users

200 concurrent voice users can be served in a Node-B without any other system or network changes. This gives a strong reason to use AAL2 multiplexing in Node-B. We used only simulation result to predict the benefit which is additional AAL2 multiplexing in a concentrator. Fig.4 shows bandwidth gain in a concentrator with various number of concurrent voice users. As you can see in the graph, increasing traffic load per Node-B from 15% to 33% (when Timer_CU value is 2 ms) results in the drastic drop of bandwidth gain from 11% to 3%. The bandwidth gain is no meaningful for small traffic load and it is negligible as traffic load increases. So this result indicates that there is no significant AAL2 multiplexing benefit in a concentrator on Iub in terms of bandwidth gain.

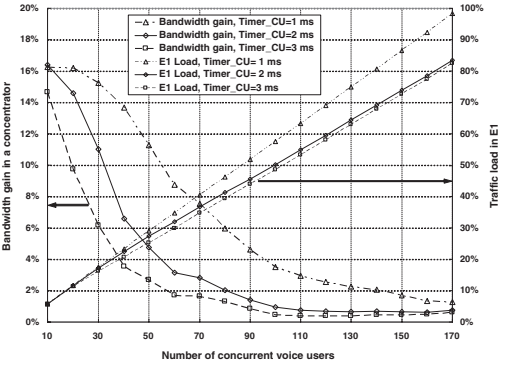


Fig. 4. Bandwidth gain in a concentrator on Iub interface (Timer_CU of Node-B fixed 1 ms and concentrator Timer_CU = 1, 2 and 3 ms, Total Voice Users = 170)

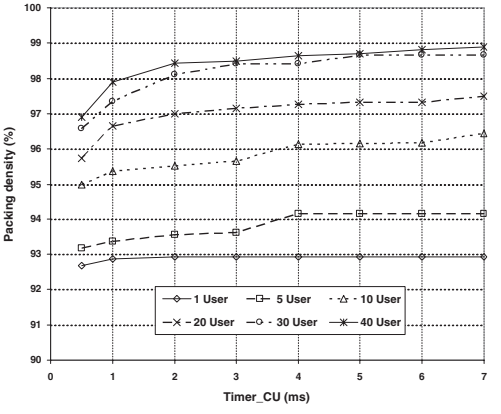


Fig. 5. The payload packing density with various Timer_CU (Web Browsing)

4.2 Data (Web Browsing) Traffic Scenario

The payload packing density results with various Timer_CU values and number of concurrent users for the data traffic are summarized in Fig.5. This figures represent downlink channel only. The traffic load generated by 40 simultaneous web-browsing sessions is about 600 Kbps (28% of E1) at the link between Node-B and concentrator after AAL2 multiplexing with Timer_CU of 1 ms at each Node-B. The traffic on this link includes DchFP and ATM protocol overhead as well. Fig.5 shows that the ATM cell packing density is larger than 92% even only one user with Timer_CU of 0.5ms. Since IP packet is already long enough to fill an ATM cell except the last fragment of the IP packet loaded in an ATM cell, the payload packing density is high even with only one user and this implies that the AAL2 multiplexing gain is very minimal even at the first multiplexing place (Node-B).

5 Conclusions

We analyzed the performance of AAL2 multiplexing in a Node-B and in a concentrator on Iub, considering UMTS specific protocols such as SCR, FP and RLC. To derive the packing density and the bandwidth gain in Node-B for voice services we applied a discrete Markov chain model. We considered not only the talkspurt packet but also the comfort noise packet in the silence period which impact traffic load and packing density. We derived the bandwidth gain in Node-B and validated these result with a detailed simulation considered UMTS specific protocol. We also evaluated the benefit of additional AAL2 multiplexing in a concentrator in Iub interface using simulation.

In this paper we concluded that the bandwidth gain of AAL2 in Node-B and in a concentrator on Iub depended heavily on the traffic load. The bandwidth gain

is no meaningful for small traffic load and it is negligible as traffic load increases. So this result indicates that there is no significant AAL2 multiplexing benefit in a concentrator on Iub in terms of bandwidth gain. For data traffic the benefit of the AAL2 multiplexing in Iub is less than that for voice service. The main contributions of this paper are twofold:

1. The performance of AAL2 in a Node-B is derived analytically considering UMTS specific protocol.
2. The decision making criterion to use AAL2 feature in a concentrator over Iub is suggested.

If a service provider or UMTS network architecture designer had an expected user traffic profile, it can be used with this paper to make a product selection decision for a concentrator.

References

[1] 3GPP TS 25.430 v.3.4.0 UTRAN Iub Interface (Release 1999),Dec. 2000.

[2] G. Eneroth , G. Fodor , A. Leijonhufvud , A. Racz . and I.Szabo , "Applying ATM/AAL2 as a Switching Technology in Third-Generation Mobile Access Networks." *IEEE Communications Magazine*, June 1999. pp.112-122

[3] C. Liu, S. Munir, R. Jain, S. Dixit, "Packing Density of Voice Trunking using AAL2", *In Proc. GlobeCom99*, December 5-9, 1999, Rio de Janeiro, Brazil,Vol. 1(B), pp. 611-615.

[4] S. Nananukul, Y. Guo, M.Holma, and S. Kekki, "Some Issues in Performance and Design of the ATM/AAL2 Transport in the UTRAN," *In Proc. IEEE WCNC 2000*,Chicago, Sep. 2000.

[5] R. Makke, S. Tohme, J.-Y. Cochennec,S. Pautonnier,"Performance of the AAL2 Protocol within the UTRAN,"*In Proc. EDUMN 2002*, pp. 92-100, 2000.

[6] 3G TS 26.093: Mandatory speech codec speech processing functions AMR speech Codec; Source Controlled Rate Operation (Release 1999),Mar. 2003.

[7] P. T. Brady, "A model for on-off speech patterns in two-way conversation," *Bell System Technical Journal*, vol 48, pp 2445-2472, Sep. 1969

[8] 3GPP2-C50-EVAL-2001022, "HTTP and FTP Traffic Models for 1xEV-DV Simulations."

[9] D. J. Houck, B. H. Kim, J. H. Kim, "End-to-end UMTS Network Performance Modelling", *In proc. Network 2002*, June, 2002, Germany

[10] ITU-T Recommendation I.363.2, B-ISDN ATM Adaptation Layer Specification: Type 2 AAL,Nov. 2000.

[11] 3GPP TS 26.101, AMR Speech Codec Frame Structure (Release 1999), Dec. 1999.

[12] 3G TS 25.322: Radio Link Control (RLC) Protocol Specification,Mar. 2003.

[13] 3G TS 25.427: UTRAN Iur and Iub interface user plane protocols for DCH data streams, Jan. 2003.

[14] TR 101.112 V3.2.0 : ETSI, selection Procedures for the Choice of Radio Transmission Technologies of the UMTS (UMTS 30.03 version 3.2.0),Apr. 1998.

An Adaptive Resource Allocation Scheme Based on Renegotiation for QoS Provisioning in Wireless Mobile Networks*

Jung-pyo Hong and Hwa-sung Kim

Department of Electronic and Communications Eng., Kwangwoon Univ., Korea
{jphong@,hwkim@daisy.}kw.ac.kr

Abstract. In the wireless mobile networks, it is important to provide the quality-of-service (QoS) guarantees as they are increasingly expected to support the multimedia applications. Although the QoS provisioning problem arises in the wire-line networks as well, the mobility of hosts and the scarcity of bandwidth make QoS provisioning a challenging task in wireless mobile networks. The resource allocation to multimedia applications of varying QoS requirement is a complex issue. In this paper, we propose a new adaptive resource allocation scheme based on the concept of the resource reservation and the renegotiation in order to guarantee the QoS of the real-time traffic. The proposed scheme is aimed at improving the performance in terms of the new call blocking rate, the handoff dropping rate, and the bandwidth utilization.

1 Introduction

The wireless mobile networks are expected to support the real-time interactive multimedia traffic and should be able to provide their users with QoS guarantees. The admission control and the bandwidth allocation scheme can help provide the QoS guarantees in wireline networks, however in wireless networks, the problem is much more complex due to the bandwidth limitations and the host mobility. The wireless systems usually use the microcellular architectures in order to overcome the limitation of the radio spectrum and to provide a higher capacity. These microcellular networks, however, have the inherent rapid handoff problem due to the smaller coverage area of cells. The rapid handoff problem leads to network congestion and higher call dropping rate. One solution to this problem in a microcellular network is to apply the distributed call admission control at the connection setup time. And, the spectrum efficiency in these networks can only be realized if there are strict rules of the admission control and the flexible resource allocation [1].

A lot of related researches about the QoS issue including the channel assignment, the bandwidth reservation and the admission control schemes have been

* This research was supported by University IT Research Center Project and the Research Grant from Kwangwoon University in 2003.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 534–543, 2004.

performed. In [2], a dynamic channel (re)assignment scheme was proposed. In this dynamic channel (re)assignment schemes, the channels are (re)assigned to different neighboring cells to reduce the interference and to increase the overall system capacity. The bandwidth reservation schemes proposed in [3] and [4], a fixed number of channels in each cell are reserved for handoff requests when none of the reserved channels is available. But, this scheme does not consider any admission control and the information regarding neighboring cells. In [5], an adaptive admission control mechanism is proposed for wireless networks. In this mechanism, the different resource sharing scheme such as complete partition is used to allocate resources to different classes of traffic. Resource allocation and admission control for wireless networks are also studied in [6].

In this paper, we propose a new resource allocation scheme based on the concept of resource reservation and renegotiation which provides the QoS guarantees for the multimedia traffic carried in wireless mobile networks. The proposed scheme is aimed at improving the performance in terms of the new call blocking rate, the handoff dropping rate, and the bandwidth utilization. This paper is organized as follows. In Section 2, the concept of the bandwidth reservation according to the classes of traffic is introduced. Section 3 describes the details of the proposed QoS provisioning scheme. In Section 4, a simulation model that was used in the simulation is described and the discussion about the simulation results will be given. Finally, Section 5 is the conclusion.

2 Overview of the Bandwidth Reservation

In this section, we provide an overview of the traffic model and the bandwidth reservation scheme related to the resource allocation schemes in cellular networks. In [7], the traffic carried in wireless networks are categorized into two classes: (a) Class I : real-time traffic, and (b) Class II : non-real-time traffic. In analogy with ATM (Asynchronous Transfer Mode) networks, Class I corresponds to the GBR (Guaranteed Bit Rate) service and Class II to the ABR (Available Bit Rate) service.

Since Class I traffic such as video and voice is highly delay sensitive, a connection should be dropped when a mobile user moves into a new cell where the minimum required QoS parameters cannot be met. On the other hand, since Class II traffic such as web and e-mail can tolerate large delay, the QoS parameters of a connection can be temporarily renegotiated and readjusted when a mobile user moves into a new cell where the original assignment cannot be provided. For example, when a mobile user receiving Class II service moves to a new cell where the originally assigned bandwidth cannot be provided, the bandwidth assigned to the connection can be reduced. Therefore, the connection receives the available bandwidth for an increased duration to compensate bandwidth reduction. Consequently, the connection does not need to be dropped. On the other hand, since Class I traffic has strict delay requirements, a priority is given to Class I over Class II. The QoS parameters for Class I traffic may include the handoff dropping probability and the minimum allowed bit rate. The QoS

parameters for Class II traffic may include the probability that a connection experiences the bandwidth readjustment. Another QoS parameter which may be important for both classes is the blocking probability of the new connections.

On the other hand, the bandwidth reservation scheme may be used to reduce the handoff dropping rate of Class I calls. Namely, the bandwidth reservation is performed in all cells adjacent to the cell in which the Class I call arrives[3][4]. The amount of bandwidth to be reserved in adjacent cells can be calculated using the various policies that may depend on many factors such as the total number of calls of the particular class or on the maximum bandwidth used by a call. The bandwidth reservation scheme can provide better QoS but there is a tradeoff between service quality and processing overhead. Besides, there are several issues that need to be resolved to efficiently apply the bandwidth reservation scheme. First issue is to decide how much bandwidth needs to be reserved in adjacent cells. If reservation is to be performed in all the cells, then only a fraction of the requested bandwidth of call needs to be reserved in the surrounding cells as the call will be hand off to one of the surrounding cells. This is the important parameter that affects the efficiency of the resource reservation schemes. Another issue is to determine when to reserve and release the bandwidth in the adjacent cells. The bandwidth reservation can be done in all the adjacent cells when a new call connection is requested, and the reserved bandwidth is released when a call finishes or hands off. These parameters are important to support QoS for a call. In the next section, we discuss careful choice of these parameters to support QoS of service. We discuss our new resource allocation scheme and show how it can be useful to support a variety of service applications.

3 A Proposed Resource Allocation Scheme

In this paper, we propose a new resource allocation scheme. The proposed scheme tries to provide the real-time traffic with QoS guarantee using the resource reservation in surrounding cells for real-time calls, while providing the non-real-time traffic with best-effort performance using the renegotiation and compensation of the required bandwidth.

3.1 Renegotiation and Compensation Scheme

The non-real-time traffic is more tolerant to delay compared to the real-time one in that it can accept a variable service rate. This property of non-real-time traffic makes resource renegotiation possible in microcellular networks. The non-real-time calls can receive higher service rates under low traffic conditions while the service rate available to them is kept at a minimum under the heavy traffic conditions. Thus, the resource renegotiation scheme enables to adapt to the changing traffic conditions in the network.

In the proposed scheme, each cell maintains a fixed reserved pool of bandwidth to serve the Class I new/handoff calls to reduce the call drop probability. On the other hand, in order to lower the request block and drop probability of

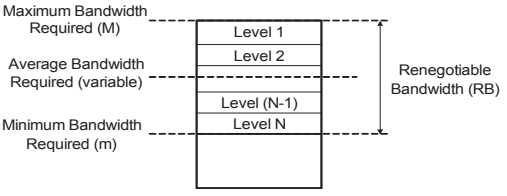


Fig. 1. Renegotiable bandwidth of a Class II call

the non-real-time calls, it has incorporated a renegotiation and compensation schemes. When a request is generated, each request specifies its desirable bandwidth M and minimum bandwidth m to the system. The difference between M and m is called renegotiable bandwidth (RB). A fraction of RB is divided into several service levels as shown in Fig. 1.

If a cell does not have enough bandwidth to serve the handoff request of Class I calls, a certain amount of bandwidth of the existing Class II calls are preempted temporarily by moving down to a lower service level. The preempted Class II calls will then be served at the lowered service level. The most important feature of the renegotiation scheme is that no request should be served below its minimum bandwidth requirement once it is admitted. The preempted bandwidth will be returned to the Class II calls later on by using the compensation scheme.

The compensation scheme is based on the max-min optimality criterion, which is both fair and efficient [8], in the sense that all degraded calls get the equal share of this surplus capacity. A call is referred to be compensated if the bandwidth of the call is changed from lower than average bandwidth (BWavg) to at least average. Bandwidth compensation to average bandwidth implies the retuning of the preempted resource. The compensation scheme will sequentially shuffle the existing bandwidths toward the average bandwidth for each call. The compensation can sometimes be very expensive and time-consuming since a large number of channel reassignments need to be performed. Hence, a partial bandwidth compensation algorithm will be required. The partial bandwidth compensation scheme checks the mobility pattern and starts when the amount of surplus resources exceeds the threshold level, as shown in Fig. 2.

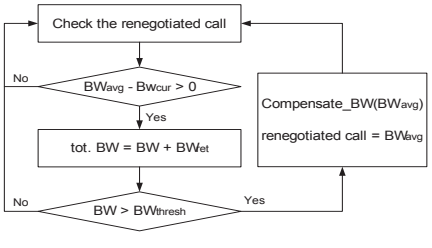


Fig. 2. Flow chart of bandwidth compensation

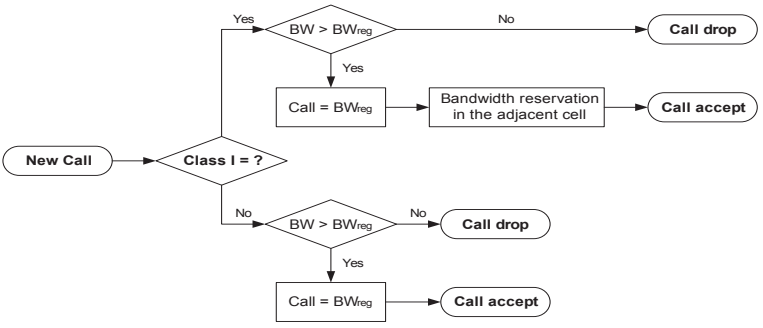


Fig. 3. Flow chart for new call setup

3.2 Resource Reservation for the New Call

A new connection, whether Class I or II, is accepted into a current cell only if its expected bandwidth (BWreg) is less than or equal to the available bandwidth of the cell (BW) excluding the reserved handoff pool. Especially in the case of establishing a Class I call connection, the resource reservation in the adjacent cells should also be performed as shown in Fig. 3.

The amount of bandwidth to be reserved in the adjacent cells for Class I call is decided according to the number of current Class I connections in each adjacent cell. In this paper, we decide the amount of reserved bandwidth according to the Fig. 4 that is optimized through the simulation. For example, if the number of Class I connection is currently 5, then 512kbps will be reserved for the Class I call. However, if the available bandwidth is less than the reserved bandwidth, the available bandwidth is reserved as the actually reserved bandwidth in the adjacent cell as the following formula:

Actual-bandwidth-reservation=min(unused-bandwidth,reserved-bandwidth)

3.3 Resource Management for the Handoff Call

The different handoff management mechanisms are applied to each of Class I and Class II handoff calls. In this paper, a Class II handoff calls are kept continued, even if the allocated bandwidth in a new cell is very small since they do not have stringent QoS requirements. Therefore, the Class II handoff calls are not dropped unless there is free bandwidth in the new cell. The reserved

Number of Class I Connections	Reserved Bandwidth
0 ~ 5	512 Kbps
6 ~ 10	1024 Kbps
11 ~ 20	2048 Kbps
21 or more	3072 Kbps

Fig. 4. Reservation function

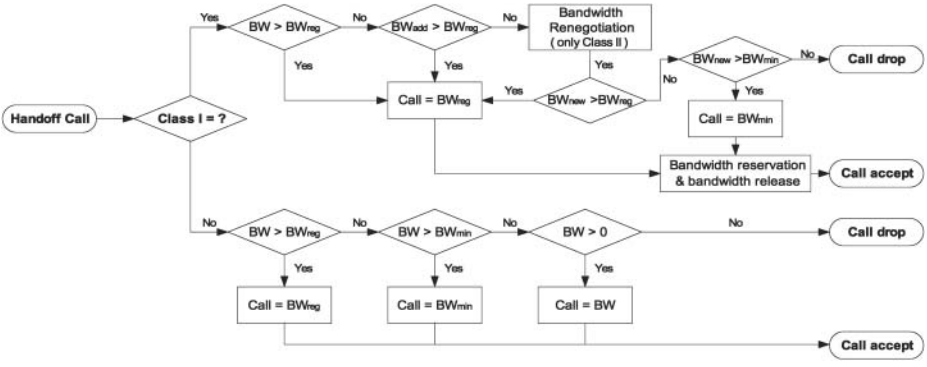


Fig. 5. Flow chart for handoff call management

bandwidth pool is for the exclusive use of Class I handoff calls and will not be available for Class II calls. On the other hand, Class I handoff call is allocated bandwidth firstly from the reserved pool in case that the expected bandwidth cannot be met using free bandwidth in the cell. If the sum of the reserved and the free bandwidth in the cell (BWadd) can not fulfill the expected bandwidth, the renegotiable bandwidth (BWnew) will be allocated to meet the expected bandwidth. Even using the renegotiable bandwidth, if the expected bandwidth can not be met, then the minimum bandwidth (BWmin) will be allocated. When even the minimum bandwidth can not be allocated, Class I call is then dropped as shown in Fig. 5.

4 Performance Evaluation

In order to evaluate the performance of the proposed scheme, we implemented and simulated three different schemes for comparison. 1) First scheme (no partition and reservation scheme) is the simplest among three. In this scheme, there is no distinction between real-time and non-real-time calls. The available bandwidth is not partitioned and any call may be served by a cell if there is enough bandwidth available. 2) Second scheme is a request-based statistical reservation scheme, namely the uniform and bandwidth based model [7]. In this scheme, when the reservations are made on behalf of a connection in neighboring cells, an equal amount of bandwidth is reserved in each neighboring cell with no consideration of the most likely cell to which the host might travel. A cell does not reserve the sum of all the bandwidth it is asked to reserve, but just the largest of all the current requests. 3) Third scheme performs the number-of-connections based bandwidth reservation in each cell for handoff. New calls are admitted if their desired bandwidth can be met. Otherwise, they are blocked. On the other hand, Class I handoff calls are admitted if their minimum bandwidth requirements can be met. If there is too little free bandwidth available, they are given enough bandwidth from the reserved pool to meet their minimum. Class

Class	Max bps	Avg bps	Min bps	Max time	Avg time	Min time	Example traffic content
I	30 Kbps	30 Kbps	30 Kbps	600s	180s	60s	Web browsing
I	256 Kbps	256 Kbps	256 Kbps	1800s	300s	60s	Audio Streams
I	6 Mbps	3 Mbps	1 Mbps	18000s	600s	300s	Video Streams
II	20 Kbps	10 Kbps	5 Kbps	120s	30s	10s	Email
II	512 bps	256 Kbps	64 Kbps	36000s	180s	30s	FTP file downloads
II	10 Mbps	5 Mbps	1 Mbps	1200s	120s	30s	FTP application downloads

Fig. 6. Traffic characteristics

II handoff calls are admitted only if there is free bandwidth in the cell excluding reserved pool.

For the simulation, we use the same traffic model, which was derived from the model used in [7]. Fig. 6 shows the traffic characteristics we simulated. Each of the six types of traffic occurs with equal probability. The desired bandwidth and cell length is generated for each new call using a geometric distribution around the minimum, maximum, and average values given. We also give each mobile host a speed characteristic (time spent in a cell) in order to simulate handoff; thus, longer calls will likely experience more handoff than shorter ones.

For the simulation, a 7 x 7 network of 49 cells is used as a network model. Each cell has a total bandwidth allocation of 30 Mbps and 10 percent of it is reserved for Class I new/handoff calls. The hosts are assumed to move the network in a directed manner, because hosts are more likely to continue move in the same direction than to turn. If a host reaches an edge of the network, the corresponding call will be terminated, hosts do not "bounce" back into the network.

In Fig. 7, the bandwidth utilization is shown as a function of number-of-clients. Bandwidth utilization for request-based reservation scheme is the lowest. The scheme without any kind of bandwidth partition or reservation has the highest bandwidth utilization because the available bandwidth in the cell can be

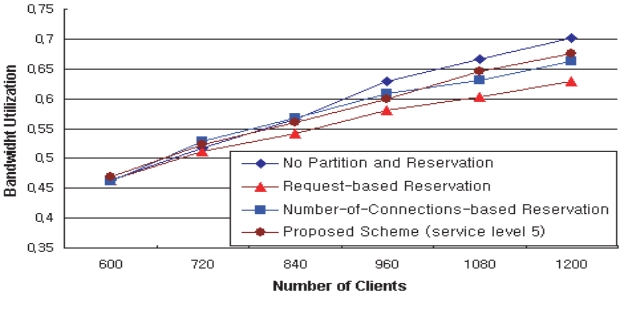


Fig. 7. Bandwidth utilization

used by any call, and hence no bandwidth is wasted. Bandwidth utilization for our scheme is fairly good than other ones.

Fig. 8 show the average blocking rate for the new call requests as a function of the number-of-clients. The No Partition and Reservation scheme produced the lowest blocking rate for the new calls. The proposed scheme based on renegotiation produces a pretty good blocking rate for the new calls compared to the other reservation based schemes. This is resulted from the same reason as explained earlier.

Fig. 9 shows the relationship between the average handoff dropping rate and the number-of-clients in case of Class 1 calls. The Handoff dropping rate is a kind of QoS Parameters for Class I calls. The simulation result shows that the proposed algorithm guarantees a predefined service quality to all existing calls in the network. No partition and Reservation scheme gives the worst average handoff dropping rate. Our scheme gives the best performance in terms of the handoff dropping rate for Class I calls. Hence the proposed scheme guarantees QoS to all existing cell. By carefully selecting the parameters - service level and amount of bandwidth reservation, the proposed scheme can provide a predefined QoS for all calls accepted in the system. There is, however, a tradeoff involved between the processing overhead and the guaranteed QoS.

Fig. 10 shows the average handoff dropping rate for Class II calls. Both of Request-based reservation scheme and the proposed scheme produce a better handoff dropping rate than other schemes. Our scheme guarantees a predefined

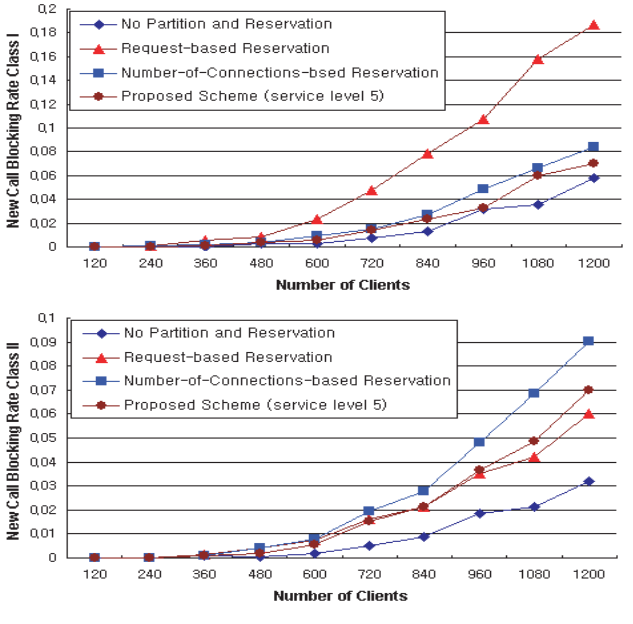


Fig. 8. New call blocking rate Class I and Class II

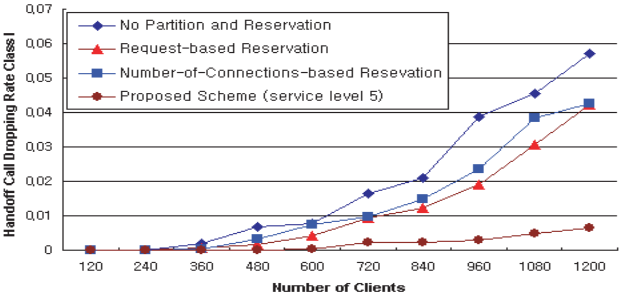


Fig. 9. Handoff call dropping rate of Class I

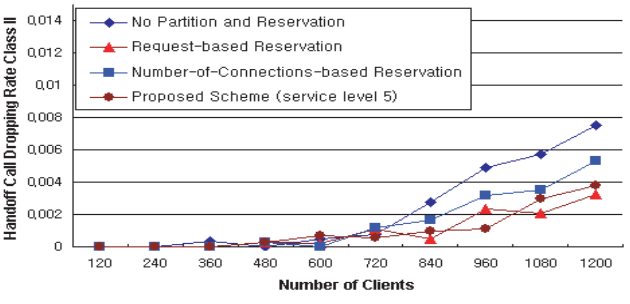


Fig. 10. Handoff call dropping rate of Class II

QoS for both real-time and non-real-time traffic. The higher blocking rate for our scheme can be reduced by increasing the dropping rate depending on the QoS metric.

5 Conclusion

With the increasing demand for wireless networks to support real-time multimedia applications along with elastic applications such as web access, QoS provisioning poses a complex problem. In this paper, an adaptive resource allocation based on renegotiation and compensation scheme has been proposed to provide QoS guarantees in wireless mobile networks. The proposed scheme overcomes the drawbacks of schemes in which only reservation is used. The proposed adaptive resource allocation scheme provides the good performance in terms of the handoff dropping and the bandwidth utilization under heavy traffic conditions.

The performance evaluation was performed through the computer simulation using 7 x 7 two-dimensional cellular network models. For the comparison, three other resource allocation schemes were implemented and simulated in addition to the proposed one. Simulation results showed that the proposed scheme provides better performance than other schemes and can be used to provide QoS

guarantee especially for the real-time traffic. The proposed scheme performs the resource reservation in the adjacent cells for real-time calls and improves the handoff dropping rate by using resource renegotiation at the time of handoff. The renegotiated resources of non-real-time traffic will be returned to them when the traffic condition gets better. Our proposed scheme gives also a better blocking rate for new calls as compared to the reservation scheme. The adaptive resource allocation scheme based on renegotiation produce the high resource utilization while remaining the blocking and dropping probability low. This feature is important in wireless mobile networks since the radio spectrum has inherent bandwidth limitations.

References

[1] A. Acampora and M. Naghshineh, "Control and quality-of-service provisioning in high-speed microcellular networks," IEEE Personal Comm., vol. 1, 1994
[2] S. Nanda and D. Goodman, "Dynamic resource acquisition: Distributed carrier allocation for TDMA cellular systems," in Third Generation Wireless Information Networks. Norwell, MA: Artech House, pp. 99-124, 1992.
[3] S. Oh, and D. Tcha, "Prioritized channel assignment in a cellular radio network," IEEE Trans. Comm., vol. 40, no. 7, July 1992.
[4] S. Tekinay and B. Jabbari, "A measurement-based prioritization scheme for handover in mobile cellular networks," IEEE J. Select. Areas Comm., vol. 10, 1992.
[5] M. Naghshineh and A. Acampora, "QoS provisioning in microcellular networks supporting multimedia traffic," in IEEE Infocom 95, BM, April 1995.
[6] B. M. Epstein and M. Schwartz, "Predictive QoS-based admission control for multi-class traffic in cellular wireless networks," IEEE J. Select. Areas Comm., vol. 158, pp. 510-522, 2000.
[7] C. Oliveira, J. Kim, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," IEEE J. Select. Areas Comm., vol. 16, 1998.
[8] A Charny, D Clark, and R. Jain, "Congestion control with explicit rate indication," Proc. ICC '95, vol. 3, pp. 1954-1963, 1995.

Practical Considerations in Trunk Engineering for Cellular Service*

Kyung Geun Lee[1], JongSuh Park[2], Ho Soo Kim[2], and Juwook Jang[2]

[1] Dept. of Information and Communication Engineering, Sejong University
kglee@sejong.ac.kr
http://nrl.sejong.ac.kr
[2] Department of Electronic Engineering, Sogang University
{netofpos,lake}@eeca1.sogang.ac.kr
jjang@sogang.ac.kr
http://eeca2.sogang.ac.kr

Abstract. We identify and address practical problems facing the engineers who are responsible for trunk engineering(determining optimal trunk requirements between switching systems in a telecommunications network) in a nation-wide cellular service. Currently, Erlang B formula is used to calculate the number of trunks to carry the estimated cellular traffic with given target grade of service(i. e. call block rate). However, our recent measurement at a nation-wide cellular service covering more than 15 million customers shows that the measured block rate is occasionally far greater than the expected block rate, as much as 8 times. Fearing this, it is a common practice for field engineers to assign far more trunks than dictated by the Erlang B formula. But the main problem is that there is no basis on how to assign more trunks. In this paper, we track the cause for excessive block rate by analyzing vast amount of call log to identify the characteristic of the recent cellular traffic. We introduce a simple but effective compensation method to adjust the Erlang B formula with random and non-random traffic. The second problem we address is that the Erlang B formula gives average block rate while the management of the cellular service demands the engineers to guarantee given upper limit to the block rate. We employ the concept of the confidence interval to guarantee given block rate with certain reliability. We develop a simulation program to derive an updated version of Erlang B table with the confidence interval and a simple heuristic method to compensate for the peakedness of contemporary cellular traffic.

1 Introduction

With explosive growth of cellular service, the cost of upgrading wired link capacity to carry the cellular traffic increases tremendously. Thus it is essential to accurately estimate the minimal capacity of the links needed for given target

* This work was supported in part by Sogang University Research Fund 2003 and uT frontier project.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 544–552, 2004.

Grade of Service (GoS), often represented by the block rate for attempted call [2,3,4].

We evaluate the performance of the current traffic engineering criterion based on Erlang B formula leveraging vast amount of recent data from an operating cellular core network with over 15 million subscribers [1]. Also, we re-examine Erlang B assumptions and discuss whether Erlang B formula is appropriate on cellular core networks. We have chosen two CGS (Cellular Gateway System)s with most traffic and the MSC (Mobile Switching Center)s connected to them. Measurement is done in time scales of hours and seconds. Billing data in user profiles were processed to obtain the traffic volume at the resolution of seconds. We also developed a Block Generating Program (BGP) to get the results using billing data, similar to real network. Block rate has been measured and compared against the expected block rate from Erlang B table. In average, the measured block rate was higher than the expected block rate. To identify the cause of deviation, the traffic characteristic of the measured traffic is analyzed at the resolution of seconds and hours. The VMR (Variance to Mean Ratio) which represents the peakedness showed from 0.45 to 3.50 for seconds traffic. Erlang B table does not guarantee the target block rate but shows an average block rates. To guarantee the target block rate, we introduce compensated Erlang B Table had to be calculated by adding a new factor called *degree of confidence*, which enables one to specify the reliability of assigned trunks. This factor proves that current Erlang's B table corresponds to 50% in terms of degree of confidence.

The rest of the paper is organized as follows. Section 2 describes the measurement setup. Evaluation of current traffic engineering is performed in Section 3 by comparing the measured block rate against the expected block rate. Experiment in real network is analyzed in Section 4. In Section 4, we used COMNET and BGP simulation to overcome experimental limit in real network and proposed a method of compensation based on confidence rate to Erlang B table for Poisson and bursty traffics. Section 5 concludes this paper.

2 Measurement Setup

Figure 1 shows a typical configuration of telecommunication network supporting cellular phone traffic. Voice traffic from cellular phone is collected from base stations. *MSC (Mobile Switching Center)* is a hardware interface between a group of base stations and the wireline network. The traffic from MSCs may be aggregated into *CGS (Cellular Gateway System)*. Trunk engineering mainly concerns determining the capacity of links between MSCs and CGSs. We measured the traffic at each link connecting an MSC to a CGS at the resolution of seconds and hours.

The call processing capability per hour of the chosen switches is shown in Table 1.Two different types of measurement are performed. The first type is to record number of incoming and outgoing calls per hour for each switch for 3 months.One problem with the first data set is that its resolution is in hours and

Table 1. Switch Specification

Specification	MSC	CGS
Call Processing Capacity per hour	315,000	1,050,000
Max. Number of E1 Lines supported	315	2,400

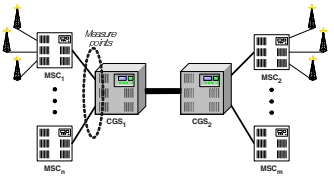


Fig. 1. General Topology of Cellular Service

only shows the aggregate number of calls for an hour. The second measurement is based on the billing data for each call and has the resolution of seconds.

3 Current Traffic Engineering Diagnostics

Current traffic engineering uses the Erlang B table to calculate the link capacity with a given target block rate. While the Erlang B table lists the amount of traffic (in the units of Erlang) to be carried for specific number of trunks at a target block rate, cellular service network uses E1 links with 31 trunks per each E1 link [3,8]. We compared the measured block rate represented in equation 1 against the expected block rate based on the Erlang B table.

$$P(\textit{block rate}) = \frac{\textit{number of TRK\_BUSY signals}}{\textit{number of call attempts per hour}} \tag{1}$$

Blocking occurs whenever the number of calls, in or out, exceeds the number of trunks available to support them. It is used primarily for determining trunk quantities in first-choice trunk groups in which, if all trunks are busy, a call overflows to another group, or never returns [10].

Figure 2 compares the measured block rate against the expected block. Note that many points are close to the origin that represents zero block rate. This is due to the fact that most links are assigned the number of trunks far greater than needed since only multiples of E1 link can be leased (E1 link consists of

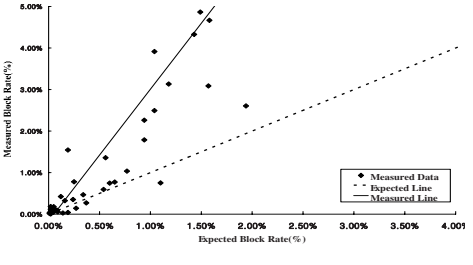


Fig. 2. Measured vs. Expected Block Rates

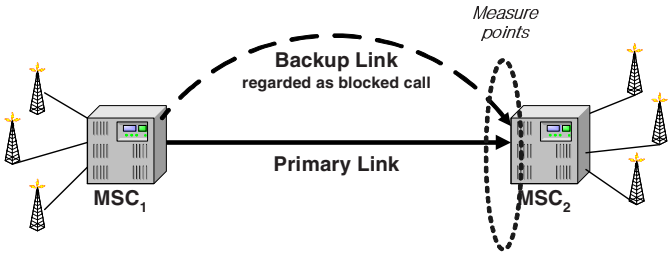


Fig. 3. Experimental Link Setup

31 trunks). The solid line represents a linear regression of the points while the broken line represents the expected line. If blocking occurs as expected from the Erlang's B formula the points should be clustered along the broken line. However, the points in Figure 2 show clear deviation from the expected line.

3.1 Methodology

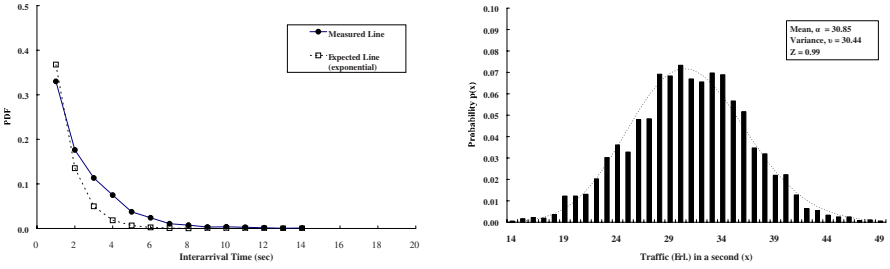
Because the current service cellular system is overprovisioned mostly to 0.1% target block rate, it is difficult to analyze the behavior the block rate. To examine the behavior of incoming calls and block rate, we have setup a experimental link as seen in Figure 3. The primary link which normally operates as main path, and the backup link which runs as alternative route that when calls are blocked from the main link they are returned to the alternative link and considered as blocked calls. The data at the resolution of seconds were gathered from the billing data that also records attempted calls (not all switching systems records attempted calls).

3.2 Analysis of Inter-arrival Times and VMR

To find this reason, we analyzed 1) Inter-arrival time distribution 2) VMR (variance to mean ratio) using billing data of each sample. It is useful for calculating skewness of nonrandom traffic [7,8,9,11]. In order to check whether the sample traffic follows Poisson distribution, we proved it with chi-square goodness-of-fit test for call inter-arrival time [5,13].

Figure 4 compares the PDF of the measured traffic against that of the inter-arrival times for the traffic measured at the resolution of seconds resembles the Poisson arrival. Chi Square verification can be used for determining if the traffic complies with the Poisson distribution, but we cannot conclude the characteristics of the traffic with it. Therefore, we analyzed VMR using billing data in order to get the traffic characteristics of the sample.

Peakedness of traffic has been found a useful characterization tool in blocking approximations and in trunk theory. The peakedness factor Z for any link is obtained by calculating the variance-to-mean ratio of the busy-hour traffic, as in



(a) Measured Traffic vs. Poisson traffic (b) PDF of the measured traffic

Fig. 4. Traffic Measurement Results

Equation 2. If Z is less than 1, the traffic is defined as *smooth* and it experiences less blocking than Poisson traffic. If the Z is larger than 1, then it is called *peaked* traffic and it experiences larger blocking than Poisson traffic [2,3,6,12,13].

$$Z = \frac{v}{\alpha} \tag{2}$$

4 Simulation

4.1 Compensation of Erlang's B Table

Erlang's B Table with Degree of Confidence for Random Traffic With the help of these results from the simulation we were able to add a new parameter to Erlang's B Table, called the degree of confidence.The block rate resulted a normal distribution which was possible to define the degree of confidence at different interval of the distribution. Table 3 shows the result of reassigned Erlang's B table adding degree of confidence from Table 2. It shows how much margins is needed to assure call blocks. Normal Erlang's B table resulted 50% of confidence which means that the target block rate can only support 50% of the measured block rate. So to allow more confidence, for example, if a service company wants the target block rate at 99% of confidence for 50Erl traffic they must add 6.5% more based on standard Erlang's B Table.

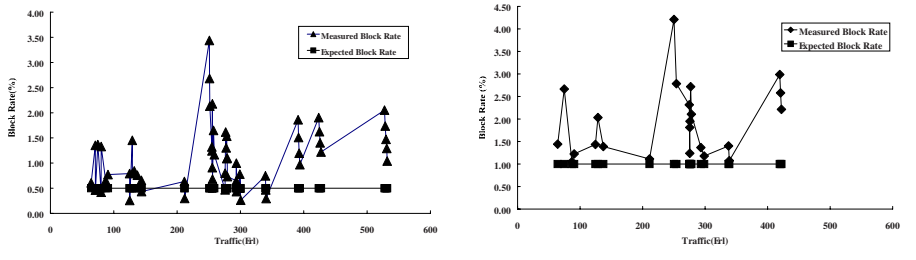
BGP (Block Generating Program) The flow of BGP process is divided into three steps. First gathering original billing data in seconds, second generating these data in BGP simulator and as a result we get attempted calls, AHT, traffic, block calls, block rate, etc from original billing data at different trunks we have set. Traffic characteristics percentage resulted as 22% for smooth 50% poisson and 28% peaked. We analyzed traffic characteristic applying VMR for billing data of total 108 samples. In results, VMR had a value from 0.48 to 3.50.

Table 2. Margin rate at different degree of confidence

1%	Margin Rate							
Block Rate	Degree of Confidence							
Traffic	99.9	99	95	90	80	70	60	50
50	7.8	6.3	4.7	3.1	3.1	1.8	1.8	0.0
1000	0.5	0.5	0.2	0.0	0.0	0.0	0.0	0.0

Table 3. Updated number of trunks using table 2

1%	Updated Number of Trunk					
Block Rate	Normal Distribution					
Traffic	Erlang B	99.9	80	70	60	50
50	64	69	66	65	65	64
1000	1029	1034	1029	1029	1029	1029



(a) Peaked traffic (0.5%) (b) Peaked traffic (1%)

Fig. 5. Measured block rate vs Expected block rate for Traffic characteristics

BGP Simulation Results Figure 5 shows one of the pattern of block rates at different traffic characteristics. It shows simulated block rates when the target block rate of 0.5% and 1%, and actually we compared the Erlang B expected block rate.

4.2 Compensation of Erlang B Table for Non-random Traffic

Heuristic Method We have derived a heuristic method in assigning trunks for peaked traffic.The three variables involved are traffic (T), block rate (B) and lines (L). Traffic (in Erlangs) is the traffic generated every hour and was collected during the busiest hour of operation of a cellular core system. Block rate is the percentage of dropped calls due to an insufficient number of lines being available. Lines are the number of trunks assigned. We proceed the following procedure to get the compensation factor.Steps of assigning L_C at diffrerent degree of confidence line L_E is the amount that Erlang B expects from the pertinent

Table 4. Compensation factor

1% Block Rate	Traffic (Erlang)	Degree of confidence (%)				
		99.9	90	80	70	60
Compensation Factor F	~180	6.7	4.5	3.3	2.0	2.1
	180~360	7.7	5.1	3.7	3.3	2.3
	360 ~	4.7	4.7	3.8	3.8	3.1

Table 5. Reassigned trunk applying compensation factor

1% Block Rate		Degree of confidence (%)				
Traffic	Erlang B	99.9	90	80	70	60
50	64	68	67	66	65	65
200	221	238	232	229	228	226
500	527	552	552	547	547	543

traffic and block rate, and line L_C is the compensation line that prevents all the possible blocks. Figure 6 shows results of block rate against traffic. Setting line L_C at different level we can calculate the degree of confidence.

$$L_E = Erl(T_M, B_E, x) \tag{3}$$

$$T_C = Erl(y, L_E, B_M) \tag{4}$$

$$L_C = Erl(T_C, B_E, z) \tag{5}$$

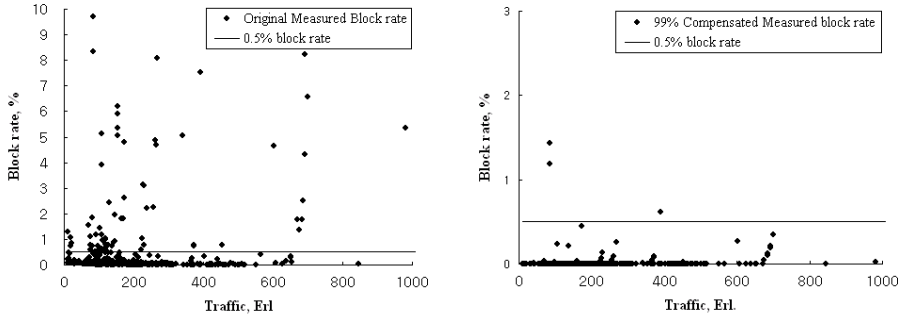
$$F = \frac{(L_C - L_E)}{L_E} * 100 \tag{6}$$

B_M stands for measured block rate, T_C is the measured traffic, B_E expected block rate, L_E expected line T_C compensated traffic and L_C for compensated line.As a result we can get the compensation factor by calculating the ratio of L_E, Erlang B expected line and L_C, derived from equation 3. F ,the compensation factor (%), is derived from Equation 6 .

Applying equation 3,4,5 and 6 the results of compensation factor are listed in Table 4. And with this factor we reassigned trunks as seen in Table 5. The results satisfied after applying heuristic method in real network. Fig 6(b) had a 96% satisfaction over 99%.

5 Conclusion

In this paper we have analyzed Erlang B theory which is currently used in cellular core networks. Experiment and analysis were made with real data in real networks. Compensation of Erlang B Table under the results of simulation.



(a) Original Measure Block Rate (b) 99% compensated

Fig. 6. Results applying in real network

Adding a new factor, degree of confidence reassigning Erlang B table. Block Generating Program was made to get results similar to real experiment using billing data. Block rate tends to be higher than the expected block rate form Erlang B currently in use. Non-poisson traffic that is bursty takes up about 30% for cellular core networks. Poisson and bursty traffic occupied 80% of total traffic, and Erlang B table needs to be compensated in order to be applied to cellular core networks.

According to our analysis, Erlang B theoretical block rate is not identical to measured block rate, and we concluded that it is due to the traffic characteristics. Therefore, we added the degree of confidence to Erlang's B table for communication quality and enables service companies to consider directly some variables such as communication quality and cost as they assign trunks. We introduced alternative calculation method,such as heuristic that considers VMR value including peakedness effects.

References

[1] SK Telecom, "Annual Report 2001", http://www.sktelecom.co.kr/english/index.html, 2001.
[2] James R. Boucher "Traffic System Design Handbook: Timesaving Telecommunication Traffic Tables and Programs", IEEE Press, pp. 1-78, 1993.
[3] Haruo Akimaru, Konosuke Kawashima, " Teletraffic- Theory and Applications", 2nd Edition, Springer Verlag, pp3-55, January 1993.
[4] Villy B. Iversen, "Teletraffic Engineering Handbook", ITU-D SG 2/16 & ITC, 2002-09-06.
[5] Robert S. Cahn "Wide Area Network Design - Concepts and Tools for Optimization", Morgan Kaufman Publishers, Inc, pp. 11-142, 1998.
[6] Derek Lam, Donald C. Cox, Jennifer Widom, "Teletraffic Modeling for Personal Communications Services", IEEE Communications Magazine, pp79-87, February 1997.

[7] S. Molnar, Gy. Miklos, "Peakedness Characterization in Teletraffic", IFIP TC6, WG6.3 conference Performance of Information and Communications Systems, PICS'98 Lund, Sweden, May 25-28, 1998
[8] Richard Parkinson, "Traffic Engineering Techniques in Telecommunications", INFOTEL SYSTEMS CORP. http://www.infotel-systems.com.
[9] D.Everitt, "Traffic capacity of cellular mobile communications systems" Computer Networks and ISDN Systems, 20, 447-454, 1990.
[10] F.Barcelo and J.Jordan. "Channel holding time distribution in cellular telephony." The 9th International Conference on Wireless Communications (Wireless'97), Vol. 1,pp.125-134,Alberta,Canada, 9-11 July, 1997.
[11] Yuguang Fang, Imrich Chlamtac. "Teletraffic Analysis and Mobility Modeling of PCS Networks" IEEE 1999.
[12] Roger I. Wilkinson, "Theories for Toll Traffic Engineering in the U. S.A", Bell System Technical Journal, vol. 35, pp. 421–513, 1956.
[13] Victor S. Frost, Benjamin Melamed, "Traffic Modeling For Telecommunications Networks", IEEE Communications Magazine, pp 70-81. March 1994.

Differentiation Mechanisms over IEEE 802.11 Wireless LAN for Network-Adaptive Video Transmission

Jaeyeon Lee and JongWon Kim

Networked Media Lab. Department of Information and Communications
Kwang-Ju Institute of Science and Technology (K-JIST)
Gwangju, 500-712, Korea
{jylee,jongwon}@netmedia.kjist.ac.kr

Abstract. In this paper, differentiation mechanisms coordinated from both application and network aspects are adapted to realize a network-adaptive video transmission over the wireless LAN. To support the required QoS differentiation for loss/delay sensitive video streams, a relative priority index (RPI) is assigned to each video packet according to its relative importance. At the MAC layer, a modified version of IEEE 802.11e EDCF (enhanced distributed coordination function) is utilized that can provide differentiated services according to access categories. Moreover, by using multiple RED (random early detection) MAC queues with WTP (waiting time priority) scheduler, loss/delay differentiation can be achieved more efficiently. A NS-2 network simulator is utilized to verify that the proposed differentiation mechanism can assist the end-to-end performance of wireless video transmission over the IEEE 802.11 wireless LAN.

Keywords: End-to-end wireless video, network adaptive video transmission, QoS of IEEE 802.11 WLAN, MAC differentiation mechanisms, joint source-channel error control, and packet prioritization.

1 Introduction

A wireless LAN (WLAN) - especially IEEE 802.11 based - is becoming popular nowadays due to wide spread of ubiquitous networking environment. Thanks to the development of wireless technology, WLAN began to deliver more bandwidth-intensive, delay-stringent media contents like streaming video. However, to guarantee successful transmission of streaming video contents over the hostile WLAN environments, where the bandwidth is scarce and channel conditions are highly fluctuating, a flexible and network-adaptive transport is required. The required network adaptation should be realized by coordinating both application and network layers. It requires the WLAN environments to provide QoS (quality of service) for video applications as being done in IEEE 802.11e [3].

In this work, by applying differentiation mechanisms in the MAC (medium access control) layer of IEEE 802.11 WLAN, a source-channel network adapta-

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 553–562, 2004.

tion is investigated employing a dynamic network adaptation framework proposed in [6]. The required network adaptation is guided by the relative priority index (RPI) for standard-based video codecs (e.g., ITU-T H.263+, ISO/IEC MPEG-4) [1]. For the required differentiation in wireless networks, a simplified version of enhanced IEEE 802.11e MAC is adopted from [4, 5]. Thus, the suggested framework is not limited to a specific layer. Instead, every layer from the application layer to the MAC layer of IEEE 802.11 WLAN has been tied by the proposed network adaptation to assure quality video delivery over the WLAN. To evaluate the proposed framework, network simulation (performed in NS-2 simulator over the distributed WLAN network with a group of access point and wireless terminals) is combined with the end-to-end video performance evaluation. Streaming transmission of prioritized packet video (H.263+ video codec with error concealment) is assumed for the evaluation and the impacts of various control parameters are experimented to verify feasibility and performance of the framework. Results show that the proposed framework can maintain transmission performance effectively, thus provide better quality of streaming video over the QoS-provisioned IEEE 802.11 WLAN environments.

The reminder of this paper is organized as follows. Based on the limitations in transmitting streaming video over the WLAN, a dynamic network adaptation framework to transmit video over the WLAN is described in Section 2. In this section, components of the framework are explained in detail. Simulation results follows in Section 3 with analysis on the results. Finally we conclude this paper in Section 4.

2 Network-Adaptive Video Transmission Framework for WLAN

To transmit streaming video over the WLAN, we need to address many problems such as unstable channel condition with fading/shading effect, power consumption of a wireless terminal, out-of-order or no packet reception during a handoff, contention for wireless channel resource, and others. Among these challenges, in this paper, we are focusing on the contention for wireless channel by investigating a basic service set (BSS) infrastructure only with an access point (AP) and multiple wireless terminals (WTs). An AP serves as a gateway between outside networks and a BSS to which it belongs. Here, packets destined to WTs are competing with other traffics to get channel access while various WTs are also waiting for their chance to deliver packets. In normal IEEE 802.11 WLAN, the AP temporarily contains packets in the single MAC queue until it obtains transmission opportunity. This single MAC queue can cause so called head-of-line (HOL) problem. Due to lower priority packets blocking the way, arrived packets can be dropped. All these burst packet loss can cause severe video degradation.

Thus, in this paper, we first focus on investigating the impact of WLAN MAC - both conventional and modified for differentiated services - to the end-to-end video performance. Also, to suggest an effective way to utilize the differentiated MAC services, we propose a network adaptation framework [6] tailored for the

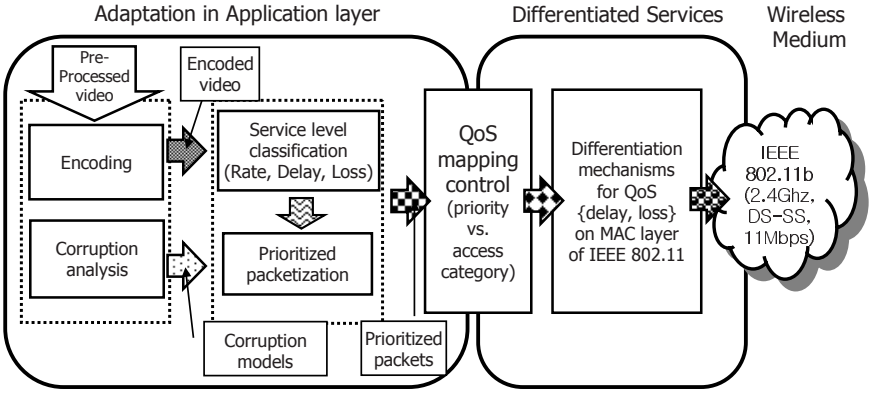


Fig. 1. Proposed framework for network-adaptive wireless video over the WLAN

IEEE 802.11 WLAN. As described in Fig. 1, the proposed framework mainly consists three major parts: prioritized packetization at the application layer, differentiation mechanisms for loss/delay in the WLAN MAC, and a network-adaptive QoS mapping control. In the followings, detailed explanations of these three parts will be given.

2.1 Prioritized Packetization and QoS Mapping

With the MAC differentiation to be explained later, the streaming video applications are subject to several classes of network services in terms of loss and delay. In this kind of streaming environment, loss and delay are most important QoS factors to decide the quality of reconstructed video [1]. Note that, although another QoS factor, bandwidth, is not explicitly considered, different loss and delay actually means differentiated availability of bandwidth per given period of time. Thus, in this paper, we assign the priority to each packet in terms for loss and delay as studied in [1] so that each packet can reflect its influence to the end-to-end video quality.

The procedure of prioritized packetization is shown in Fig. 2. Following application layer framing principle, encoded video data is packetized per each GOB (group of block). This variable-size packet is then associated with MSE (mean square error) value that represents the impact of packet loss. Assuming knowledge on the adopted error concealment scheme, the MSE is calculated per each GOB to quantify the quality degradation of reconstructed video due to packet loss. The MSE is calculated as follows:

$$MSE(n,i) = \frac{1}{N} \sum_{n \in video} |\widehat{R}_n^i(x,y) - R_n(x,y)|^2, \tag{1}$$

where $MSE(n,i)$ is the mean square error of n_{th} frame when i_{th} packet is lost and concealed. Using (x,y) as a 2-D coordinate in a video frame, $\widehat{R}_n^i(x,y)$

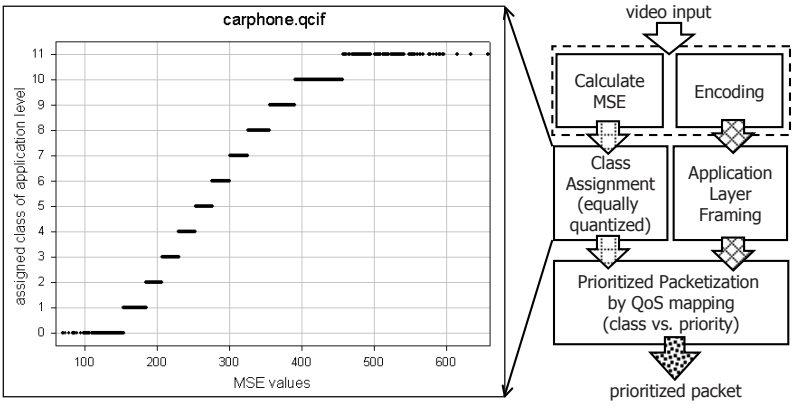


Fig. 2. Example of prioritized packetization applied in the application level

and $R_n(x, y)$ are the corresponding reconstructed n_{th} frames when i_{th} packet is lost and kept, respectively.

As shown in Fig. 2, MSE's of video packets are distributed over wide range of values. Note that these values only represent the relative importance of packets seen from this streaming application and thus it is called as relative priority index (RPI). To ease mapping to the underlying networks where only limited numbers of classes are usually provided, they are categorized into several priorities. In this paper, we are grouping the same number of packets into 12 priorities as shown in Fig. 2. These 12 categories of priorities are then uniformly mapped to three classes of differentiated MAC services (i.e., 4 categories to each class). This QoS mapping process that determines how many packets are allocated to each class is not an easy job. It is related with issues such as pricing mechanism, network status, and/or required video quality as discussed in [7]. Also, if we want to make this mapping more effective, feedback information from network or receiver can be utilized to realize dynamic QoS mapping.

2.2 Multiple MAC Queues with WTP Scheduler and Modified EDCF

As discussed in [2] and emerging IEEE 802.11e standard [3], people have started to manipulate differentiated backoff time, DIFS (distributed inter-frame space) interval, and/or maximum frame length to realize differentiated MAC. The IEEE 802.11e standard is mainly consisted with two parts - EDCF (enhanced distributed coordination function) that improves current DCF for a queue-based service differentiation and HCF (hybrid coordination function) that gives a TXOP (transmission opportunity) to a WT with higher priority by adopting PCF (point coordination function) concept [3].

Recently, by extending [2], we have tried to enhance the MAC to guarantee proportional delay differentiation according to the priority [5]. Extending this

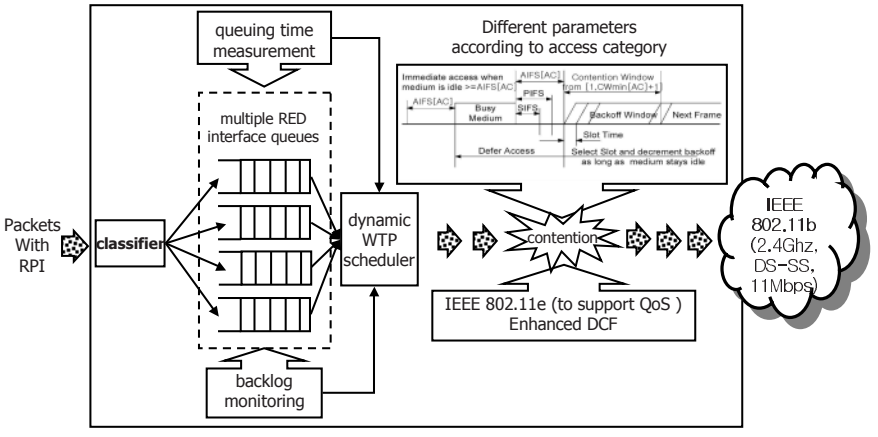


Fig. 3. Implemented MAC modifications

work, we add multiple (i.e., actually 4) MAC queues with waiting time priority (WTP) scheduling to a modified version of EDCF. Of the 8 ACs(access categories) of IEEE 802.11e - best effort, video probe, video, voice, and others, proposed modification only provides 4 ACs for the sake of simplicity: three for video and one for others. Also, the role of virtual collision handler used in the IEEE 802.11e is partly compensated by having multiple MAC queues with the WTP scheduler.

Moreover, these queues are implemented as RED (random early detection) queue. RED is known to be effective in avoid congestion by dropping enqueued packets randomly in face of possible overflow. Later we are planning to extend this queuing part further by adopting a policing mechanism linked with admission control.

Among packets located at the head of these multiple MAC queues, the WTP scheduler selects a packet that has the highest parameter $P_{(i,k)}$. By using the WTP scheduler with multiple RED queue, not only the delay differentiation is achieved, but also packet drops are differentiated due to different dequeuing speed. Parameter $P_{(i,k)}$ is calculated by Eq. (2):

$$P_{(i,k)} = \frac{W_i \times T_{(i,k)}}{B_{(q_i,t)}}, \tag{2}$$

where $P_{(i,k)}$ is the parameter of k_{th} packet with i_{th} AC, W_i is the weighting factor of i_{th} AC, $T_{(i,k)}$ is the queuing delay of k_{th} packet with i_{th} AC, and $B_{(q_i,t)}$ is the backlog of a queue with i_{th} AC at time t.

For the MAC itself, we assign parameters shown in Table 1 for the modified EDCF [3]. Among the parameters, contention window size that decides backoff time of a WT is calculated with different CW_{min} and CW_{max}. Instead of the same P_j in the conventional MAC, multiplication factor P_j value is also changed as shown in Table 1. Theoretically, WTs with the same CW_{min}, CW_{max}, and P_j

Table 1. Parameter of modified EDCF for loss/delay differentiation($A, K > 1$)

	lowest video AC	intermediate video AC	highest video AC	best-effort AC
CW_{min}	511	255	1	511
CW_{max}	1534K	511K	255K	1023K
dot11ShortRetryLimit	7	4	21	7
ShortRetransmissionLimit	7	4	21	7
Priority factor P_j	3A	2A	1A	1A
Weighting factor W_i	4	2	1	0.5

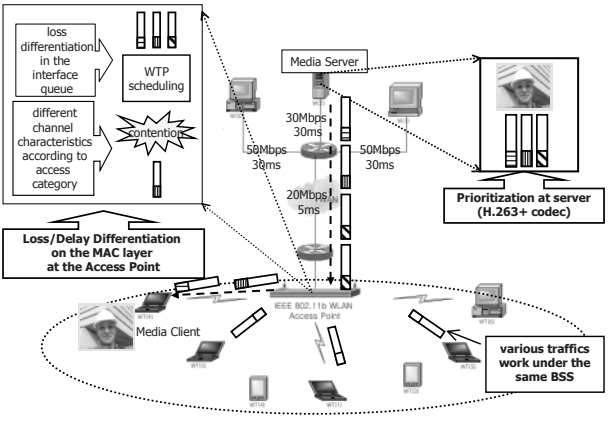


Fig. 4. Simulation topology

have the same opportunity to access the wireless medium. However, by differentiating these values, chances to access to the medium become differentiated. After all, amount of transmitted data becomes differentiated according to the AC.

3 Simulation Results and Analysis

Simulation topology to evaluate performance of the proposed framework is shown in Fig. 4. The standard-based H.263+ encoding/decoding and the network simulator NS-2 are used to evaluate the end-to-end video performance. When the H.263+ encoder encodes video, MSE value of each GOB for the packet loss corruption is calculated and stored as a data file. Because each GOB is packetized into a separate packet in this simulation, priority will be assigned to each packet according to this relative loss importance. At the receiver side, error patterns generated from the NS-2 simulations are used to decide whether the packets are lost or not. In addition, to focus on the wireless channel aspects, it is assumed that the wired network environment is ideal and there exists no packet loss.

At this stage, we have used SNR scalability mode of H.263+ and have applied these packet losses to the enhancement layer only. By doing this, we can complete

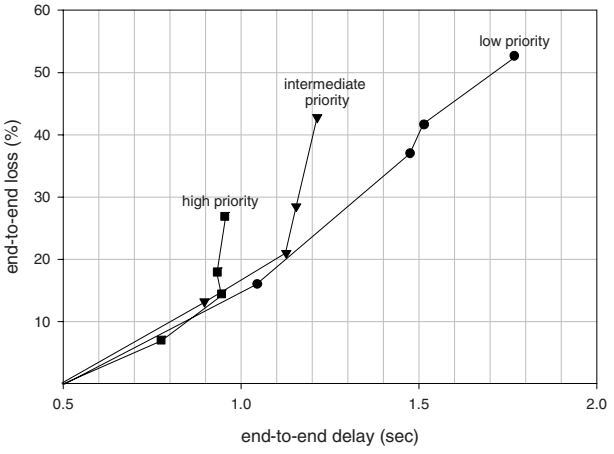


Fig. 5. Differentiated loss/delay of the network classes according to the amount of downlink traffic (video traffic: 64kbps, 5% uniform channel error, RED queue)

the video decoding procedure regardless of packet loss level that goes up to 100%. However, we also try to set the amount of base layer small compared to the total bit rate so that it does not affect the result too much. PSNR (peak signal to noise ratio) between original and reconstructed video is calculated to quantify how much video quality is degraded by the packet loss during transmission. Note that all PSNR results below are having PSNR of base-layer only as minimum.

As discussed already, modified parameters for differentiation are listed in Table 1. Even though modified EDCF used in this simulation is not the same with IEEE 802.11e standard, we obtain service infrastructure for loss/delay differentiation by adjusting them. Also, the WTP scheduler with multiple RED queues contributes to the loss/delay differentiation in the MAC layer. Then we Insert 5% uniformly distributed random error. Total length of each queue in a AP is limited to 200 packets and available bandwidth is set to 11Mbps (IEEE 802.11b).

3.1 Loss/Delay Differentiation in the Network Aspect

By using the proposed MAC modification, different network service classes are obtained as shown in Fig. 5. We can observe that both loss and delay are differentiated appropriately by increasing traffic amount in both downlink and uplink direction (uplink (Mbps): 4.18 → 5.225 → 5.747 → 6.792, downlink (kbps): 156 → 211 → 237 → 293). With Fig. 5, it is figured out that the amount of traffic is one of the factors which decide loss/delay characteristics of network. For that reason, there needs an admission control with policer for the WLAN to avoid a network congestion even with differentiated service. Consequently, enhancement of loss/delay characteristics with proposed adaptation framework is shown in Table 2. By contrasting with severe network congestion with conventional IEEE 802.11b WLAN, avg. loss and delay become profitable to stream-

Table 2. End-to-end Avg. loss, delay, jitter comparison experienced per AC

	Avg. loss (%)	Avg. delay (sec)	Avg. jitter (msec)
Conventional MAC	79.4476	0.3246	8.5679
Lowest video AC	80.9421	0.4837	18.3021
Intermediate video AC	62.3142	0.2649	9.4979
Highest video AC	27.6190	0.1663	7.2202
Best-effort AC	94.2354	1.5423	89.4125

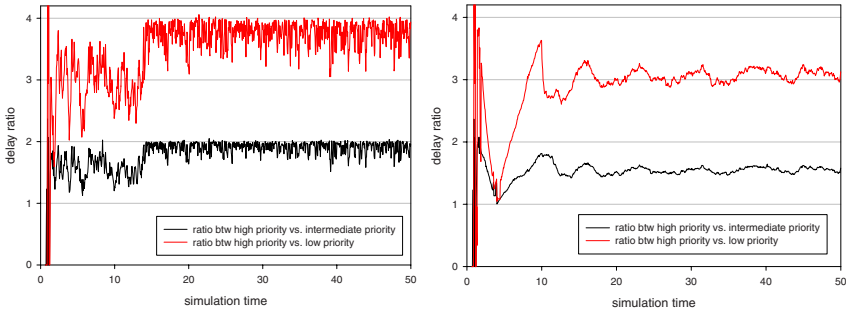


Fig. 6. Proportional end-to-end delay differentiation according to the load of the MAC queues (left: total queue length = 200 packets (heavier load), right: total queue length = 2000 packets (weaker load))

ing video with proposed network-adaptive video transmission framework (total background traffic: 4.8Mbps, no channel error, video traffic: 100kbps).

Also, differentiated delay characteristics are obtained when adding WTP scheduler with multiple MAC queues in the AP. Note that this proportional delay differentiation mechanism works better under the heavy network load [8]. Left figure of the Fig. 6 shows a result of proportional delay differentiation when total queue length is 200 packets (4 queues and the length of each queue is 50 packets.). Right one of that shows a result that the total queue is 2000 packets (the length of each queue is 500 packets.). The higher priority a packet has, the shorter average propagation delay the packet has. Additionally, jitter also becomes smaller with higher priority.

3.2 Video Quality Enhancement with Proposed Framework

As mentioned, loss and delay experienced by a video packet is main factors to decide its usability in the decoding at the receiver. For time-critical applications, experienced delay longer than allowed threshold is also considered as loss of packet. However, in this simulation, for the sake of simplicity, we assume that a sufficient size of decoding buffer is available and no late arrival packets are considered as lost. Also, packet loss is applied to enhancement layer only. Total bit rate of used sample sequence is 366.58kbps for the Carphone

Table 3. PSNR performance comparison

	Conventional MAC	Modified MAC with proposed adaptation	Base-layer only PSNR
Carphone	30.81	36.52	30.58
Glasgow	28.40	34.48	28.18

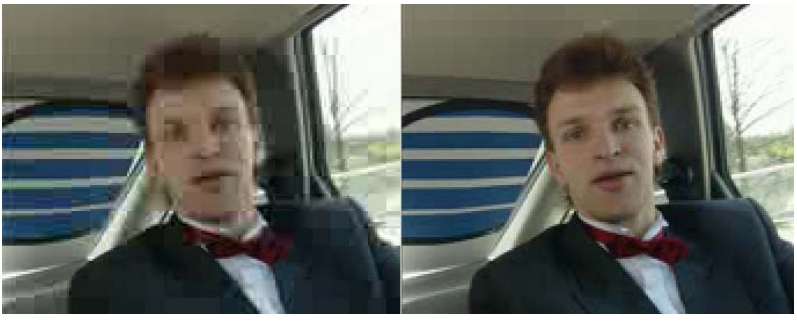


Fig. 7. Reconstructed video quality comparison (1_{st} frame of the Carphone sequence)

and 633.37kbps for the Glasgow, respectively. Bitrate for base and enhancement layers are 46.44/320.14kbps for the Carphone and 87.80/545.57kbps for the Glasgow, respectively. Under these conditions (uplink background traffic: 4.2075Mbps, downlink background traffic: 678.48kbps, 5% uniform distributed channel error, total queue size : 200 pkts), the PSNR of reconstructed video is compared in Table 3. For both sequence, more than 6 dB difference is observed. If the loss is applied to base layer, it is almost impossible to see streaming video under the conventional WLAN environment. Finally, reconstructed Carphone video is compared in Fig. 7. As expected, subjective video quality is significantly improved with the proposed approach.

4 Conclusion and Future Works

In this paper, a network-adaptive video transmission framework over IEEE 802.11-based WLAN has been introduced. To support the required QoS differentiation for loss/delay sensitive video stream, a relative priority index (RPI) is assigned to each video packet according to its relative importance. At the MAC layer, a modified version of IEEE 802.11e EDCF (enhanced distributed coordination function) and multiple RED (random early detection) MAC queues with WTP (waiting time priority) scheduler are used for efficient loss/delay differentiation. By adopting the proposed framework, it's shown that streaming video transmission can be feasible even over the WLAN environment. As a result, higher PSNR of reconstructed video was shown as the proof of quantitative performance improvement. As a future work, we need to refine the simulation framework further and simulate more error-resilient but delay-constrained video

applications. Also, by adding end-to-end feedback mechanism, improved QoS management (including dynamic QoS mapping) will be exercised to combat dynamic changes in the source characterization and underlying network status.

Acknowledgement

This research was supported in part by University IT Research Center Project and in part by K-JIST.

References

[1] J.-G. Kim, J. Kim, and C.-C. J. Kuo, "Video packet categorization for priority delivery to enhance end-to-end QoS performance," in *Proc. Packet Video Workshop '2002*, May 2002.
[2] I. Aad and C. Castelluccis, "Differentiation mechanisms for IEEE 802.11," in *Proc. IEEE INFOCOM*, Apr. 2001.
[3] IEEE 802.11 standard working group, "Medium access control (MAC) enhancement for quality of service (QoS)," IEEE 802.11e/D4.1 draft, 2002.
[4] S. Choi, J. del Prado, S. Mangold, and S. Shankar, "IEEE 802.11e contention-based channel access (EDCF) performance evaluation," in *Proc. IEEE ICC'03*, May 2003.
[5] K. Yoon and J. Kim, "Dynamic proportional delay differentiation scheduling over IEEE 802.11 wireless network," in *Proc. IEEE HSNMC 2003*, July 2003.
[6] J. Kim and J. Shin, "Dynamic network adaptation framework employing layered relative priority index for adaptive video delivery," in *Proc. IEEE Pacific-Rim Conference on Multimedia (PCM'2002)*, Dec. 2002.
[7] J. Shin, J.-G. Kim, J. Kim, and C.-C. J. Kuo, "Dynamic QoS mapping framework for relative service differentiation-aware video streaming," in *European Transactions on Telecommunications*, 2001.
[8] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: Delay differentiation and packet scheduling," in *Proc. ACM SIGCOMM '99*, Sept. 1999.
[9] UCB/LBNL/VINT, *Network simulator – ns (version 2).* http://www.isi.edu/nsnam/ns, 1998.

Design of ABEL Route Recording System Base on BGP for Network Management and Application Software

Masayuki Tabaru[1], Koji Okamura[2], Seomee Choi[3], Jaehyuk Ryu[3], and DaeYoung Kim[3]

[1] Graduate School of Information Science and Electrical Engineering
Kyushu University, Japan
tabaru@dontaku.csce.kyushu-u.ac.jp
http://dontaku.csce.kyushu-u.ac.jp/~tabaru/
[2] Computing and Communications Center
Kyushu University, Japan
[3] Chungnam National University, Korea

Abstract. Recently many advanced communication applications which need wide bandwidth appear by the advancing the development of Internet environment. And the demand for real-time video transmission for conference system is increased. Currently end users of these application focus on only network bandwidth. Stable real-time transmission needs such Internet environment that has enough wide bandwidth and is controlled by stable routing information. If routing control is not so stable, bandwidth, delay and jitter of communication are not stable and communication would be sometimes disconnected. Then statistics information of routing can be useful to know the network is stable or not. We focus on inter domain routing information such as routing information of BGP between some AS and the other AS. We use BGPView to collect such routing information. But the routing information is usually large amount of data size. We make method for process large data and make database system named ABEL based on that information. Because of that method this database system respond quickly. It's useful for network administrators who want to know the stability of Internet and some application which need knowledge about stable network.

1 Introduction

Many advanced communication applications which need wide bandwidth have appeared in recent years and the demand for real-time video transmission for one of such applications are increased. In past days, we couldn't use such the communication softwares over the Internet because its bandwidth was too small. The advanced development of network technologies enabled this situations. Network bandwidth has become gigabit class in recent years. Because of this improvement of networking environment, real-time transmission for video conference system have spread for commercial, medicine and so on. In this situation, users of these

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 563–572, 2004.

applications focus on only network bandwidth. But stable real-time transmission needs such Internet environment that has enough wide bandwidth and is controlled by stable routing information. If routing control is not so stable, delay and jitter of communication are not stable and communication would be sometimes disconnected.

For example, if someone use Digital Video(DV) format for real-time video transmission for conference presumptively, he needs a network which has enough wide bandwidth to through the DV format data which sizes are about 30Mbps. He had researched the route which is own network to peer network have enough wide bandwidth to through DV format data. But if the route changes during video conference and that route can not have enough wide bandwidth to through data which sizes are about 30Mbps, the network bandwidth becomes overflow and he can not continue the conference. On the other hand if that route has enough wide bandwidth to accept 30Mbps data, he can not see peer for a minutes or two because network configuration takes a minutes or two to change route and the connection is disconnected for that time.

Therefore we collect AS-PATH informations which have route information between some AS and the other AS to communication applications. And we analyze that routing informations and show statistic information of route stability information. After that we design database system used that information to respond AS-PATH stability. AS-PATH statistic informations are important for DV connection for the previous reasons. The administrator who uses DV connection can avert the time of which AS-PATH has been unstable to use these informations.

2 How to Collect and Manage Routing Informations

We exchange informations in the Internet based on routing information. Routing informations include the route of each IP address prefix and routers use these informations for decision of selecting next hop.

Routers are put on bound of networks and holds routing informations to send packets for right direction. Routers use protocols to introduce routing informations. These protocols are classified into Interior Gateway Protocols(IGPs) and Exterior Gateway Protocols(EGPs). IGPs are protocols that control routing in Autonomous system(AS) and RIP2 and OSPF are representative examples. EGPs control routing that between ASes and BGP-4[1] is representative example. AS is the set of networks which has same management policy. In general, routing informations are structured based on messages exchange between routers. Sending routers introduce and send messages under the order of the protocol. Receiving routers analyze messages under same order and make or update routing informations. Each router send informations that it has. Message exchange continue to one to one and finally all of routers received messages. Because of that each router has routing informations that include all of networks.

To collect routing informations, one way is pick up that informations by routing informations which routers have directly. Routers use software to exchange

messages and manage that informations to introduce routing informations. To access that software and use command to show that informations. Another way is use software which behaves in a router. It can exchange messages between routers. If this software can record logs of message exchange, we analyze logs of messages and we can get routing informations.

In the Internet, networks state changes momentarily and the routing informations are also changed dynamically. Then you have picked up routing informations once, routing informations may already be changed. If you want to collect current routing informations, you have to continue to observe or pick up that informations.

In this paper we consider AS-PATH's stability. AS-PATH is the route of an AS to the other AS. That information is updated by routers which exchange BGP-4 messages. AS-PATH informations are included in UPDATE messages ordered BGP-4. These messages have list of AS-PATH of each IP address prefix and withdrawn IP address prefixes. These informations are processed by routers and introduced routing informations. If we can pick up all of routing informations easily, it is a best way to analyze AS-PATH stability. Only if we can pick up only contents of messages which are exchanged by routers, we can pick up routing informations. Because the format of these messages are ordered the protocol, we have introduced routing informations ourselves to order the protocol and get routing informations.

2.1 BGPView

We need to correct AS-PATH informations to bring out AS-PATH stability. To realize that thing we use software that named BGPView[2]. BGPView is the software that observes BGP routers and takes log of messages which have exchanged by routers created by Internet Initiative Japan Inc. It makes peer connection to each one of BGP routers and exchanges messages like router to log messages. We use this software to collect contents of messages which exchanged by routers to introduce routing informations and analyze these informations to check AS-PATH stability.

2.2 Basic Files and Methods of Analysis

We have introduced some files based on BGPView-log files to introduce database system which responds AS-PATH stability.

We assume that this database system accept queries which introduced by users who need those informations and has number of destination AS or IP address and optionally has time data and the day of week which user wants to know the stability. When it accepts query, it is necessity responds quickly and exactly. Therefore informations based on database system are desired to collect short time interval and for a long time period. Then we collect and process data per ten minutes and preserve these data for a year. To realize that situation our algorithm needs a technique to introduce files which used by database system to answer queries.

```
3                                   :number of routes
1       2523 2497 2828 11796        :AS-PATH
2       2523 2497 3356 3908 11796
3        WITHDRAW
```

Fig. 1. Route information file example

First, BGPView-log files have only list of correspondences of IP address prefix to AS-PATH and withdrawn IP address prefixes which is recorded per ten minutes. This list consist of informations which are included UPDATE massages of BGP and have exchanged by routers in user-established interval time. Then this list may include several informations of same IP address prefix and does not have informations about all of IP address prefix. Therefore we introduce AS number to AS-PATH file which has list of correspondences of AS number to AS-PATH and IP address prefix to AS number file which has list of correspondences of IP address prefix to AS number based on BGPView-log files to compare the data which are in old AS number to AS-PATH and IP address prefix to AS number file with data which are in BGPView-log file and update old informations or add new informations.

Next, we introduce some files which data size is small. Because there are a lot of networks and ASes in the Internet. AS number to AS-PATH file and IP address prefix to AS number file become large amount of data and need large disc space. If we use these files for database system which responds route stability, it takes long time to access files and process data. Then it can not respond quickly.

We have introduce two kind of files for each ASes based on AS to AS-PATH file. One is the route information file. This one records list of AS-PATH which has be recorded ever since the database system have started to collect routes. We read informations of AS number to AS-PATH file one by one and take AS number and AS-PATH which recorded per ten minutes. If we find new AS-PATH of an AS in AS number to AS-PATH file to reference old route information file, we have added new route to route information file.

The format of route information files are number of routes that is integer number and list of AS-PATH with route number follows with this number. When we will introduce new route information files, if an AS number which was recorded before but we can not find this AS number in AS number to AS-PATH file, then we think the route to the AS is withdrawn and we record "WITHDRAW" in route information file of the AS.

At the same time we introduce number information files that record informations which are how often each route which is recorded in route information files. We collect which route of each AS is used in this time and make number information files with this information.

Each number information file format is 144í7 matrix. Rows of matrix hold informations of one day because one day is equal 1440 minutes and we process data per ten minutes. Columns of matrix mean 7 days of the week. The place of each element shows time and the day of week. The detail of an element is

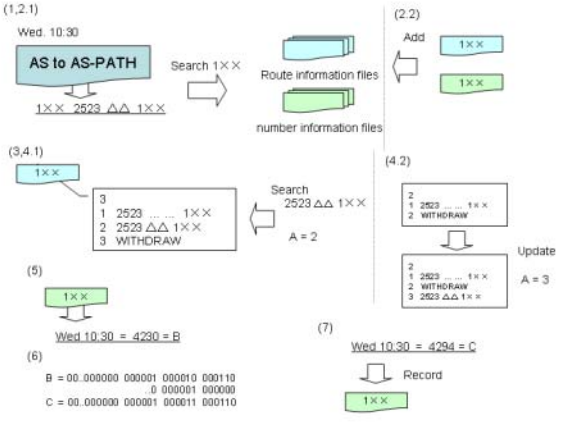


Fig. 2. How to introduce number information files

integer of 64bit. We collect what a number of times each route which recorded in route information file are recorded in six bits. Because of one year has about 53 weeks. Six bits can show from zero to sixty-three. It can collect data for a year and 64bit can collect ten routes in one integer number. We assume AS-PATHs of many ASes have changed under ten times for a year. If AS-PATHs of an AS change over ten times, we will introduce files which has additional informations.

How to introduce number information files:

1. Take from each AS data from AS number to AS-PATH file and time information.

2.1. Open and read the route information file and the number information file that correspond data's AS number.

2.2. If these files do not exist, introduce new route information file and number information file of which name correspond data's number. Elements of the matrix in this number information file are 0 without element that correspond time information. The value of the element is 1.

3. Search route informations whether the route which is same as route of data exists in the route information file or not.

4.1. If same route exists, take the number of the route(that displayed A).

4.2. Else add new route with number in the route information files and take this number(that displayed A).

5. Take data from number information file that corresponds time information(that displayed B).

6. Shift value 1 $(A - 1) \times 6$ left. And add this value with B. The answer displayed C.

7. Record C in number information file instead of B.

We analyze AS-PATH stability based on these files. If an element of the number information file of an AS is [123][3] of matrix is equal 4230.

```
[123][3] = 4230

Time        : 123/6 = 20 and 123 mod 6 = 3
Day of week : 0:Sun,1:Mon,2:Tue,3:Wed,4:Thu,5:Fri,6:Sat
4230 = 00 ... 000000 000001 000010 000110
 (63 = 00 ... 000000 000000 000000 111111)
```

Fig. 3. An example of the data which is in number information file

First, rows of matrix indicate time of a day. That information is found by simple calculation. If the first element of matrix is equal 123, 6 into 123 is 20, remainder 3. The quotient of the number which divided by 6 indicates hour and the remainder multiple 10 indicates minutes. Second, columns of matrix indicate the day of week that figure shown. Finally, we process decimal to binary conversation and each 6 bit shows the number of recorded times. In the instance, number 1 route is recorded six times, number 2 route is recorded twice and number 3 route is recorded once. These numbers match the numbers in route information file. To pick up each value of route uses bit shift operation and logic operation AND with 63. In this way we have got informations of one period and we have done iterative addition of this information for some periods which are needed for stability information and calculated usage rate of each route. This results are basic responses of AS-PATH stability informations. But this calculation takes very short time because of the simple operations as we mentioned above.

We will plan to collect this information for one year. But we will be able to backup these files easily because of the size of these files are small. It's important for long time usage of the database. If basic routing informations are used for the database system, it needs large amount of file recorded spaces. And the database system needs to record of long standing, the amount of data is too large. It is difficult to keep and backup data. But the files have been introduced by our method are used integer number can record one year informations, because disc space of these files needs are little or nothing increased. To use this method of analyze Bgpview-log files, the database system does not need to search large amount of data. Because of that we designed database system which responds quickly and does not need large amount of disc space.

3 Design of ABEL

A Route Recording System based on BGP for Network Management and Application Software(ABEL) is database system of AS-PATH. The data is collected per ten minutes. The database can respond what route and what ratio of each route is used and this information has indicated AS-PATH stability.

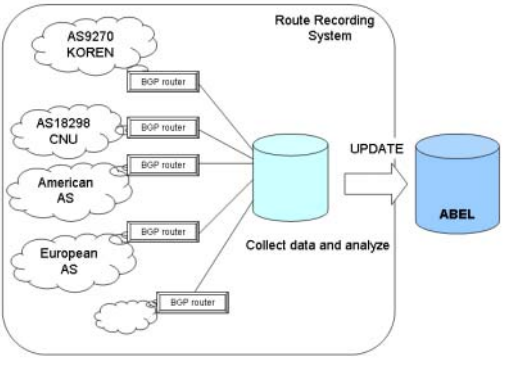


Fig. 4. Route Recording

3.1 Route Recording

A method introduced in section 2 records route stability of only one AS to the other ASes. The database has limit because it has informations of route stability of only one AS to the other AS. More informations from other source ASes are need for expanded informations giving of the database. We will collect some other ASes informations and introduce the system which accept informations of other ASes and update database system.

We will record informations of the other ASes soon, for example KOREN and CNU, and make it more useful. After that database system's capability is more growing.

For example, own AS to destination AS's AS-PATH is not always same as reverse direction of destination AS to own AS's AS-PATH. It is influenced by policy of intermediary ASes. That fact is important for a kind of the softwares. For example, a software which use DV connection for conference. This software needs AS-PATH informations of both direction to keep stable connection during video conferences. ABEL responds one direction route is stable, but reverse direction route information is not in database. It's one of the limit of informations of one AS. Therefore this update is important. This update will introduce only few addition which is source AS number to the name of route information files and number information files.

3.2 Inputs and Outputs of ABEL

We assume the first type of query format of ABEL has destination AS number and optionally has time data which have information of start time, end time and the day of the week which user want to know the stability. If ABEL accept query, it search for route information file and number information file of destination AS. If it can find these files, to read collected informations about AS-PATH and calculate collecting ratio of the time which is start time to end time of

```
0 : 00 - 23 : 50 Fri.
1  2523 7660 11537 10886 102

24.427481%
2  2523 2497 209 102

75.572519%
Main Route of AS 102 is 2  2523 2497 209 102
```

Fig. 5. An example of the response of ABEL

the day of the week which user sent. That calculation used bit shift operation and logic operation AND to select values we mentioned section 2.2. And the response is route informations with collecting ratio and main route of that time. That calculation is takes informations by number information file and when it outputs the response, it references to route information file. Second query type use IP address prefix instead of AS number. It takes similar process about AS number query. But it has to do IP address prefix to AS number conversion. We have introduced that information based on BGPView-log files. After that it takes same process as AS number query.

The interface of ABEL is assumed Web based interface. When you need informations, you can access the database and get informations readily on the Internet and optionally ABEL supplies informations of intermediate AS which are in main route which is answered by ABEL. That informations supplies image of AS-PATH in the Internet.

In addition we will plan to update ABEL will be able to accept the queries by softwares automatically.

4 Examples of the ABEL's Usage

ABEL supplies the information which is AS-PATH stability of destination AS. That information is useful for network administrators. In the Internet, there are two kind of network. One is the network for research which is used for work of network development. Another one is the network for commercial. The network environment for research is glowing to gigabit or tens of gigabit class. Because of that environment we can use software that uses wide bandwidth. For example, DV format is high-quality and high-resolution format. If it is used for video conference, it needs network which is enough to through 30Mbps bandwidth. But that network environment has no problem. But it is in multi homing AS that connect research network and commercial network. Only the research network has wide bandwidth. In this situation, video conference used DV use closed research network to keep enough wide bandwidth to use DV format. But during video conference if research network, for example, has trouble or maintenance. That network will be disconnected and the route will change to commercial network by dynamic route configuration. The commercial network has not have

Time	Message	Update	Notific	Open	Keepali
14:34:57	192	172	0	0	20
14:44:57	161	141	0	0	20
14:54:57	355	335	0	0	20
15:04:57	7218	7198	0	0	20
15:14:57	1549	1529	0	0	20
15:24:57	367	347	0	0	20

Fig. 6. A part of BGPView number of messages file

enough wide bandwidth to through DV stream. The video conference over the Internet will be stopped.

If the cause of that problem is maintenances of network and it has implemented periodic time schedule. In this situation, ABEL will be useful for users of softwares to avert communication will be disconnected. The solution of that problem is the administrator will access ABEL and get AS-PATH stability of destination AS before video conference. The administrator may find the information of maintenances time or the route become unstable irregular time or not by the responses of ABEL. The administrator will make video conference to avert this time or abandon video conference.

For example, fig.5 shows one of answers of ABEL. It shows AS-PATH information for own AS to destination AS which number is 102 of Friday. It shows that route is unstable that time and the administrator has got information which is ill-fitted for DV connection. Then the administrator has got information of short interval by ABEL. If only one period shows different AS-PATH from AS-PATH which shows another time period, the administrator can presume that the time is periodic maintenance time. And if some periods shows other path, the administrator can presume that the path is unstable and decide to abandon video conference.

In addition, if ABEL will become to accept queries by software and responds automatically. Softwares can get informations about AS-PATH stability and will take best time and route automatically to avert the situation which is communication disconnected by the responses of ABEL. Administrators will follow the instruction of the software.

5 Consideration about Automatic Analysis and Visualization

BGPView also can record a number of each messages of BGP-4 per user-established time.

We can find burst of messages sometimes in this file. BGPView usually receives too small amount of messages than that situation. It means something happened in this time. The way to resolve that problem is to check BGPView-log files which correspond that time. BGPView-log files record received time, IP

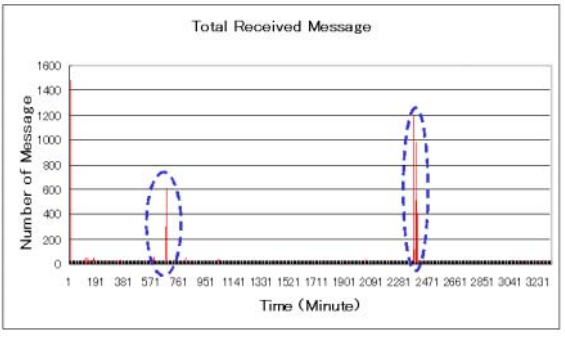


Fig. 7. Number of messages graph by automatic analysis and visualization

address prefix and AS-PATH. We may find destination AS number or intermediate AS number which causes of that problem in BGPView-log files. Automatic analysis observes BGPView number of messages file and if message amount has become large amount, it will check BGPView-log files around that time and visualization is able to show the number of messages time-number graphs automatically.

This analysis is other way to check AS-PATH stability. The method which we have shown aforementioned focuses destination AS number. It is available for destination AS number is clearly. If query has only time data, it will not respond the AS-PATH stability informations. But this method focuses time informations to analyze AS-PATH stability. Automatic analysis and visualization can respond instability route of the time which is included in query.

If we can combine two types of method, we can provide more useful information.

6 Conclusion and Future Works

We have introduced the database system named ABEL that responds AS-PATH stability of AS-PATH information. Routing information about AS-PATH is the large amount and stability analysis needs long time data. But we introduced the method for that problem and made ABEL to respond quickly.

Future, we are planning to make cooperation for many researchers and take a lot of data of AS-PATH stability. And we will update ABEL's functions, make new stability analysis method, to be connected by communication softwares automatically and concurrently enhancement of data.

References

[1] T. J.Watson Research center, IBM corp, Y.Rekhter, Cisco systems, T Li:A Border gateway Protocol 4(BGP-4), RFC 1771. Mar.1995. 564

[2] BGPView Home Page, <http://bugest.net/software/bgpview/> 565

Part III

High-Speed Network Technologies

Multicast Algorithms Using Status of Receivers in WDM Broadcast Network for CDN

Kyohong Jin[1], JongWook Jang[2], and Won-Joo Hwang[3]

[1] Dongeui University, Department of Multimedia Engineering
24 Gaya-dong, Busanjin-gu, Busan 614-714, Korea
khjin@deu.ac.kr
[2] Dongeui University, Department of Computer Engineering
24 Gaya-dong, Busanjin-gu, Busan 614-714, Korea
jwjang@deu.ac.kr
[3] Inje University, School of Electronics and Telecommunication Engineering
607 Obang-dong, Gimhae, Gyungnam 621-749, Korea
ichwang@inje.ac.kr

Abstract. Multicasting the multimedia content usually requires broader bandwidth than unicasting service and accordingly WDM broadcast network has been highly recommended for the infrastructure network of CDN(Content Delivery Network). Multicast Service can be implemented by unicast scheduling algorithm, but unnecessary multiple transmissions of a multicast message may result in a waste of bandwidth. To reduce the number of transmissions, multicast service transmits only one message to all destinations, but this may result in excessive receiver waiting time due to complicated transfer scheduling protocol. Although the WDM broadcast network easily supports the multicast service, multicast group partitioning problem must be resolved to reduce the receiver's waiting time and the number of transmissions. In this paper, we propose methods for partitioning a multicast group into smaller subgroups and for scheduling a separate transmission for each of these subgroups. The proposed algorithms reduce the receivers' waiting time by using the previous status of receivers. We analyze the proposed algorithms comparing with the conventional research through the computer simulation.

1 Introduction

Application of content delivery network(CDN), one of the hot topics in the networking and the biggest IP trends going, is quickly branching out. The CDN is a network optimized to deliver specific content, such as static web pages, transaction-based web sites, streaming media, or even real-time video or audio. Its purpose is to quickly give end users the most current content from the Content Provider(CP)'s server system. In other words, the goal is to push content as close to the user as possible to minimize content latency, jitter and to maximize available bandwidth speed. Figure 1 shows a typical content distribution system.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 575–584, 2004.

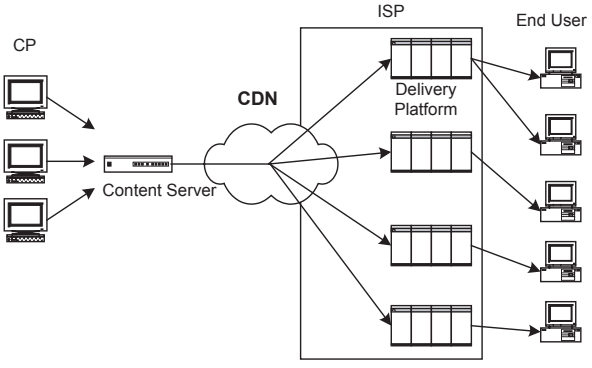


Fig. 1. CDN in Content Distribution System

As shown in figure 1, the CP makes various types of content and stores them at content server. The content according to the customer's demand is delivered to delivery platforms that is geographically distributed cache servers located at Internet Service Provider(ISP) facilities. When a user requests some contents to CP, the specific contents are delivered from the delivery platform not from the content server.

Recently, as the rich media content like audio and video streaming over the Internet is becoming more and more popular, the broader bandwidth of CDN becomes necessary. Since the fiber optic technology becomes available and supports a few Gbps in a single data channel, the WDM(Wavelength Division Multiplexing) broadcast network is highly recommended to a solution of CDN. Also, because the CDN should deliver the content frequently from the content server to several delivery platforms in multicast manner, the architectural advantage of WDM broadcast network would be well matched to distribution service of CDN.

The multicast service can be implemented by unicast scheduling algorithm. However unnecessary multiple transmissions of a multicast message may result in a waste of bandwidth. To reduce the number of transmissions, multicast service transmits only one message to all destinations, but this may result in excessive receiver waiting times due to complicated transfer scheduling protocol. By partitioning a multicast transmission into multiple subgroups, an efficient balance between the number of transmissions and the receiver waiting time may be achieved. Several multicast scheduling algorithms with the feature of partitioning multicast group are proposed in [1, 2, 3, 4].

In [1], greedy heuristics are proposed. One of them, the EAR(Earliest Available Receiver) schedules a transmission by the source node to the first receiver which becomes free. If additional receivers become available during this transmission, a transmission by the source to these receivers is scheduled immediately after the completion of the first one. In [2], random scheduling algorithms are studied. The random scheduling algorithm selects C nodes out of N nodes and schedules the multicast transmissions. If two or more nodes attempt to transmit

message to the same destination node, the receiver selects one message among the transmitted messages with equal probability. It is shown that, if the number of channels, C, is small, then network performance is limited by insufficient bandwidth. However, if the number of channels is relatively large, performance is limited by the occurrence of destination conflicts, and thus, employing multiple receivers per node can significantly increase the throughput and decrease the average delay. Unlike in [2], the [3]'s algorithm are designed for a centralized architecture in which a master scheduler maintains complete information about the state of the network, and instructs transmitters and receivers to tune to the appropriate channels. And in [4], the virtual receiver concept was developed as a novel way to perform fanout splitting that overcomes the overhead incurred when a partitioning and scheduling decision has to be made for each packet.

As described above, the existing multicast scheduling algorithms attempt to reduce the delay time through partitioning multicast group into several subgroups. However most of those algorithms do not consider the receiver's tuning latency and previous status of receiver. If partitioning algorithm used the previous status of receiver, the preceding tuning process of receiver could be eliminated and accordingly the receiver's waiting time and the number of transmissions could be reduced. Therefore, in this paper, we propose heuristic multicast group partitioning algorithms that partition receivers into subgroups using the information of receiver's previous status. And also we try to minimize the transmission delay of multicast message.

The rest of this paper is organized as follows. The system model is described in the next section. The partitioning problem is explained and the proposed heuristics are presented in section 3. The heuristics are tested on randomly generated test cases in section 4, and finally some concluding remarks are given in section 5.

2 System Model

The typical CDN network based on the WDM broadcast network consists of a passive star coupler and N nodes [5] as shown in figure 2.

Each node connects to the passive star coupler via a fiber link consisting of a pair of fibers. There are $W + 1$ communication channels in the system, where $W \leq N$. One of the channels is used as control channel that is shared by all nodes. The rest of the channels are data channels that are used for data transmission. Each node is equipped with one fixed transmitter, one fixed receiver and one tunable transceiver. The fixed transmitter and fixed receiver are on the control channel. The tunable transceiver is used on the data channels.

A number of multicast scheduling algorithms have been developed and they are generally including the multicast group partitioning algorithm. The multicast group partitioning algorithm plays an important roll in the scheduling algorithm and makes the average receiver's waiting time and the number of transmission processes smaller.

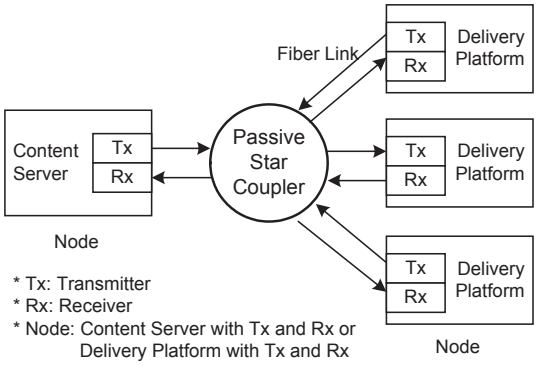


Fig. 2. System Model based on WDM Broadcast Network

The next section shows the heuristic multicast group partitioning algorithms proposed in this paper.

3 Proposed Multicast Group Partitioning Algorithms

In this paper, two heuristic algorithms are proposed to resolve the multicast group partitioning problem. Before describing the algorithms, we present that the multicast group partitioning problem is NP-complete. This works are based on the EAR[1]. The EAR algorithm is a greedy heuristic that schedules a transmission from the first available receiver. If additional receivers become available during this transmission, a transmission by the source to these receivers is scheduled immediately after the completion of the first one. However, the EAR does not consider the previous status of receivers and accordingly the tuning latency of receivers is included in every transmission scheduling.

The objective of partitioning multicast group into several subgroups is to schedule a separate transmission for each subgroup for minimizing the average receiver waiting time. Followings are assumed to simplify the partitioning problem.

- Source node's transmitter and at least one data channel is available before the first receiver in the multicast destination group becomes available.
- Before transmission scheduling, the data channel number of receiver previously used is known to all nodes in the network.

The first is valid assumption by noting that multicast messages use an equal amount of transmitter and channel resources, but consume a higher amount of receiver resources, since each transmission is received by multiple receivers. Thus receivers are likely to be the bottleneck of the network. The second assumption is reasonable, since every node has a fixed transmitter and a fixed receiver to exchange the control packet to each other for planning transmission time. The problem can be formalized by following parameters:

- L: Message duration time of the multicast message
- G: Multicast group size
- τ: Tuning latency of transmitter and receiver
- $P_i, i = 1, 2, \ldots, G$: Time when receiver at the destination node i finishes to receive a previous message before tuning to transmitter's data channel for this time scheduling. P_is are ordered such that $P_1 \leq P_2 \leq \ldots \leq P_G$
- $X_j, j = 1, 2, \ldots, G$: Time when receiver at the destination node j becomes available to receive after tuning to transmitter's data channel. X_js are ordered such that $X_1 \leq X_2 \leq \ldots \leq X_G$ and the following relation stands up; $X_j = P_j + \tau$.

The output using above parameters, S_j, $j \in \{1, 2, \ldots, G\}$ is $j-th$ scheduling time to transmit a message. The minimum average receiver's delay time($\overline{w}$) is given by

$$\overline{w} = \frac{1}{G} \cdot \sum_{i=1}^{G} \min_{S_j \geq X_i} (S_j - X_i) \tag{1}$$

And the equation (1) can be extended as below using the $X_i = P_i + \tau$.

$$\overline{w} = \frac{1}{G} \cdot \sum_{i=1}^{G} \min_{S_j \geq P_i + \tau} (S_j - P_i - \tau) \tag{2}$$

If the tuning latency(τ) is deleted from equation (2), then the minimum average delay time of receiver would be reduced. Our algorithms are able to remove the τ by using the previous status of receivers.

The transmission starting time S_j, $j \in \{1, 2, \ldots, G\}$ has two constraints, such that

$$S_G \geq X_G \geq P_G \tag{3}$$

$$S_j + L \leq S_{j+1}, \quad \text{for} j = 1, 2, \cdots, G-1 \tag{4}$$

The constraint in (3) guarantees that every receiver is able to have at least one chance to be scheduled. The constraint in (4) is due to the fact that there is only one transmission from occurence at the source. This prevents more than one transmission from occurring at the same time.

If $S_j < X_1$, then any transmission scheduling is not occurred because no receivers will be ready in time to receive and also if $S_j \geq X_G + L$, then the earlier scheduling($\leq S_{j-1}$) may be applied. Since the term between X_is is not fixed value and it could be less than the message length, there will be available receivers during the transmission of a message. The equation (5) shows this situation.

$$\begin{aligned} &S_1 = X_1 \\ &S_j + L < X_i, \quad \text{where } 2 < i \leq G, 2 \leq j \leq G, i \geq j \end{aligned} \tag{5}$$

Example of equation (5) is following. If L is 5 and the available time of multicast destinations in a certain request of message transfer are (X_1, X_2, X_3, X_4, X_5) = (1,2,4,7,10), then the first transmission is scheduled at $S_1(= X_1)$. And

the next transmission scheduling(S_2) is occurred at time 6($=X_1 + L$). Therefore the transmission for receivers X_2, X_3 schedules at S_2 because of $S_2 \geq X_2$ and $S_2 \geq X_3$.

From the equation (5), we can conclude that the multicast group partitioning problem is to find a minimum-transmissions. Thus, if the minimum-transmissions problem can be solved, the transmission-number problem can also be solved but not vice versa. Therefore, the minimum-transmissions problem is at least as hard as the transmission-number problem. If the transmission-number problem is NP-complete, the minimum-transmission problem is also NP-complete. The transmission-number problem can be proved to be NP-complete by a reduction from the minimum-cover problem defined in [6] which is NP-complete.

Based on the above conclusions, we present two heuristics. The heuristic multicast partitioning algorithms proposed in this paper partition the multicast group into subgroups and transmit multicast message according to the status information of receivers.

3.1 PTL

PTL(Partitioning with Tuning Latency) is a greedy algorithm. When multicast message is transferred, in order to partition the multicast group, PTL generates a few pseudo groups. Pseudo group consists of one or more receivers, and the receivers in this group are scheduled separated with other pseudo groups. The pseudo group is configured by the time extent, message duration time(L) and tuning latency(τ). The first pseudo group starts at the P_1, and if there is such P_i, $P_i \leq P_{i-1} + L + \tau$, $i = 2, \ldots G$, then the P_i belongs to the pseudo group. The first pseudo group is ended at the time of P_{j-1} where $P_j > P_{j-1} + L + \tau$. And the next pseudo group starts at P_j. Until all receivers belongs to one of the subsequent pseudo groups, this process goes on. After configuration of pseudo groups, receivers in a pseudo group tune their receiver to the last receiver's data channel in the pseudo group. Therefore the data channels used in the pseudo groups can be different each other.

After tuning the receivers in a pseudo group, PTL schedules the first transmission to the earliest available receiver in a pseudo group. The next transmission is either scheduled immediately after the first transmission if any of the remaining receivers become available during the first transmission, or, if no receivers became available during first transmission, whenever the next receiver becomes available. And after scheduling of pseudo group, the transmitter changes the data channel to corresponding next pseudo group's data channel.

The PTL is designed to minimize the delay time of each receiver by using the tunable transmitter when the term between P_is are longer than the $L + \tau$. The running time for this heuristic is $O(G)$.

When a multicast message transfer is generated at the content server, the group partitioning and transfer-time scheduling are executed according to the following example. Figure 3(a) shows the finished time of previous transmission(P_i) and data channel number having been used(W_j). And figure 3(b) presents the configuration process of pseudo groups. In figure 3(b), the pseudo group #1

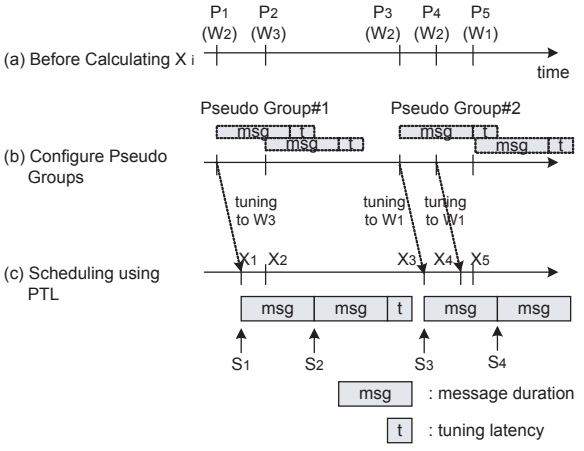


Fig. 3. PTL Algorithm

covers the P_1 and P_2, and P_3, P_4, and P_5 belongs to pseudo group #2. The nodes at P_1 and P_2 tune their receivers to data channel W_3 and P_3, P_4 and P_5 to W_1. After configuration of pseudo groups, the first transmission, S_1 starts at X_1. During the first transmission, X_2 becomes available. The second transmission starts immediately after the first transmission. Since X_3 is belonging to the second pseudo group, transmitter changes the data channel to W_1. The third transmission starts at X_3 and during this transmission the X_4 and X_5 became available. The last transmission includes the X_4 and X_5. As shown in the figure 3(c), the multicast group is partitioned as with (X_1), (X_2), (X_3), and (X_4, X_5).

3.2 M-PTL

M-PTL(Modified-PTL) algorithm is modified version of the PTL algorithm to decrease the transmission-number. The procedure of calculating the X_i is exactly same with that of PTL, but scheduling is occurred only once at the time of final receiver's available time in each pseudo group. The running time for this heuristic is $O(G)$.

Like in PTL algorithm, figure 4(a) shows the finished time of previous transmission (P_i) and W_j indicates the data channel number used at P_i. And figure 4(b) shows the configuration procedure of pseudo groups. The first pseudo group covers P_1 and P_2, and the second includes P_3, P_4 and P_5. The members of the first pseudo group tune their receivers to W_3 and the members of the second tune to W_1 and accordingly the all X_is are calculated.

As shown in figure 4(c), after generating the pseudo groups, M-PTL algorithm schedules the transmission at last X_i in each pseudo group. Since X_1 and X_2 belong to the pseudo group #1, the transmission S_1 starts at X_2 with data channel W_3, and at the X_5 the transmission S_2 begins with W_1. Therefore

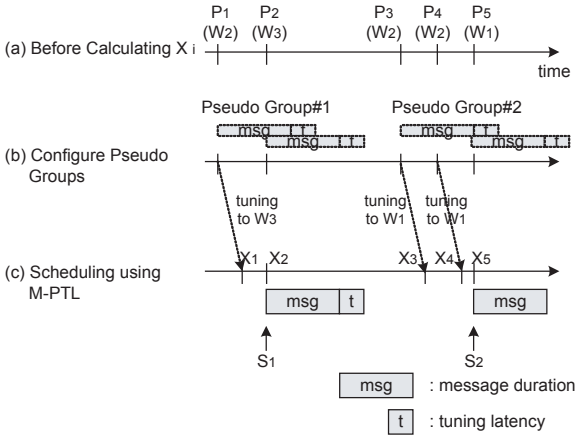


Fig. 4. M-PTL Algorithm

the subgroups are composed of (X_1, X_2), (X_3, X_4, X_5) and the transmission-number is minimized.

4 Simulating Result

The PTL and M-PTL algorithms were tested on a number of randomly generated test cases. Each test case consisted of generating P_1, P_2,..., P_G randomly according to a geometric distribution with parameter ρ, fixing L and G. In our simulation the ρ is set to 0.5(ρ=1 is the case in which all receivers become available at the same time, while ρ=0 is the case in which the receivers become available at intervals infinitely spaced apart). Also the data channels' number previously used at receivers is generated randomly. The EAR algorithm [1] was included in performance analysis to be compared with proposed algorithms.

Following system is considered.

- $N = 50$: The number of nodes in the WDM broadcast network
- $W = 50$: The number of data channels used in the WDM network
- In the network, one control channel exists to exchange the nodes' status information. This control channel is separated from the data channels.
- Each node has one fixed transmitter and one fixed receiver that are tuned to the control channel.
- Each node has one tunable transmitter and one tunable receiver that can be tuned to any of the data channels.
- Multicast group size, G, is uniformly distributed over $(1, 2, \ldots, N)$.
- We assume the tuning time of transmitter and receiver τ is much smaller than message length.
- Each node maintains P_i and W_{Pi}, $i = 1, 2, \ldots, N$. W_{Pi} means the data channel number used to previously receive a message. W_{Pi} is initialized randomly.

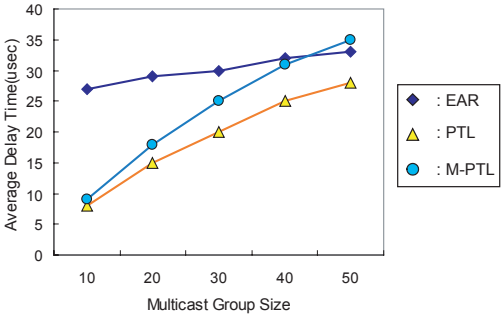


Fig. 5. Average Delay Time of Receivers(Message Length=1,500 bytes)

The goal of the simulation experiment is to investigate the receiver's average delay time and the number of transmissions as multicast group size is growing. The receiver's delay time is defined as the amount of time that a receiver must wait before it begins to receive a message. The delay time is measured from the point at which the receiver finishes the last scheduling. It is supposed that the bandwidth of one data channel is 1Gbps, tuning latency is 10μsec(when using the acousto-optic elements) and the message length is 1,500 bytes and 3,000 bytes. Figure 5 shows the results of average delay time according to the number of destinations in multicast group where L is message duration time of 1,500 bytes-long message.

Since the PTL and M-PTL algorithms consider the previous status of receivers, it is not necessary receiver's tuning in every time. And between pseudo groups the transmitter tunes to receiver's data channel, they also decrease the average delay time. However in M-PTL, as multicast group size grows to the number of nodes, the average delay time becomes long because the number of pseudo group is gradually decreasing.

Next, the number of transmissions is measured by how many transmissions are performed until all receivers in multicast group finish the message receiving. This number is equal to the number of subgroup and is an important measurement parameter of multicast group partitioning algorithm. Figure 6 shows the number of transmissions according to the number of destinations in multicast group where L is message duration time of 1,500 bytes-long message.

As shown in figure 6, the number of transmission in PTL is slightly smaller than that of EAR with such a reason that in EAR all receivers have to be tuned to transmitter's data channel before scheduling, but in PTL, the last receiver in a pseudo group does not have to be tuned. Also in the M-PTL, the transmission is occurred only once in each pseudo group and so the number of transmissions in M-PTL is smaller than the others.

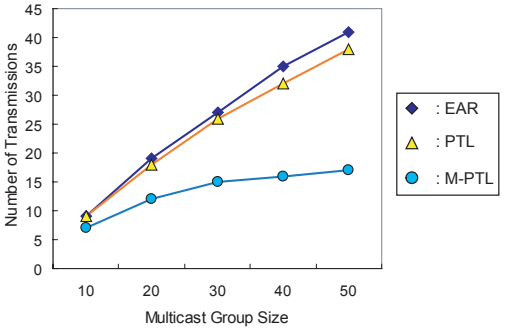


Fig. 6. Number of Transmissions(Message Length=1,500 bytes)

5 Conclusions

By partitioning a multicast transmission into multiple subgroups, an efficient balance between the number of transmission and the receiver waiting times may be achieved. Heuristic algorithms proposed in this paper reduce the average delay time of receivers' and the number of transmissions. Since the PTL and M-PTL consider the previous data channel number used at the last scheduling, they reduce the average delay time and the number of transmission using the pseudo groups. In particular, the M-PTL sends only one message in each pseudo group, it minimize the number of transmissions. Therefore the proposed algorithms could be used as the solution of the partitioning problem in CDN where multicast service is frequently occurred.

References

[1] J. Jue, B. Mukherjee, "The advantages of partitioning multicasting transmissions in a single-hop optical WDM network," Proc. of ICC'97, pp.427-431, 1997 576, 578, 582

[2] A. Mokhtar, M. Azizoglu, "Packet Switching Performance of WDM Broadcast Networks with Multicast Traffic," Proc. of SPIE'97, pp.1-29, 1997 576, 577

[3] E. Modiano, "Random Algorithms for Scheduling Multicast Traffic in WDM Broadcast-and-Select Networks," IEEE/ACM Transactions on Networking, Vol. 7, No. 3, pp.425-434, June, 1999 576, 577

[4] Z. Oritz, G. N. Rouskas, H. G. Perros, "Scheduling of Multicast Traffic in Tunable-Receiver WDM Networks with Non-negligible Tuning Latencies," Proc. of SIGCOMM'97, pp.301-310, Sept., 1997 576, 577

[5] R. Ramaswami, K. M. Sivarajan, *Optical Networks, A Practical Perspective*, Morgan Kaufmann, San Francisco, 1998 577

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability*, San Francisco, CA: Freeman, 1979 580

Improving Data Distribution in Branching Point Based Multicast Protocols

Mozafar Bag-Mohammadi, Siavash Samadian-Barzoki, and Nasser Yazdani

Router Lab., Dept. of EE & Computer Eng., Univ. of Tehran
Tehran, Iran {mozafarb,s.samadian}@ece.ut.ac.ir
yazdani@ut.ac.ir

Abstract. In several multicast routing protocols, the multicast distribution tree is identified by its branching points where multicast data is delivered from one branching point to another using native unicast. As a result, these protocols can be deployed incrementally and have low memory requirements compared to the traditional multicast routing protocols. The main drawback of these approaches is excessive lookup process in both multicast and unicast forwarding engines when handling data packets. To avoid this, we propose a minor modification in the parser code of routers and the IP header of the multicast data packets. Simulation results show that our techniques reduce the overall number of required lookups at least by 45.89% compared with non-optimized branching point based protocols like REUNITE [6] and HBH [7].

1 Introduction

Many applications like video conferencing use multicast services to reduce the network load and data distribution delay. Almost all multicast routing protocols construct multicast distribution trees in order to deliver multicast data packets. Traditional multicast routing protocols [4, 10] require group-related state maintenance at all on-tree routers, commonly known as Multicast Forwarding Table (MFT). A branching point (BP) in a multicast tree is a router which forwards the multicast data packets to multiple next-hop routers on the tree. Therefore, a BP has one incoming interface and more than one outgoing interfaces in the multicast tree. In BP-based protocols [6, 7, 1, 12], only the BP routers keep MFT entries. All non-branching routers in these protocols simply forward multicast data packets using unicast forwarding engine.

In BP-based protocols, when a multicast data packet arrives at an on-tree router, it looks up for a matching entry in the MFT. If an entry is found, the packet is sent to the corresponding next hop routers. Otherwise, the packet is handled by unicast forwarding engine. Hence, multicast data packets in non-branching routers require two successive lookups in order to be forwarded since they don't match any entry in MFT. These additional lookups are also needed for unicast data packets because the routers can not distinguish them from multicast ones. Generally, the problem arises as the routers have no prior knowledge whether the packets will match an entry in MFT or not.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 585–595, 2004.

Despite the aforementioned drawback, the BP-based protocols have many interesting features that make them very suitable to deploy multicast services. Their main beneficial characteristics are as follows [6]:

Incremental deployment: Many multicast routing protocols like PIM-SM (Protocol Independent Multicast-Sparse Mode) [10] and DVMRP [4] require every router in the network to implement the protocol. In contrast, BP-based protocols like REUNITE [6] and HBH [7] have native support for incremental deployment. Since all packets have unicast destination addresses, routers that not implemented the protocol will forward the packets in unicast. Despite the fact that these routers can not act as branching points, they still can take part in multicast data distribution [6].

Enhanced scalability: Works in [6] and [11] show that for sparse multicast groups, the multicast data delivery tree is likely to have a large number of non-branching routers. Since only the branching nodes need to maintain multicast state, BP-based protocols can support larger number of multicast groups in comparison with traditional multicast solutions [4, 10].

Unique group identification: In REUNITE, a multicast tree is identified with the IP address of its tree root and a port number. Thus, generating globally unique group identification is trivial in REUNITE since the root just needs to generate locally unique port number [6]. SEM [1] and HBH [7] use the channel abstraction in EXPRESS [13] to identify a group. As a consequence, combining with its own IP address, each source selects a unique 24-bit channel identifier which forms a unique group identifier.

Constructing the tree in forward direction: Traditional multicast protocols like PIM-SM [10] consider the SPT (Shortest Path Tree) as the target. However in practice, they build an approximation of the SPT using the shortest reverse path selection for each new member. The resulting tree in this case is a poor approximation in the presence of asymmetry, which is prevalent in existing networks [8]. In contrast, BP-based protocols build the multicast tree in the forward direction trading off more join latency.

In this paper, we propose a method to eliminate unnecessary lookups performed in the BP-based protocols. In our mechanism, a unicast packet never goes into multicast forwarding engine and is completely forwarded by the unicast engine. Furthermore, when a multicast packet arrives at a non-branching router, it is directly handled at the unicast engine without having to pass through the multicast engine. In the branching points, multicast packets are directed to the multicast forwarding engine which is obviously necessary. Accordingly, our mechanism reduces the number of lookups by half for unicast data packets. Moreover, multicast packets only enter the multicast engine where necessary.

While improving data forwarding in the BP-based protocols, the proposed method does not require any changes in their tree construction processes. Our modifications consist of two parts. First, we propose to assign a special value to the *Protocol* field in the IP header of the multicast data packets. This is required in BP based protocols which have no clue to distinguish multicast data packets from unicast ones (like REUNITE [6]). Second, we recommend changing

the parser code of the routers that implement the BP-based protocol to pass data packets to their corresponding forwarding engine correctly.

In section two, we briefly introduce BP-based protocols. We discus our mechanisms in section three. Next in section four, we present the simulation results. Finally, section five concludes the paper.

2 Related Work

REUNITE (REcursive UNIcast TrEes) [6] implements multicast distribution based on the unicast routing infrastructure. REUNITE 's basic motivation is that in typical sparse multicast distribution trees, the majority of routers are relay routers which simply forward incoming packets to an outgoing interface. In other words, the minority of routers are branching nodes. Nevertheless, all multicast protocols keep per group information in all routers of the multicast tree. Hence, they separate multicast routing information in two tables: a Multicast Control Table (MCT) that is stored in the control plane (slow path [14]) and a Multicast Forwarding Table (MFT) installed in the data plane (fast path [14]). Non-branching routers simply keep group information in their MCT and the branching nodes keep MFT entries which are used to recursively create packet copies to reach all group members. A multicast session is denoted by a $< S, P >$ tuple in REUNITE, where S is the unicast address of the source and P is a 16-bit port number allocated by the source.

Reference [15] proposes a scheme to achieve a same state reduction at non-branching nodes as REUNITE. However, it requires dynamically setting up tunnels between adjacent branching routers in a multicast tree. Using an additional layer of IP header introduces 20 more bytes overhead in each packet. In addition, to support dynamic membership, a sophisticated and complex control protocol is needed to dynamically set up and tear down tunnels.

Reference [7] showed that REUNITE fails to construct SPT in the presence of unicast route asymmetries [8]. Asymmetries may also lead REUNITE to unnecessary packet duplications on certain links. They also showed that the departure of one receiver may change the route for another one. Consequently, they propose HBH to solve these deficiencies. In this protocol, the tree is completely represented by its branching points and the routers attached directly to the receivers. Furthermore, HBH identifies a multicast session using the channel concept existing in EXPRESS [13].

Simple Explicit Multicast (SEM) [1] is a BP-based method with less tree construction complexity than REUNITE and HBH. SEM uses the receivers' list to construct the tree. SEM packets are forwarded according to unicast forwarding paradigm between SEM router pairs. The MFT structure in SEM is similar to HBH. When a new member joins the multicast session or one of the existing members leaves the session, the whole multicast tree must be constructed again.

Originally proposed to reduce the forwarding cost in Xcast [9], Sender Initiated Multicast (SIM) [12] is a BP-based protocol as well. SIM packets are forwarded between each SIM router pair in unicast. Basically, SIM has two for-

warding modes: list mode and preset mode. In the list mode, an SIM sender always attaches the receivers' list to multicast data packets. In the preset mode which is the prevalent SIM mode, the SIM sender periodically attaches the receivers' list to multicast data packets. SIM uses an MFT-like table to forward the packets in the preset mode. Our mechanism can be applied to that mode.

3 Improved Forwarding Mechanism

3.1 Current Method

We classify on-tree nodes in a multicast tree into three distinct categories based on the number of their branches [11]:

Member nodes: Examples of these nodes are leaf receivers and occasionally the senders. These nodes have degree one on the multicast distribution tree graph. In Fig. 1a, nodes S and r1 to r5 are member nodes.

Relay nodes: These nodes have degree two and just relay the multicast data packets from an incoming interface to an outgoing interface. They are called non-branching points and their presence is ignored in BP-based protocols. Traditional multicast schemes maintain multicast states in non-branching routers consuming invaluable memory space in their data paths [10, 4]. On the contrary, in BP-based protocols, some protocols maintain these states in control plane [6, 7] and some do not require them at all [1, 12, 15]. These states may only be stored for tree maintenance purposes. Relay nodes are shown with Ri in Fig.1a which i varies between 1 and 9.

Branching points: As stated before, these nodes have degree more than two and include more than one outgoing interfaces. These nodes are responsible for making copies from multicast data packets and sending them to next higher level branching points in multicast distribution tree. In BP-based protocols, only these nodes are allowed to keep entries in their MFT in data path. H1 to H4 are example BPs in Fig. 1a.

Since the relay nodes are ignored in multicast distribution tree of BP-based protocols, the notion of tree in these protocols is different from other multicast protocols. Therefore, we use the terms "complete tree" and "reduced tree" when

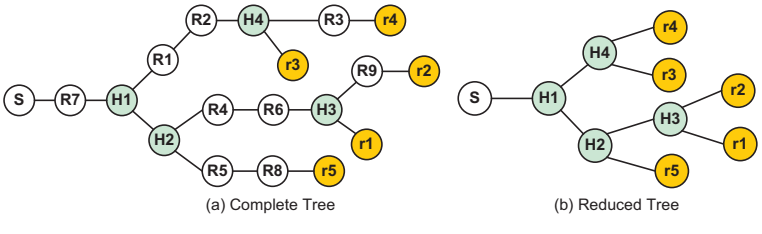


Fig. 1. Ordinary protocols use complete trees and BP-based protocols use reduced trees for multicast data distribution. The reduced trees do not contain the relay nodes

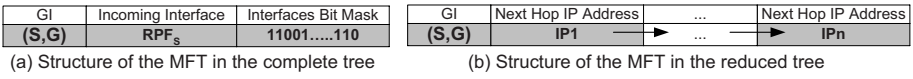


Fig. 2. The MFT structure in complete and reduced trees

referring to the multicast tree in ordinary protocols and BP-based protocols respectively [11]. These two kinds of tree are shown in Fig. 1. The complete tree may contain all three types of on-tree nodes. On the other hand, the reduced tree only consists of member nodes and branching points. In both kinds of the tree, on-tree nodes maintain entries in their MFTs except for the leaf nodes.

Fig. 2 depicts the differences between MFT structure in complete tree and reduced tree. The MFT in complete tree consists of incoming link, outgoing links and group identifier (GI). Usually, the GI is (S, G) or $(*, G)$ pair where S is the source IP address, G is the group address and * is don't care. In a reduced tree, the MFT contains the GI and the IP addresses of directly attached next hop BPs or leaf receivers. Here, GI may be (S, P) or (S, G) tuples, where P is the port number allocated by source. Even though, the size of MFT is less in a complete tree, the number of routers requiring MFT maintenance is smaller in a reduced tree. As a result, the total memory consumption in a reduced tree is less [6].

Since the MFT structure in a reduced tree contains IP addresses of the next hops, multicast data forwarding uses unicast routing. This allows BP-based protocols to adapt themselves with route changes and instabilities. This criterion makes some inefficiency for unicast data packet forwarding since these packets should also be tested in MFT prior to IP lookup in unicast forwarding engine. In the case of multicast data packet, if an MFT entry is found, the packet is sent to the unicast destinations that exist in the entry. Otherwise, the packet is sent to the unicast destination in the IP header using normal unicast IP lookup. Therefore, for all unicast packets, the router must perform an additional lookup in the MFT. Together with the extra MFT lookup for multicast data packets in relay nodes, this problem makes the architecture of the existing BP-based protocols inefficient regarding the packet forwarding. The packet forwarding mechanism in these protocols is depicted in Fig. 3. To the best of our knowledge, there is no other proposal except our solution to address this inefficiency problem.

The original BP-based proposals like REUNITE [6] and HBH [7] do not consider this problem very serious. They argue that the MFT lookup uses the exact matching algorithm which is not time-consuming in comparison with the longest prefix matching algorithm in existing IP lookup solutions [19]. Nevertheless, as stated in reference [16], since the entries in MFT are not easily aggregated as opposed to unicast forwarding tables, the MFT size can be too large. The huge number of entries in MFT makes the exact matching algorithm a time-consuming process. They propose a leaky aggregation scheme in [16] which decreases the MFT size considerably trading off some network bandwidth. However, their approach replaces the exact matching algorithm needed for MFT lookup to a longest prefix matching algorithm. Another proposal exists that

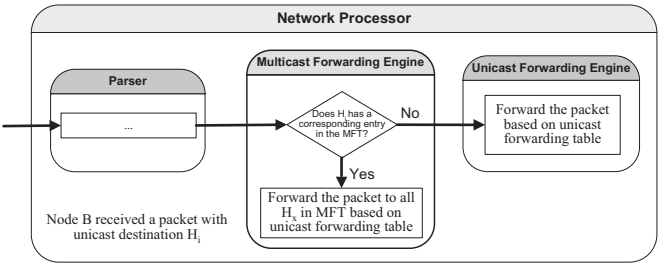


Fig. 3. The current forwarding mechanism in BP-based protocols

achieve some aggregation in MFT without sacrificing bandwidth and lookup substitution [17]. Yet, they can't shorten the size of MFT completely.

3.2 Our Solution

In order to eliminate the impact of multicast forwarding on unicast packets, we set the *Protocol* field in IP header of a multicast data packet to a special value named BP_PROT. The value of BP_PROT can be different for each BP-based protocol, but the value must be known to all routers implementing the protocol. By this way, multicast packets are easily distinguished among other packets. Hence, the parser [14] can partition incoming packets based on the value in this field. As a result, unicast packets do not further go to the multicast engine. This eliminates duplicate and unnecessary lookups which exist in the forwarding procedure of BP-based protocols for unicast packets.

Using this idea, the only inefficiency which remains is at relay nodes. In these nodes, multicast packets still have to pass through the multicast engine without matching any MFT entry. We can do minor changes in the parser to further facilitate the packet forwarding process as follows. In each router that implements the BP-based protocol, the parser first checks the *Protocol* field in the IP header of received packet. If the packet is a multicast packet, the

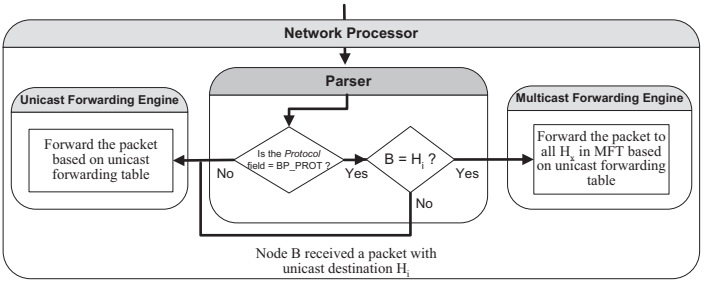


Fig. 4. Improved forwarding mechanism for BP-based protocols

parser checks the destination address of packet. The packet goes to the multicast engine if it is destined to this router and the unicast engine otherwise. This way, a multicast packet is forwarded by the unicast engine at relay nodes since the destination address in the packet is different from that of the relay node. The new revised forwarding mechanism in BP-based protocols is illustrated in Fig. 4. Moreover, the proposed solution to remove the unnecessary MFT lookups in relay nodes for multicast packets does not apply to REUNITE since this protocol does not use the branching point addresses as destination address.

3.3 Disscusion

Our changes are only applied to the data forwarding plane and do not affect the control plane of the BP-based protocols. This means that all BP-based protocol messages remain intact. One can argue that the proposed mechanisms introduce extra overhead in the parser code. We believe that this is not an important issue and one can implement the extra conditional experiments in the parser code considering the condition prediction in such a way that the most usual case be performed as a default process and the other case as an exception. Besides, the parser code already examines the value in the *Protocol* field and the *Destination* field of the packet for other purposes. Therefore, one can rewrite the parser code combining the required changes with other cases in the parser code which further reduces the overhead of our modifications.

4 Simulation Results

To prove the effectiveness of the proposed mechanisms, we have evaluated the number of required lookups using NS-2 environment [18] for both unicast and multicast packets in comparison with previous mechanism. We considered network and group size as two main parameters that affect the performance of proposed mechanisms. We performed two sets of experiments, one in small networks and another in large one. Since varying the network size in large networks is not a trivial task [5], we used the small network experiment set to evaluate the network size effects. On the other hand, the large network experiment set is mainly used to show the efficiency of the mechanisms for various group sizes.

For the small networks, we used two different flat random graph generation methodologies namely Locality and Doar-Leslie [5]. The graphs are generated using GT-ITM network topology generator [5]. We changed the network size from 20 to 200 nodes while the average node degree is fixed approximately at 3.5. For each generated network, we intentionally introduced 50% asymmetry in the network links. For the large networks, we generated two different sets of random graphs based on two famous network topology models, Barabasi-Albert [3] and transit-stub [5], using BRITE [2] and GT-ITM network topology generators respectively. The Barabasi-Albert model takes the power law relationship of Internet into account, while the transit-stub model considers the hierarchical transit-stub relationships between numerous ASs. The main difference between

these two models is the distribution of nodes' degree. The average node degree for transit-stub and Barabasi-Albert topologies are fixed approximately at 3.5 and 4 respectively. Since we used the session-level mode of NS-2 for these simulations, it was not possible to add artificial asymmetry into the generated graphs.

4.1 Small Networks

For the sake of simplicity, we use E_BP and BP referring to the BP-based protocol after and before applying our mechanisms. Fig. 5 shows the comparison between the required number of lookups in E_BP and BP for Doar-Leslie based topologies when forwarding multicast data packets. Since BP-based protocols are intended to support sparse groups, the group size is fixed at 30% of the network size in all small networks experiments. Each point in each graph is resulted from 10 simulation runs for each 10 different random topologies. This creates 100 different simulation runs for each point. The identities of the group members, i.e. the sender and the receivers, are selected randomly in each run. Finally, the plots are normalized to BP to achieve the relative improvement factor.

Fig. 5 shows the reduction gain achieved from multicast data packet forwarding in E-BP which we call MG (Multicast Gain). However, E_BP reduces the number of required lookups for unicast packets exactly by half. Assuming α percentage of the total traffic is unicast, we can calculate the overall reduction gain in the number of lookups as follow:

$$Gain = \alpha/2 + (1 - \alpha).MG \tag{1}$$

If we suppose that α is 90%, then the overall reduction gain is 0.5365 considering the MG factor equal to 0.865 from Fig. 5. The simulation results for Locality topology model are the same and omitted from here due to space limit.

4.2 Large Networks

We fixed the network size at 1204 nodes and performed the simulations with various group sizes ranging from 10 to 300. The simulation results with Barabasi-

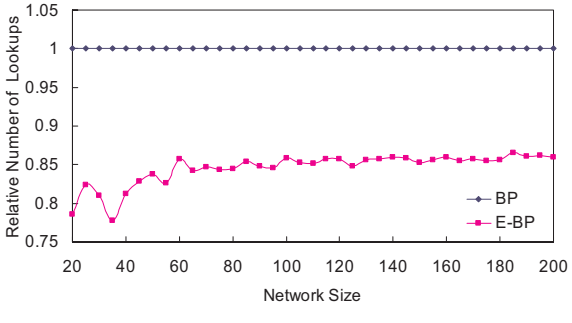


Fig. 5. The number of required lookups for multicast data packets changing the network size from 20 to 200 in Doar-Leslie flat random topology model

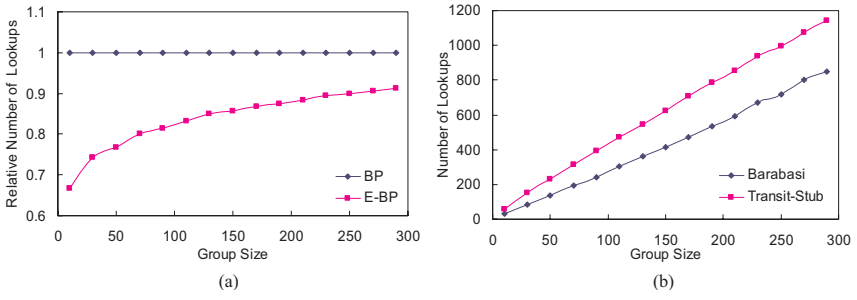


Fig. 6. The number of required lookups changing the group size from 10 to 300 in (a) Barabasi-Albert flat random topology model (b) both topology models

Albert graph model are shown in Fig. 6a. As can be seen in this figure, E-BP reduces the number of required lookups more for larger group sizes compared to BP. The comparison between simulation results of transit-stub and Barabasi-Albert network models is shown in Fig. 6b. The transit-stub graphs consist of four transit domains connecting 12 stub domains each with 100 nodes. The transit-stub simulations show less reduction in the number of required lookups for E-BP compared to Barabasi-Albert simulation results. Several explanations exist for this behavioral distinction. The node degree distribution, the amount of member nodes and the limited number of transit domains in the generated transit-stub graphs are some of them. These factors may increase the number of branching points in transit-stub simulation compared to Barabasi simulation. If we assume that α is 90%, then the overall gain using our mechanisms will be 0.5411 and 0.5386 for the Barabasi-Albert and transit-stub models respectively.

These simulations indicate that the MG decreases when the ratio of the group size to the topology size increases. We call this ratio group density. An increase in the group density increases the number of branching nodes which indeed decreases the number of relay nodes. Consequently, the MG will be decreased. The size of simulation scenario (1204) is orders of magnitude smaller than Internet. Therefore, the group density is much lower in Internet than simulation scenario. So, we expect that the MG on Internet to be higher than the simulation results.

5 Conclusion

The BP-based protocols like REUNITE and HBH are promising kinds of multicast routing protocols with numerous fascinating features like incremental deployment. They use unicast routing mechanism in the forwarding process of multicast data packets. They require less memory space for MFT construction in the whole network compared with traditional multicast routing protocols. As a main drawback, BP-based protocols deteriorate the forwarding of unicast packets. We propose a new mechanism which eliminates the superfluous lookups

in the packet forwarding of the BP-based multicast routing protocols. Simulation results show that our techniques reduce the overall number of required lookups at least by 45.89% for both unicast and multicast packets. We believe that the competence of BP-based protocols must be revaluated considering their beneficial features and the improved forwarding mechanism in this proposal.

References

[1] A. Boudani, B. Cousin, "SEM: A New Small Group Multicast Routing Protocol", Proc. of IEEE ICT2003, Tahiti, Feb. 2003. 585, 586, 587, 588

[2] A. Medina, A. Lakhina, I. Matta, J. Byers "BRITE: An Approach to Universal Topology Generation" , In Proc. of MASCOTS'01, Cincinnati, Ohio, August 2001. 591

[3] A. L. Barabasi, R. Albert, "Emergence of Scaling in Random Networks". Science, 286:509-512, October 1999. 591

[4] D.Waitzman, C.Partridge, S.Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, Nov.1988 585, 586, 588

[5] E. W. Zegura, K. Calvert, S. Bhattacharjee., "How to model an Internetwork.", Proc. of IEEE Infocom'96, San Francisco, CA 591

[6] I. Stoica, T. S. Eugene Ng, H. Zhang, "REUNITE: A Recursive Unicast Approach to Multicast", IEEE INFOCOM'2000, Mar. 2000. 585, 586, 587, 588, 589

[7] L. H. M. K. Costa, S. Fdida, O. C. M. B. Duarte, "Hop By Hop Multicast Routing Protocol", SIGCOMM'01, San Diego, USA, August 2001. 585, 586, 587, 588, 589

[8] Paxson, "End-to-End Routing Behavior in the Internet", In Proceedings of SIGCOMM'96 , Stanford, CA, August 1996. 586, 587

[9] R. Boivie, et al, "Explicit Multicast (Xcast) Basic Specification, IETF Internet-Draft, draft-ooms-xcast-basic-spec-04.txt, 2003 587

[10] S.Deering, et al, "The PIM architecture for wide-area multicast routing", IEEE/ACM Trans. on Networking, Vol.4, No.2, April 1996 585, 586, 588

[11] J. Pansiot and D. Grad, "On routes and multicast trees in the Internet", ACM Computer Communication Review, vol. 28, no. 1, pp. 41-50, Jan.1998. 586, 588, 589

[12] V. Visoottiviseth, H. Kido, Y. Kadobayashi, S. Yamaguchi, "Sender-Initiated Multicast Forwarding Scheme", Proc. of IEEE ICT'2003, Tahiti, Feb. 2003. 585, 587, 588

[13] H. W.Holbrook, D. R.Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications", Proc. of ACM SIGCOMM'99, Sept.1999. 586, 587

[14] J. Aweya, "IP Router Architectures: An Overview", Journal of Systems Architecture, 46 (2000) pp.483-511, 1999. 587, 590

[15] J. Tian, G. Neufeld, "Forwarding state reduction for sparse mode multicast communication", Proc. of INFOCOM'98, San Francisco, California, Mar. 1998. 587, 588

[16] P. I. Radoslavov, D. Estrin, R. Govindan, "Exploiting the Bandwidth-Memory Tradeoff in Multicast State Aggregation", TR 99-697, USC-CS, July 1999. 589

[17] D. Thaler, M. Handley, "On the Aggregatability of Multicast Forwarding State", Proceedings of INFOCOM'2000, Mar. 2000. 590

[18] "The Network Simulator - ns - 2", http://www.isi.edu/nsnam/ns/ 591

[19] H. Mohammadi, N. Yazdani, B. Robatmili and M. Nourani, "HASIL: Hardware Assisted Software-based IP Lookup for Large Routing Tables", Proceeding of ICON'2003, pp. 99-105, Sydney, Australia. 589

Hierarchical Overlay Data Delivery Tree Construction Adopting Host Group Model and Topology-Awareness

Dong-Kyun Kim[1], Ki-Il Kim[2], Il-Sun Hwang[1], and Sang-Ha Kim[2] *

[1] KISTI, 52 Eoeun-Dong, Yusong-Gu, Taejon 305-806, Korea
{mirr,his}@kreonet2.net
http://www.kisti.re.kr
[2] Department of Computer Science, ChungNam National University
220 Gung-dong, Yusong-Gu, Taejon, 305-764, Korea
{kikim,shkim}@cclab.cnu.ac.kr
http://cclab.cnu.ac.kr

Abstract. We propose a scheme for hierarchical overlay multicast based on IP topology awareness and host group model called PAM (Practical topology-Aware overlay Multicast). PAM is remarkably different from the previous overlay multicast or ALM (Application Layer Multicast) schemes in the following aspects. First, instead of end-host, DR (Designated Router) builds overlay DDT (Data Delivery Tree) by adopting host group model [1], in order to improve scalability within a subnet, robustness of group members and transparency compatible with traditional multicast. Second, hierarchical overlay tree is constructed. That is, intra-domain DDT and inter-domain DDT are built hierarchically for more scalable and practical deployment. In addition, PAM constructs overlay DDT with IP topology information derived from DRs to localize group members and gain performance enhancement such as low first join latency and reduced control overhead. Compared with other principle ALM schemes, those distinct advantages of PAM are evaluated by simulation.

1 Introduction

Traditional IP multicast is yet to take off on Internet in spite of its efficient data delivery for group communication, since there have been many barriers against its deployment, for example inter-domain routing problem, forwarding state scalability on core networks, and full router dependency [2]. As the alternatives, Overlay multicast or ALM schemes [6]- [15] have been suggested. However, most of them are based on end-hosts and still need to solve some problems in practical deployment aspects as follows. 1) Inefficient operation and consequent higher link stress within a subnet, 2) Construction of not-optimized DDT, 3) Unscalability and complexity due to large number of control packets, 4) Instability

* The author is corresponding author

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 595–605, 2004.

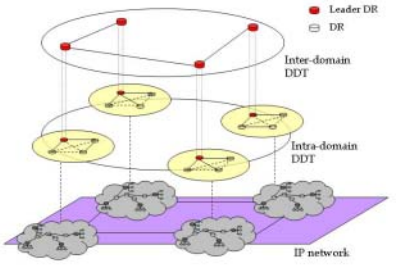


Fig. 1. Hiearchical PAM Architecture

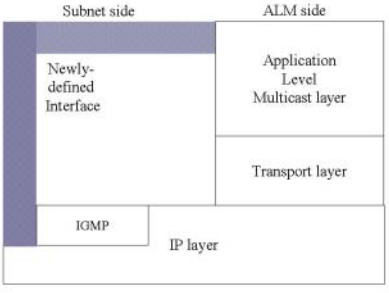


Fig. 2. Protocol Stack and Interface

resulted from frequent failure or misbehavior of hosts, 5) A lot of dependency between each group member.

In this paper, we propose a new ALM scheme, PAM, which is designed by the following principles: host group model, IP network topology awareness and hierarchical DDT architecture. Based on host group model of PAM, IP multicast is applied between end-hosts and a DR within a subnet by using IGMP (Internet Group Management Protocol) and overlay DDT is constructed between DRs. When PAM makes overlay DDT, it deals with intra-domain and inter-domain of IP network separately so that DDTs can be built hierarchically based on topology awareness and consequent localization of group members (Figure 1). With these properties, PAM makes significant performance enhancements based on scalability, simplicity, robustness, optimality and so on, which are described more in detail in the following chapters.

The remainder of this paper is organized as follows. Section 2 discusses the contributions of PAM. In section 3, procedure for dynamic group membership and DDT maintenance is described. The simulation result is presented in section 4. Finally, we make a conclusion in section 5.

2 Contributions of PAM

PAM is scalable to large number of group members due to adopting host group model and hierarchical DDT. That is, even though many group members (end-hosts) intend to join or leave a multicast group, it does not cause any extra control and data forwarding overhead at all in intra-domain because each DR only reacts to the first group member's join and the last member's leave via IGMP (Internet Group Management Protocol) within a subnet and these reactions are bounded and localized within an intra-domain. Also, inter-domain DDT is affected only by LDR (Leader DR)'s changes due to LDR's join and leave. Since the number of LDRs is limited as constant number, the efforts to construct and maintain this DDT does not cause scalability problem. In addition, PAM requires the minimum measurement for building optimized DDT. In intra-domain, DDT is constructed using metrics based on routing protocols such

as RIP, OSPF, and IS-IS, because nodes that belong to DDT are routers (DR). Therefore, it does not require any network measurement for DDT construction. While in inter-domain, it makes use of minimized measurement to select a close-by LDR in other domain.

The stability and robustness of DDT mostly depends on component's stability and robustness property or how frequently DDT should be updated. In PAM, DDT consists of routers, which are more robust and stable than general hosts. So, it can make network configuration simple and additionally get reduced control overhead. As another effort to gain more performance and reduce control overhead, PAM distinguishes inter-domain from intra-domain based on IP network topology information. This hierarchical architecture prevents changes over intra-domain DDT from being propagated to the whole networks. Hence, PAM can make network configuration simple and scalable by removing close dependency between each member. The optimality in PAM can be obtained by fully making use of metrics based on routing protocol. Using this information, PAM can grasp IP network topology. Once accurate topology information is obtained, optimality problem can be solved by Steiner-tree algorithm as well as shortest path algorithm. Also, in case of PAM that is equipped with host group model, duplicates of packets can be drastically reduced within a subnet and consequently link stress can decrease as well.

With above reasons, PAM can solve many deployment issues with common and scalable architecture. However, it is not a complete solution to the alternative of traditional IP multicast. The major deficit of PAM is the centralized overhead on each DR. DR should be equipped with ALM functionalities for group management and data forwarding. Also, it requires a special application for the purpose of strong and distributing list of group members and LDRs. So, single point of failure problem exists. In addition, it still has transmission cost, relative delay penalty, and session delay penalty.

3 Group Communications in PAM

Prior to detailed description, we assume that there is an application, which records the list of being joined DR and a leader DR (LDR) of a domain. Address of this application can be advertised via e-mail or session advertisement in traditional IP multicast. In this paper, we call this application RP (Rendezvous Point). This RP is different from RP in traditional IP multicast in a point that it is not involved with data forwarding, but just building DDT.

3.1 Extensions of DR Functionality

In order to support ALM in the DR level, the functionality of DR should be extended. These extensions must include scheme to control dynamic group membership. A DR participating in the group is equipped with two protocols to support ALM and IP multicast as shown in Figure 2. ALM protocol needs to be added between application layer and transport layer. The most basic function of

ALM layer is to transmit data by constructing DDT between DRs participating in the group. In the proposed design of DR, a newly defined interface exists between ALM layer and IP layer (subnet). Using this interface, IP multicast datagram sent from a subnet is transmitted directly to the ALM layer without looking up general multicast routing table. To the contrary, ALM layer uses the new interface to transmit original IP multicast datagram extracted from ALM packets that are sent from neighbor DRs in the group, directly toward IP layer at subnet side. IP datagram here includes IP header, which encodes a multicast group address as destination address. Furthermore, hosts can notify their join and leave to DR via the interface between IGMP and ALM. DR needs to deliver these IGMP messages to ALM when it receives new group join or leave messages. In other worlds, the interface between application layer and IP layer is designed for data transmission, while the newly defined interface between ALM and IGMP is a control interface for dynamic group management.

3.2 Operations for Group Join and Leave

A host that intends to join multicast group requests GROUP JOIN through IGMP REPORT message. If a requesting host issues the first join to a DR, the DR requests the ALM protocol to join specified multicast group. Also, ALM protocol sends REGISTRATION REQUEST with GID (Geographic ID) to the RP. This GID can be automatically set as Autonomous System number for BGP routing or manually predetermined value. GID should be unique and assigned with consideration of physical IP network topology. After receiving requested from DR, RP searches a recorded LDR corresponding to the GID. The RP operates differently as the following situations.

- If a LDR is present, RP replies the requesting DR with other members' addresses. At the same time, other members are informed the address of the new DR from RP. After this procedure, each DR computes and establishes a new intra-domain DDT including a new member. In practical, it is not necessary to define additional control messages for establishing new DDT.
- Otherwise, if there is no recorded LDR corresponding to GID, it means that it is the first requesting DR in the corresponding domain. In this case, RP selects some LDRs from a set of LDRs and then replys with REGISTRATION REPLY including chosen LDRs' addresses. After receiving them, the chosen DR as LDR measures RTT between other LDRs and itself in order for registration to the closest neighbor LDR. For other approaches, various other ALM schemes [6]-[9] can be applied. In case of group leave, it is more complex than group join. When there is no more group participant on a subnet, a DR requests GROUP LEAVE to the RP. The included information in GROUP LEAVE message is different from each case depending on the role of DR, that is, LDR and non-LDR.
- If a DR requesting group leave is a LDR for a group in intra-domain, RP selects one DR as a new LDR among remaining DRs and updates it as the LDR corresponding to a GID. The RP should address the LDR's change to

other LDRs of inter-domain DDT. In this case, the inter-domain DDT has the same delivery path regardless of fluctuating group membership. While, if the leaving DR is a LDR and the last one in the domain, all other LDRs are informed of the situation by RP, and only do neighboring LDRs with the leaving DR try to rebuild their inter-domain DDT.
– If a DR requesting leave is not a LDR in a domain, as we expect, the inter-domain DDT has no effect of DR's leave. It is handled by rebuilding intra-domain DDT.

3.3 Data Dissemination

Data dissemination in PAM is accomplished along established inter-domain and intra-domain DDT. When a member intends to transmit a data packet, it sends an original multicast packet toward DR. Then, the DR encapsulates the original multicast packet in the form of unicast packet and sends it to a logical neighboring DR on intra-domain DDT. This data dissemination is continued until all member DRs receive the data packet. When a LDR receives data packet, it performs data forwarding along intra-domain DDT as well as inter-domain DDT. As a result, the data packet sent from a host is gradually propagated along following orders; intra-domain DDT, LDR, and inter-domain DDT.

4 Performance Evaluation

In this section, we analyze the performance of the proposed mechanism using simulation. The used simulation tool is NS-2 [17]. The simulation is performed toward two directions; Performance enhancement gained by adapting 1) host group model and 2) topology-awareness property. The simulation results are compared with End-system multicast and HMTP since the former is a principle mesh-based ALM scheme and the latter is a major tree-based ALM.

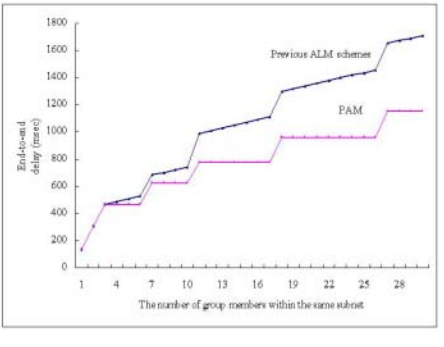


Fig. 3. End-to-End Delay

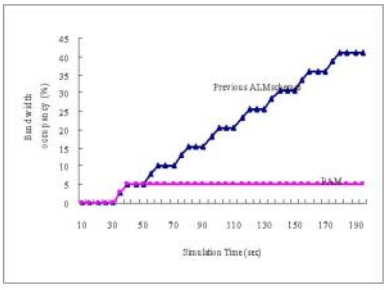


Fig. 4. Total number of path computation

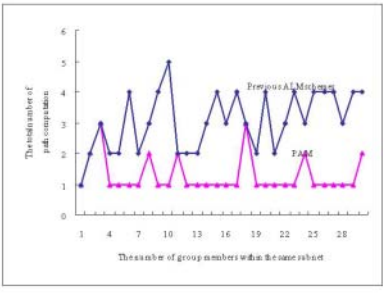


Fig. 5. Bandwidth utilization

4.1 Simulation Scenario

For simulation, we model the example networks, which consist of 1000 nodes and 35 sub networks. The 500 nodes are intended to join a multicast group. Each node is connected to randomly chosen DR. With random designation, the number of group members within a subnet ranges from 1 to 37. Each node requests multicast group join at arbitrary time and remains members throughout simulation time.

4.2 Performance Improvement by Adopting Host Group Model

Figure 3 shows the comparison result of average end-to-end delay vs. group members in an intra-domain. In Figure 3, previous ALM schemes have the longer latency and delay than PAM. In case of the previous ALM schemes, the delay increases whenever a group member joins the multicast group. On the other hand, in case of PAM, the delay increases only when new DR requests GROUP JOIN. If there is at least one member of multicast group within a subnet, a host's join does not affect the end-to-end delay at all in our scheme. It is shown as horizontal red line in Figure 3. The difference between End-system multicast and ours gets bigger and bigger as the number of group members increases. The number of path computation is shown in Figure 4. In PAM, the number of path computation increases when a new DR on different subnet joins the group. In case of the previous ALM schemes, new intra-domain DDT is computed even though there is a member within a same subnet. The last performance evaluation is to measure how much bandwidth is used for multicast within a subnet. In case of the previous ALM schemes, about 40% of total bandwidth is used for multicast. It is mainly because data packet is issued in the form of unicast. With PAM, the data packet is transmitted exactly once toward a subnet. This is because the number of group member within a subnet does not affect the usage of bandwidth.

Table 1. Terminology

n	Total number of group members
D_i	Delay time to receive response from i
k	A set that consists of several group members among all members
l	Total number of LDRs (l <= n)

Table 2. Simulation Result for First Join Delay

	HMTP			End-system multicast			PAM			
(1)	D_t	D_m	D_{ls}	D_t	D_m	D_{ls}	D_t	D_m	D_{ls}	(2)
10	128.117	128.117	0	154.693	96.9932	57.6998	40.0709	21.3477	18.723	1
20	233.734	233.734	0	176.5008	79.5076	96.9932	104.4223	83.0973	21.325	3
30	361.484	361.484	0	298.683	101.67	197.013	169.7446	129.73	40.015	7
50	583.318	583.318	0	377.034	108.547	268.487	198.8652	141.231	57.634	10
70	870.585	870.585	0	675.64	117.099	558.541	358.41	154.234	204.18	12
100	1192.61	1192.61	0	705.01	318.9	386.11	534.057	201.232	332.83	16
150	1765.18	1765.18	0	869.244	109.533	759.711	556.331	189.232	367.10	19
200	2349.34	2349.34	0	1502.933	369.343	1133.59	583.685	238.123	345.56	21
300	3543.12	3543.12	0	1520.759	92.5893	1428.17	879.174	289.324	589.85	22
400	4476.23	4476.23	0	2456.298	76.6083	2379.69	884.124	254.23	629.89	24
500	5923.98	5923.98	0	3514.101	520.461	2993.64	965.639	343.324	622.32	27

D_t = Total first join delay (1) # of group members
D_m = Delay time for measurement and group join (2) # of LDRs
D_{ls} = Delay time for link state exchange

4.3 Performance Improvement by Adopting Topology-Awareness

Table 2 and Figure 6 shows the simulation result of first join delay, which is defined as elapsed time from the time at which a host (DR in case of PAM) requests group join to the time at which a member's join is addressed to all related hosts (DRs). After this period, the optimal DDT can be established through proposed algorithm in HMTP and End-system multicast. So, the first join delay can be considered as real completion time for group join. First join delay in each scheme can be expressed as following equations, (1) - (3).

- HMTP: $D_{RP} + \sum_{i=root}^{n/2} D_i + D_{selected\_node}$ (1)
- End system multicast: $D_{RP} + D_{neighboring\_node} + D_{selected\_node}$ (2)
- PAM: $D_{RP} + \sum_{i=0}^{l} D_i + D_{selected\_LDR}$ (if LDR == NULL), D_{RP} (if LDR != NULL) (3)

According to the equations, (1) - (3), all schemes commonly include response time from RP (D_{RP}). In case of HMTP, it requires following additional times; measurement time for each member + response time from ultimately selected node. In the worst case, the measurement is accomplished as many as the total number of group members / 2 since it is based on tree structure. Different from

Table 3. Simulation Result for Control Overhead

	HMTP			End-system multicast			PAM		
(1)	C_t	C_m	C_{ls}	C_t	C_m	C_{ls}	C_t	C_m	C_{ls}
10	17	7	10	33	1	32	19	5	14
20	29	8	21	107	1	106	32	6	26
30	40	8	32	129	1	128	43	5	38
50	64	10	54	139	1	138	47	5	42
70	90	15	75	157	1	156	62	5	57
100	126	18	108	187	1	186	74	6	68
150	180	21	159	261	1	260	78	7	71
200	241	25	216	361	1	360	83	8	75
300	351	27	324	507	1	506	84	4	80
400	462	30	432	707	1	706	90	7	83
500	574	34	540	1002	1	1001	104	8	96

C_t = Total number of control packets (1) # of group members
C_m = Control packets for measurement
C_{ls}= Control packets for link state exchange

HMTP, End-system multicast requires following additional times; delay time to address a new join by periodically link state exchange + response time from ultimately selected node. This difference is easily seen in Table 2. Since HMTP does not need to exchange link state information at all, the delay for state exchange is 0. However, the delay for measurement increases as the group member size increases. On the other hand, End-system multicast requires shorter measurement time and longer delay time for link state exchange than HMTP does. In particular, delay time for measurement varies rapidly in End-system multicast. It is mainly dependent on round trip time between requesting member and the chosen candidate member among list of group member addressed from RP. But, in case of End-system multicast, link state exchange for constructing underlying mesh takes a long time. On the other hand, PAM has different first join delay depending on existence of LDR. If a LDR is not present, it incurs following additional times; measurement time for LDRs + responses time from ultimately selected LDR. Otherwise, it requires only response time from RP. For better understand, the number of LDRs is shown in the last column. One fact worth mentioning is that PAM has longer measurement and delay time for state exchange than End-system multicast even though the number of LDRs in PAM is similar to the number of group members in End-system multicast. This example is easily seen that the delay time for state exchange in PAM is 622.315 msec, on the other hand, End-system takes 197.013 msec where there are 27 LDRs and 30 members. It is because LDRs are located in each different domain so it takes longer time to reach each LDR. As a result, the three protocols can be arranged according to first join delay such as *PAM* < *End-system multicast* < *HMTP* in worst case.

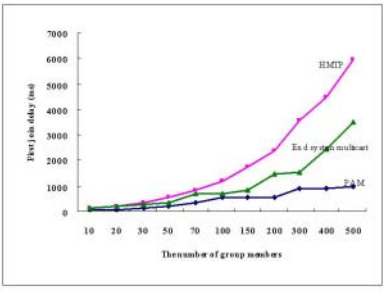


Fig. 6. First Join Delay

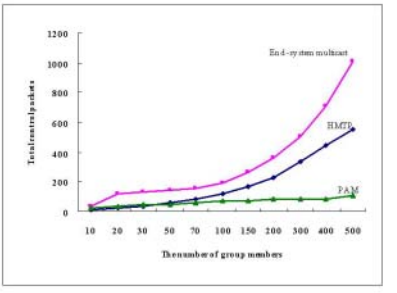


Fig. 7. Control Overheads

The total number of control packets in all cases consists of control packet to RP + control packet for measurement in join process + control packet for DDT maintenance. So, the total number of control packets in scheme is largely dependent of latter two factors. That is, the number of control packets is incurred differently according to how to construct the underlying DDT, in particular, control packets for DDT maintenance. In case of HMTP, it equals to control packet for measurement on tree + control packets for tree maintenance. In case of End-system multicast, it equals to control packet for join request (=1) + control packets for mesh maintenance and DDT creation. On the other hand, the required number of control packet in PAM is defined as control packets for LDR measurement + control packets for update in intra-domain + control packets for LDR maintenance and inter-domain DDT creation. Similar to first join delay time, total number of control packets on each protocol is defined as equation (4) - (6).

- HMTP: $C_{RP} + n/2 + C_{m\_tree}$ (4)
- End system multicast: $C_{RP} + 1 + C_{m\_mesh}$ (5)
- PAM: $C_{RP} + C_l + C_{intra\_domain} + C_{m\_LDR}$ (6)

In terms of control packets for DDT maintenance, it is mostly dependent on the number of components of underlying structure such as tree and mesh. So, the control packets for maintenance is roughly expected to be arranged in order such as $C_{m\_mesh} > C_{m\_tree} > C_{m\_LDR}$ because mesh in End-system multicast consists of all group members (n), on the other hand, members in HMTP maintains $n/2$ members in the worst case. In case of PAM, only one LDR exists among all member DRs of each domain, therefore it is easily expected that PAM generates the fewest number of control packets in terms of DDT maintenance.

Table 3 and Figure 7 show the simulation result for total control packets depending on the number of group members. As you can see in Table 3, control packets for measurement are fixed as 1 in case of End-system multicast. For End-system multicast, we assume that the RP replies with the list of four members and then a new member wants to request join the multicast group to one of them. Interesting fact in Table 3 is that there is a little difference in the number

of control packets for measurement rather than in terms of link state exchange. Thus, it is clear that the total number of control packets is mainly influenced by the number of control packets for maintenance. Due to adapting locality property and host group model, PAM incurs the smallest number of control packets.

5 Conclusion and Future Works

In this paper, we propose a new ALM scheme (PAM), which adapts host group model and topology-awareness property. Due to above properties, PAM has many advantages in terms of scalability, simplicity, practicality, stability, and reduced link stress. Using simulation results, these advantages are evaluated. For all cases, PAM significantly improves the performance.

For further study, we plan to design more concrete architecture where host group model is not applicable. Also, automatic GID generation and multicast address allocation are another issue of PAM to be considered.

References

[1] C. Diot et al., "Deployment Issues for the IP Multicast Service and Architecture," IEEE Network, Vol. 14, Jan.-Feb. 2000, pp. 78 - 88. 595
[2] D. Kosiur, "IP Multicasting : The Complete Guide to Interactive Corporate Networks, John Wiley & Sons, Inc.," 1998, pp. 57 - 58. 595
[3] H. Holbrook and B.Cain, "Source-Specific Multicast IP," IETF Internet-Draft, draft-ietf-holbrook-ssm-arch-00.txt, 2000.
[4] R. Boivie et al., "Explicit Multicast (Xcast) Basic Specification," IETF Internet-Draft, draft-ooms-xcast-basic-spec-02.txt, 2001.
[5] M. Shin et al., "Explicit Multicast Extension(Xcast+) for Effcient Multicast Packet Delivery," ETRI journal, Vol. 23, No. 4, Dec. 2001.
[6] Y. Chu et al., "A Case for End System Multicast," IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast, Oct. 2002. 595, 598
[7] P. Francis, "Yoid: Extending the Internet Multicast Architecture," ACIRI Technical Re-port, Apr. 2000.
[8] B. Zhang et al., "Host Multicast: A Framework for Delivering Multicast To End Users," IEEE INFOCOM'02, Jun. 2002.
[9] S. Banerjee et al., "Scalable Application Layer Multicast," ACM SIGCOMM'02, Aug. 2002. 598
[10] S. Ratnasamy et al., "Application-Level Multicast Using Content-Addressable Networks," 3rd International Workshop on Networked Group Communication, Nov. 2001.
[11] A. Rowstron et al., "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," IFIP/ACM ICDCP'01, Nov. 2001.
[12] B. Y. Zhao et al., "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing," Technical Report, UCB/CSD-01-1141, University of California, Berkeley, CA, USA, Apr. 2001.
[13] C. G. Plaxton and A. W.Richa, "Accessing Nearby Copies of Replicated Objects in A Distributed Environment, " In ACM Symposium on Parallel Algorithms and Architectures, Jun. 1997.

[14] M. Castro et al., "SCRIBE: A Large-Scale and Decentralized Application-Level Multicast Architecture," IEEE Journal on Selected Areas in communications (JSAC), 2002.
[15] M. S. Kwon et al., "Topology-Aware Overlay Networks for Group Communication," NOSSDAV'02, May 2002. 595
[16] S. Deering, "Host Extensions for IP Multicasting," IETF RFC-1112, Aug. 1989.
[17] NS-2 Simulator, `http://www.isi.edu/nsnam/ns/`. 599

A New TCP Congestion Control for High-Speed Long-Distance Networks

Byunghun Song, Kwangsue Chung, and Seung Hyong Rhee

School of Electronics Engineering, Kwangwoon University, Korea
byungh@adams.kw.ac.kr
{kchung,shrhee}@daisy.kw.ac.kr

Abstract. It has been reported that TCP Reno underutilizes network bandwidth, especially in High-Speed Long-Distance (HSLD) networks, while it has been widely adopted the dominant transport protocol in current Internet. Moreover, the Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm of the TCP Reno has equilibrium and dynamic problems that become more and more severe as the bandwidth-delay product increases. An ideal TCP congestion control algorithm that achieves a high utilization, a small queueing delay, a stable behavior and fairness in bandwidth allocation has been major objectives of networking research in recent years. In this paper, we propose a new congestion control protocol, called TCP KWIK, a modification to the TCP Reno for HSLD networks. TCP KWIK uses delay-based feedback information to address these problems in order to stabilize a network around a fair and efficient operating point. We argue that the delay-based approach is better than the end-to-end congestion control as the networks capacity increases. Their advantage is small at low speed but decisive at high speed. The simulation results demonstrate the effectiveness of our proposed algorithm.

1 Introduction

Congestion control of the Internet requires a distributed algorithm to share network resources among competing users. The mechanism is very useful in the situations where the availability of resources and the set of competing users dynamically vary, yet efficient sharing is desired. These constraints, unpredictable supply and demand and efficient operation, necessarily lead to feedback control as the only approach, where traffic sources dynamically adapt their rates to congestion in their paths. On the Internet, this is performed by the TCP at both source and destination computers involved in data transfers. The congestion control algorithm in the current TCP, which we refer to as Reno in this paper, were developed in 1988 and have gone through several enhancements since.

TCP Reno has adopted the loss-based algorithm. However, it has been reported that TCP Reno substantially underutilizes network bandwidth in High-Speed Long-Distance (HSLD) networks. TCP Reno increases its congestion window by one at every round trip time (RTT) and reduces it by half at a loss

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 606–615, 2004.

event. For example, in order for TCP to increase its window for full utilization of 10Gbps with 1500-byte packets, it requires more than 83,333 RTTs. With 100ms RTT, it takes approximately 1.5 hours, and for full utilization in steady state, the loss rate cannot be more than 1 loss event per 5,000,000,000 packets which is less than the theoretical limit of the network's bit error rates [1].

In this paper, we argue that the solution to these problems requires a delay-based algorithm that is scalable to the capacity. Delay-based congestion control has been proposed, e.g., in [4]. Its advantage over loss-based algorithms is small at low speed, but decisive at high speed, as we will see below. However, as pointed out in [5], delay can be a poor or untimely predictor of packet loss. Therefore using a delay-based algorithm to augment the basic Additive Increase Multiplicative Decrease (AIMD) mechanism of TCP Reno may be a wrong approach to resolve problems in the networks with large bandwidth-delay products. Instead, a new approach that fully exploits delay as a congestion measure, augmented with loss event information, is required.

The rest of this paper is organized as follows: We present the background and related work in Section 2. In Section 3, we describe proposed new congestion control algorithm. Section 4 illustrates the performance evaluation of the proposed algorithm. Conclusions and future work are presented in Section 5.

2 Background and Related Work

A congestion control algorithm can be designed at two levels. The *flow-level* design aims to achieve the goals such as high utilization, low queueing delay and loss, fairness, and stability. The *packet-level* design implements these *flow-level* goals with the constraints imposed by end-to-end control. TCP Reno, a typical loss-based algorithm, considered the *packet-level* implementation first. The resulting *flow-level* properties, such as fairness, stability, and the relationship between equilibrium window and loss probability, were then understood as an afterthought. In contrast, the *packet-level* design of our proposed algorithm is explicitly guided by *flow-level* goals. It is important to distinguish between *packet-level* and *flow-level* difficulties, because they must be addressed by different means. In this section, we first describe TCP Reno at both the *packet-level* and *flow-level*, and then discuss what seems to be the problem in HSLD networks.

2.1 TCP Reno Congestion Control

The congestion avoidance algorithm of TCP Reno and its variants have the form of AIMD in Figure 1. The AIMD algorithm for window adjustment is a *packet-level* model, but its *flow-level* properties such as throughput, fairness, and stability are poor. These properties can be understood with a *flow-level* model of the Reno, e.g., [6]. In each RTT, the window $w_i(t)$ of source i increases by 1 packet, and decreases by

$$x_i(t)p_i(t) \cdot \frac{1}{2} \cdot \frac{4}{3} w_i(t) packets \ (x_i(t) \approx \frac{w_i(t)}{T_i(t)} pkts/sec) \tag{1}$$

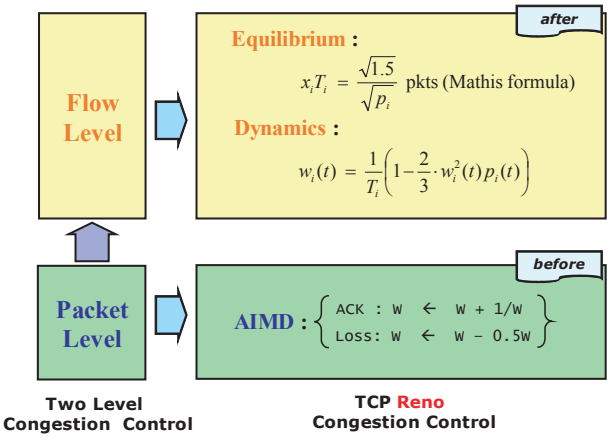


Fig. 1. TCP Reno Congestion Control

$T_i(t)$ is the round-trip time and $p_i(t)$ is the end-to-end loss probability in period t. Here, $4w_i(t)/3$ is the peak window size that gives the average window of $w_i(t)$. Setting $w_i(t) = 0$ in the dynamics equation yields the well-known $1/\sqrt{p}$ formula for TCP Reno discovered in [7], which relates loss probability to window size in equilibrium.

In summary, two equations of Figure 1 describe the *flow-level* dynamics and the equilibrium, respectively, for TCP Reno. They differ in their choices of marginal dynamic and equilibrium equation, and whether the congestion measure $p_i(t)$ is loss probability or queueing delay.

2.2 Problems in HSLD Networks

As the bandwidth-delay product continues to grow, TCP Reno will eventually become a performance bottleneck itself. The following two problems contribute to the poor performance of TCP Reno in networks with large bandwidth-delay products, as described by Jin *etal* [8].

i) **Dynamic Oscillation Problem:** The cause of the oscillatory behavior in TCP Reno lies in its design at both the *packet-level* and *flow-level*. At the *packet-level*, the choice of loss $p_i(t)$ congestion signal necessarily leads to oscillation, and the parameter setting in Reno worsens the situation as bandwidth-delay product increases. At the *flow-level*, the system dynamics equation is unstable at large bandwidth-delay product.

ii) **Equilibrium Problem:** The end-to-end loss probability must be exceedingly small to sustain a large window size at large capacity, making the equilibrium difficult to maintain in practice, as bandwidth-delay product increases. Even though equilibrium is a *flow-level* notion, this problem manifests itself at the *packet-level*, where a source increments its window too

slowly and decrements it too drastically. Therefore, TCP Reno substantially under utilizes network bandwidth in HSLD networks.

2.3 Existing Approach for HSLD Networks

After recognizing limitation of Reno, the networking research community responded quickly. Several promising new loss-based congestion control protocols have been put forward: High Speed TCP (HSTCP) [2], Scalable TCP (STCP) [3], these protocols adaptively adjust their increase rates based on the current window size. So the larger the congestion window is, the faster it grows. In other words, the increment functions of loss-based protocols such as HSTCP and STCP are more aggressively than Reno. And they decrement less drastically. At the *flow-level*, this means that, in equilibrium, both HSTCP and STCP can tolerate larger loss probabilities than TCP Reno. It does not, however, solve the dynamic problems at the *packet-level* and the *flow-level*.

At the *packet-level*, as mentioned above, a natural way to avoid oscillation is to use delay-based approach, as opposed to loss-based approach, congestion signal. Without explicit feedback, this means adjusting window based either on loss probability or on queueing delay. Queueing delay can be more accurately estimated than loss probability both because packet losses in networks with large bandwidth-delay product are rare events (probability on the order 10 or smaller), and because loss samples provide coarser information than queueing delay samples. Indeed, each measurement of packet loss (whether a packet is lost) provides raw information for the filtering of noise, whereas each measurement of queueing delay provides proactive information. This allows an equation-based implementation to stabilize a network in a steady state with a target fairness and high utilization.

At the *flow-level*, the dynamics of the feedback system must be stable in the presence of delay, as the network capacity increases. Here, again, queueing delay has an advantage over loss probability as a congestion measure: the dynamics of queueing delay seems to have the right scaling with respect to network capacity. This helps maintain stability as network speed grows.

2.4 RTT Unfairness Problem of HSTCP and STCP

Our study reveals that notwithstanding their scalability and TCP friendliness properties, HSTCP and STCP have a serious RTT unfairness problem when multiple flows with different RTT delays are competing for the same bottleneck bandwidth. We define the RTT unfairness of two competing flows to be the ratio of windows in terms of their RTT ratio. Under a completely synchronized loss model, STCP does not even find a convergence point such that shorter RTT flows eventually starve off longer RTT flows. We also find that in HSTCP has RTT unfairness problem. RTT unfairness stems from the adaptability of these protocols - ironically the very reason that makes them more scalable to large bandwidth - in these protocols, a larger window increases faster than a smaller

window. Compounded with a delay difference, RTT unfairness gets worse as the window of a shorter RTT flow grows faster than that of a longer RTT flow.

3 A New TCP Congestion Control

In this paper, we propose a new congestion control protocol, called KWIK[1], for HSLD networks. Unlike the approach taken by Reno, HSTCP, and STCP, this approach provides proactive information for the end-to-end congestion measure $p_i(t)$. As argued in Section 2.3, in the absence of explicit feedback, queueing delay seems to be the only viable choice for congestion measure, as network capacity increases.

This approach, with proper choice of *flow-level* and *packet-level* designs, can address the four difficulties of Reno at large capacity. First, by accurately estimating how far the current transmission rate is from the equilibrium value, the scheme can drive the system rapidly, yet in a fair and stable manner, toward the equilibrium.

3.1 Architecture

As the Internet scales up in speed and size, KWIK's stability and performance become harder to control. The fluid network theory that allows us to understand the equilibrium and stability properties of large networks under end-to-end control forms the foundation of KWIK. It plays an important role in its implementation by providing a framework to understand issues, clarify ideas and suggest directions, leading to a more robust and better performing implementation.

Figure 2 illustrates the operating points chosen by KWIK, using the single-link single-flow case. It shows queueing delay as a function of queue length. The queueing delay starts to build up after point A where window equals bandwidth propagation-delay product, until point C where the queue overflows. Since Reno oscillates around point C, the peak window size goes beyond point C, and the amount of overshoot depends on the feedback delay. The minimum window in steady state is half of the peak window. This is the basis for the rule of thumb that bottleneck buffer should be at least one bandwidth-delay product: the minimum window will then be above point A, and buffer will not empty in steady operation, yielding full utilization. Full utilization, even if achievable, comes at the cost of severe oscillations and potentially large queueing delay. The KWIK scheme proposes to oscillate around point B, the midpoint between A and C. The KWIK scheme increases congestion window linearly by λ packet per RTT, until point B and proposed scheme increases congestion window linearly by one packet per RTT. The $q_i(t)$is queueing delay. $baseRTT$ is the minimum of all measured RTT. δ is tuning value by loss event rate. In the loss-based schemes, the congestion signal is raw information, and hence congestion window must

[1] KWIK(QUICK[kwik] : KwangWoon In Korea) means that this algorithm quickly keeps high utilization at HSLD networks.

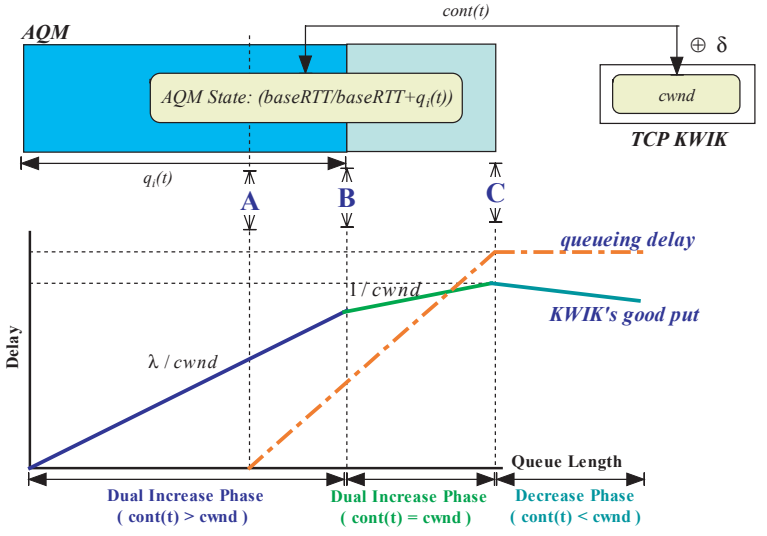


Fig. 2. Architecture of KWIK

oscillate. The congestion window can be stabilized only if proactive information, such as queueing delay, is used. Therefore, using queueing delay as the congestion measure $p_i(t)$ allows the network to stabilize in the region below the overflowing point, around point B in Figure 2, when the queue size is sufficiently large. The stabilization at this operating point eliminates large queueing delay and unnecessary packet loss.

3.2 Algorithm

In KWIK algorithm, the sender measures condition variable *Con(t)* in equation (2) to detect status of AQM(Active Queue Management).

$$Con(t) = \frac{1}{2} \cdot \left(cwnd_i(t) \cdot \frac{BaseRTT}{BaseRTT+q_i(t)} + \delta_i(t) \right)$$

$$\left[\begin{array}{l} if(cwnd(t) < Con(t)) \\ cwnd(t+1) = cwnd(t) + \lambda/cwnd(t) \\ if(cwnd(t) = Con(t)) \\ cwnd(t+1) = cwnd(t) + 1/cwnd(t) \\ if(cwnd(t) > Con(t)) \\ cwnd(t+1) = cwnd(t) - 1 \end{array} \right] \quad (2)$$

The window is reduced proactively before network buffer can overflow. By this "ideal" window adjustment, KWIK can avoid packet overflowing and meanwhile reaches maximum utilization of the bottleneck link. Thus, all packet losses in KWIK are assumed to occur by non-congestion reasons. Similar to the fact that

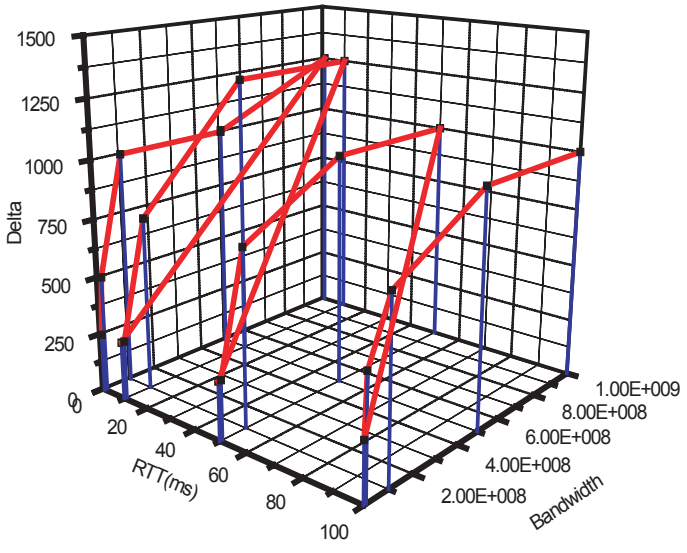


Fig. 3. Change in Optimal δ Value by Bandwidth and RTT

Reno rigidly interprets all packet losses are due to congestion, KWIK rigidly interprets all packet losses are due to other reasons. Thus we have a protocol which resembles TCP Vegas, but KWIK algorithm is designed to provide the stability guarantees and high performance.

The parameter δ affects both the equilibrium and dynamic behavior of a network. According to bandwidth(or loss event rate) and RTT, δ determines weight value for the window increment. So it use to tune the aggressiveness of the KWIK's window increase by condition variable *Con(t)*. The change in optimal value of δ by bandwidth and RTT is illustrated in Figure 3. Therefore, in order to maintain optimal throughput of each KWIK connection, δ should be set according to the Figure 3.

4 Performance Evaluation

In this section, we compare the performance of KWIK using simulation with that of HSTCP, STCP, and AIMD. Every experiment uses the same simulation setup described in Section 4.1. Unless explicitly stated, the same amount of background traffic is used for all the experimental runs. We evaluate KWIK, AIMD, HSTCP, and STCP for the following properties: throughput, oscillation, RTT unfairness.

4.1 Simulation Network Topology

Figure 4 shows the ns2 [9] simulation setup that we use throughout the paper. Various bottleneck capacity and delays are tested. The buffer space at the bottle-

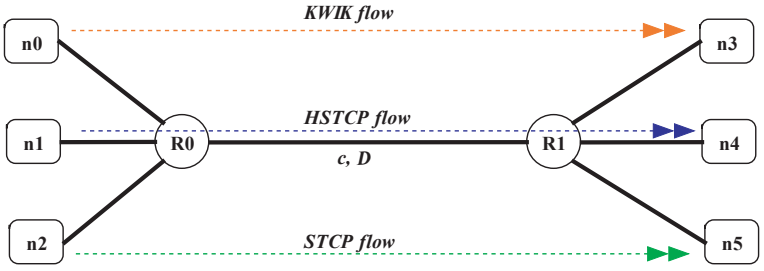


Fig. 4. Simulation Network Topology

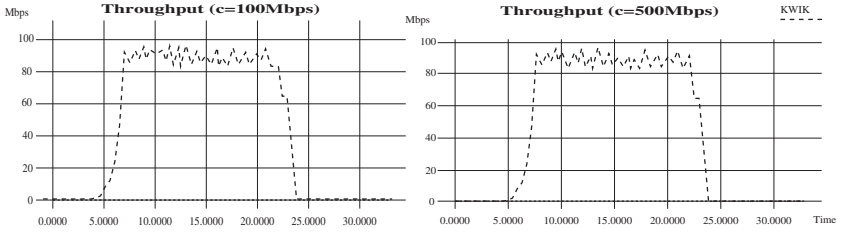


Fig. 5. Throughput of KWIK

neck router is set to 100% of the bandwidth and delay products of the bottleneck link. Every traffic passes through the bottleneck link; each link is configured to have different RTTs and different starting times and end times to reduce the phase effect.

4.2 Throughput and Oscillation

We first show the steady-state performance of the KWIK in comparison with TCP Reno. All the simulations in this part use ftp traffic on a single bottleneck link. When the capacity varies from 100Mbps to 500Mbps, the KWIK flows always keeps high utilization as expected and very small queues. The throughput of KWIK flows are illustrated in Figure 5. Moreover, we can see that our proposed KWIK provides smoother transmission rate than TCP Reno in Figure 6.

4.3 RTT Unfairness

In this experiment, two high-speed flows with a different RTT share the bottleneck. The RTT of flow 1 is 40ms. We vary the RTT of flow 2 among 40ms, 120ms, and 240ms. We run two groups of simulation, each with different bottleneck bandwidth: 100Mbps and 1Gbps. This setup allows the protocols to be tested for RTT fairness. According to the results, KWIK shows the acceptable RTT unfairness. We show only the results of drop tail. Tables 1 and 2 show the results for the runs in 100Mbps and 1Gbps respectively. It can be seen that

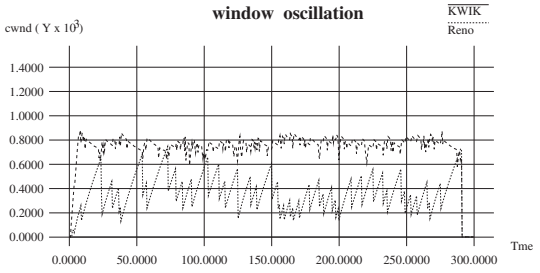


Fig. 6. Oscillation Behavior of KWIK

Table 1. RTT Unfairness Comparison in 100Mbps

RTT Ratio	**1**	**3**	**6**
AIMD	0.99	7.31	26.12
HSTCP	1.13	10.42	51.09
STCP	1.12	27.84	72.74
KWIK	1.22	2.36	23.46

Table 2. RTT Unfairness Comparison in 1Gbps

RTT Ratio	**1**	**3**	**6**
AIMD	1.05	6.56	22.55
HSTCP	0.99	47.42	131.03
STCP	0.92	140.52	300.32
KWIK	1.19	2.26	24.10

the RTT unfairness of KWIK is relatively comparable to AIMD. In Table 1 and 2, the performance of KWIK does not deteriorate much while HSTCP and STCP have improved RTT unfairness. Therefore, the RTT unfairness of KWIK is much better than HSTCP and STCP. HSTCP and STCP tend to starve long RTT flows under HSLD network environments.

5 Conclusion

This paper aims at designing a congestion control system that scales gracefully with network capacity, providing high utilization, low queueing delay, dynamic stability, and fairness among users.

In this paper, we propose the TCP KWIK, a modification to the TCP Reno for HSLD networks. TCP KWIK uses delay-based feedback information to address these problems in order to stabilize a network around a fair and efficient operating point. Therefore, we develop packet-level implementations of KWIK, using queueing delay with loss event information, as means of communicating the congestion measure from links to sources.

We compare the performance of KWIK using simulation with that of HSTCP, STCP, and AIMD. Consequently, the throughput, oscillation, and RTT unfairness of KWIK are much better than HSTCP and STCP.

Acknowledgement

This research has been conducted by the Research Grant of Kwangwoon University in 2003. It is also supported in part by Korea Science and Engineering Foundation under contract number R01-2002-000-00179-0(2002).

References

[1] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High-Bandwidth Delay Product Networks," *Proceeding of ACM Sigcomm*, August 2002. 607
[2] S. Floyd, "HighSpeed TCP for Large Congestion Windows," Internet draft draft-oyd-tcp-highspeed-02.txt, February 2003. 609
[3] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks," Submitted for publication, `http://www-lce.eng.cam.ac.uk/~ctk21/scalable/`, December 2002. 609
[4] L. Brakmo and L. Peterson, "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet," *Proceeding of IEEE Communications*, October 1995. 607
[5] J. Martin, A. Nilsson, and I. Rhee, "Delay-Based Congestion Avoidance for TCP," *Proceeding of ACM Networking*, June 2003. 607
[6] F. Kelly, A. Maulloo, and D. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Proceeding of IEEE Communications*, March 1998. 607
[7] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Proceeding of ACM Computer Communication Review*, July 1997. 608
[8] C. Jin, D. Wei, and S. Low, "TCP FAST: Motivation, Architecture, Algorithms, Performance," *Proceeding of IEEE Infocomm*, July 2003. 608
[9] UCB LBNL VINT, "Network Simulator ns (Version 2)", `http://www.isi.edu/nsnam/ns/`. 612

A Scalable Parallel Lookup Framework Avoiding Longest Prefix Match*

Zhiyong Liang, Ke Xu, and Jianping Wu

Department of Computer Science, Tsinghua University
Beijing, P.R.China, 100084
{lzy,xuke}@csnet1.cs.tsinghua.edu.cn
jianping@cernet.edu.cn
http://netlab.cs.tsinghua.edu.cn

Abstract. Fast routing lookups are crucial for the forwarding performance of IP routers. Longest prefix match makes routing lookups difficult. This paper proposes a method to partition a routing table. The method can divide all prefixes in a routing table into several prefix sets where prefixes don't overlap. Based on the method, this paper also presents a common parallel lookup framework(PRLF) that reduces "longest prefix matching" in all the prefixes to "only prefix matching" in several prefix sets. The framework can effectively simplify the design of lookup algorithms and improve lookup performance. The framework is suitable for most lookup algorithms. For simple binary search algorithm, the framework can reach $log_2 2N/B$ lookup complexity (where N is prefix number in a routing table and B is an integer bigger than 4). Also, the framework can scale to IPv6 easily.

1 Introduction

Today, ports with high rates are applied widely by high-performance routers, such as OC48, OC192 and even higher rates. For a port with OC192, routers need approximately forward 30 million packets and do 30 million lookups in a second (assuming that packet length is 40 bytes). Fast lookups become crucial for improving the forwarding performance of routers. In 1990s, IETF proposed a new addressing scheme-CIDR [1]. CIDR allows routing aggregation. So a routing lookup must find the longest matching prefix and then become complex and difficult. The next generation Internet adopts IPv6 addressing scheme[2]. In IPv6, address length is expanded from 32 bits to 128 bits and the size of routing tables will be larger. Routing lookups will face a more rigorous challenge.

To solve fast lookup problem, researchers have proposed many different mechanisms and algorithms [3]-[8]. At present TCAM is a popular device for fast lookups. TCAM can reach O(1) lookup complexity. However, TCAM still has

* Supported by: (1) the National Natural Science Foundation of China (No. 90104002; No. 69725003); (2) the National Basic Research Program of China No. 2003CB314801.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 616–625, 2004.

some disadvantages: high cost, high power consumption. Some works has been done to overcome these disadvantages [10].

"Longest prefix match" makes routing lookups very tough. Since prefixes can overlap, traditional lookup algorithms(such as hash lookup, binary search) cannot directly be applied. To reduce the difficulty of routing lookups, researchers proposed some methods to transform "longest prefix match" into "only prefix match".

Srinivasan et al. suggested the leaf pushing method [5]. Leaf pushing can reduce "longest prefix match" to "only prefix match". However, it brings two problems. First, lots of duplicated routing info exists. Second, update is complex.

Waldvogel et al. presented another converting method for "longest prefix match"[4]. All prefixes are divided into several sets according to prefix length. A lookup in all prefixes can be divided into several lookups in several sets. And "longest prefix match " is also be transformed into "only prefix match ".

Mohammad et al. proposed a port-based method to partition prefixes [6]. All routing prefixes are partitioned into several sets according to different egress ports. Assuming that routers don't connect to any shared link, prefixes in each set will not overlap. "Longest prefix match" can be converted to "only prefix match".

The contribution of this paper mainly includes two parts. Firstly, we propose a method to partition prefixes. The method is based on a binary trie. With it, all prefixes can be divided into several prefix sets where prefixes don't overlap. Then we can transform "longest prefix match " problem in all prefixes to "only prefix match " problem in several sets. Secondly, we propose a common parallel lookup framework that is suitable for most lookup algorithms. The framework performs lookups parallel in several non-overlapping prefix sets. "Only prefix match " in a set can simplify the design of lookup algorithms. If simple binary search is employed, the framework can achieve O($log_2 2N/B$) lookup complexity.

2 Prefix Partition and Parallel Lookups

2.1 A Binary Trie

A binary trie is a tree-based structure that can be used to do "longest prefix match". Fig. 1 shows a simple binary trie. In Fig. 1, ten prefixes (prefix a to j) are stored in the trie. Gray nodes hold these prefixes. For example, gray node b represents a bit string "01000" that is just prefix b.

Nodes with prefixes are called prefix nodes that store routing info, for example, all gray nodes in Fig.1. Nodes without prefixes are called non-prefix nodes that don't contain routing info, for example, all white nodes in Fig.1.

From the binary trie structure, we can derive its two important attributes:

1. If prefix node n_1 is the ancestor of prefix node n_2, prefixes represented by n_1 and n_2 must overlap, for example, node a and c in Fig.1.
2. If prefix p_1 and prefix p_2 overlap, nodes with p_1 and p_2 must be the relation of ancestor to descendant, for example prefix d and g or prefix d and f in Fig.1.

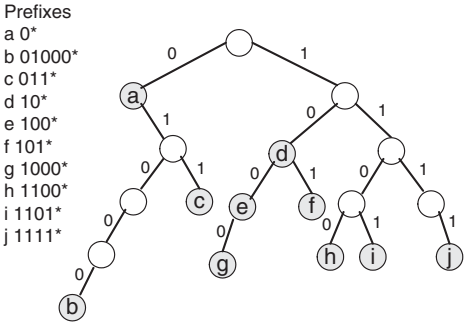


Fig. 1. A binary trie

From the above attributes, we can see that the relation of ancestor to descendant between prefix nodes is equivalent to overlapping between prefixes. If all prefix nodes are partitioned into several sets and any two nodes in each set aren't the relation of ancestor to descendant, prefixes represented by nodes in each set must be non-overlapping. Therefore, we can transform prefix partition into prefix node partition.

2.2 Prefix Partition

To partition prefix nodes, we introduce a new description parameter for trie nodes - height.

Definition 1. *Height.*

For a given node n (prefix node or non-prefix node), there are multi paths from n to n's multi descendant leaf nodes. Among these paths, let p_{max} be the path containing most prefix nodes. The number of prefix nodes (not including node n) in p_{max} is called n's height.

From Def. 1, we can conclude that the height of all leaf nodes is 0.

Definition 2. *Height set h.*

A height set h is defined as a set that contains all prefixes whose nodes in the trie have height h.

We denote height set h by HeightSet(h). h is also called set height. According to prefix node height, we can partition prefixes in Fig.1 into several following height sets: HeightSet(0) = {b, c, g, f, h, i, j}; HeightSet(1) ={a,e}; HeightSet(2)={d}.

With two attributes of a binary trie, we can derive three theorems about height sets.

Theorem 1. *All prefixes in a height set are non-overlapping.*

Theorem 2. *For a given IP address, a high set contains only one match prefix or doesn't contain any match prefix.*

Theorem 3. *For a given IP address, HeightSet(i) has one match prefix, $prefix_i$; HeightSet(j) also has one match prefix, $prefix_j$(i!=j). If $i < j$, length of $prefix_i$ is longer than length of $prefix_j$. Otherwise, length of $prefix_i$ is shorter than length of $prefix_j$.*[1]

According to Theo.1, a height set is a non-overlapping prefix set. So we can partition a routing table into several prefix sets with different height. We perform prefix partition in five real routing tables [9]. And we find two character. First, the number of height sets is very few. All partitioned routing tables have only five height sets. Second, height sets are very different in prefix number. The large difference of prefix number is unbeneficial for parallel lookups in several height sets. In the latter section, we will discuss and solve this problem.

2.3 Parallel Lookups

We can do parallel lookups in height sets to improve lookup performance. Then we will transform a "longest prefix match" lookup in all prefixes into several "only prefix match" lookups in several height sets.

Parallel lookups have the several following advantages: 1.Parallel lookups in height sets can increase lookup speed; 2."Only prefix match " can simplify the design of lookup algorithms; 3.Simplified structures can reduce update and storage complexity.

3 Parallel Routing Lookup Framework

3.1 Architecture

Based on prefix partition, we propose a parallel routing lookup framework (abbreviated as PRLF). Fig.2 shows the architecture of PRLF that includes four parts: dispenser, banks, selector and nexthop mapping table.

The dispenser dispatches a destination IP address to all the banks so that banks can perform lookups parallel.

A bank takes responsibility for doing "only prefix match" lookups in its prefixes. A bank stores a height set. Banks are numbered and the bank with smaller number stores the height set with lower height. For example, HeightSet(0) is stored in bank 0, HeightSet(1) in bank 1, HeightSet(2) in bank 2, and so on. From Section 2.2, real routing tables can be partitioned into at most five height sets. Hence, only five banks are contained in the architecture. However, bank number can also be increased.

Besides routing prefixes, a bank should also store forwarding info-(nexthop IP, egress port). However, in the architecture forwarding info is stored in the

[1] Due to limited pages, we will not give the proofs of the above three theorems in this paper. If you need them, you can mail to us or visit our homepage.

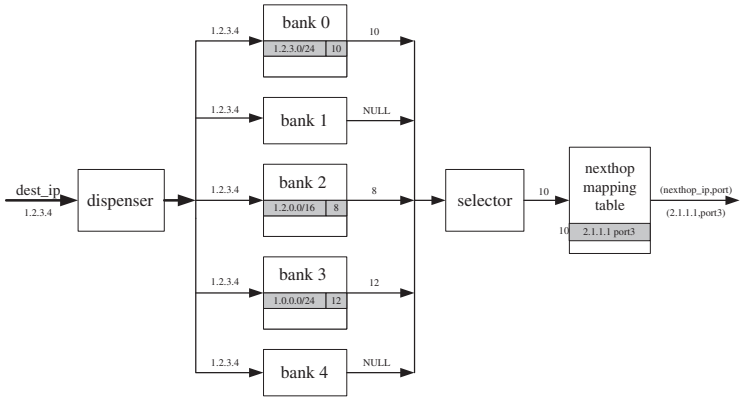


Fig. 2. PRLF architecture

nexthop mapping table (abbreviated as NMT). A bank only stores indexes of the NMT entries. This way can reduce memory requirement of banks.

After a bank completes a lookup, it will return a NMT index and send it to the selector. Then the selector will choose the result from the smallest number bank as the final lookup result.

The nexthop mapping table (NMT) holds real forwarding info. Each entry in the NMT stores one pair of (nexthop IP, egress port). According to the entry index, the NMT can output the corresponding (nexthop IP, egress port).

From the architecture, we can conclude its two traits:

1. Prefixes stored in a bank don't overlap.
2. For a given IP address, if multi banks contain matching prefixes, the smallest number bank holds the longest matching prefix.

3.2 Lookup

In the PRLF, a lookup is done as the four following steps:

1. The dispenser dispatches a destination address to all banks.
2. All banks perform lookups in parallel and send lookup results to the selector.
3. From multi NMT indexes, the selector chooses the index from the smallest bank.
4. According to the chosen index, the NMT output the final forwarding info-(nexthop IP, egress port).

For example, for a given IP 1.2.3.4 in Fig.2, firstly the dispenser dispatches 1.2.3.4 to all five banks. Secondly, five banks perform lookups. And three banks contain match prefixes that are marked by gray color, including 1.2.3.0/24 in bank 0, 1.2.0.0/16 in bank 2 and 1.0.0.0/8 in bank 3. The corresponding NMT indexes to the there match prefixes are 10, 8 and 12, respectively. These indexes

are sent to the selector. Thirdly, since bank 0 has the smallest number among bank 0, bank 2 and bank 3, the selector chooses index 10 from bank 0 as the final lookup result. Finally, the NMT outputs forwarding info in the 10^{th} entry-(2.1.1.1, port3).

3.3 Nexthop Mapping Table

To design the NMT, we examine five real routing tables [9] and observe the number of different forwarding info.

We find that different forwarding info is very few (at most 33 in the five routing tables). Based on it, the NMT size is chosen as 64 in the PRLF. 6 bits ($2^6 = 64$) can satisfy the storage for the NMT index. Since bank holds lots of prefixes and corresponding indexes, smaller storage for an index can reduce bank memory requirement greatly. In addition, an index is transferred between banks and selector, selector and NMT. A shorter index can also decrease circuit complexity in hardware.

3.4 Lookup Algorithm in a Bank

Non-overlapping in bank prefixes simplifies the design of lookup algorithms. In the latter experiment, we apply simple binary search to implement lookups in a bank. Besides binary search, other most lookup algorithms are also suitable for the PRLF, such as tries, hash lookup, multiway search [3], etc.

4 Improvement for PRLF

4.1 Balancing Prefixes in Different Banks

In the PRLF, the selector can't work till all banks finish their lookups. Hence, parallel lookup speed is entirely determined by the slowest bank. When designing lookup algorithms in banks, we should guarantee that all banks have close lookup speed. However, if prefix number has large difference between banks, that will be difficult. For example, for binary search, lookup complexity is $log_2 2N$. The large difference of N can directly results in the large difference of lookup speed. In addition, for some algorithms (such as tries and hash lookup), their lookup complexity is independent of N. However, N can indirectly affects lookup speed. In tries, if N increases, nodes in tries will increase. Average height of tires will rise correspondingly. That will slow down lookup speed. In hash table, increasing N can bring more collision. Lookup speed will also be affected.

From the above, we can see the difference in prefix number is unhelpful for parallel lookups. To solve this problem, we'll present an efficient method to balance prefixes in different banks and to reduce the difference in prefix number.

Our method has two objectives: first, balancing prefixes in different banks; second, keeping the traits of the PRLF. For the second objective, we need ensure that prefixes in a bank don't overlap and the bank with the smallest number holds the longest match prefix. Our method is based on a kind of special prefix nodes in a binary trie - orphan nodes.

Definition 3. *Orphan node*

A leaf node is called as an orphan node, if the path from it to root node doesn't contain other prefix nodes except itself.

Obviously the height of an orphan node is zero and all its ancestor nodes are non-prefix nodes. In addition, since a leaf node must be a prefix node in a binary trie, an orphan node is also a prefix node.

For example, in Fig.1 node h is just an orphan node.

Definition 4. *Orphan set*

The orphan set is defined as a set that contains all prefixes whose nodes are orphan nodes.

We denote the orphan set by OrphanSet. Since the height of an orphan node is zero, OrphanSet is a part of HeightSet(0). However for OrphanSet's particularity, we separate OrphanSet from HeightSet(0). In the latter section, if not explicitly say, HeightSet(0) won't include OrphanSet.

In Fig.1, all prefixes can be repartitioned into the following sets: OrphanSet = {h, i, j}; HeightSet(0)={b, c, g, f}; HeightSet(1)={a, e}; HeightSet(2)={d}.

Theorem 4. *The prefix represented by an orphan node doesn't overlap any other prefix.*[1]

According to Theorem 4, any prefix in the orphan set doesn't overlap other prefixes. So we can make prefix balance in banks with the orphan set.

To prove that orphan set is effective, we examine one routing tables [9]. And section 6 will show our experiment results.

4.2 Increasing Banks

In Fig. 2 the PRLF has five banks each of which stores a corresponding height set. To improve parallel lookup performance, we can increase banks and store a height set with several banks. As bank number increases, prefixes in each bank will decrease and lookup speed in each bank will be faster. Lookup performance of the PRLF will also increase correspondingly.

As bank number increases, we can also balance prefixes between banks with the orphan set. As shown in the latter experiments, this method can reach a perfect result. Prefix difference between banks can be kept in a little range.

As bank number is equal to prefix number in a routing table, a bank store only one prefix. In this case, the PRLF has a similar structure with TCAM. However, in contrast to TCAM, the advantage of our PRLF is that we can flexibly adjust bank number to satisfy different lookup speed, cost, consumption and density.

[1] Due to limited pages, we will not give the proof of this theorem in this paper. If you need them, you can mail to us or visit our homepage.

4.3 Scalability

Although the PRLF is proposed based on IPv4, it can scale to IPv6 easily. However, the PRLF need make some adjustments in the following aspects.

1. Bank number. In IPv6 routing tables, prefixes will be more than IPv4. The number of height sets will increase. Therefore, banks holding height sets should also increase.
2. NMT. Since IPv6 address is 128 bits, the storage for one entry should be enlarged. In a router, if port number doesn't increase, the number of different forwarding info in a routing table will not be changed. Therefore, even for IPv6, 64 entries are enough for the NMT.

5 Routing Update

In the PRLF, a routing update can be divided into two parts: update for the NMT and update for bank prefixes. The update for the NMT mainly includes adding or deleting forwarding info. Note that multi routes may be associated with the same forwarding info. We need a reference count to remember the number of multi routes. Only if the count is 0, we can delete this forwarding info from the NMT.

Prefixes in banks are constrained by their node's height. In a bank, if a prefix is added or deleted, the height of multi prefixes may be changed. In other banks, some prefixes should be added or deleted correspondingly.

6 Performance Analysis

6.1 Balancing Banks

We choose one real routing tables - mae-east [9]. We partition them into five height sets and an orphan set. Then with the orphan set, we balance prefixes in different banks. Two bank number are chosen, B=5 and B=10. As B=5, a bank holds a height set. As B=10, we determines the number of banks holding a height set with prefix number in different height sets. In this routing table, HeightSet(0) is stored with 5 banks, HeightSet(1) with 2 banks, other height sets with 1 bank respectively.

In Fig.3, black bars presents prefix number in non-balanced banks; gray bars denotes the number in balanced banks. As B=5, prefix difference between non-balanced banks are very large. Maximum bank contains tens of thousands prefixes and minimum bank only has less than 5 prefixes. After balancing, difference is reduced greatly. Prefix number in bank 1 to bank 4 are very close. However, prefixes between bank 0 and other four banks have still rather difference. The reason for that is prefixes in HeightSet(0) take more than 20% of all prefixes. Therefore, even though balanced, the difference still exists. For B=10, before balanced, prefix number in multi banks for the same height set is very close.

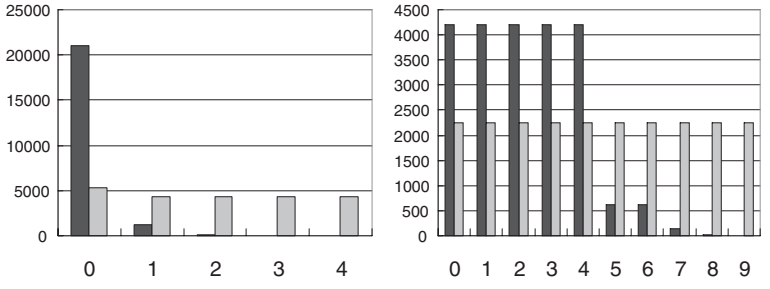


Fig. 3. Mae-east routing table

However, it has a great gap between banks for different height sets. After balanced, the gap is entirely eliminated so that maximum difference between two banks is less than or equal to 1. The reason is that we use five banks (1/2 of total banks) to hold HeightSet(0). Although prefixes in HeightSet(0) are quite a few, it doesn't reach 1/2 of total prefixes yet. Hence, the orphan set can entirely eliminate prefix difference between banks.

6.2 Lookup Performance

With the PRLF, we can increase lookup speed of algorithms. In this experiment, we test lookup performance of parallel binary search. Our hardware platform is a PC with PIII 933M CPU and 256M SDRAM. Five real routing tables [9] are adopted.

To simulate a parallel lookup, we perform a lookup in each bank independently. Then a slowest bank is chosen and its lookup time is regarded as parallel lookup time.

For destination addresses, we generate 500 IPv4 addresses randomly. For each address, we will perform 1000 lookups and get the average lookup time. Tab.1 shows total lookup time for 500 addresses in different routing tables. Note that time unit is ms.

Without the PRLF, a lookup is "longest prefix match". Binary search cannot be directly applied. Hence, we adopt binary search on intervals [3]. From Tab. 1,

Table 1. Lookup time for 500 addresses

Routing table	Mae-east	Mae-west	Aads	Paix	Pb
Prefix number	22443	33953	30716	16880	42348
Binary search on intervals	8.231	8.562	8.395	8.162	9.454
Parallel binary search (B=5)	7.321	7.341	7.330	7.261	7.921
Parallel binary search (B=10)	6.790	6.800	6.795	6.279	7.331
Parallel binary search (B=500)	3.710	4.220	3.766	3.705	4.456

we can see the PRLF can improve lookup performance. And as B (bank number) increases, lookup speed becomes faster. For B=500, comparing parallel binary search with binary search on intervals, lookup performance is increased by more than 100%.

As known, binary search isn't a very fast algorithm although it is simple. If more efficient algorithms are adopted (such as hash lookup, tries and multiway search), we will get much better lookup performance. However, the discussion about these algorithms goes beyond this paper.

7 Conclusion

"Longest prefix match" makes routing lookup difficult. In this paper, we proposed a common framework - PRLF. It can be suitable for most lookup algorithms. For binary search, PRLF can reach $log_2(2N/B)$ lookup complexity.

TCAM is a popular device to implement fast lookups. As bank number is big enough, the PRLF can have a similar structure and lookup performance with TCAM. However, in contrast to TCAM, the PRLF is more flexible and can achieve different lookup performance, cost, and power consumption by adjusting bank number.

For future work, we plan to do two things: first, implementing more efficient algorithms than binary search; second, implementing the PRLF with hardware.

References

[1] Vince, et al, Classless Inter-Domain Routing (CIDR): an address assignment and aggregation strategy, RFC1519, 1993. 616

[2] R. Hinden andS. Deering, IP Version 6 Addressing Architecture, RFC 2373, 1998. 616

[3] B. Lampson, V. Srinivasan, and G. Varghese. IP Lookups Using Multiway and Multicolumn Search. IEEE/ACM Transactions on Networking, 1999, 7(3):324-334. 616, 621, 624

[4] M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, Scalable High Speed IP Routing Lookups, in: Proceedings of ACM SIGCOMM'97, Cannes, France, 1997. 617

[5] V. Srinivasan and G. Varghese, Fast IP Lookups Using Controlled Prefix Expansion, IEEE/ACM Transactions on Computer Systems, 1999,17(1): 1-40. 617

[6] M. J. Akhbarizadeh and M. NouraniAn, An IP Packet Forwarding Technique Based on Partitioned Lookup Table, in: Proceedings of IEEE ICC '02. 617

[7] I. Ioannidis and A. Grama, Mikhail Atallah Adaptive Data Structures for IP Lookups, in: Proceeding of IEEE INFOCOMM'03.

[8] S. Dharmapurikar,P. Krishnamurthy,et al. Longest Prefix Matching using Bloom Filters, in:Proceedings of ACM SIGCOMM'03. 616

[9] Michigan University, Merit Network, Internet performance and analysis (IPMA) Project. Available WWW: `http://www.merit.edu/~ipma` 619, 621, 622, 623, 624

[10] G. Narlikar, A. Basu and F.ZaneFast, CoolCAMs: Power-Efficient TCAMs for Forwarding Engines, in: Proceedings of IEEE INFOCOMM'03. 617

Admission Control and Resource Allocation with Improved Effective Bandwidth/Buffer Calculation Method

Yongjin Kim

University of Southern California
Los Angeles, CA, USA
yongjkim@usc.edu

Abstract. Connection Admission Control(CAC) and network resource allocation are important problems in the design and control of high-speed communication networks. The problems become very complex when combined with Quality of Service(QoS) guarantee for traffic with different characteristics. CAC and network resource allocation are real-time traffic control procedures. Processing load should be minimized to reduce delay. At the same time, network resources should be utilized efficiently to accommodate more users. However, reducing processing load and obtaining high network resource efficiency has been considered to be contradictory matter. In addition, a research on CAC and resource allocation scheme, which considers multiple QoS criteria – loss and delay – simultaneously has not been adequately done. We propose an improved effective bandwidth/buffer calculation method based on a new channel/buffer separation analysis scheme. We show that our method based on effective bandwidth and effective buffer can achieve high network resource efficiency with reduced processing load. Moreover, we show that our scheme allows for simultaneous consideration of multiple QoS criteria, loss and delay.

1 Introduction

Guarantee of QoS in high-speed communication networks is one of the most challenging issues. CAC and network resource allocation are key mechanisms to provide QoS guarantee. CAC determines "whether" a new connection can be accepted or not based on existing connections and network resources [9],[14]. In other words, CAC determines whether sufficient bandwidth is available to satisfy required QoS at the connection setup time. On the other hand, the network resource allocation scheme decides "how" to accept incoming connection requests [1,6,8,10,15]. For statistical QoS guarantee, it is necessary to know how much additional bandwidth needs to be reserved for a new connection. However, there exists complex problems to provide such functions. In high-speed communication network architectures, several classes of traffic streams with widely varying traffic characteristics are statistically multiplexed and share common transmission resources. Because of the differences in the statistical behavior of connections, the problems of CAC and network resource allocation create difficulties which

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 626–636, 2004.

are very different from the ones present in traditional circuit-switched networks. Because all connections are statistically multiplexed at the physical layer and the bit rate of connections is varied, a challenging problem is to characterize, as a function of the desired QoS, the network resource requirements. Another major problem is to provide these traffic control functions in real-time, upon the arrival of a new connection. The corresponding procedures must be computationally simple enough so their overall complexity is consistent with real-time requirements. One more thing that has not been adequately studied so far in CAC and resource allocation is considering multiple QoS criteria simultaneously. With the rapid growth of network users, current network traffic has diverse QoS demands. Most traffic has multiple QoS demands such as delay and loss level. We consider two QoS criteria, loss and delay. Even though packet delay in the network can be expressed as the sum of the processing, queuing, transmission, and propagation delays, we will focus on queuing delay in this paper.

This paper is organized as follows. In section 2, we briefly explain about previous works on CAC and resource allocation. In section 3, we propose a new system analysis scheme based on virtual channel/buffer and present a loss probability calculation method for a multiplexing system. In section 4, we propose new effective bandwidth/buffer calculation method for a multiplexing system that allows for efficient CAC, network resource allocation, simultation consideration of multiple QoS requirement. In In section 5, we perform a numerical study to verify the adequacy of our proposal. Finally, in section 6, we conclude the paper.

2 Previous Works

To date, many CAC and network resource allocation schemes have been proposed [1-8],[10-13],[15,16]. Each of the many methods proposed so far has its own strengths and weaknesses. Here, they can be classified from several different standpoints to provide insights for choosing the most fitting method.

The first basis is whether it is a deterministic or a stochastic scheme. Deterministic admission control uses a source's peak rate. If the sum of a requestor's peak rate is less than the channel capacity, it accepts the request. However, if the sum of rates exceeds the channel capacity, it rejects the resource request. Even though the scheme is pretty simple, it underutilizes the network resources because a source does not send packets in peak rate constantly. The average rate may be far less than the peak rate. On the other hand, the stochastic scheme is based on statistical behavior of network traffic. Effective bandwidth concept is often used for a stochastic CAC and resource allocation. Effective bandwidth satisfies the given QoS such as loss probability with minimum amount of bandwidth. Stochastic allocation makes economic sense when dealing with bursty sources, but it is difficult to carry out effectively because it is a CPU-intensive job requiring high processing load.

The second basis for classification is whether the buffering effect is taken into account in evaluating performance level or not. If the buffering effect is

not considered, CAC and resource allocation function could simply be made by allocating required amount of resources to the connections. A new connection will be accepted if the sum of the required rate of all existing connections and required rate of a new connection is less than the capacity of the link. However, under that scheme, network resources cannot be utilized efficiently because they ignore the buffering effect. On the other hand, if the buffering effect is considered, we need to model the queuing process. A queue will build up in the buffer as long as the total arrival rate exceeds the link capacity. When the queue length reaches buffer capacity, packet loss occurs. A strength of this method is that it can achieve high resource efficiency. However, it requires a considerable amount of processing power.

As we can see in the above classfication, there exists common tension between achieving resource efficiency and reducing processing load. In addition, previous work does not consider simultaneous satisfaction of multiple QoS criteria, loss and delay. Packet loss probability can be reduced by increasing delay. If we don't consider loss probability and delay simultaneously, multimedia traffic could suffer long delay satisfying only loss probability level, which is meaningless for most real-time application. In addition, existing scheme does not consider security aspect in CAC and resource allocation. In this paper, we use a stochastic scheme and consider the buffering effect to increase resource efficiency. Furthermore, to address the high processing load problem, take security into consideration and achieve simultaneous satisfaction of multiple QoS criteria, we propose a new system analysis scheme based on a virtual buffer and channel system[6]. Based on this new analysis scheme, we propose a new effective bandwidth/buffer calculation method which results in efficient CAC and bandwidth/buffer allocation with light processing load satisfying multiple QoS criteria simultaneously.

3 Loss Probability Calculation for a Multiplexing System

We propose a new channel/buffer separation analysis scheme based on virtual channel/buffer system. It begins with the characterization of the ith connection of class i by stationary random arrival process a_i that represents instantaneous loads. We assume the statistical independence of traffic sources. We can split the traffic process $a_i(t)$ into two separate processes of $u_i(t)$ and $v_i(t)$ representing respectively, the bandwidth requirement and buffer requirement at time t. Its relation can be expressed as the following equation.

$$a_i(t) = u_i(t) + v_i'(t) \tag{1}$$

Throughout this paper, u_i and v_i are defined as two random variables that represent, respectively, the instantaneous bandwidth requirement and buffer requirement of source i at a random time t. In addition, we assume u_i and v_i, $1 \leq i \leq I$, are mutually independent.

By obtaining buffer requirement and bandwidth requirement, virtual channel/buffer system can be obtained as in Fig.1. This separation makes it possible to separate a system into two independent systems, virtual channel system and

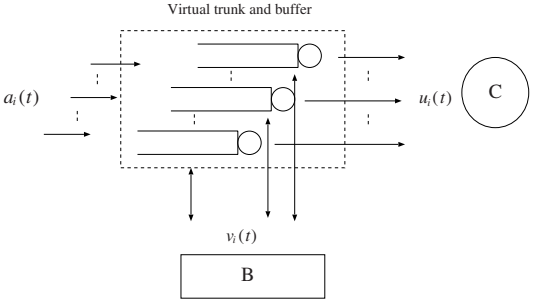


Fig. 1. multiplexing system

virtual buffer system. Consequently, complicated calculation of loss probability of one buffered model can be converted into two simple independent models. A loss probability event occurs when all of the two resources, virtual channel and virtual buffer, are exhausted simultaneously. Consequently, we can calculate loss probability by the product of two independent events, the virtual buffer overflow probability event and the virtual channel overflow probability event as follows.

$$P_{loss} \approx \Pr\{\sum_{i=1}^{I} K_i u_i > C\} \Pr\{\sum_{i=1}^{I} K_i v_i > B\} \tag{2}$$

where K_i, is number of each class i. Using Chernoff's bound,

$$P_{loss} \leq e^{\tilde{F}_{\mathbf{K}}(s)} e^{\hat{F}_{\mathbf{K}}(s)} \tag{3}$$

where,

$$\tilde{F}_{\mathbf{K}}(s) = \sum_{i=1}^{I} K_i \log \tilde{M}_i(s) - sC \tag{4}$$

$$\hat{F}_{\mathbf{K}}(s) = \sum_{i=1}^{I} K_i \log \hat{M}_i(s) - sB \tag{5}$$

where, $\tilde{M}_i(s), \hat{M}_i(s)$ is moment generating function of u_i and v_i, respectively.

Based on large deviation approximation, we can approximate loss probabilty as follows.

$$P_{loss} \approx \inf_{s \geq 0}\{e^{\tilde{F}_{\mathbf{K}}(s)}\} \inf_{s \geq 0}\{e^{\hat{F}_{\mathbf{K}}(s)}\} \tag{6}$$

Loss probability calculation becomes simple(low processing load) and accurate(high resource efficiency) using our scheme, which make it possible to be used for traffic control.

4 Effective Bandwidth and Effective Buffer for a Multiplexing System

In this section, we propose effective bandwidth and effective buffer calculation scheme based on the system analysis method proposed in the previous section. So far, the proposed effective bandwidth concept has been considered to be the best CAC and resource allocation scheme. Under the effective bandwidth approach, the amount of bandwidth required by a connection is estimated individually by the answer to the following question – What should the service rate for a link with buffer capacity be to achieve a declared QoS if a connection is input to this link? Here, we propose an effective bandwidth and effective buffer calculation scheme, which answers the following question. What should the service rate and buffer capacity be for a link to achieve a declared QoS if the connection is input to this link? By answering this question, the amount of bandwidth and buffer required to satisfy multiple QoS criteria, loss and delay, by a connection is estimated individually. The constraint on loss probability, L, will be satisfied if,

$$P_{loss} \approx \inf_{s\geq 0}\{e^{\tilde{F}\mathbf{K}^{(s)}}\} \inf_{s\geq 0}\{e^{\hat{F}\mathbf{K}^{(s)}}\} \leq L \tag{7}$$

In other words, it can be expressed as follows.

$$\inf_{s\geq 0}\{\tilde{F}_{\mathbf{K}}(s)\} \inf_{s\geq 0}\{\hat{F}_{\mathbf{K}}(s)\} \leq \log L \tag{8}$$

Here, we can consider various combinations. We select constraints as follows.

$$\inf_{s\geq 0}\{\tilde{F}_{\mathbf{K}}(s)\} = L_c \tag{9}$$

$$\inf_{s\geq 0}\{\hat{F}_{\mathbf{K}}(s)\} = L_b \tag{10}$$

where,

$$L_c + L_b = \log L \tag{11}$$

For lossless multiplexing, u_i and v_i that satisfies relation Eq.(1) should be selected for the moment generating function of $\log \tilde{M}_i(s)$ and $\log \hat{M}_i(s)$ respectively. For statistical multiplexing, smaller values than u_i and v_i should be selected for statistical multiplexing gain. On the other hand, when delay criterion is added, the ratio of u_i to v_i should be determined depending on delay constraint. That is, larger u_i and smaller v_i should be selected for stringent delay constraint.

Now, we use the value (u_i, v_i) and (L_c, L_b) to define effective bandwidth and effective buffer, respectively. That is, effective bandwidth is defined as the amount of bandwidth that satisfies the constraint of L_c and effective buffer is defined as the amount of buffer that satisfies the constraint of L_b. Effective bandwidth and effective buffer are in a trade-off relation as shown in Eq.(1) and Eq.(11). This trade-off relation between effective bandwidth and effective buffer makes it possible to consider multiple QoS criteria simultaneously for CAC and resource allocation.

With the constraint of (L_c, L_b), Eq.(8) can be expressed as the following.

$$\inf_{s\geq 0}\{\sum_{i=1}^{I} K_i \log \tilde{M}_i(s) - sC\}+$$

$$\inf_{s\geq 0}\{\sum_{i=1}^{I} K_i \log \hat{M}_i(s) - sB\} \leq L_c + L_b \tag{12}$$

Where, $\tilde{M}_i(s)$ is the moment generating function of u_i and $\hat{M}_i(s)$ is the moment generating function of v_i. The tangent plane at a point $\dot{\mathbf{K}}$ on the boundary of the region is,

$$\sum_{i=1}^{I} K_i \log \tilde{M}_i(\dot{s}) - \dot{s}C+$$

$$\sum_{i=1}^{I} K_i \log \hat{M}_i(\ddot{s}) - \ddot{s}B = L_c + L_b \tag{13}$$

where $\dot{s}$ and $\ddot{s}$ attains the infimum in (15) with $\mathbf{K}$ replaced by $\dot{\mathbf{K}}$. One of the choice for the above condition is,

$$\sum_{i=1}^{I} K_i \log \tilde{M}_i(\dot{s}) - \dot{s}C = L_c \tag{14}$$

$$\sum_{i=1}^{I} K_i \log \hat{M}_i(\ddot{s}) - \ddot{s}B = L_b \tag{15}$$

It can also be expressed as the following.

$$\sum_{i=1}^{I} K_i e_i = C \tag{16}$$

$$\sum_{i=1}^{I} K_i b_i = B \tag{17}$$

where e_i is effective bandwidth and b_i is effective buffer.

$$e_i = \frac{\log \tilde{M}_i(\dot{s})}{\dot{s} + L_c/C} \tag{18}$$

$$b_i = \frac{\log \hat{M}_i(\ddot{s})}{\ddot{s} + L_b/B} \tag{19}$$

Consequently, when, loss and delay QoS criteria are given, the following two conditions should be satisfied.

$$\sum_{i=1}^{I} K_i e_i \leq C \text{ and } \sum_{i=1}^{I} K_i b_i \leq B \tag{20}$$

In above equation, loss probability can be guaranteed by selecting (L_c, L_b) that satisfy the relation of Eq.(11) and and consequent effective bandwidth e_i and effective buffer b_i. In addition, delay constraint can be guaranteed by selecting proper combination of (u_i, v_i), (L_c, L_b) and consequent e_i and b_i to satisfy given delay constraint. The trade-off relationship between (u_i, v_i), (L_c, L_b) and consequent e_i and b_i makes it possible to consider multiple QoS criteria, loss and delay. More specifically, satisfying Eq.(20) guarantees that the following loss and workload(virtual delay in the buffer) requirement of connection i will be satisfied.

$$P_{loss}^{i} \leq L \tag{21}$$

$$W_i \leq b_i \tag{22}$$

On the other hand, when there is only loss criterion, we can induce the following equation, By summing up Eq.(16), and Eq.(17), we can approximate,

$$\sum_{i=1}^{I} K_i e_i + \sum_{i=1}^{I} K_i b_i \approx B + C \tag{23}$$

Accordingly, by satisfying following condition, we can meet loss QoS requirement.

$$\sum_{i=1}^{I} K_i (e_i + b_i) \leq B + C \tag{24}$$

5 Numerical Study

We present numerical results in this section to verify the adequacy of the proposed method by comparing analysis results and simuation results and show how our method can be used for system design. We consider a multiplexing system with single class and multiple connections. Each connection is regulated by a leaky bucket regulator. We prepare a node that has a total amount of bandwidth C and buffer space B, shared by the sources.

To exploit the bursty nature of traffic sources regulated by leaky buckets, we assume that the regulated sources are extremal on-off periodic processes after going through leaky bucket regulators. On-off periodic source represents uncompressed audio/video traffic and an extremal on-off periodic source is one which, when active, generates data at the peak rate P until the depletion of its token bucket. It then stays inactive until the token bucket is completely filled again. Such processes account for the worst-case statistical behavior in the sense that they maximize the average loss rate. Note that our scheme can be generally applied to other traffic models such as on-off model Pareto distribution which has heavy-tailed distribution. T_{on} is the maximal time that a regulated source can

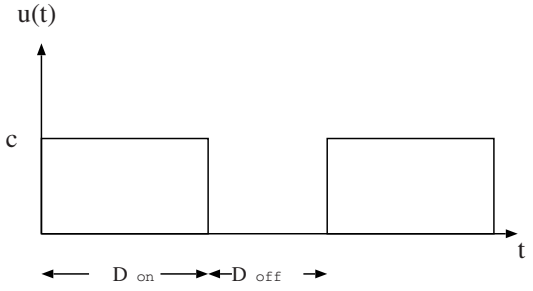


Fig. 2. Process for virtual channel

send traffic at the peak rate P. This happens when the leaky bucket is full with σ amount of tokens. T_{off} is the time to fill an empty buffer with the token rate ρ. This process is appropriate for our analsis because it represents congestion status well and one of the ultimate goals of CAC and resource allocation is to prevent congestion. Assume that traffic sources are grouped into I classes and for $1 \leq i \leq I$, there are K_i sources of class i. Each source of class i, is an extremal on-off periodic process with leaky bucket parameters (ρ_i, σ_i, P_i) where ρ_i is token generation rate, σ_i is token bucket size and P_i is peak rate. As explained in the above section, loss probability can be estimated using the Chernoff bound. The moment generating function for the process of virtual channel(Fig.2) can be obtained as follows.

$$\tilde{M}_i(s) = 1 - \omega_i + \omega_i e^{sc_i} \tag{25}$$

In this equation, ω_i is the probability for each state, which is ρ_i/c_i. We set v_i as on-off process requiring maximum buffer size, q_i, for conservativeness and computational simplicity. Then, the moment generating function for the process of virtual buffer(Fig.3) is expressed in the same way.

$$\hat{M}_i(s) = 1 - \omega_i + \omega_i e^{sq_i} \tag{26}$$

We set 45Mbps of C and 1000 cells of B. Q, 250 Cells, is the amount of data generated during period T_{on}. In the above equation, c_i and q_i is selected among combinations satisfying Eq.(1). D_{on} is the time cycle that the buffer is not empty.

Then, we can obtain the number of admissible sources(K) for a given QoS criterion(loss probability) as shown in table 1. That is, to satisfy each loss probability, maximum number of K sources can be admitted. . The significance of above result is that our analysis scheme is fairly close to simulation results(less than 6 percent difference). In addition, it shows conservativeness which is desirable to give a definit QoS guarantee.

Effective bandwidth and effective buffer can be obtained easily using calculation method[Eq.(18), Eq.(19)] outlined in section V. Effective bandwidth and effective buffer make it possible to consider multiple QoS criteria(loss and delay) simultaneously. In Fig.4., we show different combination of effective bandwidth

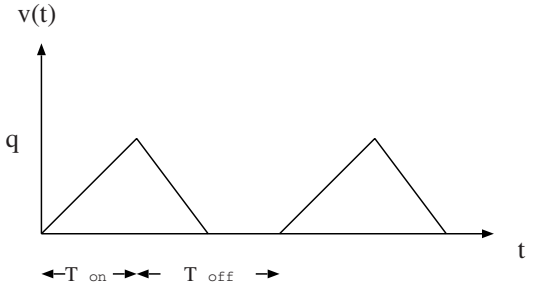


Fig. 3. Process for virtual buffer

Table 1. Comparison with simulation

Loss Prob.	K(By simulation)	K(By analysis)
2.2×10^{-8}	150	143
1.7×10^{-7}	160	151
1.3×10^{-6}	170	164
6×10^{-6}	180	171

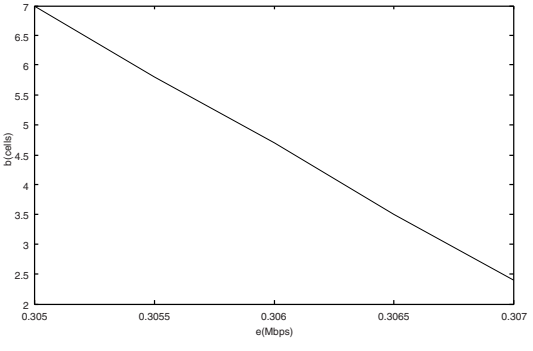


Fig. 4. Effective bandwidth(e) vs. effective buffer(b)

and effective buffer for a single source satisfying the same loss probability constraint of 10^{-7}. Now, for stringent delay constraint, larger effective bandwidth is needed. In contrast, smaller effective bandwidth is required for the traffic with loose delay constraint. This trade-off allocation can be utilized using our scheme to satisfy diverse QoS demands of current networks.

6 Conclusion

In this paper, we proposed a new system analysis scheme based on virtual channel/buffer. Comparison between analysis result and simulation result showed that our scheme accurately estimates simulation network. We made the following major contributions in this paper. (1)We proposed new system analysis scheme based on virtual channel/buffer. We calculated loss probability as the product of two independent probability events – virtual channel overflow probability event and virtual buffer overflow probability event.Our system analysis method allows efficient and computationally light traffic control. (2) Based on our analysis scheme, we calculated effective bandwidth and effective buffer, which gives useful information for CAC and network resources (bandwidth and buffer) allocation. Our calculation allows for high resource efficiency with light processing load. (3)Our scheme has merit that allows for simultaneous consideration of multiple QoS criteria – loss and delay.

References

[1] K. Belkacem and S. Mischa, "Bandwidth Allocation Strategies in Wide-band Integrated Networks", IEEE J. Select. Areas Commun., Vol.SAC-4, No.9, pp.869-878, Sept.1986.
[2] A. W. Berger and W. Whitt, "Effective Bandwidth with Priorities", IEEE/ACM Trans. on Networking., Vol.6, No 4, August 1998, pp. 447-460.
[3] C. S. Chang and J. A. Thomas, "Effective Bandwidth in High-speed Digital Networks", IEEE J. Select. Areas Commun., Vol.13, No.9, pp.1091-1100, Sept. 1995.
[4] G. L. Choudhury, D. M. Lucantoni and W Whitt, "Squeezing the Most Out of ATM", IEEE Trans. Commun., Vol.44, pp.203-217, 1996.
[5] A. I. Elwalid. and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks", IEEE/ACM Trans. on Networking., Vol.1,No 3, June 1993, pp. 329-343.
[6] A. I. Elwalid, D. Mitra and R. H. Wentworth, "A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node", IEEE Trans. on Selected Areas in Communications., August 1995, pp. 1115-1127.
[7] R. J. Gibbens and P. J. Hung, "Effective Bandwidth for the Multi-type UAS Channel", Queueing Syst., Vol.9, pp.17-28, 1991.
[8] R. Guerin, H. Ahmadi and M. Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High-speed Networks", IEEE J. Select. Areas Commun., Vol.9, No.7, pp.968-981, Sept., 1991.
[9] G.P. Harry and M.E. Khaled, "Call Admission Control Schemes:A Review", IEEE Communication Magazine., November 1996, pp.82-91.
[10] J. Y. Hui, "Resource Allocation for Broadband Networks", IEEE J. Select. Areas Commun., Vol.6, No.9, pp.1598-1698, Dec.1988.
[11] F. P. Kelly, "Effective Bandwidths at Multi-type queues", Queueing Syst, Vol.9, pp.5-15, 1991.
[12] F. P. Kelly, "Note on Effective Bandwidths", Stochastic Networks., Oxford, U.K. Clarendon, pp.141-168, 1996.
[13] G. Kesidis, J. Walrand and C. S. Chang, "Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources", IEEE/ACM Trans. Networking, Vol. 1, No 4, pp.424-428, 1993

[14] K. Shiomoto, N. Tamanaka, and T. Takahashi, "Overivew of Measurement-Based Connection Admission Control Methods in ATM Networks", IEEE communications surveys., First quarter 1999.
[15] G. Veciana, G. Kesidis and J. Walrand, "Resource Management in Wide-area ATM Networks Using Effective Bandwidth", IEEE J. Select. Areas Commun., Vol.13, No.9, pp.1081-1090, Sept.1995.
[16] W Whitt, "Tail Probabilities with Statistical Multiplexing and Effective Bandwidths in Multi-class Queue", Telecommun. Syst., Vol.2, pp.71-107, 1993.

Distributed Scheduling Policies for Networks of Switches with a Configuration Overhead

Claus Bauer

Dolby Laboratories, San Francisco, CA, 94103, USA
cb@dolby.com

Abstract. Optical switching cores are fast gaining importance for deployment in internet switches/routers. The reconfiguration of these switches requires a large timely switching overhead. The design of efficient algorithms that take into account the configuration overhead has been widely researched. However, all previous research solely focuses on switching algorithms that optimize the performance features of a single switch with configuration overhead. This paper is the first which designs classes of switching policies for a network of switches with a configuration overhead that guarantee the stability of the network. We also show that networks of switches with configuration overhead are stable if different classes of policies are deployed at different switches simultaneously.

1 Introduction

The introduction of new optical transmission technologies such as Dense Wavelength Division Multiplexing (DWDM) have dramatically increased the transmission capacity of optical fibers. As a consequence, there is a need for switches and routers that work at or above the speed of the high-speed optical links connecting them.

Today, most high-performance routers/switches use an electronic core that deploys a Virtual Output Queueing Scheme and a crossbar switching core. It is not expected that electronic switches will be able to meet the performance requirements imposed by future optical transmission capacities. Thus, optical switching cores have increasingly gained importance. At this time, it is neither possible to buffer packets in the optical domain nor to evaluate the packet header in the optical domain. Therefore, most researchers ([7, 12])) propose *hybrid* electronic-optical architectures: Packets that arrive on optical input links are converted into an electric signal. The header evaluation and the eventual buffering are performed in the electronic domain. In order to forward the packet through the optical switching core, the packet is reconverted into the optical domain and sent through the switch. If the switch uses output buffers, the packet is again reconverted into electronics at the output, buffered electronically, and reconverted into optics when it leaves the switch. Obviously, it is desirable to find ways to perform the header evaluation and buffering in the optical domain in order to save the numerous electronic-optical and optical-electronic conversions.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 637–647, 2004.

The MEMS technology is a favorite candidate for an optical switching core ([6, 13]). Compared to electronic switches, that reconfigure themselves in few nanoseconds or less, the reconfiguration time of a MEMS based switch is typically quoted as being between 1 and 10 ms. Depending on the link speed, these switches take up to 20,000 cell times to reconfigure. This long reconfiguration time introduces large packet delays and requires scheduling algorithms that take this configuration overhead into account and optimize the delay and loss characteristics of the resulting schedule.

Recently, various researchers have proposed different scheduling algorithms for switches for a configuration overhead. Two approaches can be distinguished: In the Timeslot Assignment based approach ([9, 10, 15]), incoming traffic is accumulated during a predefined accumulation period. During each cycle, the arriving traffic is buffered in an $N \times N$ traffic matrix. Using different algorithms, the traffic matrix is decomposed into permutation matrices that determine the configurations of the switch. The Single Scheduling approach ([10, 11]) can be considered as a slow version of scheduling algorithms for switches without configuration overhead. Similar to packet-based based scheduling ([14]), a schedule is generated and maintained for several timeslots. Most approaches require the switching core to work at a speedup $S > 1$ compared to the linkspeed.

Previous research on scheduling algorithms for switches with configuration overhead has investigated scheduling algorithms that optimize the performance features stability and delay for a single switch. So far, no research on scheduling algorithms that stabilize networks of switches with a configuration overhead has been performed. For switches without configuration overhead, it has been shown in [3] for the example of a maximum weight matching algorithm ([7]) that scheduling algorithms that guarantee the stability of individual switches do not necessarily guarantee the stability of networks of switches. Following the argument in [3], it can be shown that the Single Scheduling $LQF + holding$ algorithm proposed in [10] can lead to instabilities in a network of switches with configuration overhead.

For switches without configuration overhead, in [2] and [3], local scheduling algorithms that stabilize networks of switches, but require signaling traffic between adjacent switches have been proposed. Only recently, in [1] a scheduling algorithm has been proposed that does not require signaling traffic, but stabilizes a network of switches. This algorithm requires non-local information to be transported in the packet header.

This paper is the first effort to investigate local scheduling algorithms for networks of input-queued switches with a configuration overhead that stabilize the entire network.

The rest of the paper is organized as follows. In the next section, we develop a model for a network of switches. In section 3, we define local scheduling policies and prove the stability of networks that deploy any of those policies at all switches. In section 4, we prove that networks of switches that deploy different classes of these policies simultaneously at different switches of the network are stable as well. We conclude in section 5.

2 Terminology and Model

2.1 Model of a Network of Queues

In this section, we follow an approach in [1] to describe our model of a queueing system. We assume a system of J physical queues $\tilde{q}^j$, $1 \leq j \leq J$ of infinite capacity. Each physical queue consists of one or more logical queues, where each logical queue corresponds to a certain class of customers within the physical queue. Whenever a packet moves from one physical queue to another, it changes class and therefore also changes logical queue. We denote a logical queue by q^k, $1 \leq k \leq K$, where $K \geq J$. A packet enters the network via an edge switch, travels through a number of switches and leaves the network via another edge switch. We define a function $L(k) = j$ that defines the physical queue $\tilde{q}^j$ at which packets belonging to the logical queue q^k are buffered. The inverse function $L^{-1}(j)$ returns the logical queues q^k that belong to the physical queue $\tilde{q}^j$.

Throughout this paper, the time t is described via a discrete, slotted time model. Packets are supposed to be of fixed size and a timeslot is the time needed by a packet to arrive completely at an input link.

We define a row vector $X_n = (x_n^1, ..., x_n^K)$, where the k-th vector x_n^k represents the number of packets buffered in the logical queue q^k in the n-th timeslot. We define $E_n = (e_n^1, ..., e_n^K)$, where e_n^k equals the number of arrivals at the logical queue q^k in the n-th timeslot. Analogously, we define $D_n = (d_n^1, ..., d_n^K)$, where d_n^k expresses the number of departed packets from q^k in the n-th timeslot. Thus, we can describe the dynamics of the system as follows:

$$X_{n+1} = X_n + E_n - D_n. \tag{1}$$

Packets that arrive at a logical queue q^k either arrive from outside the system or are forwarded from a queue within the system. Thus, we can write:

$$E_n = A_n + T_n,$$

where $A_n = (a_n^1, ..., a_n^K)$ denotes the arrivals from outside the system and $T_n = (t_n^1, ..., t_n^K)$ denotes the arrivals from inside the system.

We define a routing matrix $R = [r_{i,j}]$, $1 \leq i, j \leq K$, where $r_{i,j}$ is the fraction of customers that depart from the logical queue q^i and are destined for the logical queue q^j. Assuming a deterministic routing policy, there holds, $r_{i,j} \in \{0, 1\}$, $\sum_{1 \leq i < K} r_{i,j} < 1$, $\sum_{1 \leq j \leq K} r_{i,j} \leq 1$. We set $r_{i,j} \neq 0$, if q^j follows q^i along the route. Noting that $T_n = D_n R$ and writing I for the identity diagonal matrix, we find

$$X_{n+1} = X_n + A_n - D_n(I - R). \tag{2}$$

We assume that the external arrival processes are stationary and satisfy the Strong Law of Large Numbers. Thus,

$$\lim_{n \to \infty} \frac{\sum_{i=1}^{n} A_i}{n} = \Lambda \qquad \text{w.p.1}, \tag{3}$$

where $E[A_n] = \Lambda = (\lambda^1, .., \lambda_K), \forall n \geq 1$[1]. Noting that $(I-R)^{-1} = I+R+R^2+...$, we find that the average workload $W = (w^1, ..., w^K)$ at the logical queues q^k is given by $W = \Lambda(I - R)^{-1}$.

Finally, we give a stability criteria for a network of queues as proposed in [2].

Definition 1: A system of queues is rate stable if

$$\lim_{n \longrightarrow \infty} \frac{X_n}{n} = \lim_{n \longrightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} (E_i - D_i) = 0 \qquad \text{w.p.1.}$$

A necessary condition for the rate stability of a system of queues is that the average number of packets that arrive at any physical queue $\tilde{q}^j$ during a timeslot is less than 1. We formalize this criteria as follows:

Definition 2: For a vector $Z \in \mathbb{R}^K, Z = (z^1, .., z^K)$, and the function $L^{-1}(k)$ as defined in this subsection, we set:

$$||Z||_{maxL} = \max_{j=1,..,J} \left\{ \sum_{k \in L^{-1}(j)} z^k \right\}. \tag{4}$$

The necessary condition for rate stability can now be formalized as follows:

$$||W||_{maxL} < 1. \tag{5}$$

2.2 Model of a Network of Switches

In this section, we apply the terminology of the previous section to a network of switches. We assume that the switching core is an $N \times N$ input-queued or combined input/output-queued (IQ/CIOQ) switch that deploys a Virtual Output Queue buffer structure ([7]. A network of IQ/CIOQ switches can be conceived as a queueing system as defined in the previous section where the virtual output queues are considered as the physical queues. In this model, we neglect the output queues of the switches because instability can only occur at the Virtual Output Queues (see [1]).

We say that packets that enter the network via the input of a given switch and leave the network via the output of a given switch belong to the same flow. Packets belonging to the same flow travel through the same sequence of physical queues and are mapped to the same logical queues at each physical queue, i.e., a flow can be mapped biunivocally to a series of logical queues.

We assume that each logical queue behaves as a FIFO queue and assume a *per-flow* scheduling scheme. It has been shown in [1] how stability results for *per-flow* scheduling schemes can be used to design less complex and stable *per-virtual output queue* schemes.

[1] Throughout the paper, we abbreviate "with probability 1" by "w.p.1."

The network consists of B switches and each switch has N_b, $1 \leq b \leq B$, inputs and outputs. If the total number of flows in the system is C, we do not have more than N_b^2 physical queues and CN_b^2 logical queues at switch b. We can model the whole network of switches as a system of $\sum_{1 \leq b \leq B} CN_b^2$ logical queues. For the sake of simplicity, we suppose that $N_b = N$, $1 \leq b \leq B$ and set $K = CN^2B$. Finally, we define $Q_I(b,i)$ as the set of indexes corresponding to the logical queues at the i-th input of the b-switch. Analogously, $Q_O(b,i)$ denotes the set of indexes corresponding to the logical queues directed to the i-th output of the b-switch. We use these definitions to adapt the norm $||Z||_{maxL}$ to a network of switches:

Definition 3: Given a vector $Z \in \mathbb{R}^K$, $Z = \{z^k, k = CN^2b + CNi + Cj + l,\ 0 \leq b < B,\ 0 \leq i, j < N, 0 \leq l < C$, the norm $||Z||_{IO}$ is defined as follows:

$$||Z||_{IO} = \max_{\substack{b=1,..,B \\ i=1,..,N}} \left\{ \sum_{m \in Q_I(b,i)} |z^m|, \sum_{m \in Q_O(b,i)} |z^m| \right\}.$$

As we assume a deterministic routing policy, the necessary condition for rate stability given in (5) can be written for a network of switches as follows:

Definition 4: For a network of IQ/CIOQ switches, a traffic and routing pattern W is admissible if and only if:

$$||W||_{IO} = ||\Lambda(I - R)^{-1}|| < 1. \tag{6}$$

In the rest of this paper, we will only consider traffic and routing patterns that satisfy the condition (6). We will say that a network which is rate stable under condition (6) achieves 100% throughput.

3 Local Scheduling Policies

3.1 Weight Function

All scheduling policies introduced in this paper are matching policies. Any matching policy is defined relative to a specific weight. For the definition of the weights, we will make use of a family of real positive functions $f_k(x) : \mathbb{N} \to \mathbb{R}$, $1 \leq k \leq K$, that satisfy the following property:

$$\lim_{n \to \infty} \frac{f_k(n)}{n} = \frac{1}{w^k} \qquad \text{w.p.1.} \tag{7}$$

We define $\overline{d}^k(n) = \sum_{m \leq n} d_m^k$ as the cumulative number of services at queue q^k up to time n. We define the weights of the queues q^ks at time n by

$$\phi_n^k = n - f_k(\overline{d}^k(n)) \quad \Phi_n = (\phi_n^1, .., \phi_n^K). \tag{8}$$

In [2], an example for $f_k(n)$ is given. The cumulative function of external arrivals for the logical queue q^k is given by $\overline{a}^k(n) = \sum_{m \leq n} a_m^k$. The inverse function $[\overline{a}^k]^{-1}(p)$ maps the packet number p to the arrival slot. Setting $f_k(p) = [\overline{a}^k]^{-1}(p)$, the weight $\phi_n^k = n - [\overline{a}^k]^{-1}(p)$ denotes the time the packet has already spent in the network. At its departure time n, the age of the p-th packet is $n - [\overline{a}^k]^{-1}(\overline{d}_n^k)$.

3.2 The Fluid Methodology

The main proof of this section will make use of the fluid methodology as introduced in [4, 5]. As in [1], we consider an extension of the fluid model to a network of switches. First, we define three vectors: $X(t) = (X_{1,1}(t), ..., X_{N,N}(t))$ denotes the number of packets in the $VOQs$ at time t, $D = (D_{1,1}(t), ..., D_{N,N}(t))$ denotes the number of packet departures from the $VOQs$ until time t and $A = (A_{1,1}(t), ..., A_{N,N}(t))$ denotes the number of packet arrivals at the $VOQs$ until time t. We define $\Pi = \{\pi\}$ as the set of all possible network-wide matchings and denote a specific scheduling algorithm by $\mathcal{S}$. For all $\pi \in \Pi$, we denote by $T_\pi^{\mathcal{S}}(t)$ the cumulative amount of time that the matching π has been used up to time t by the algorithm $\mathcal{S}$. Obviously, $T_\pi^{\mathcal{S}}(0) = 0 \; \forall \pi \in \Pi$. Using (2), we obtain the fluid equations of the system as follows:

$$X(t) = X(0) + \Lambda t - D(t)(I - R), \tag{9}$$

$$D(t) = \sum_{\pi \in \Pi} \pi T_\pi^{\mathcal{S}}(t), \tag{10}$$

$$\sum_{\pi \in \Pi} T_\pi^{\mathcal{S}}(t) = t. \tag{11}$$

The first two equations model the evolution of the logical queues, whereas the third counts the total number of departures from the $VOQs$. The third equation reflects the fact that in each timeslot, each input is connected to some output.

3.3 Maximum Weight Matching Policies

In this section, we define a class of maximum weight matching policies that guarantee the stability of a network of IQ/CIOQ switches with configuration overhead. We introduce a set of functions $\mathcal{G}$ as follows:

Definition A real function $\mathcal{F}$ is said to belong to the set $\mathcal{G}$ if

a) $\dot{\mathcal{F}}(x)$ exists for all $x > 0$.
b) $\mathcal{F}$ and $\dot{\mathcal{F}}(x)$ are strictly monotonically increasing, non-negative and $\mathcal{F}(\prime) = \prime$, $\dot{\mathcal{F}}(0) = 0$.
c) $\dot{F}$ satisfies a Lipschitz condition: $\exists C_F \; s.t. \; |\dot{\mathcal{F}}(x) - \dot{\mathcal{F}}(y)| \leq C_{\mathcal{F}}|x - y|$, $\forall x, y \in \mathbb{R}$.

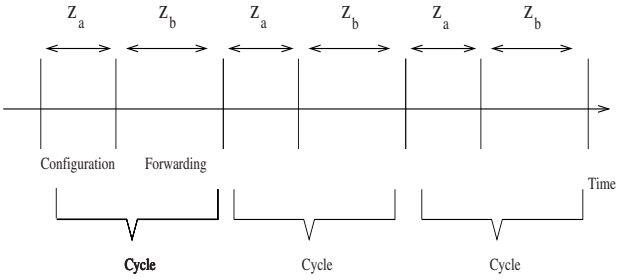


Fig. 1. Operation mode of the $MWM_c^{\mathcal{F}}$ algorithm

Using the fluid methodology, we define a function $\Phi(t)$ based on the definition of the function Φ_n in (8). We set

$$\mathcal{F}(\Phi(t)) = \sum_{k=1}^{K} \mathcal{F}(\phi^k(t)).$$

We define $\Gamma = [\gamma^{(i,j)}]$ as the diagonal matrix with $\gamma^{(k,k)} = w^k$, and let Γ^{-1} be the inverse of Γ. Further, we write the scalar product for two vectors v_1 and v_2 as $\langle v_1, v_2 \rangle = v_1 v_2^T$. Following an idea in [1], we first define a scheduling policy for input-queued switches without configuration overhead. For every function $F \in \mathcal{G}$, we define a scheduling algorithm $MWM^{\mathcal{F}}$ as follows. At each time t, the scheduling algorithm $MWM^{\mathcal{F}}$ chooses the schedule $\pi^{\mathcal{F}}$ which is defined as:

$$\pi^{\mathcal{F}}(t) = \arg\max_{\pi} \left\{ \langle \pi, \dot{\mathcal{F}}(\phi(t)) \rangle \right\}. \tag{12}$$

For fixed $\mathcal{F}$, we denote the value of the matching achieved by $MWM^{\mathcal{F}}$ as $M(t) = \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\phi(t)) \rangle$. Using this terminology, we now define a class of scheduling policies $MWM_c^{\mathcal{F}}$ for switches with configuration overhead. As shown in fig. 1, a switch operates in cycles of constant length. Each cycle consists of a configuration phase of length z_a-timeslots, during which the switch is reconfigured and no packets are forwarded, and a forwarding phase of length z_b timelslots, during which the switch configuration remains unchanged and packets are forwarded. Scheduling decisions are made according to the policy $MWM^{\mathcal{F}}$ at instants t_n, where $t_{n+1} - t_n = z_a + z_b$. The chosen schedule $\pi^{\mathcal{F}}(t_n)$ is kept constant during the interval $[t_n + z_a, t_{n+1}[$. We set

$$M_c(t) = \langle \pi^{\mathcal{F}}(t_n), \dot{\mathcal{F}}(\phi(t)) \rangle \qquad \text{for } t \in [t_n, t_{n+1}[. \tag{13}$$

Now, we can formulate the main result of this section:

Theorem 1. *For any function $\mathcal{F} \in \mathcal{G}$, a network of IQ/CIOQ switches with configuration overhead that implements a $MWM_c^{\mathcal{F}}$ – policy, in which the weight ϕ_n^k of queue q^k at time n is defined as in (8) , and which deploys a speedup $S \geq \left\lceil \frac{z_a + z_b}{z_b} \right\rceil$ achieves 100% throughput.*

3.4 Proof of Theorem 1

Before proving the theorem, we introduce the notion of a $MWM_m^{\mathcal{F}}$ scheduling policy for switches without configuration overhead. We define the $MWM_m^{\mathcal{F}}$ scheduling policy as the scheduling policy that applies an $m - timeslots\, old$ matching of the $MWM^{\mathcal{F}}$ scheduling policy to configure the switch: If we set $\pi_m^{\mathcal{F}}(t) = \pi^{\mathcal{F}}(t-m)$, then the $MWM_m^{\mathcal{F}}$ policy calculates the match: $M_m(t) = \langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle$. We set $\Pi^{m,\mathcal{F}}(t) = \{\pi' : \langle \pi', \dot{\mathcal{F}}(\Phi(t))\rangle = \max_{\pi} \langle \pi, \dot{\mathcal{F}}(\phi(t-m))\rangle\}$, and follow an argument in [4] to derive from (11):

$$\sum_{\pi \in \Pi^{m,\mathcal{F}}(t)} \dot{T}_{\pi}^{MWM_m^{\mathcal{F}}}(t) = 1. \tag{14}$$

Lemma 1.

$$M_m(t) \geq M(t) - C(W,m), \tag{15}$$

where $C(W,m)$ is a constant that depends on m and the matrix W.

Proof. We note that by (7) $\lim_{t\to\infty} f_k(t) \to t/w^k$, and that $\overline{d}^k(t) \to \infty$ for $t \to \infty$. Thus

$$\phi^k(t) \to t - \frac{\overline{d}^k(t)}{w^k}, \tag{16}$$

and

$$|\phi^k(t+m) - \phi^k(t)| \leq m\left(1 + \frac{1}{w_k}\right) =: C(W,m), \tag{17}$$

From (17) and the Lipschitz condition of the function $\dot{\mathcal{F}}$, we obtain $\langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle \geq \langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle - KC_{\mathcal{F}}C(W,m)$, and $\langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle \geq \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle - KC_{\mathcal{F}}C(W,m)$. Further, by (12) $\langle \pi_m^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle \geq \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t-m))\rangle$. Combining the estimates, the lemma follows with $C(W,m) =: 2KC_{\mathcal{F}}C(W,m)$.

We note that by (13) and lemma 1,

$$M_c(t) \geq M(t) - C(W, z_a + z_b). \tag{18}$$

For technical reasons, we first prove theorem 1 for the case $z_a = 0$ and then show the general case $z_a > 0$. We define the Lyapunov function: $G(t) = \langle \mathbb{I}, \mathcal{F}_1(\Phi(t))\rangle$, where $\mathcal{F}_1(x) = \Gamma\mathcal{F}(x)$. By (16),

$$\dot{\Phi}(t) = \mathbb{I} - \dot{D}(t)\Gamma^{-1}. \tag{19}$$

We want to show that for an absolute constant $B > 0$ there is $\forall t \geq 0$,

$$||\Phi(t)||_1 \leq B, \tag{20}$$

where $||\ ||_1$ denotes the L_1 norm. Noting that $G(0) \geq 0$, we see that if

$$\frac{d}{dt}G(t) \leq 0 \tag{21}$$

$\forall t$ such that $\langle \mathbb{I}, \Phi(t)\rangle \geq K_0$ for a fixed $K_0 > 0$, then $\forall t \geq 0$, there holds $G(t) \leq \max_{\substack{L\in\mathbb{R}^K,\\ \langle \mathbb{I},L\rangle \leq K_0}} \langle \mathbb{W}, \dot{\mathcal{F}}(L)\rangle$, which in turn implies (20) for a certain $B > 0$. Thus, we derive (21), from (10), (14), (18), and (19):

$$\begin{aligned}
&\frac{d}{dt}G(t) = \langle 1, \dot{\mathcal{F}}_1(\Phi(t))\dot{\Phi}(t)\rangle \\
&= \langle 1, \Gamma\dot{\mathcal{F}}(\Phi(t))\rangle(I - \dot{D}(t)\Gamma^{-1}) = \langle 1, \dot{\mathcal{F}}(\Phi(t))\rangle(\Gamma - \dot{D}(t)) \\
&= \langle \Gamma, \dot{\mathcal{F}}(\Phi(t)\rangle - \sum_{\pi\in\Pi^{z_a+z_b,\mathcal{F}}(t)} \dot{T}_\pi^{MWM^{\mathcal{F}}_{z_a+z_b}} \langle \pi_{z_a+z_b}(t), \mathcal{F}(\dot{\Phi}(t))\rangle \\
&\leq \langle \Gamma, \dot{\mathcal{F}}(\Phi(t)\rangle - \langle \pi^{\mathcal{F}}(t), \dot{\mathcal{F}}(\Phi(t))\rangle + C(W, z_a + z_b).
\end{aligned} \tag{22}$$

As by (6),$||W||_{IO} < 1$, we argue as in [7] to find a W_1 satisfying $||W_1||_{IO} < 1$ and $W \leq (1-\epsilon)W_1$ for $\epsilon > 0$. We define Γ_1 based on W_1 analogously to Γ. We know from [7] that (12) implies $\langle \Gamma_1, \dot{\mathcal{F}}(\Phi(t))\rangle < \langle \pi(t), \dot{\mathcal{F}}(\Phi(t))\rangle$. Thus, by (22):

$$\begin{aligned}
\frac{d}{dt}G(t) &\leq \langle \Gamma_1, \dot{\mathcal{F}}(\Phi(t))\rangle - \langle \pi(t), \dot{\mathcal{F}}(\Phi(t))\rangle - \epsilon\langle \Gamma_1, \dot{\mathcal{F}}(\Phi(t)\rangle + C(W, z_a + z_b) \\
&\leq -\epsilon \min_{\substack{1\leq k\leq K\\ w^k>0}} w^k \max_{1\leq k\leq K} \dot{\mathcal{F}}(\phi^k(t)) + C(W, z_a + z_b).
\end{aligned}$$

As $\dot{\mathcal{F}}$ was supposed to be non-negative and strictly monotonically increasing, (21) follows. The relations (16) and(20) implies that: $0 < t - \frac{\overline{d}^k(t)}{w}^k + C \leq B$. Whence, $\lim_{t\to\infty} \frac{\overline{d}^k(t)}{t} = w^k$, i.e., $\lim_{t\to\infty} \frac{D(t)}{t} = W$, w.p.1, which corresponds to the rate stability condition for $X(t)$.

Following an argument in [14], we now treat the general case $z_a > 0$. For any switch that operates with a speedup S and has a configuration period of length $z_a = 0$, the equation (11) changes to

$$\sum_{\pi\in\Pi} T_\pi^{\mathcal{F}}(t) = St. \tag{23}$$

As for the $MWM_c^{\mathcal{F}}$ policy a fraction $\frac{z_a}{z_a+z_b}$ of the bandwidth is lost during the configuration phase, we adjust the RHS of (23) by a factor of $\frac{z_b}{z_a+z_b}$ and obtain the equality $\sum_{\pi\in\Pi} T_\pi^{\mathcal{F}}(t) = S\frac{z_b}{z_a+z_b}t$. Differentiating this equation, we follow an argument in [4] as in the derivation of (14), and find that for $S > z_a + z_b/z_b$,

$$\sum_{\pi\in\Pi^{z_a+z_b,\mathcal{F}}(t)} \dot{T}_\pi^{MWM^{\mathcal{F}}_{z_a+z_b}}(t) = S\frac{z_b}{z_a+z_b} > 1. \tag{24}$$

Arguing as for $z_a = 0$ with (24) instead of (14), the theorem follows for $z_a > 0$.

4 Networks of Switches that Deploy Different Scheduling Policies

In the previous section, we introduced scheduling policies for networks that provide stability for a network of switches where all switches implement the same scheduling policy. Here we prove that a network of switches in which each switch deploys any of those policies also achieves 100 % throughput.

Theorem 2. *A network of IQ/CIOQ switches with configuration overhead where each switch deploys any MWM_c^F policy, $\mathcal{F} \in \mathcal{G}$, in which the weight is defined as in (8), and which deploys a speedup $S \geq \left\lceil \frac{z_a+z_b}{z_b} \right\rceil$ achieves 100% throughput.*

Proof. We divide the switches in the network into M groups G_i, $i \in \{1,..,M\}$ where G_i contains the switches that deploy the switching policy $MWM_c^{F_i}$. Accordingly, we can divide the departure vector $D(t)$ and the arrival rate vector W in M subvectors, i.e., we write $D(t) = (D_1(t),..,D_M(t))$ and $W = (W_1,..,W_M)$. In order to prove rate stability, it is obviously sufficient to show that $\forall i \in \{1,..,M\}$, $\lim_{t\to\infty} \frac{D_i(t)}{t} = W_i$, w.p.1. This relation can be proved by applying the proof of theorem 1 to each group of switches G_i separately.

5 Conclusions

This paper investigates scheduling policies for networks of switches with a configuration overhead. It proposes a class of switching policies that are based on maximum weight matchings. It is shown that network of switches that deploy either one or any mixture of those policies with a speedup of $S \geq \lceil \frac{z_a+z_b}{z_b} \rceil$ achieve 100% throughput.

References

[1] Ajmone, M.,Giaccone, P., Leonardi, E., Mellia, M., Neri, F., *Local scheduling policies in networks of packet switches with input queues,* Proc. of Infocom 2003, San Francisco, April 2003. 638, 639, 640, 642, 643

[2] Ajmone, M.,Leonardi, E., Mellia, M., Neri, F., *On the throughput achievable by isolated and interconnected input-queued switches under multicalss traffic,* Proc. of Infocom 2002, New York City, June 2002. 638, 640, 642

[3] Andrews, M., Zhang, L., *Achieving stability in networks of input queued switches* Proc. of Infocom 2001, Anchorage, Alaska, April 2001. 638

[4] Dai, J. G., Prabhakar, B., *The throughput of data switches with and without speedup,* Proc. of IEEE Infocom 2000, Tel Aviv. 642, 644, 645

[5] Dai, J. G., *Stability of fluid and stochastic processing networks,* Miscellanea publ. n.9, Centre for Mathematical Physics and Stochastic, Denmark, Jan. 9. 642

[6] Lin, L. Y., *Micromachined free-space matrix switches with submillisecond switching time for large-scale optical crossconnect,* OFC Tech. Digest, 1998. 638

[7] Keslassy, I., McKeown, N., *Achieving 100% throughput in an input queued switch,* IEEE Transactions on Communications, vol. 47, no. 8, Aug. 1999, 1260 - 1272. 637, 638, 640, 645

[8] Keslassy, Chuang, S. T., Yu, K., Miller, d., Horowitz, M., Solgaard, M., McKeown, N., *Scaling Internet Routers Using Optics,* Proc. of ACM SIGCOMM, Karlsruhe, Germany, Aug. 2003.

[9] Li, X., Hamdi, M., *λ-adjust algorithm for optical switches with reconfiguration delay,* Proc. of ICC'03, Anchorage, Alaska, May 2003. New York City, June 2002. 638

[10] Li, X., Hamdi, M., *Design and analysis of scheduling algorithms for switches with reconfiguration overhead,* Proc. of High Performance Switching and Routing (HPSR'03), Torino, Italy. June 2003. 638

[11] Li, X., Hamdi, M., *Analysis of reduced rate scheduling for switches with reconfiguration overhead,* Proc. of Global Communications Conference (Globecom'03), San Francisco, Dec. 2003. 638

[12] McKeown, N., *Optics inside Routers,* Proc. of ECOC 2003, Rimini, Italy, September 2003. 637

[13] A. Neukermans and R. Ramaswami, *MEMS Technology for Optical Networking Applications*, IEEE Communications Magazine January 2001, p.62 - 69. 638

[14] Shah, D, Gangali, Y., Keshavarzian, A., *Input-queued switches: Cell switching versus packet switching*, Proc. of IEEE Infocom 2003, San Francisco. April 2003. 638, 645

[15] Towles, B., Dally, W. J., *Guaranteed scheduling for switches with configuration overhead,* Proc. of Infocom 2002, New York City, June 2002. 638

Location Management with Dynamic Anchor Scheme in Wireless ATM Networks

DongHo Kim[1], KangWoo Lee[1], Wonjong Noh[2],
Sinam Woo[2], and Sunshin An[2]

[1] School of Information and Communication Engineering, Halla University
San 66, Heungup-Li, Heungup-myon, Wonju-shi, Kangwon-do, Korea
{imi,kwlee}@hit.halla.ac.kr
[2] Computer Network Lab., Dept. of Electronic Eng., Korea University
Seoul, Korea
{nwj,niceguy,sunshin}@dsys.korea.ac.kr

Abstract. This paper presents a dynamic anchor scheme in wireless ATM (asynchronous transfer mode) networks, which improves the location registers (LR) scheme, in which LR's are hierarchically structured. This dynamic anchor scheme introduces the CCR (communication-to-computation cost ratio) term in the LR scheme. The proposed scheme reduces computation cost by introducing the active LR. There are at most two active LR's in a hierarchical arrangement of LR's and the location of one active LR is static and the location of another active LR called as anchor LR is dynamically selected among the LRs in the hierarchy in order to minimize a total location management cost. The total location management cost is calculated depending on the CCR as well as the CMR (call-to-move ratio). Numerical results show that the proposed scheme can reduce the location management cost compared with the original LR scheme for the different values of the CCR and the CMR.

Keywords: Wireless ATM, Location Management, anchor scheme

1 Introduction

In wireless communication, the point of attachment to infrastructure can be changed over time. Due to the mobility, wireless connection allows mobile terminals to move freely in the network and possibly also between the networks. Mobility management is one of the most important and challenging problems for wireless communication independent of network. Mobility management [1] is a set of functions, which enables telecommunication networks to track and locate the current point of attachment of a mobile user device or network entity and to provide seamless connectivity to mobile user device or network, while the terminal is moving into a new service area. Mobility management can be categorized into two areas: handoff management and location management. Handoff management is essentially based on wireless aspects and main difficulties in improving the performance of handoff management come from unpredictable and

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 648–657, 2004.

highly fluctuating radio link and the randomness of user's mobility. Location management has to face the tradeoff between mobile tracking and mobile locating.

This paper considers the location management for wireless ATM networks. There are basically two different ways of location management: a two-tier scheme, and a hierarchical scheme [2][3][4]. In the two-tier scheme, HLR (Home Location Register) stores temporary address based on the MSC (Mobile Switching Center)/VLR (Visitor Location Register), each VLR of serving MSCs stores location information on the MT (Mobile Terminal), and the temporary address is used to route the connection from the calling switch to the mobile's current switch. In the hierarchical scheme, location registers are hierarchically structured, and the temporary location of the MT is registered up the higher location register beyond which there is no change of information regarding the MT's current location. There is tradeoff between computation cost and communication cost. Long-distance query to HLR for the current location of the MT incurs larger communication costs. The hierarchical structure removes the need for the long-distance query. However, it increases the computation costs due to much larger number of location registers. Location management schemes in between these two extreme schemes in terms of computation and communication costs are studied. The forwarding scheme [5], the anchor scheme[6], the location registers (LR) scheme and the mobile PNNI scheme [4] are proposed in wireless ATM networks.

The LR scheme [4] distributes the LRs hierarchically with the scope S. The LR stores the location information on each of the mobile terminals (MT) being serviced by the peer group, while the higher peer group does not have the exact location information of the MT. If the value of the CCR is low, the location management cost of the LR scheme can be more expansive than that of the 2-tier scheme due to computation cost of each LR in the hierarchy. Furthermore, the scope value, which optimizes the location management cost, strongly depends on the CCR as well as the CMR. Unfortunately, the LR scheme does not consider the computation cost. In this paper, we propose dynamic anchor scheme, which improves the location registers (LR) scheme, in which LRs are hierarchically structured.

In this paper, we propose dynamic anchor scheme, which improves the location registers (LR) scheme, in which LRs are hierarchically structured. This dynamic anchor scheme introduces the CCR term in the LR scheme. Furthermore, in order to reduce computation cost, the proposed scheme introduces active LR term. There are at most two active LRs in the hierarchy; the level-L LR and the anchor LR. The anchor LR is dynamically selected, depending on the CCR as well as the CMR, among the LRs in the hierarchy in order to minimize a total location management cost. The proposed scheme is in between the LR scheme and the 2-tier scheme. We analyze the location management cost of the proposed scheme, and compare it with that of the original LR scheme with the fixed scope value. Numerical results shows that the proposed scheme can reduce the location

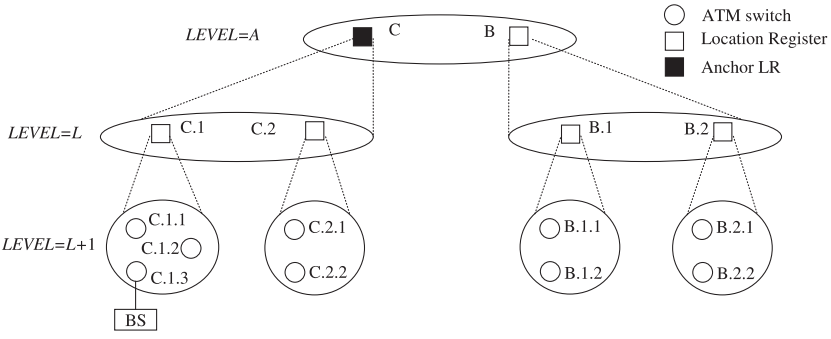


Fig. 1. Reference architecture for the dynamic anchoring scheme

management cost compared with the original LR scheme for the different values of the CCR and the CMR.

The rest of this paper is organized as follows. We discuss the proposed dynamic anchor scheme in section 2. We analyze the scheme in section 3, and present the numerical results of the scheme in section 4. Finally, section 5 concludes the paper.

2 Dynamic Anchor Scheme

We design a dynamic anchor scheme, in which location registers (LRs) are hierarchically organized in order to reduce communication cost by localizing the impact of mobility in wireless ATM networks. Fig. 1 shows the reference architecture for the dynamic anchor scheme. LR's are are hierarchically constructed from level-L up to level-1 with the some scope level-A. The level-A indicates anchor point. Thus, the architecture is the same as that of the location registers scheme [7].

However,the proposed scheme reduces computation cost by introducing the active LR, which are actually used. It essentially uses at most two active LRs for each MT. One active LR is level-L LR, the other active LR is level-A LR. The level-L LR stores the exact location information on an MT. The level-A LR called as an anchor LR has the pointer to the current level-L LR of the MT instead of the exact location information of the MT. The anchor LR is dynamically selected among the LRs in the hierarchy in order to minimize the total location management cost. The HLR of the MT only tracks the anchor LR.

2.1 Mobile Tracking

Mobile tracking is the procedure by which the information about the attachment point of a MT is updated, when the MT powers on/off or moves to a new BS. Fig. 2 shows how location registers are updated in the dynamic anchor scheme. a_{ij} is the ancestors-are-siblings level of nodes i and j at which both of nodes i

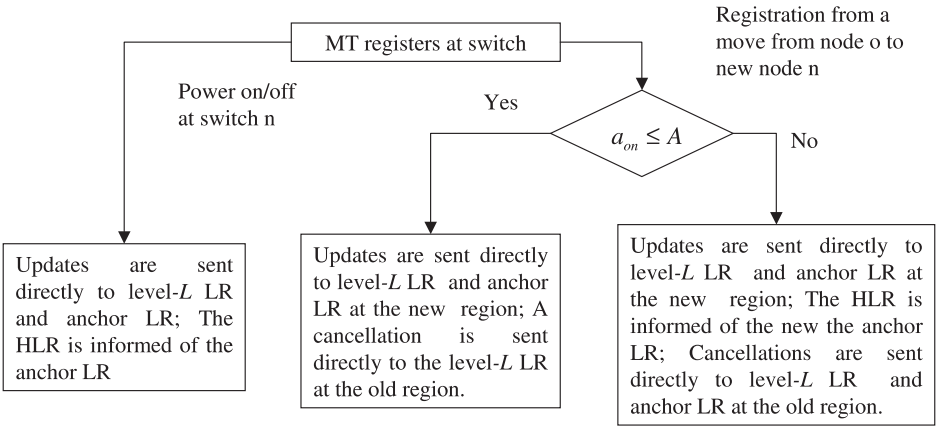


Fig. 2. Flowchart representing how updates are propagated

and j has a common ancestor. Subscripts o, n, v, h, and c of a_{ij} represent the old, new, visiting, home and calling party location of an MT, respectively.

If an MT powers on/off, the MT sends a message to the current base station (BS). The BS sends power-on/off registration notification (REGNOT) message to the level-L LR to which the BS belongs. The level-L LR, in turn, sends the message to the an-chor LR to notify that the MT is existing/away currently in its domain. The anchor LR also relays the message to the HLR of the MT. An MT has one HLR based on its permanent address.

If an MT moves to a BS of another switch within the same level-L domain ($a_{on} \geq L$), the MT sends a REGNOT message upward to its level-L LR. When an MT moves to another level-L domain within level-A domain ($A \leq a_{on} < L$), the MT sends a REGNOT message upward to its level-L LR. The level-L LR, in turn, sends the message to its anchor LR to notify that the MT is currently in its domain. On re-ceiving the information of the move, the old switch sends a registration cancelation (REGCANC) message to the old level-L LR to notify that the MT is away in the old level-L domain.

If the MT moves to another level-A domain ($a_{on} < A$), the REGNOT message is propagated upward to a new anchor LR. Since anchor LR changes due to the move, the REGNOT message is sent to the HLR of the MT in order to inform about new anchor LR. On receiving the information of the move, the old switch sends a REGCANC message upward to the old anchor LR.

2.2 Mobile Locating

Mobile locating is the procedure by which the target MT is found. Fig. 3 shows the flowchart representing how first call is propagated. When a call occurs, a chain of LRs is traced to find the called MT prior to a call setup.

If the called MT is at the same level-L domain of the calling MT, the BS of the calling MT generates a location request (LOCREQ) to its level-L LR

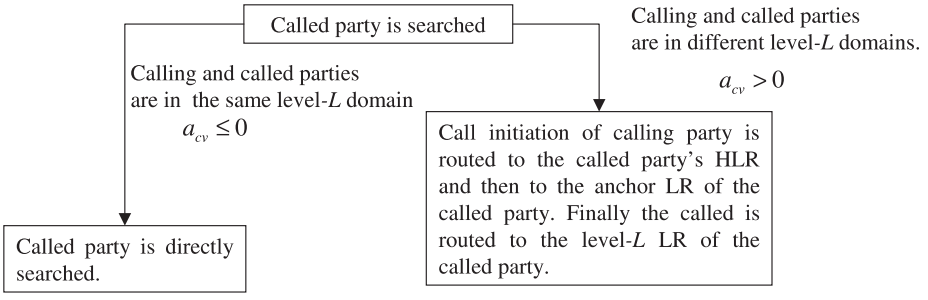


Fig. 3. Flowchart representing how updates are propagated

which then responds to the calling MT's switch. If calling and called parties are in different level-A domains, a LOCREQ message is sent directly to the HLR of the called party. This scheme referred to as the HLR last scheme is feasible when the calling party is close to the called party. The original LR scheme adopts the HLR last scheme. However, we assume that the distribution of calls to an MT follows a uniform distribution. Thus, it is probable that the HLR of the called party is away from the level-A domain of calling parties. If calling and called parties are in different level-L domains, the level-L LR of the calling party sends the call initiation directly to the HLR of the called party. This scheme is referred to as the HLR first scheme. We will consider the HLR first scheme only. On receiving the message, the HLR sends the call directly to the level-A of the called party. Finally, the call is routed to the level-L LR of the called party. The level-L LR sends an acknowledgement message to the calling party.

3 Performance Analysis

We assume that messages traverse the chain of switches in hierarchy. Since only at most two active LRs are really used for an MT in the proposed scheme, we define the cost of updating/querying LRs in the path from the level-L LR to the level-k LR as follows

$$R_k = \delta^c_{kL}\left(\sum_{i=A}^{L-1} c_i + P_A\right) + c_L + P_L \tag{1}$$

where

$$\delta^c_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases},$$

P_i is the the computation cost and c_i is the communication cost for updating/querying a level-i LR. We assume that $P_1 = P_2 = \cdots = P_A$. We denote the CCR as σ $(=c_i/P_i)$.

3.1 Tracking Cost

The average tracking cost per move is given by

$$\begin{aligned}\overline{M} = P(a_{on} = L)(R_L + 1) + \sum_{i=A}^{L-1} P(a_{on} = i)(2R_A - P_A + 1) \\ + \sum_{i=1}^{A-1} P(a_{on} = i) \cdot \left\{ \sum_{j=A}^{L} P(a_{hn} = j)(2R_A + h + 1) \right. \\ \left. + \sum_{j=A+1}^{H} P(a_{hn} = j)(2R_A + 2h) \right\} \end{aligned} \tag{2}$$

Then, the first term and the second term are the case of $a_{on} \leq A$. The level-L LR and the level-A LR relative to the new location are updated, and the information of the old level-L LR in the old location is deleted. When $a_{on} > A$, The level-L LR and the level-A LR relative to the new location are updated, and information of the old level-L LR and the level-A relative to the new location are deleted. We assume that the cost incurred to update the HLR is given by a long-distance message cost h, if the home switch is not in the neighborhood of the MT. Otherwise, the cost is given by 1. Therefore, the cost of updating the home switch depends on a_{hn}.

3.2 Mobile Locating

The average search cost per call is given by

$$\begin{aligned}\overline{S} = P(a_{cv} = L)(R_L + 1) + \sum_{i=A}^{L-1} P(a_{on} = i) \cdot \left\{ \sum_{j=A}^{L} P(a_{hv} = j)(R_L + R_A + 3) \right. \\ \left. + \sum_{j=1}^{A-1} P(a_{hv} = j)(R_0 + R_A + 2h + 1) \right\} \\ + \sum_{i=1}^{A-1} \cdot \left\{ \sum_{j=A}^{L} P(a_{hv} = j)(R_0 + R_A + 2h + 1) \right. \\ \left. + \sum_{j=1, j\neq i}^{A-1} P(a_{hv} = j)(R_0 + R_A + 3h) \right. \\ + P(a_{hv} = i) \cdot \left\{ \sum_{j=i}^{A-1} P(a_{ch} = j \mid a_{hv} = a_{cv} = i)(R_0 + R_A + 3h) \right. \\ \left. \left. \left. + \sum_{j=A}^{L} P(a_{ch} = j \mid a_{hv} = a_{cv} = i)(R_0 + R_A + 2h + 1) \right\} \right\} \end{aligned} \tag{3}$$

The first term is the case that the calling party and the called party are located in the same level-L domain. Then, the calling party sends a LOCREQ

message to the level-L LR relative to its location. Moreover, the final response is sent directly from the level-L LR to the switch of the called party. The second and the third terms are the case that the calling party and the called party are located in the same level-A domain. Furthermore, the second term is the case that the called party and the HLR of the calling party are located in the same level-A domain, and the third term is the other case. The level-L LR sends the LOCREQ message directly to the HLR. The HLR forwards the LOCREQ message to the current level-A LR, which, in turn, generates downward a LOCREQ message directly to the level-L LR of the called party. The level-L LR responds directly to the calling party's switch.

The other terms are the case that the calling party and the called party are located in different level-A domains. The fourth term is the case that the called party is in its home level-A domain. The other terms are the case that the called party is not in its home level-A domain. The fifth and the sixth terms are the case that the calling party is outside the home level-A domain, and the seventh term is the case that the calling party is in the home level-A domain. Costs for the terms can be similarly reasoned. The probability distributions of $P(a_{cv})$, $P(a_{ch})$, $P(a_{hv})$, $P(a_{hn})$, $P(a_{ho})$, $P(a_{on})$ and other conditional probability distributions for $\overline{M}$ and $\overline{S}$ are given by [4]. The total cost per unit time is given by

$$T = \lambda_m \overline{M} + \lambda_c \overline{S} \tag{4}$$

where λ_c is the rate of call arrival at an MT, and λ_m is the rate at which the MT moves between BSs. The average total cost per move is given by

$$T^m = \lambda_m \overline{M} + \lambda_c \overline{S} \tag{5}$$

where $\rho(= \lambda_c/\lambda_m)$ is call-to-mobility ratio (CMR).

4 Numerical Results

We show that the optimal anchor value A_{opt}, which optimizes the average total cost per move, strongly depends on the long-distance message cost h, the values of the CCR and the CMR. Furthermore, we compared the proposed location management scheme with the original LR scheme to evaluate the performance of the proposed scheme.

4.1 Optimal Value A_{opt}

We assume that L is 10, each P_i has the same value with 1, and each c_i has the same value based on the CCR (σ) value. Fig. 4 shows the effects of value A on the average total cost per move against CCR when h =8 and ρ =0.1. The results provide significant insight for the optimal value of A (=A_{opt}). If CCR is 0.1, A_{opt} is 7. But if CCR is 5, A_{opt} is 8. Fig. 5 shows that average total cost per move against h with CCR=0.3 and ρ =0.1. The results show that if h is 6, A_{opt} is 8. But if h is 8, A_{opt} is 7. Table 1 shows A_{opt} for the different values of h, ρ and σ, which minimize the average total cost per move. The table shows that A_{opt} increases as increases, and A_{opt} decreases as ρ increases or h increases.

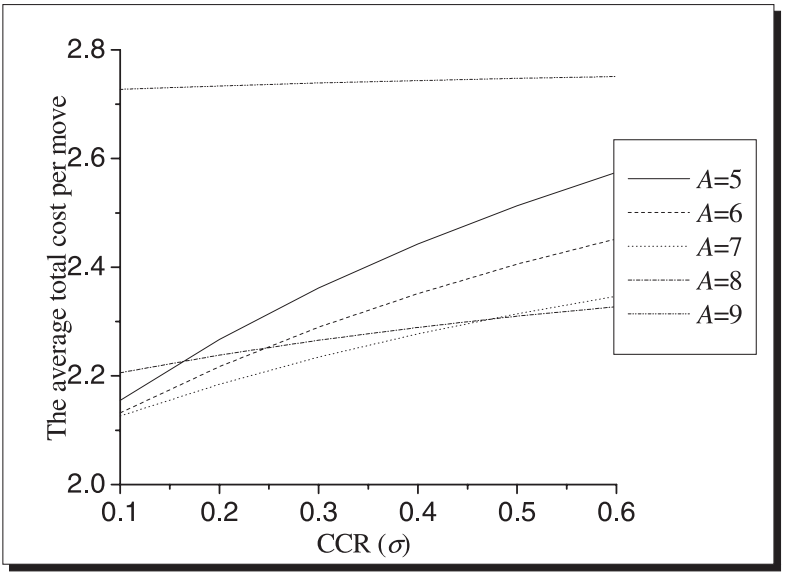


Fig. 4. Average total cost per move against CCR with h =8 and CMR=0.1

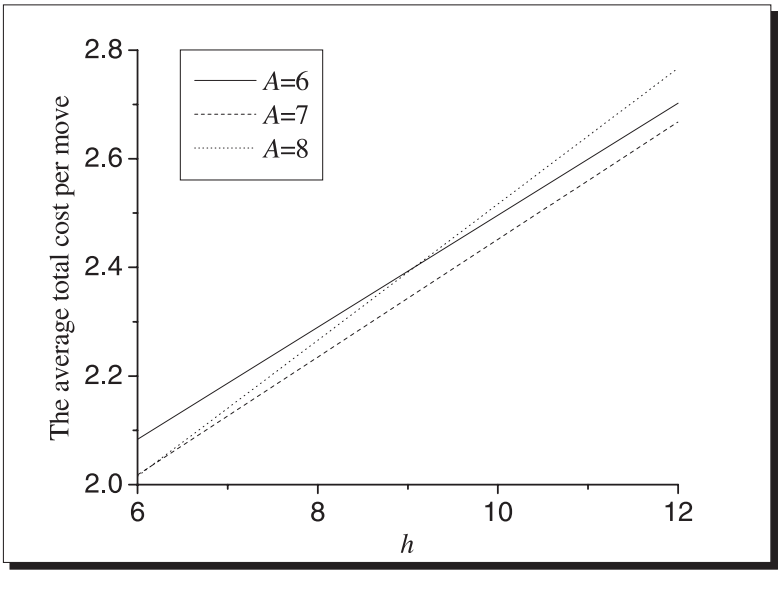


Fig. 5. Average total cost per move against h with CCR=0.3 and CMR=0.1

4.2 Comparison of the Proposed Scheme and the LR Scheme

Fig. 6 shows the average total cost per move of the proposed scheme and the LR scheme against CCR with h=8 and CMR=0.1. In the figure, S denotes scope-

Table 1. A_{opt} for the different values of h, CMR (ρ) and CCR (σ)

A_{opt}	ρ	0.1	0.2	0.3	0.4	0.5	0.6
h=6	0.01	7	7	8	8	8	8
	0.1	7	7	8	8	8	8
	1.0	6	7	8	8	8	8
h=8	0.01	7	7	7	7	8	8
	0.1	7	7	7	7	8	8
	1.0	6	7	7	7	7	7
h=10	0.01	6	7	8	8	8	8
	0.1	6	7	8	8	8	8
	1.0	5	6	7	8	8	8

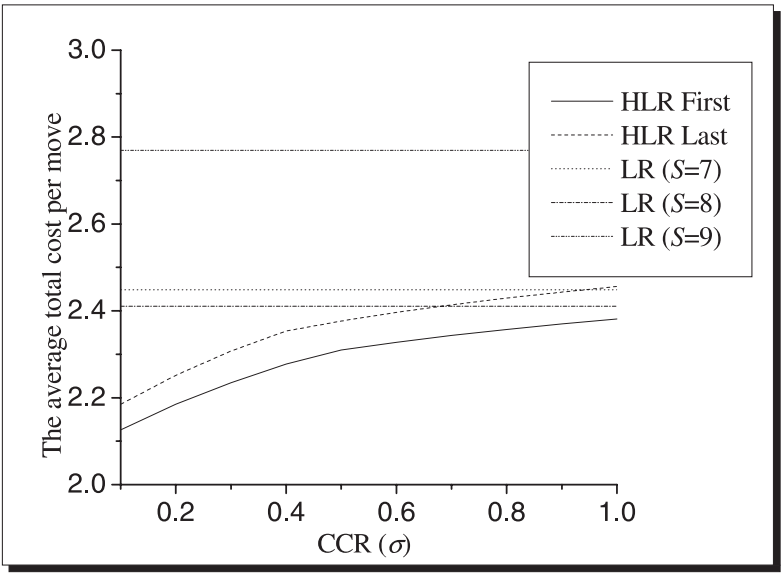


Fig. 6. Average total cost per move against CCR with h =8 and CMR=0.1

limiting parameter of the original LR scheme, and the average total cost per move is constant independent of the CCR in the LR scheme. The reason is that each P_i has the same value with 1. The figure shows that the proposed scheme improves the LR scheme. Furthermore, the figure also shows that performance of the HLR first is better than that of the HLR last in the proposed scheme.

5 Conclusions

In this paper we proposed the dynamic anchor scheme, which reduces average total cost dependent on the mobile tracking cost and mobile locating cost. Furthermore, it essentially uses at most two hierarchical location registers. One

location register co-locates with the level-L LR, the location of the other location register is dynamically determined according to the values of the CCR and the CMR. Numerical results showed that the optimal anchor level increases as CCR increases, and the optimal anchor level decreases as CMR increases or h increases. Numerical results also showed that the proposed scheme reduces the average total cost per move compared with the original LR scheme with the optimal anchor level. The proposed scheme can be extended to the ad-hoc networks.

References

[1] I. F. Akyildiz, J. Mcnair, J. Ho, H. Uzunalioglu, and W. Wang, Mobility management in current and future communications networks, *IEEE Network*, vol.12 no.4, pp. 39-49, 1998. 648

[2] J. Z. Wang, A fully distributed location registration strategy for universal personal communication system, *IEEE J. Select. Areas Commun.*, vol. 11, no. 2, pp 850-860, Aug. 1993 649

[3] Y.-B. Lin, Determining the user locations for personal communications services networks, *IEEE Trans. Veh. Technol.*, vol. 43, pp 466-473, Aug 1994 649

[4] . M. Veeraraghavan and G. Dommety, Mobile Location Management in ATM Networks, *IEEE J. Select. Areas Commun.*, vol. 15, no. 8. Oct, 1997 649, 654

[5] R. Jain and Y.-B. Lin, An auxiliary user location strategy employing forwarding pointers to reduce network impacts of PCS, *ACM/Baltzer Wireless Networks J.*, vol. 1, pp. 197-210, July 1995. 649

[6] J. S. M. Ho, and I. F. Akylidiz, Local anchor scheme for reducing location tracking costs in PCN's, *IEEE/ACM Trans. Networking*, vol. 4, pp. 709-725, Oct, 1996. 649

[7] The ATM Forum, Technical Committee, ATM User-Network Interface (UNI) Signalling specification, Version 4.0, *af-sig-0066.000*, ATM Forum/95-1434R13, June 1996. 650

[8] M. A. Marsan, C. Chiasserini, R. L. Cigno, and M. Munafo, Local and global handovers for mobility management in wireless ATM networks., *IEEE Personal Communications*, vol. 4, no. 5, pp. 16-24, 1997.

[9] Campbell, A., Liao, R.-F. and Shobatake, Y., Supporting QoS controlled handoff in mobiware, *the Sixth WINLAB Workshop on Third Generation Wireless Information Networks*, March 20-21, 1997.

Throughput and Delay Bounds for Input Buffered Switches Using Maximal Weight Matching Algorithms and a Speedup of Less than Two

Claus Bauer

Dolby Laboratories, San Francisco, CA, 94103, USA
cb@dolby.com

Abstract. Two main performance features of high-bandwidth switches are stability and delay. This paper investigates these performance features for input buffered switch architectures that deploy maximal weight matching algorithms to determine the configurations of the switching core. For this purpose, a novel mathematical model of the dynamics of a maximal weight matching algorithm is developed. Based on this model, it is shown that certain maximal weight matching algorithms provide stability when the switching core runs at a speedup of less than two. In addition, bounds on the expected average delay and on the absolute delay are established.

1 Introduction and Motivation

The core of most existing IP routers is based on a cell-based switching fabric. Because output queued switches become increasingly impractical due to the high speedup required in the switching core, most switches are based on either a pure input (IQ) or a combined input and output (CIOQ) buffered architecture. A typical CIOQ $N \times N$ switch is shown in figure 1. For each input i, there are N virtual output queues $VOQ_{i,j}$, $1 \leq j \leq N$. The cells arriving at input i and destined for output j are buffered in $VOQ_{i,j}$. The switching core itself is modeled as a crossbar requiring that not more than one packet can be sent simultaneously from the same input or to the same output. It works with a speedup of $S, S \geq 1$, i.e., it works at a speed S times faster than the speed of the input links. The choice of the scheduling algorithm is a major design criteria for switches. The scheduling algorithm should optimally provide guarantees on throughput and on average or absolute delay. In [4, 6, 7], it has been shown that for a speedup of $S = 1$ there exist scheduling algorithms that provide guarantees for the throughput and the average delay of a switch. These algorithms solve the scheduling problem by finding a maximum weight matching of a bipartite $N \times N$ graph. The weights are chosen to be either the queue length of the $VOQs$ or the actual waiting time experienced by the head of line cells of the $VOQs$. Let $\lambda_{i,j}$ define the arrival rate of cells for $VOQ_{i,j}$. Traffic is said to be admissible if

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 658–668, 2004.

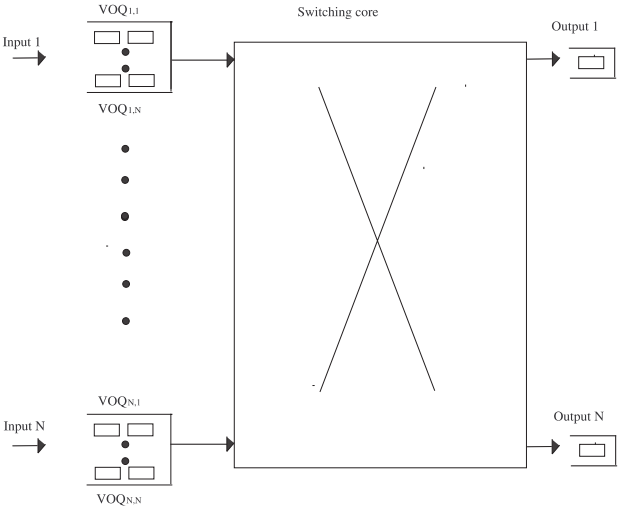


Fig. 1. Architecture of an input queued switch

$$\sum_{j=1}^{N} \lambda_{i,j} \leq 1, \qquad \sum_{i=1}^{N} \lambda_{i,j} \leq 1, \qquad \forall\, i, j,\ 1 \leq i, j \leq N. \tag{1}$$

However, the implementation of maximum weight algorithms is impractical as they have a complexity of $O(N^3 \log N)$. Therefore, the less complex class of maximal weight matching algorithms has been widely researched ([2], [8]). A maximal weight matching algorithm is defined for a set of weights $Q_{i,j}$, $1 \leq i, j \leq N$, where $Q_{i,j}$ is the weight assigned to $VOQ_{i,j}$, as follows:

1. Initially, all $VOQ_{i,j}$ are considered potential choices for a cell transfer.
2. The VOQ with the largest weight, say $VOQ_{a,b}$, is chosen for a cell transfer and ties are broken randomly.
3. All $VOQ_{i,j}$ with either $i = a$ or $j = b$ are removed.
4. If all $VOQ_{i,j}$ are removed, the algorithm terminates. Else go to step 2.

It has been shown in [3] and [5] that under the assumption of admissible traffic, every maximal weight matching algorithm deployed with a speedup of two guarantees the stability of the switch. In [1] these results were improved by showing that a switch that deploys any maximal weight matching algorithm is stable even with a speedup $S \geq R$, where

$$R = \max_{i,j} \left(\lambda_{i,j} + \sum_{k \neq j} \lambda_{i,k} + \sum_{l \neq i} \lambda_{l,j} \right). \tag{2}$$

We note that due to (1), there holds $R < 2$. Under the additional assumption that ties are broken in a specific way, the stability of the $MM-LQF$ algorithm,

the maximal weight matching algorithm with the weights chosen as the VOQ length, was even shown for $S = 1$ in [8]. (The $MM - LQF$ is denoted as $iLQF$ in [8].) Both papers describe the switch behavior via an approximate fluid model.

In contrast, this paper uses a discrete model to provide an analysis of switches deploying the $MM - LQF$ or the $MM - OCF$ algorithm, which uses the actual waiting time of the head of line cells as the weights, with a speedup $S > R$. It develops a new model to describe the dynamics of maximal matching algorithms. Based on this model, we derive results on stability and - in contrast to the work quoted above - also bounds on the expected average delay a cell experiences at the input buffer. In addition, an absolute delay bound for the $MM - OCF$ algorithm deployed with a speedup $S > R$ is established for the first time.

The rest of the paper is organized as follows. Section 2 introduces the terminology to describe the dynamics of a switch and gives some preliminary results. In section 3, an inequality that characterizes the behavior of maximal weight matching algorithms is derived. In the sections 4 - 6, results on stability, the average and the absolute delay for the $MM - LQF$ and/or the $MM - OCF$ algorithms are proven. Our conclusions are presented in section 7.

2 Terminology and Model

Throughout this paper, the time t is described via a discrete, slotted time model. Cells are supposed to be of fixed size. An external timeslot is the time needed by a cell to arrive completely at an incoming link. As the switching core works at a speedup $S \geq 1$, an internal timeslot is defined as the time needed to transfer a cell through the switching core from an input to an output. Thus, the external timeslot from time t to $t+1$ consists of the S internal timeslots $[t+(k-1)/S, t+k/S]$, $1 \leq k \leq S$. For the sake of simplicity, we always assume that S is an integer. All proofs in this paper can be easily generalized for non-integer S by considering the dynamics of the switch over g external timeslots, where gS is an integer, instead of over one external timeslot as done in this paper. We suppose that cells arrive at the beginning of an external timeslot t and are transferred instantly at the end of an internal timeslot. We abbreviate $\sum_{i,j} = \sum_{1 \leq i,j \leq N}$ and define the norm of an $N \times N$ matrix as $||x||_1 = \sum_{i,j} x_{i,j}$. We define the arrival matrix $A(t)$, representing the arrivals at each VOQ:

$$A_{i,j}(t) = \left\{ \begin{array}{l} 1, \text{ if an arrival at } VOQ_{i,j} \text{ occurs at at time } t, \\ 0, \text{ else.} \end{array} \right\}$$

As there cannot arrive more than one cell at an input during an external timeslot, there holds

$$\sum_{1 \leq j \leq N} A_{i,j}(t) \leq 1, \qquad \forall i,\, 1 \leq i \leq N,\, \forall t. \tag{3}$$

The service matrix $S^k(t)$, indicating which queues are served at the end of the k-th internal timeslot of the t-th external timeslot is defined as:

$$S^k_{i,j}(t) = \left\{ \begin{array}{l} 1, \text{ if } VOQ_{i,j} \text{ is served at the end of the } k\text{-th internal time slot} \\ \quad \text{of the } t\text{-th external time slot,} \\ 0, \text{ else.} \end{array} \right\}$$

Due to the crossbar structure of the switch, the following inequalities hold:

$$\sum_{1 \le i \le N} S^k_{i,j}(t) \le 1, \quad \sum_{1 \le j \le N} S^k_{i,j}(t) \le 1, \quad \forall i, j,\ 1 \le i, j \le N,\ \forall k,\ 1 \le k \le S, \forall t. \quad (4)$$

Now, we formalize the notion of the stability of a switch and give a sufficient criterion for stability which we will use later to establish some of our results.

Definition: Let $Y_n = (y_n(1), ..., y_n(M))$ be the row vector of a system of M queues at time n, where $y_n(i)$ is the length of the queue i at time n. A system of queues is said to be strongly stable if, for every $\epsilon > 0$, there exists $B > 0$ such that $\lim_{n\to\infty} P\{||Y_n||_1 > B\} < \epsilon$, and $\lim_{n\to\infty} \sup E||Y_n||_1 < \infty$.

Theorem 1. *Given a system of queues whose evolution is described by a discrete time Markov chain with state vector $X_n \in \mathbb{N}^M$, if a lower bounded function $V(X_n)$, called Lyapunov function, $V : \mathbb{N}^M \to \mathbb{R}$ can be found such that*

$$E[V(X_{n+1})|X_n] < \infty \ \forall\, X_n,$$

and there exist $\epsilon \in \mathbb{R}^+$ and $B \in \mathbb{R}^+$ such that

$$E[V(X_{n+1}) - V(X_n)|X_n] < -\epsilon \ \ \forall ||X_n|| > B,$$

then the system of queues is strongly stable.

Proof. This is a special instance of theorem 1 in [4].

3 A Characterization of Maximal Weight Matching Algorithms

In this section, we develop an inequality that describes the dynamics of a maximal weight matching algorithm. We define $Q(t) = (Q_{1,1}(t), ..., Q_{N,N}(t))$ as the weights at the beginning of the t-th external timeslot and $Q^k_{i,j}(t)$, $1 \le k \le S$, as the weight of $VOQ_{i,j}$ at the beginning of the k-th internal timeslot of the t-th external timeslot. Thus, $Q_{i,j}(t) = Q^1_{i,j}(t)$, $\forall\, i, j,\ 1 \le i, j \le N$. We show:

Theorem 2. *For any input buffered switch that applies a maximal weight matching algorithm and a speedup-up S, there holds at any time t*

$$\frac{1}{R}\sum_{k=1}^{S}\sum_{i,j} Q_{i,j}^{k}(t)\lambda_{i,j} \le \sum_{i,j}\sum_{k=1}^{S} Q_{i,j}^{k}(t)S_{i,j}^{k}(t).$$

Proof. In the first internal timeslot, in its first iteration, the algorithm selects the queue with the largest weight, say Q_{a_1,b_1}^{1}, for transfer. Thus, from (2):

$$\begin{aligned} Q_{a_1,b_1}^{1}(t)R &\ge Q_{a_1,b_1}^{1}(t)\left[\lambda_{a_1,b_1} + \sum_{k\neq b_1}\lambda_{a_1,k} + \sum_{l\neq a_1}\lambda_{l,b_1}\right] \\ &\ge Q_{a_1,b_1}^{1}(t)\lambda_{a_1,b_1} + \sum_{k\neq b_1} Q_{a_1,k}^{1}(t)\lambda_{a_1,k} + \sum_{l\neq a_1} Q_{l,b_1}^{1}(t)\lambda_{l,b_1}. \end{aligned}$$

All queues with either $i = a_1$ or $j = b_1$ are removed. In the second iteration, the remaining queue with the largest weight is chosen. Thus,

$$Q_{a_2,b_2}^{1}(t)R \ge Q_{a_2,b_2}^{1}(t)\lambda_{a_2,b_2} + \sum_{k\neq b_1,b_2} Q_{a_2,k}^{1}(t)\lambda_{a_2,k} + \sum_{l\neq a_1,a_2}\lambda_{l,b_2}Q_{l,b_2}^{1}(t). \quad (5)$$

The matching algorithm stops after k, $k \le N$ iterations when only empty queues remain. For each of these k iterations, an inequality analogous to (5) holds. For the remaining empty queues, additional $(N-k)$ inequalities as in (5) hold where both sides are equal to zero. Summing over all N inequalities, we obtain

$$\sum_{m=1}^{N} Q_{a_m,b_m}^{1}(t) = \sum_{i,j} Q_{i,j}^{1}(t)S_{i,j}(t) \ge R^{-1}\sum_{i,j} Q_{i,j}^{1}(t)\lambda_{i,j}. \quad (6)$$

Applying this analysis to all S internal timeslots, we get the theorem.

4 Stability

In this section, we use the result of the previous section to prove stability for the $MM-LQF$ and $MM-OCF$ algorithms.

We set $L(t) = (L_{1,1}(n), .., L_{N,N}(t))$ where $L_{i,j}(t)$ defines the occupancy of $VOQ_{i,j}$ at the beginning of t-th external timeslot. We define $L_{i,j}^{k}(t)$, $1 \le k \le S$ as the occupancy of $VOQ_{i,j}$ at the beginning of the k-th internal timeslot in the t-th external timeslot. Therefore, $L_{i,j}(t) = L_{i,j}^{1}(t)$, $\forall\, i, j$, $1 \le i, j \le N$. The development of the VOQ occupancy between consecutive internal timeslots is described by the following equations:

$$L_{i,j}^{k}(t) = [L_{i,j}^{k-1}(t) - S_{i,j}^{k-1}(t)]^{+},\ \forall k,\ 2 \le k \le S, \quad (7)$$

$$L_{i,j}^{1}(t) = [L_{i,j}^{S}(t-1) - S_{i,j}^{S}(t-1)]^{+} + A_{i,j}(t), \quad (8)$$

where $[a]^+ = \max(0, a)$. Combining (7) and (8), we see:

$$L_{i,j}(t+1) = [L_{i,j}(t) - \sum_{k=1}^{S} S_{i,j}^k(t)]^+ + A_{i,j}(t+1).$$

For technical reasons, we also introduce the approximate next-state vectors:

$$\tilde{L}_{i,j}^k(t) = L_{i,j}^{k-1}(t) - S_{i,j}^{k-1}(t), \ \forall k, \ 2 \le k \le S, \tag{9}$$

$$\tilde{L}_{i,j}^1(t) = L_{i,j}^S(t-1) - S_{i,j}^S(t-1) + A_{i,j}(t), \tag{10}$$

$$\tilde{L}_{i,j}(t+1) = L_{i,j}(t) - \sum_{k=1}^{S} S_{i,j}^k(t) + A_{i,j}(t+1). \tag{11}$$

We see from (4) and (7):

$$L_{i,j}^S(t) \le L_{i,j}^k(t) \ \forall k, \ 1 \le k \le S, \tag{12}$$

$$||L^S(t)||_1 \ge ||L(t)||_1 - (S-1)N. \tag{13}$$

Now we prove theorem 3 which was shown via a different approach in [1]:

Theorem 3. *The $MM-LQF$ algorithm with a speedup $S > R$ is stable for all admissible i.i.d. arrival processes.*

Proof. We define the Lyapunov function as $V(L(t)) = \sum_{i,j} L_{i,j}^2(t)$. We will give an upper bound for the expected value of the expression $V(\tilde{L}(t+1)) - V(L(t))$. By (7) and (9) there is, $V(\tilde{L}^k(t)) \ge V(L^k(t))$. Thus, we get from (9), (10), and (4):

$$\begin{aligned}
&V(\tilde{L}(t+1)) - V(L(t)) \\
&= V(\tilde{L}(t+1)) - V(L^S(t)) + \sum_{k=2}^{S} \left[V(L^k(t)) - V(L^{k-1}(t))\right] \\
&\le V(\tilde{L}(t+1)) - V(L^S(t)) + \sum_{k=2}^{S} \left[V(\tilde{L}^k(t)) - V(L^{k-1}(t))\right] \\
&= 2\sum_{i,j} \left(A_{i,j}(t+1) - S_{i,j}^S(t)\right) L_{i,j}^S(t) + \sum_{i,j} \left(A_{i,j}(t+1) - S_{i,j}^S(t)\right)^2 \\
&\quad + \sum_{k=1}^{S-1} \left[-2\sum_{i,j} S_{i,j}^k(t) L_{i,j}^k + \sum_{i,j} \left(S_{i,j}^k(t)\right)^2\right].
\end{aligned} \tag{14}$$

Rearranging the last two sums $\sum_{i,j} .. + \sum_{k=1}^{S-1} ..$ in (14), we obtain from (3) and (4):

$$\sum_{i,j} A_{i,j}^2(t+1) - 2\sum_{i,j} A_{i,j}(t+1) S_{i,j}^S(t) + \sum_{k=1}^{S} \sum_{i,j} \left(S_{i,j}^k(t)\right)^2 \le N(S+1). \tag{15}$$

Applying (12), (15) and theorem 2 with $Q_{i,j}^k(t) = L_{i,j}^k(t)$, we obtain from (14):

$$\begin{aligned}
&E\left[V(\tilde{L}(t+1)) - V(L(t))|L(t)\right] \\
&\leq N(S+1) + 2\sum_{i,j}\lambda_{i,j}L_{i,j}^S(t) - 2\sum_{k=1}^{S}\sum_{i,j}S_{i,j}^k(t)L_{i,j}^k(t) \\
&\leq N(S+1) + 2\sum_{i,j}\lambda_{i,j}L_{i,j}^S(t) - \frac{2}{R}\sum_{k=1}^{S}\sum_{i,j}\lambda_{i,j}(t)L_{i,j}^k(t) \\
&\leq N(S+1) + 2\left[\sum_{i,j}\lambda_{i,j}L_{i,j}^S(t) - \frac{2S}{R}\sum_{i,j}\lambda_{i,j}(t)L_{i,j}^S(t)\right] \\
&\leq N(S+1) + 2\left(1 - \frac{S}{R}\right)\min_{\substack{i,j \\ \lambda_{i,j}>0}}\lambda_{i,j}||L^S(t)||_1. \qquad (16)
\end{aligned}$$

From (13) and (16), we see

$$\begin{aligned}
E\left[V(\tilde{L}(t+1)) - V(L(t))|L(t)\right] \leq N(S+1) + 2\left(1 - \frac{S}{R}\right)\min_{\substack{i,j \\ \lambda_{i,j}>0}}\lambda_{i,j}||L(t)||_1 \\
+2\left(\frac{S}{R} - 1\right)\min_{\substack{i,j \\ \lambda_{i,j}>0}}\lambda_{i,j}\,N(S-1). \qquad (17)
\end{aligned}$$

By definition, we note that for a value $b, 0 \leq b \leq S$,

$$L_{i,j}(t+1) - \tilde{L}_{i,j}(t+1) = \left\{\begin{array}{l} 0, \text{ if } L_{i,j}(t) \geq S, \\ b, \text{ else }. \end{array}\right\} \qquad (18)$$

Thus, $E[V(L(t+1)) - V(\tilde{L}(t+1))|L(t)] \leq N^2S^2$. Now, the theorem follows from theorem 1, because we obtain by (17):

$$E[V(L(t+1)) - V(L(t))|L(t)] \leq -\epsilon||L(t)||_1, \qquad \forall L(t) : ||L(t)|| > B.$$

For the analysis of the $MM-OCF$ algorithm, we denote by $C_{i,j}^k(t)$ the head of line cell of $VOQ_{i,j}$ at the beginning of the k-th internal timeslot of the t-th external timeslot. We define the interarrival vector $\tau^k(t) = (\tau_{1,1}^k(t), .., \tau_{N,N}^k(t))$, where $\tau_{i,j}^k(t)$ is the interarrival time between $C_{i,j}^k(t)$ and the cell behind it in line. $W_{i,j}^k(t)$ denotes the actual waiting time of $C_{i,j}^k(t)$. We set $W_{i,j}(t) = W_{i,j}^1(t)$, and describe the evolution of $W_{i,j}^k(t)$ between consecutive internal timeslots:

$$W_{i,j}^k(t) = [W_{i,j}^{k-1}(t) - S_{i,j}^{k-1}(t)\tau_{i,j}^{k-1}(t)]^+, \qquad (19)$$

$$W_{i,j}^1(t) = [W_{i,j}^S(t-1) - S_{i,j}^S(t-1)\tau_{i,j}^S(t-1) + 1]^+, \qquad (20)$$

$\forall k, 2 \leq k \leq S$. Combining the last two equations, we obtain:

$$W_{i,j}(t+1) = \left[W_{i,j}(t) - \sum_{k=1}^{S}S_{i,j}^k(t)\tau_{i,j}(t) + 1\right]^+. \qquad (21)$$

The approximate next-state vectors are defined $\forall k,\ 2 \le k \le S$ as follows:

$$W_{i,j}^{k}(t) = W_{i,j}^{k-1}(t) - S_{i,j}^{k-1}(t)\tau_{i,j}^{k-1}(t), \tag{22}$$

$$W_{i,j}^{1}(t) = W_{i,j}^{S}(t-1) - S_{i,j}^{S}(t-1)\tau_{i,j}^{S}(t-1) + 1, \tag{23}$$

$$\tilde{W}_{i,j}(t+1) = W_{i,j}^{k}(t) - \sum_{k=1}^{S} S_{i,j}^{k}(t)\tau_{i,j}^{k}(t) + 1.$$

We argue as in (12) and (13) and use the relation $E[\tau_{i,j}^{k}(t)] = \lambda_{i,j}^{-1}$ to obtain:

$$W_{i,j}^{S}(t) \le W_{i,j}^{k}(t)\ \forall k,\ 2 \le k \le S, \tag{24}$$

$$E\left[||W^{S}(t)||_1\right] \ge E\left[||W(t)||_1\right] - (S-1)N \min_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j}^{-1}. \tag{25}$$

Theorem 4. *The $MM-OCF$ algorithm with a speedup $S > R$ is stable for all admissible i.i.d. arrival processes.*

Proof. (Sketch): We follow the proof of theorem 3. As in [6], we use the Lyapunov function $V(W(t))$ defined as $V(W(t)) = \sum_{i,j} W_{i,j}^{2}(t)\lambda_{i,j}$. Applying (22) and (23) instead of (9) and (10), we obtain

$$E\left[V(\tilde{W}(t+1)) - V(W(t))\right] = 2\sum_{i,j}\lambda_{i,j}W_{i,j}^{S}(t) - 2\sum_{k=1}^{S} S_{i,j}^{k}(t)W_{i,j}^{k}(t) + \sum_{i,j}\left[\lambda_{i,j} - 2\sum_{k=1}^{S} S_{i,j}^{k}(t) + \sum_{k=1}^{S}\frac{S_{i,j}^{k}(t)}{\lambda_{i,j}}\right].$$

The sequel of the proof follows the proof of theorem 3. Instead of the relations (12) and (13), we use the relations (24) and (25). Instead of (18), we use:

$$W_{i,j}(t+1) = \left\{\begin{matrix} \tilde{W}_{i,j}(t+1), & \text{if } W_{i,j}(t+1) \ge 0, \\ 0, & \text{else.} \end{matrix}\right\}$$

□

5 Bounds on Average Delay

In this section, we derive bounds on the average delay experienced by cells at the VOQs. We first consider the $MM-LQF$ algorithm and prove the following bound on the average sum of the length of all VOQs.

Theorem 5. *Under the assumption of i.i.d. admissible traffic, for the $MM-LQF$ algorithm with a speedup $S > R$, the expected sum of all the queue lengths $E[||L(t)||_1]$ is bounded as follows:*

$$E[||L(t)||_1] \le \frac{\sum_{i,j}\lambda_{i,j}}{\left(\frac{S}{R}-1\right)\min\limits_{\substack{i,j \\ \lambda_{i,j}>0}}\lambda_{i,j}} + N. \tag{26}$$

Proof. Following an argument in [7], we revisit the proof of theorem 3. We assume that $L(t+1)=\tilde{L}(t+1)$. An analysis of the proof below shows that the exact state vector would incur an additional term of $N^2(S+1)$ to the delay bound. The approximate state vector will give a bound C. We will see that in general $C \geq N^2(S+1)$, and hence we will not consider the term $N^2(S+1)$ in this section. Arguing as in (14) and (16), without using the estimate (15), we obtain:

$$V(L(t+1)) - V(L(t)) \leq 2\left(1-\frac{S}{R}\right) \min_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j} ||L^S(t)||_1 + \sum_{i,j} A_{i,j}^2(t+1) \\ -2\sum_{i,j} A_{i,j}(t+1) S_{i,j}^S(t) + \sum_{k=1}^{S} \sum_{i,j} \left(S_{i,j}^k(t)\right)^2. \quad (27)$$

Following an argument from [7], we see that for large enough t.

$$E[A_{i,j}(t)] = \lambda_{i,j}, \quad E[A_{i,j}^2(t)] = \lambda_{i,j}, \quad (28)$$

As the $MM-LQF$ algorithm is stable, the switch state becomes a discrete time Markov chain with a stationary distribution. Thus, for large enough t,

$$E\left[\sum_{k=1}^{S} S_{i,j}^k(t)\right] = E\left[\sum_{k=1}^{S} \left(S_{i,j}^k(t)\right)^2\right] = E[A_{i,j}(t)] = \lambda_{i,j}. \quad (29)$$

In summary, from (27)-(29) we obtain:

$$E[V(L(t+1)) - V(L(t))|L(t)] \leq 2\sum_{i,j} \lambda_{i,j} + 2 \min_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j} \left(1-\frac{S}{R}\right) E[||L(t)||_1].$$

We use the last equation to obtain:

$$E[V(L(t+1))] = E[V(L(t+1)) - V(L(t)) + V(L(t))] \\ \leq 2\sum_{i,j} \lambda_{i,j} + 2 \min_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j} \left(1-\frac{S}{R}\right) E[||L^S(t)||_1] + E[V(L(t))].$$

Summing over $t=0$ to $t=T-1$, we get

$$E[V(L(T))] \leq 2T\sum_{i,j} \lambda_{i,j} + 2 \min_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j} \left(1-\frac{S}{R}\right) \sum_{t=0}^{T-1} E\,||L^S(t)||_1 \\ +E[V(L(0))].$$

We see from (3) and (10) that $||L_{i,j}(t+1)||_1 \leq ||L_{i,j}^S(t)||_1 + N$. Thus, noting that $V(\cdot) \geq 0$ and assuming $E[V(L(0))] = 0$, we derive from the last equation:

$$\frac{1}{T}\sum_{t=0}^{T} E[||L(t+1)||_1] \leq \frac{\sum_{i,j} \lambda_{i,j}}{\left(\frac{S}{R}-1\right) \min_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j}} + N. \quad (30)$$

As we assume an i.i.d. arrival process, the switch state is a discrete, irreducible aperiodic Markov chain. Thus, it is ergodic, and the left hand side of (30) converges to the expected value of $||L(T+1)||_1$ in the equilibrium state, i.e., $\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} E[||L(t+1)||_1] = \lim_{T\to\infty} E[||L(T+1)||_1]$. Inserting this relation in (30), (26) follows.

For the specific case of uniform arrival traffic for all VOQs, i.e., $\lambda_{i,j} = \lambda \forall i,j,\ 1 \le i,j \le N$, there holds $E[L(t)]/N^2 = E[L_{i,j}(t)]$. Using Little's law, we derive from theorem 5 that the expected delay $E[T]$ is bounded as:

$$E[T] \le \frac{1}{\left(\frac{S}{R}-1\right)\lambda} + \frac{1}{N\lambda}. \tag{31}$$

For the $MM-OCF$ algorithm, the following theorem holds:

Theorem 6. *Under the assumption of i.i.d. admissible traffic, for the $MM-OCF$ algorithm with a speedup $S > R$, the expected sum of all the delays in all virtual output queues $E[W(t)]$ is bounded as follows:*

$$E[||W(t)||_1] \le \frac{N^2 + \sum_{i,j} \lambda_{i,j}}{2\left(\frac{S}{R}-1\right) \min\limits_{\substack{i,j \\ \lambda_{i,j}>0}} \lambda_{i,j}} + N.$$

Proof. The proof follows the proof of theorem 5. □

Assuming uniform traffic, a bound on the average delay is obtained by dividing the right side of theorem 6 by N^2. This differs from the derivation of the delay bound (31) from theorem 5 which was performed using Little's law.

6 Absolute Delay Bound for the $MM-OCF$ Algorithm

In this section, we use techniques first introduced in [2] to establish a bound on the absolute delay for the $MM-OCF$ algorithm.

Theorem 7. *There exists a constant F, such that under the assumption of i.i.d. admissible traffic, for the $MM-OCF$ algorithm with a speedup $S \ge R$, the absolute maximum delay D a cell can experience at an input queue is bounded as $D \le \frac{(2N-1)F+1}{S-R}$.*

Proof. We follow the proof of theorem 5 in [2]. Whereas in [2] the arrival rate of all cells that compete for resources with a given cell was bounded by 2, we bound it by R. The leaky bucket conditions and the constant F needed for the proof in [2] can be derived from (1).

7 Conclusions

This paper presents a new approach to establish results on stability and delay bounds for maximal weight matching algorithms. The proof is based on a new mathematical model of the dynamics of a maximal weight matching algorithm and on the theory of the Lyapunov function. This approach allows for the first time to give delay bounds for maximal weight matching algorithms with a speedup of less than two. Furthermore, an absolute delay bound for a maximal weight matching algorithm with a speedup of less than two is proved.

References

[1] Benson, K., *Throughput of crossbar switches using maximal matching algorithms,* Proc. of IEEE ICC 2002, New York City. 659, 663

[2] Charny, A., Krishna, P., et al., *Algorithms for providing bandwidth and delay guarantees in input buffered crossbars with speedup,* Proc. of IWQoS, Napa, CA, 1998. 659, 667

[3] Dai, J. D.; Prabhakar,B., *The throughput of data switches with and without speedup,* Proc. of IEEE Infocom 2000, Tel Aviv. 659

[4] Leonardi, E., Mellia, M., Neri, F., Marsan, M. A., *Bounds on average delay and queue size averages and variances in input queued cell-based switches,* Proc. of IEEE Infocom 2001, Anchorage, Alaska. 658, 661

[5] Leonardi, E., Mellia, M., Neri, F., Marsan, M. A., *On the stability of input-buffered cell switches with speed-up,* Proc. of IEEE Infocom 2001, Anchorage, Alaska. 659

[6] N. McKeown, A.Mekkittikul, V. Anantharam, J. Walrand, *Achieving 100% throughput in an input-queued switch,* IEEE Transactions on Communications, vol. 47, no. 8, Aug. 1999, 1260 - 1272. 658, 665

[7] Shah, D.; Kopikare, M., *Delay bounds for approximate maximum weight matching algorithms for input queued switches,* Proc. of IEEE Infocom 2002, New York City, June 2002. 658, 666

[8] Shah, D,; *Stable Algorithms for input queued switches,* Proc. 39th Annual Allerton Conference on Communication, Control and Computing, Oct. 2001. 659, 660

Research on Protection Mechanisms of Resilient Packet Ring Network

Krzysztof Nowicki[1], Pawel Wojnarowicz[1], and Kamil Ratajczak[2]

[1] Gdansk University of Technology, 80-952 Gdansk, Poland
know@pg.gda.pl, pawel.wojnarowicz@pf.pl
[2] Pukyong National University, Busan 608-737, Korea
kamrat7@wp.pl

Abstract. Paper describes iSteering – a new method to control traffic in case of RPR network failure. There's comparison of packet wrapping, steering and our proposed method iSteering, which can be used in case of transmitter/receiver or fiber failure. Numerical analysis for even and random traffic flow has been made in order to compare packet wrapping, steering and iSteering. The results of all the tests are included.

1 Introduction

Resilient Packet Ring is a new technology for LAN, MAN and WAN networks and currently is in research and development stage. It has its ancestors in Cisco DPT (Dynamic Packet Transport) [1] and Nortel Optera Packet Edge [2]. Idea is based on two counter-rotating fiber rings providing fairness, spatial reuse and protection. Network is composed of nodes connected by two fiber rings as shown on Fig.1.:

Rings are called "inner" and "outer". Data packets are transmitted in one direction (downstream) whereas control packets are transmitted in the opposite

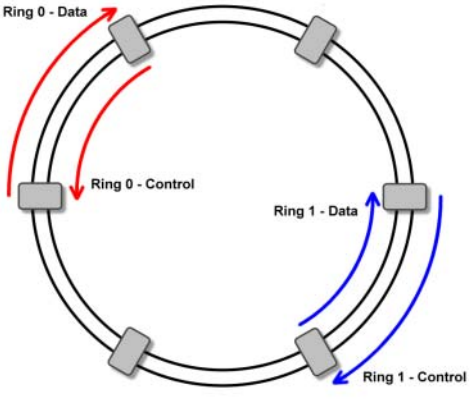


Fig. 1. Idea of RPR network

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 669–678, 2004.

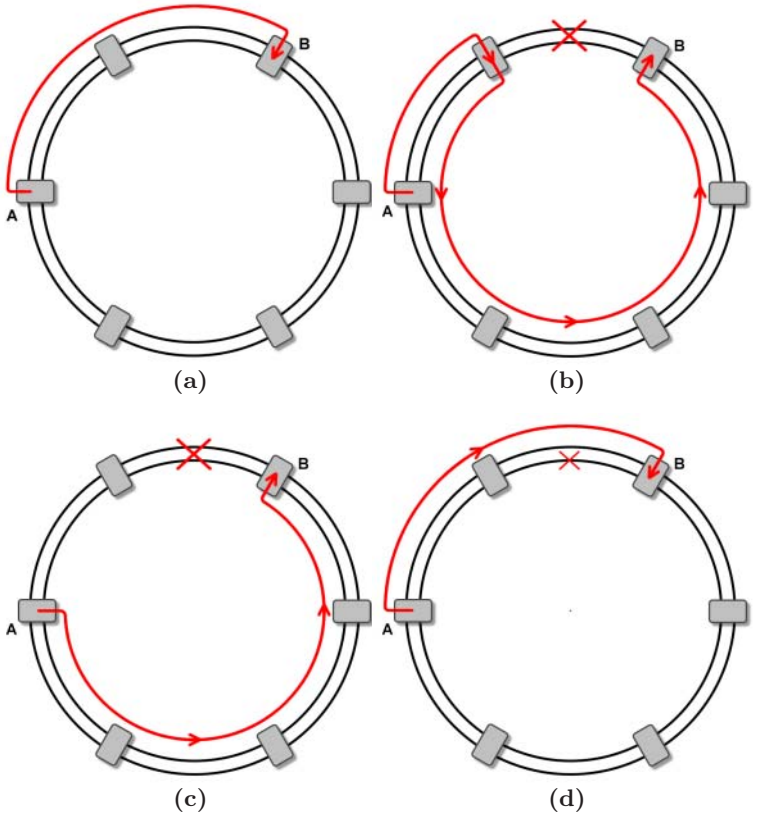


Fig. 2. Protection mechanisms (a) Normal work (no network malfunction) (b) Packet wrapping mechanism (c) Packet steering mechanism (d) iSteering mechanism

direction (upstream) in the other fiber. Any packet transmission between two nodes can be realized using one of two available paths – shortest path is chosen by default. One of the features of RPR is protection mechanism used in case of malfunction of a network providing fast resilience (topology recovery within 50 ms) – we focus on comparison of network performance before and after failure.

2 Protection Switching

2.1 Packet Wrapping and Packet Steering

Let's consider transmission between nodes A and B if there is no network malfunction (normal work). Packet are send through shorter path (outer ringlet in this case) as shown on Fig.2a.

There are two main methods to cope with network failure: packet wrapping and packet steering.

Packet wrapping is inherited from FDDI [3]. Fig.2b. shows how packet wrapping works [4]. If utilizing packet wrapping only nodes adjacent to failure point wrap traffic, whereas remaining nodes don't have to redirect packets. Node A sends packets to node B as if there is no failure (depicted as X). As you can see in wrapping mechanism packets may be directed to the other ring, which can unfavorably influence bandwidth utilization.

Packet steering is a more complex method. Every node is aware of network malfunction and transmits packets using shortest path. Fig.2c. explains packet steering mechanism [4]. Node A transmits packets to node B through inner ring because this path is now shorter than previous path (outer ring was used before network malfunction). Packet steering is proposed as a default protection mechanism, which is justified by the fact that even in packet wrapping every node knows about the point of malfunction.

2.2 iSteering (intelligent Steering)

Both packet wrapping and packet steering assume dual fiber or both transmitter and receiver failure. We propose a new way to control traffic in case of network malfunction.

In case of failure of both ringlets iSteering works exactly like packet steering. Difference appears if only one transmitter/receiver or one ring fails. iSteering takes advantage of the remaining ring or transmitter/receiver in order to transmit data (shorter path is chosen).

Fig.2d. illustrates how iSteering works. Let's assume single fiber failure as above (inner ring failure). Node A can transmit data to node B using outer ring now, which would be impossible with packet wrapping or steering. iSteering allows to utilize remaining fiber, which can improve network efficiency. This can be really significant for nodes which accessibility is crucial for network such as access nodes for SAN or databases.

Another advantage of iSteering is easy and cheap implementation. Packet wrapping as well as packet steering know where malfunction is and if there's single or dual fiber/interface failure. Control messages carry information about network malfunction. The remaining fiber/interface is used to transmit/receive control messages, but is not used to transmit/receive data.

Implementing of iSteering is particularly justified for transmitter or receiver module breakdown. It's rather less possible for single ring to fail especially it's mostly placed together with other fibers. But failure of single transmitter or receiver module is much more probable (for example – transmitter's LED failure) – and here iSteering can be really useful. It's still possible to receive data although node is unable to transmit data through the other fiber. It requires to notify downstream node that node is unable to send keep-alive packets, but it still can receive data.

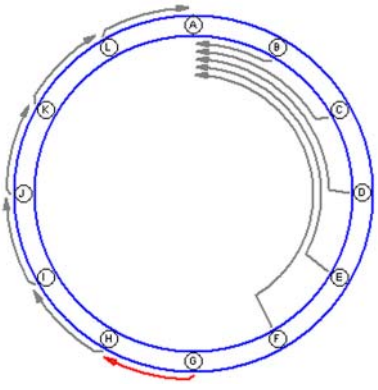


Fig. 3. Traffic flow before malfunction

3 Examples

We have carried a short analysis of two examples of traffic flow - worst-case and network with central node. Following assumptions were made: (1) buffers of nodes are unlimited (2) every node wants to transmit data with maximal transmit speed (full line rate) (3) for every transmission shortest path is always chosen (minimal hop number) (4) to simplify example one node can transmit only once (this is necessary due to clarity of pictures)

3.1 Example 1 – Worst Case

Let's consider network composed of 12 nodes and traffic flow as on Fig.3.:

Node A is a central node and nodes B, C, D, E and F want to transmit data directly to node A. Nodes G, H, I, J, K and L transmit data only to its adjacent neighbors. Table 1 shows bandwidth utilization by nodes.

Let's explain two first rows of Table 1. Node A does not transmit data, so its bandwidth utilization is 0%. Node B has to share bandwidth with 4 nodes (C, D, E, F), so bandwidth utilization is 100/5 % = 20%. Total bandwidth, which can be defined as total throughput of a network is $B_{TOTAL} = 700\%$.

Now let's deliberate bandwidth utilization in case of fiber failure between nodes A and B when packet wrapping, packet steering and iSteering are used.

1. *packet wrapping*
   Applying of packet wrapping is shown on Fig.4a. Packet from nodes C, D, E, F traverse through inner ring as there is no failure and packets are wrapped by node B. Bandwidth has to be divided among nodes B, C, D, E, F and additionally one of G, H, I, J, K, L. This is very ineffective and causes serious bandwidth utilization decrease(Table 1), hence:

$$B_{TOTAL(wrapping)} = \frac{1100}{6}\% = 183,3\% \tag{1}$$

Table 1. Bandwidth utilization by nodes

node	bandwidth utilization by node [%] **before malfunction**	bandwidth utilization by node [%] **with packet wrapping**
A	0	0
B	20	100/6
C	20	100/6
D	20	100/6
E	20	100/6
F	20	100/6
G	100	100/6
H	100	100/6
I	100	100/6
J	100	100/6
K	100	100/6
L	100	100/6

and remaining bandwidth[1]:

$$B_{REMAINING} = \frac{B_{TOTAL(wrapping)}}{B_{TOTAL}} \cong 26\% \tag{2}$$

2. *packet steering*
   Although packet steering mostly works better than packet wrapping [4] that in worst-case results are the same (see Fig.4b.).
   Packets traverse shorter way than in packet wrapping so latency can be smaller (except for node B) but remaining bandwidth is the same as in packet wrapping.
3. *iSteering*
   Utilization of iSteering has to be divided into two examples: inner and outer fiber failure

a. *inner fiber failure*
   In this case traffic flows as depicted on Fig.4c. Packets are send in the same way as before (data can be transmitted only in one direction in one ringlet) – so remaining bandwidth is the same as previously.

$$B_{REMAINING} = \frac{B_{TOTAL(iSteering-inner\_failure)}}{B_{TOTAL}} \cong 26\% \tag{3}$$

b. *outer fiber failure*
   In this case iSteering shows its potential (Fig.4d.) – packets can be send through inner ring and network malfunction does not influence bandwidth utilization:

$$B_{REMAINING} = \frac{B_{TOTAL(iSteering-outer\_failure)}}{B_{TOTAL}} = 100\% \tag{4}$$

[1] Remaining bandwidth is a ratio of total bandwidth after failure and total bandwidth before failure.

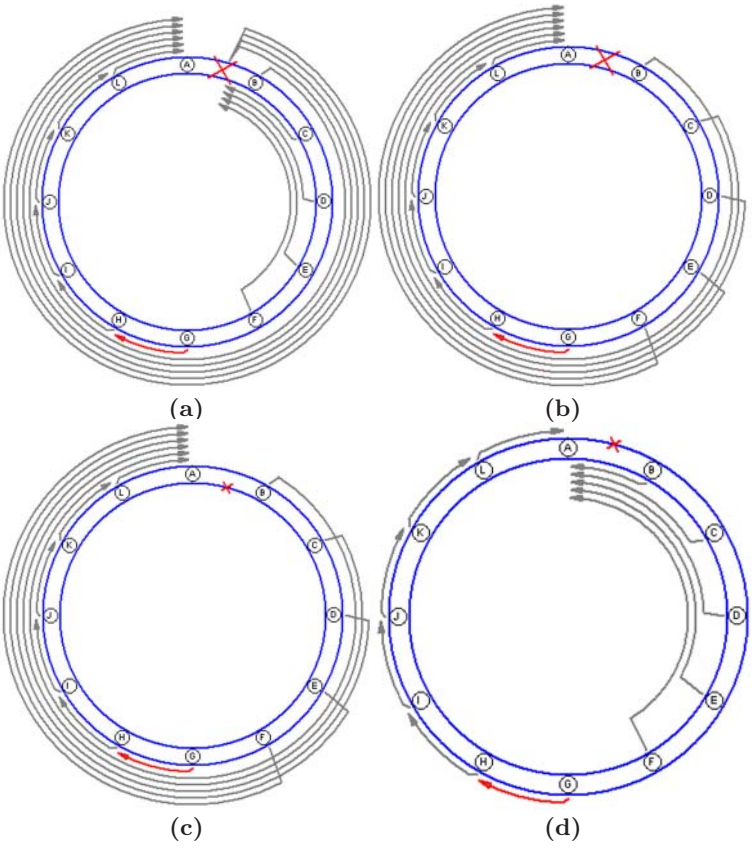


Fig. 4. Protection mechanisms – example 1 – traffic flow (a) packet wrapping (b) packet steering (c) iSteering – inner ring failure (d) iSteering – outer ring failure

According to [5], for both wrapping and steering:

$$B_{REMAINING} = \frac{4(N-1)}{(N+1)^2} \tag{5}$$

N – total number of nodes. Network build of 128 nodes in worst-case traffic flow can utilize only 3% of bandwidth with packet wrapping/steering, whereas it can work like there's no malfunction with iSteering.

3.2 Example 2 – Central Node

Let's consider network with 7 nodes (Fig.5.). Traffic is directed from all nodes to the central node (A) (for example – it could be FTP server or data base server). hence $B_{TOTAL} = 200\%$.

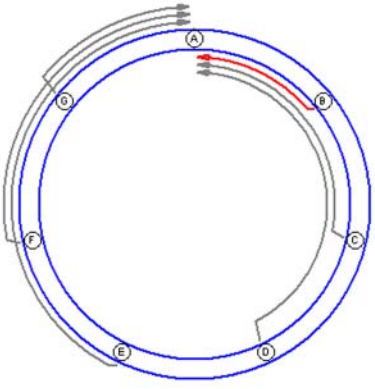


Fig. 5. Traffic flow before malfunction

Table 2. Bandwidth utilization by nodes

node	bandwidth utilization by node [%] **before malfunction**	bandwidth utilization by node [%] **with packet wrapping**
A	0	0
B	100/3	100/6
C	100/3	100/6
D	100/3	100/6
E	100/3	100/6
F	100/3	100/6
G	100/3	100/6

1. *packet wrapping/steering*
   Let's consider network malfunction between nodes A and B. Utilization of packet wrapping means that we lose half of the total bandwidth (Fig.6a.): hence:
   $$B_{TOTAL(wrapping)} = \frac{600}{6}\% = 100\% \tag{6}$$
   and remaining bandwidth:
   $$B_{REMAINING} = \frac{B_{TOTAL(wrapping)}}{B_{TOTAL}} = 50\% \tag{7}$$
   Usage of packet steering (Fig.6b.) results in the same remaining bandwidth. Loss of bandwidth is constant and does not change with number of nodes in this example.
2. *iSteering*
   Again, we divide this situation into two examples: inner and outer fiber failure.

a. *inner fiber failure*
   In this case traffic flows as depicted on Fig.6c. Packets are send in the same

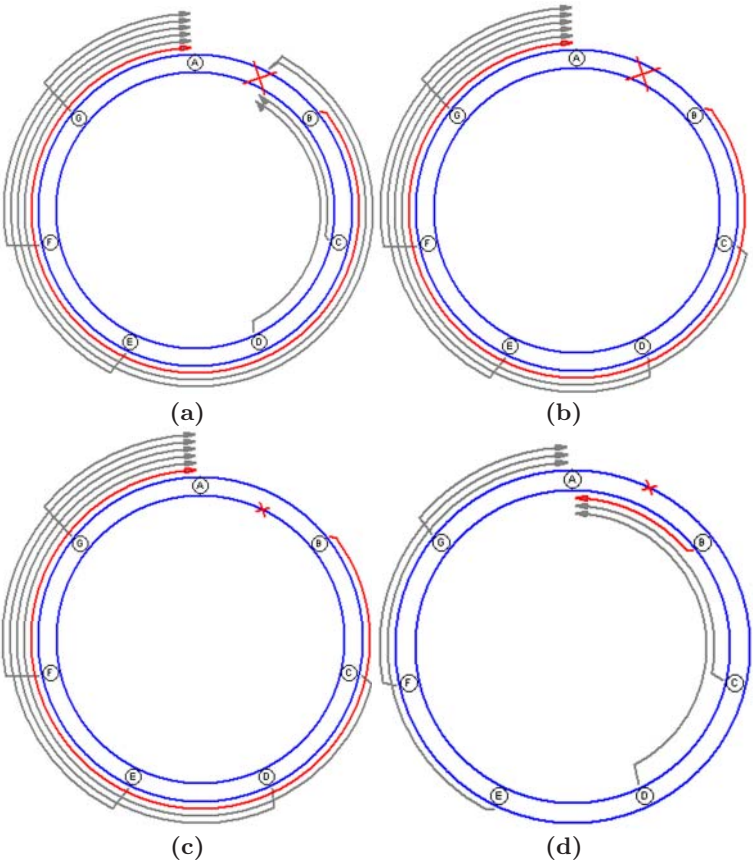


Fig. 6. Protection mechanisms – example 2 (a) traffic flow – packet wrapping (b) packet steering (c) iSteering – inner ring failure (d) iSteering – outer ring failure

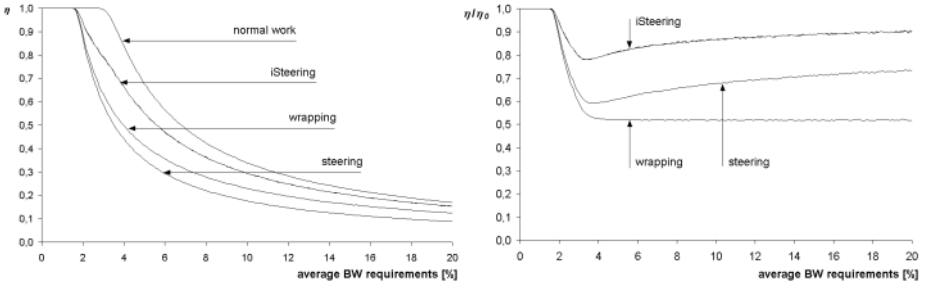
way as before (packet wrapping/steering) – so remaining bandwidth is the same as previously.

$$B_{REMAINING} = \frac{B_{TOTAL(iSteering-inner\_failure)}}{B_{TOTAL}} = 50\% \tag{8}$$

b. *outer fiber failure*
In this case packets can be send through inner ring (Fig.6d.) and network malfunction does not influence remaining bandwidth:

$$B_{REMAINING} = \frac{B_{TOTAL(iSteering-outer\_failure)}}{B_{TOTAL}} = 100\% \tag{9}$$

Utilization of iSteering in case of outer fiber failure allows us to transfer data as there is no network malfunction.



(a) **(b)**

Fig. 7. (a) Efficiency of protection switching mechanisms for random traffic flow (b) Relative efficiency of protection switching mechanisms for random traffic flow

4 Numerical Analysis

In order to compare packet wrapping, steering and iSteering numerical methods were used. We have developed a useful tool for quick calculating of bandwidth utilization in normal work and in case of malfunction. We have calculated bandwidth utilization for even and random traffic flow. For random traffic flow every node draws its bandwidth (BW) requirements for transmissions to all remaining nodes. Drawing was made by a random generator with average value z and for this average value traffic was drawn multiple times. Mean was then computed. Fig.7. illustrates network efficiency of network with 15 nodes for even traffic flow and random traffic flow if network efficiency is defined as:

$$\eta = \frac{\sum_{m \in N} \sum_{r \in \{0,1\}} s(m,r)}{\sum_{m \in N} \sum_{r \in \{0,1\}} req(m,r)} \tag{10}$$

s(m,r) – transmission speed of interface "r" in node number "m" (every node has 2 interfaces)
req(m,r) – bandwidth requirements of interface "r" in node number "m" N – total number of nodes

Results for even traffic flow are similar – the charts look very much alike so we presented only random traffic work. Fig.7a. shows efficiency of protection switching mechanisms (packet wrapping, steering and iSteering) for random traffic flow compared with normal work. Fig.7b. shows relative (to normal work) efficiency of protection switching mechanisms for random traffic flow.

There is value of bandwidth requirements for which both normal work and protection switching mechanisms (packet wrapping, packet steering and iSteering) results in serious efficiency decrease. This happens if bandwidth requirements exceed network transmission abilities.

As you can see generally iSteering is more efficient than packet steering, whereas packet steering is more efficient than packet wrapping.

5 Conclusions

We have introduced a new protection method, which can be used in case of single fiber or single transmitter/receiver failure. If single fiber fails then iSteering can greatly improve bandwidth utilization. We have proved it's significance in two examples (worst-case and central node) and in even/random traffic flow. iSteering shows it's excellent efficiency and it's quite simple to realize.

Notice that we have carried only numerical analysis and it isn't so accurate as complete RPR network simulation.

In the following work, we would like to implement iSteering on RPR simulator and compare the results with our expectations.

References

[1] Tsiang D., Suwala G.: The Cisco SRP MAC Layer Protocol, IETF Networking Group, RFC 2892 (2000) 669
[2] Gandalf Proposal for IEEE Standard 802.17, Draft 0.4 (2001) 669
[3] Nowicki K., Wozniak J.: Przewodowe i bezprzewodowe sieci LAN, OWPW, Warszawa (2002) 671
[4] A Summary and Overview of the IEEE 802.17 Resilent Packet Ring Standard, v.2.2, IEEE 802.17 Study Group (2003) 671, 673
[5] Jajszczyk A., Szymanski A, Wajda K.: Issues of DPT efficiency in various traffic conditions, Proc. 8th Polish Teletraffic Symposium, Zakopane (2001) 674

An Efficient Video Prefix-Caching Scheme in Wide Area Networks

Hyotaek Lim[1,⋆], DaeHun Nyang[2], and David H.C. Du[3]

[1] Department of Computer Engineering, Dongseo University
Busan, 617-716, Korea
htlim@dongseo.ac.kr
[2] Graduate School of Information Technology and Telecommunication
Inha University, Inchon, 402-751 Korea
nyang@inha.ac.kr
[3] Department of Computer Science and Engineering, University of Minnesota
Minneapolis, MN 55455, U.S.A
du@cs.umn.edu

Abstract. Web proxy caching provides an effective way to reduce access latency and bandwidth requirement. In particular, prefix caching is considered as an alternative for improving video delivery over wide area networks because video objects are usually too large to be cached in their entirety. Many studies have pointed that the user-perceived latency is often not dominated by object transmission time, but rather by setup process such as TCP connection time that precedes it. We propose *pre-connecting* techniques which hide the setup process such as TCP connection time in prefix caching proxy and show that the pre-connection can be used efficiently in TCP splicing. Our analysis shows the pre-connection significantly reduces start-up latency and TCP connection time in simple analytical model and hierarchical caching model, respectively. The deployment of the proposed pre-connection does not require protocol modification or the cooperation of other entities.

1 Introduction

As the web continues to grow exponentially, streaming media delivery is gaining popularity as indicated by dramatically increased deployment of commercial products for playback of stored video and audio over the Internet [1]. However, the service of quality as perceived by the end user is still very poor because the support is lacking in the Internet to meet delay and jitter requirements for realtime traffic. In particular, start-up latency is the central performance problem of streaming media delivery in the Internet today.

Caching of web objects for improving end-to-end latency and reducing network load have been studied extensively starting with CERN httpd, followed by improvements in hierarchical caching and co-operative caching in the Harvest

⋆ The first author was partially supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2002.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 679–688, 2004.

project and the Squid project, respectively [9]. Recently, some works have been presented on proxy caching architectures for improving video delivery over wide area networks [9, 10, 11]. Video objects are usually too large to be cached in their entirety. Video files in size are usually much larger than other web documents. HTML documents and embedded images are usually in the range of 1Kbyte to 10Kbyte, however, video data such as mpeg, AVI, QuickTime typically exceed 1Mbyte [10]. The large size of video data makes conventional proxy caching techniques inappropriate.

As an alternative for video caching proxy, a prefix-caching proxy can be used efficiently in case of the large streams such as video files. A prefix-caching proxy caches the initial portion of the stream at proxy and transmits the prefix from the proxy to the client. During the transmission of the prefix to client, the proxy requests the remainder from the server. Fortunately, most of streaming protocols, such as HTTP/1.1 and RTSP, allow the proxy to fetch the remainder of the stream by random access.

A key realization is that streaming latency is often not dominated by object transmission time, but rather by the setup process that precedes it [7]. The dominating latency factors of the user perceived latency include name-to-address resolution, TCP connection establishment time, HTTP request-response time, server processing and finally transmission time. Information exchange between a client and a proxy, a proxy and a server in the Web is performed using HTTP, which typically uses TCP as the underlying transport protocol. Hence HTTP request and response messages are carried on a TCP connection established between the two entities. When the requested object is of small size, it is often the case that only two round trips are required to fetch the object: one for connection-establishment and the other for the object transfer itself.

This paper proposes a pre-connecting scheme to hide TCP connection establishment time, which is significant relative to HTTP request-response times when the prefix-caching proxy requests the remainder to the server [11, 10]. It allows the proxy to save time to support additional functions such as transcoding. We also show that the proposed protocols can be used efficiently in TCP splicing. The analytical results exhibit that the proposed scheme substantially reduces the start-up latency in a simple model and the connection time in hierarchical caching model.

The remainder of this paper is organized as follows. Section 2 proposes the pre-connection for prefix-caching video proxy and describes the results of applying the pre-connection to TCP splicing. Section 3 analyzes the proposed pre-connection. The analysis is concentrated on start-up delay and connection time in simple analytical model. Finally, Section 4 summarizes and concludes.

2 Pre-connection for Prefix-Caching

2.1 Connection for Prefix-Caching

Web clients and servers use HTTP which in turn uses TCP as its underlying reliable transport protocol. A TCP connection needs to be established and

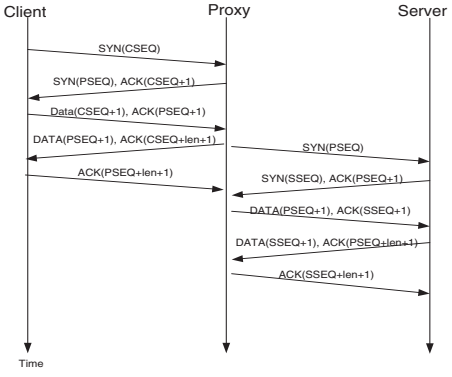


Fig. 1. Protocol Sequences in Normal Case

acknowledged prior to transporting HTTP messages. Fig. 1 illustrates the protocol timeline of TCP segments in conventional prefix caching when the proxy has cached the prefix of the video data. TCP connections are established with a 3-way handshaking: The client sends a SYN segment to the proxy or server, receiving the proxy or server's SYN segment with the ACK flag on , and then acknowledging the server's SYN. We assumed that a data segment(HTTP GET request) is sent on the 3rd segment of an initial 3-way handshaking to simplify the timing sequence. The 4rd segment contains the HTTP response to send the prefix of an object to client. CSEQ, PSEQ and SSEQ in the figure indicate client, proxy and server sequence number which is included in TCP segment, respectively. In most systems, the process to send the prefix to client and the process to request the remainder of a stream from the server are started concurrently. Nevertheless, as we can see in Fig.1, the proxy has to wait the TCP connection time before starting to fetch the remainder of a video stream from the server by invoking the range request specified in HTTP/1.1 [2, 13]. In real systems, the round-trip time for TCP connection between the proxy and the server may not be negligible because the proxy and server exist in WAN such as Internet. The round-trip time in hierarchical caching in which caches are placed at multiple levels of the networks degrades the quality of service significantly.

So, we propose the protocol sequence to hide the round-trip time for TCP connection as shown in Fig.2. It is possible for the prefix-caching proxy to pipeline the handshake by sending out the SYN segment to the server immediately after receiving the SYN segment from the client. The initial SYN segment contains the destination IP address and port number. The proxy has all of the necessary information for TCP connection setup. In the prefix-caching proxy, the sequence described in Fig.2 allows the proxy to have the time to support additional functions such as transcoding. This scheme is transparent to the client and server and follows a standard protocol. As the number of caching proxies increases, it significantly reduces the round-trip time for TCP connection.

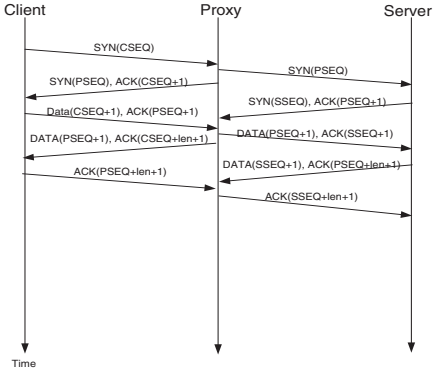


Fig. 2. Protocol Sequences in Proposed Case

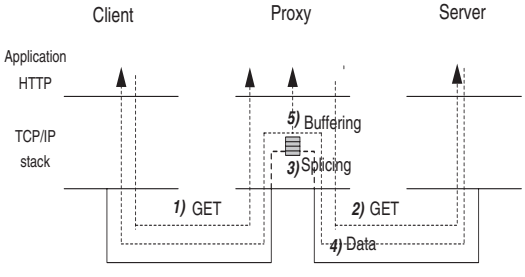


Fig. 3. Steps to process HTTP GET request in TCP-Splicing

2.2 Pre-connection in TCP-Splicing

The pre-connection proposed in the previous Section can be used in TCP splicing. TCP splicing is a technique to splice together two TCP connections that were independently set up with a proxy [3, 4, 12]. In conventional caching systems, received packets in the proxy are passed up through the protocol stack to application layer, where they are then passed back down again in order to be sent out to the requesting client. After the splice is created, however, these two connections should be one connection actually as shown in Fig. 3. The TCP splicing can improve the performance of the video prefix-caching proxy with the proposed protocols.

Fig. 3 shows all steps for the proposed prefix-caching proxy to process a cache miss in TCP splicing. Assuming that the prefix of the requested data is not in the proxy cache, the steps of the TCP splicing including the proposed protocols are as follows: (1) Proxy receives a GET from the client after TCP connection establishment, (2) The proxy sends a GET request immediately after receiving the 2nd segment for TCP connection from the server because the requested data is not in the cache, (3) The proxy splices the two connections, (4) The data

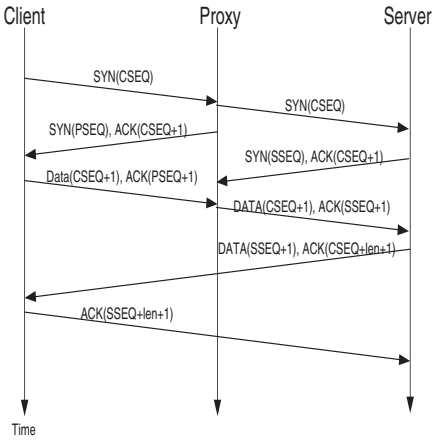


Fig. 4. Protocol Sequence in TCP-Splicing

flows through the splice from server to client, (5) Proxy application reads the data from tap buffer and spools into the proxy cache. With TCP splicing, the requested data can be sent from the server to the client without going up the application layer. During sending the data to client, at the same time, the proxy can cache the data from the server using an additional tap buffer in the proxy. The proposed pre-connection is used efficiently in the case of a cache miss or a system initialization.

Fig. 4 illustrates the TCP sequence which sends the data from the server to the client using the TCP splicing in the case of a cache miss. The sequence shows that the proxy sends out the SYN segment to the server immediately after receiving the SYN segment from the client as in the previous section. The proposed pre-connecting technique improves the performance of the TCP splicing by hiding the TCP connection time between the proxy and the server.

3 Analysis of Pre-connecting Prefix Caching

3.1 Analysis of Start-Up Delay

To analyze client start-up latency, we use a simple model as shown in Fig. 5. The model involves a caching proxy between the client and the server [6]. Client start-up latency can be defined as the time difference between sending a request and starting to play the media object at the client. We assume that the server send out data packet according to its playback rate r bytes/second, and each client keeps a buffer of K seconds. The client does not start playing the object until its buffer is filled. Also, we assume the delay between the server and the proxy is d_1, and between the proxy and the client is d_2. In Fig. 5, without a proxy the start-up latency, L_0 is $4(d_1 + d_2) + K$, where rational for the 4 is due to the

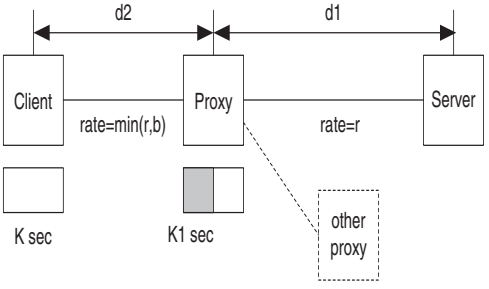


Fig. 5. Analysis Model

three-way handshaking of TCP connection and the delay to deliver an object. We assumed a HTTP request is sent on the 3rd segment to simplify the timing delay. Assuming the proxy has K_1 seconds of data in its cache, let's consider the start-up latency. At first, it takes $3d_2$ msec for the client request to arrive at the proxy because we assumed that a HTTP GET request is sent on the 3rd segment of the initial three-way handshaking. And then, the proxy starts two processes concurrently.

One process is to download the existing K_1 seconds of data to the client. Assuming the rate between the client and the proxy is b bytes/second, it takes $(K_1 \cdot r)/b$ seconds. The other process is to request $(K - K_1)$ seconds of data from other sources such as other proxy or the server. This process takes $4d_1$ because of the three-way handshaking of TCP connection. However, the proposed pre-connection takes $2d_1$ assuming that d_1 is equal to d_2 to simplify timing problem. It is because the proposed pre-connection hides the TCP connection time between the proxy and the server somewhat. Thus the time for both processes to finish is $max(K_1 \cdot r/b, 4d_1)$ and $max(K_1 \cdot r/b, 2d_1)$ in normal and proposed prefix caching, respectively. And then, the time for the remaining part of data to arrive at the client is $d_2 + (K - K_1) \cdot r/min(r, b)$. During this step, the buffer at the proxy is filled with rate r and drained with rate b. In order to avoid buffer underflow the actual draining rate for the buffer is set to $min(r, b)$.

The resulting start-up latency, L_1 is $4d_2 + max(K_1 \cdot r/b, 4d_1) + (K - K_1) \cdot r/min(r, b)$ and $4d_2 + max(K_1 \cdot r/b, 2d_1) + (K - K_1) \cdot r/min(r, b)$ in normal and proposed case, respectively. Fig. 6 shows the start-up latency for K_1 and b/r in normal case. As shown in the figure, the start-up latency is decreases as K_1 increases, and the start-up latency is decreases as b/r increases. Fig. 7 compares the start-up latency of normal and pre-connecting prefix caching. The figure shows that the start-up latency in proposed prefix caching indicates lower latency than in normal prefix caching as b/r increases.

3.2 Connection Time in Hierarchical Caching

With hierarchical caching, caches are placed at multiple levels of the network. We shall make the reasonable assumption that the cache levels consist of three

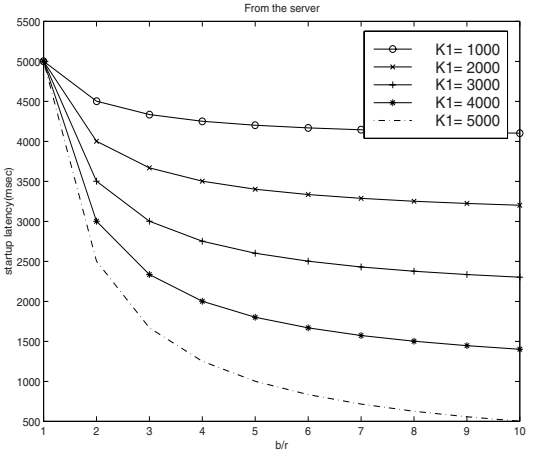


Fig. 6. Start-up Delay from Server

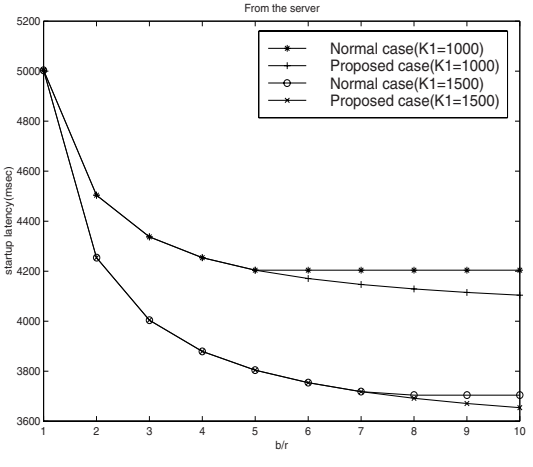


Fig. 7. Comparison of Normal and Pre-connecting Prefix Caching

levels of institutional, regional, national caches [8]. All the clients are connected to the institutional networks which contain their caches. When a request is not satisfied by the client cache, the request is redirected to the institutional cache. If the object is not found at the institutional level, the request is then forwarded to the regional level cache which in turn forwards unsatisfied requests to the national level cache. If the object is not found at any cache level, the national level cache contacts directly the original server, it travels down the hierarchy, leaving a copy at each of the intermediate caches along its path.

We model the network topology as a full O-ary tree, as shown in Fig. 8 [5]. The following notation is used in the model.

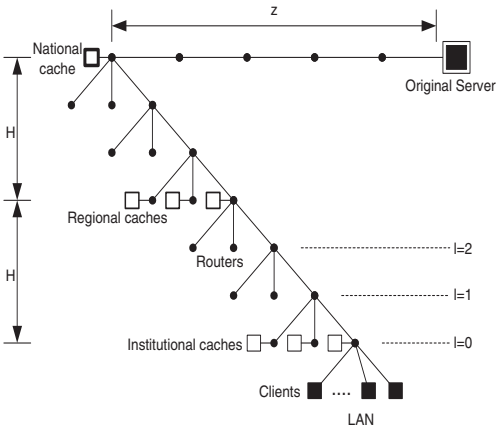


Fig. 8. Caching Hierarchy

- O: nodal outdegree of the tree
- H: number of network link between the root node of a national network and that of a regional network
- z: number of links between a origin server and root node
- l: level of the tree, $0 \leq l \leq 2H + z$
- d: per-hop propagation delay
- L: number of links that a request travels to hit a object in the caching hierarchy
- T_c: Connection time

Actually, the connection time depends on the number of links from the client to the cache containing the desired object. The connection time in normal caching can be formulated as follows [5]:

$$E[T_c] = 4d \sum_{l=\{0,H,2H,2H+z\}} P(L = l)(l + 1) \tag{1}$$

In the above equation, the factor of 4d, which is due to the three-way handshaking of TCP connection, is the dominant factor of the overall connection time. The proposed pre-connection as described in the previous Section can be used efficiently in hierarchical caching as well. It reduces the round-trip delay of the TCP connection time by connection hiding or pipelining. As the level of the hierarchical caching increases, the proposed pre-connection reduces the TCP connection time much more.

The connection time in the proposed prefix caching is given by

$$E[T_c^*] = 4d \cdot P(L = 0) + 2d \sum_{l} P(L = l)(l + 1) \tag{2}$$

where $l = \{H, 2H, 2H + z\}$.

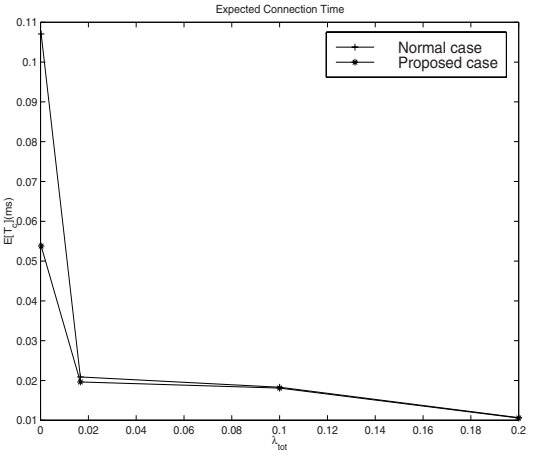


Fig. 9. Expected Connection Time

The first part of the above equation is the connection time to request a prefix in the institutional cache and the second part is the connection time to request the remainder in the caches of higher levels than institutional cache. The factor of $2d$ is due to pre-connecting for TCP connection. To calculate $P(L = l)$ we use $P(L = l) = P(L \geq l) - P(L \geq l + 1)$ as described in [5].

Fig. 9 shows the connection time of the normal and proposed case for different object's popularity. Assuming that requests for document i are uniformly distributed between all O^{2H} institutional caches, there are $\lambda_{tot} = \lambda_{I,i} \cdot O^{2H}$ total requests for document i. We observe that for unpopular object(small λ_{tot}) both normal and proposed case experience high connection times because the request needs to travel to the origin server. As the number of requests for an object increases(large λ_{tot}), the average connection time decreases since there is a higher probability to hit an object at closer caches than the origin server. For all the objects which λ_{tot} is smaller than 0.1, the proposed prefix caching gives shorter connection times than the normal prefix caching.

4 Conclusion

Prefix caching offers an effective way for improving video delivery over wide area networks. The prefix caching can be supported by the range request specified in HTTP/1.1 and RTSP which allows the proxy to fetch the remainder or suffix of the stream by random access. However, many studies have pointed that the user-perceived latency is often not dominated by object transmission time, but rather by setup process such as TCP connection time that precedes it. We have proposed a simple pre-connecting technique to hide the TCP connection time in prefix caching proxy and showed that the proposed technique can be used efficiently in TCP splicing. Our analysis has also showed the pre-connection

significantly reduces start-up latency and TCP connection time in simple model and hierarchical caching model in which caches are placed at multiple levels of the networks, respectively. The deployment of the proposed pre-connection does not require protocol modification or the cooperation of other entities. As part of our ongoing part, we are pursuing a good design and implementation of the video prefix-caching proxy system including the proposed technique. In addition, we are investigating how to combine the proposed technique with connection caching to reduce the latency of servicing user request.

References

[1] S. Gruber, J. Rexford, A. Basso, "Protocol considerations for a prefix-caching proxy for multimedia streams," *Computer Networks*, Vol. 33, 2000, pp.657-668. 679

[2] J. Jung, D. Lee, K. Chon, "Proactive web caching with cumulative prefetching for large multimedia data," *Computer Networks*, Vol. 33, 2000, pp.645-655. 681

[3] D. A. Maltz, P. Bhagwat, "Improving HTTP caching proxy performance with TCP tap", *Proc. of the Fourth International Workshop on High Performance Protocol Architectures (HIPPARCH'98)*, June 1998, pp.98-103. 682

[4] G. Apostolopoulos, D. Aubespin, V. Peris, P. Pradhan, D. Saha, "Design, implementation and performance of a content-based switch," *Proc. of the IEEE Infocom*, Mar. 2000. 682

[5] P. Rodriguez, C. Spanner, E. W. Biersack, "Web Caching Architectures: Hierarchical and Distributed Caching", *Proc. of the 4th International Caching Workshop*, San Diego, California. Apr, 1999. 685, 686, 687

[6] E. Bommaiah, K. Guo, M. Hofmann, S. Paul, "Design and Implementation of a Caching System for Streaming Media over the Internet", *IEEE Real-Time Technology and Applications Symposium (RTAS)*, Washington D. C., USA, May 31-June 2, 2000. 683

[7] E. Cohen, H. Kaplan, "Prefetching the Means for Document Transfer: A New Approach for Reducing Web Latency", *Proc. of the IEEE Infocom*, Mar. 2000. 680

[8] Jia Wang, "A Survey of Web Caching Schemes for the Internet", *ACM CCR* Vol. 29, Nov. 1999. 685

[9] Markus Hofmann, T. S. Eugene Ng, Katherine Guo, "Caching Techniques for Streaming Multimedia over the Internet", *Bell Labs Technical Memorandum*, April, 1999. 680

[10] Soam Acharya, "Techniques for Improving Multimedia Communication Over Wide Area Networks", Ph.D. thesis, Cornell University, Dept. of Computer Science, 1999. 680

[11] S. Sen, J. Rexford, and D. Towsley, "Proxy Prefix Caching for Multimedia Streams", *Proc. of the IEEE Infocom*, Mar. 1999. 680

[12] Cohen, A., S. Rangarajan, and H. Slye. On the Performance of TCP Splicing for URL-aware Redirection. *Proc.of the USENIX Symposium on Internet Technologies and Systems*, pp. 117-125, October 1999. 682

[13] Hyotaek Lim, David H. C. Du, "Protocol Considerations for Video Prefix-Caching Proxy in Wide Area Networks", To appear, *IEE Electronics Letters*, 2001. 681

A Hierarchical LSP Management Architecture for MPLS Traffic Engineering*

Daniel Won-Kyu Hong[1], Choong Seon Hong[2], and Dongsik Yun[1]

[1] Operations Support System Lab., R&D Group, KT
463-1 Jeonmin-Dong Yuseong-Gu, Daejeon 305-811 KOREA
{wkhong,dsyun}@kt.co.kr
[2] School of Electronics and Information, Kyung Hee Univerity
1 Seocheon Giheung Yongin, Gyeonggi 449-701 KOREA
cshong@khu.ac.kr

Abstract. In this paper, we propose a scalable Label Switched Path (LSP) management architecture for Multiprotocol Label Switching (MPLS) traffic engineering using a hierarchical network model. MPLS introduces the concept of a label hierarchy to support scalability by lessening the complexity of transit routers for the computation of complete routing tables. However, this hierarchical network model creates a critical disadvantage in providing a globally-optimal route because of topology abstraction or aggregation. This paper proposes a hierarchical routing scheme that can provide a globally-optimal route in hierarchical MPLS networks using propagation from the Condensed Subordinate Route Information (CSRI). CSRI is summarized route information among border nodes in the lower layer network and is reflected in the process of the LSP path computation in the higher layer network. We also propose the algorithms for generation of CSRI, for reflection CSRI to higher layer network topology, and for computation of an optimal LSP in a higher layer.

1 Introduction

The Internet is becoming an ideal platform to support modern communications including voice, data, and multimedia transmissions. However, standard IP routing protocols were developed on the basis of a connectionless model with routing decisions based on simple metrics, such as delay or hop count, which lead to the selection of shortest path routes [1,2,3]. Despite its ability to scale to very large networks, this approach provides only basic Quality of Service (QoS) capabilities, which are unable to provide scalable-service level agreements for bandwidth intensive applications in modern networks. Multi-protocol Label Switching (MPLS) extends IP destination-based routing protocols to provide new and scalable routing capabilities.

MPLS traffic engineering is inherently uses explicitly-routed paths. LSPs are created independently, specifying different paths based on user-defined policies;

* This research was supported by ITRC Project of MIC.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 689–698, 2004.

however, this may require extensive operator intervention. RSVP and CR-LDP are two possible approaches to supply dynamic traffic engineering and QoS in MPLS [4,5,6]. Conversely, MPLS has been introduced into LSP hierarchy. LSP hierarchy is the idea that LSPs can be nested inside other LSPs, giving rise to an LSP hierarchy. This is achieved by considering an LSP as a link in the IS-IS or OSPF link state database [6], which is termed a layering concept in this paper. LSP hierarchy has been introduced to enhance scalability by reducing the complexity of transit routers from the composition of all network routing tables. Alternatively, a network can be logically or physically partitioned according to the service provider (SP) domain and the administrative domain within the same service provider, such as access and backbone networks, which will be defined as a partitioning concept in this paper. Layering and partitioning concepts are very useful when deploying a large-scaled network. However, these concepts bring a disadvantage in that they cannot provide a globally-optimal route in a hierarchical network model because of network topology abstraction and aggregation [8,9,10,11,12]. Consequently, we need a new scheme for guaranteeing the provisioning of a globally-optimal route in a hierarchical network model: the major goal of MPLS traffic engineering.

This paper proposes a scalable LSP management architecture that can provide a globally-optimal route in the hierarchical MPLS network. We will define the hierarchical network model applicable for a MPLS network. In addition, we will define the LSP management framework based on proposing a hierarchical network model for provisioning optimal LSPs in terms of MPLS traffic engineering. The paper is organized as follows: In section 2, we identify problems of the hierarchical model and define the hierarchical network model applicable for MPLS traffic engineering. In section 3, the LSP management framework for MPLS traffic engineering is discussed in detail. In section 4, we discuss performance issues of the proposed hierarchical LSP management framework. Finally, resulting conclusions are presented.

2 A Hierarchical Network Model

In this section, we discuss the generic hierarchical network model and identify problems of MPLS traffic engineering in a hierarchical network. In addition, we propose a new hierarchical network model that can solve the problems associated with layering and partitioning concepts.

2.1 Generic Hierarchical Network Model

A hierarchical network model is a traditional solution to the scaling problem [8]. The layer network (*LN*) represents a network boundary that can transfer certain kinds of network traffic without adaptation. Typical LNs can be IP, ATM, SDH, MPLS, et al. There can be client/server relationships among different LNs. Layer networks (*LNs*) are organized into different interconnected sub-networks (*SN*) called domains. An *SN* consists of multiple interconnected network nodes such

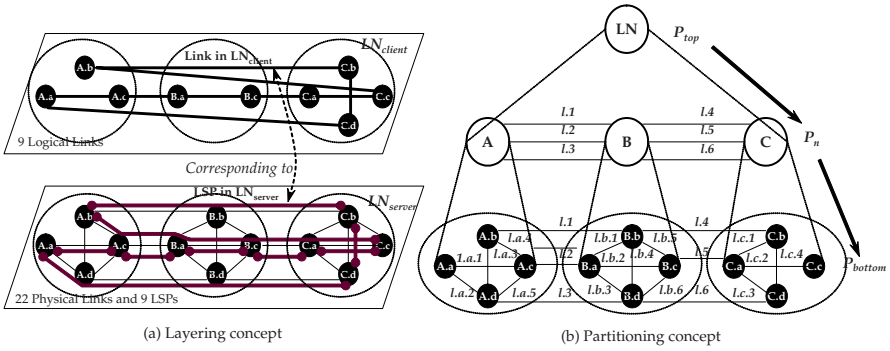


Fig. 1. Layering and Partitioning Concepts

as IP routers, ATM switches, MPLS node, etc. An *SN* can be further partitioned until the lastly partitioned *SN* corresponds to a network node.

Fig. 1 shows layering and partitioning concepts. We apply the layering concept to the LSP hierarchy. As shown in Fig. 1 (a), the server layer network (LN_{server}) consists of 12 MPLS nodes, 22 physical links (Lp) and 9 LSPs. In LN_{server}, there are two kinds of LSPs: one is the end-to-end LSP (LSP_e) that can directly deliver customer MPLS packets, the other is the trunk LSP (LSP_t) that contains a number of subordinate LSPs. Fig. 1 (a) shows only LSP_ts. The LSP_ts correspond to the logical link (Ll) in the client layer network(LN_{client}). There is a *"corresponding to"* relationship between LSP_t in LN_{server} and Ll in LN_{client}.

An LN_{client} consists of a number of interconnected MPLS nodes that terminate the LSP_t at LN_{server}, and a number of Lls corresponding to the LSP_ts at LN_{server}. An LN_{client} is purely logical, whereas an LN_{server} is purely physical. Fig. 1 (b) shows the partitioning concept. An LN can be partitioned according to the different service provider domains or the different administrative domains within the same service provider domain. There may also be N-partitioning levels. The top partitioning level (P_{top}) represents the LN itself. The lowest partitioning level (P_{bottom}) corresponds to the entire network topology of an LN. There can be a number of subsequent partitioning levels between P_{top} and P_{bottom}, determined by the administrative policy of each network service provider. In Fig. 1 (b), the LN (P_{top}) is composed of three *SNs (A, B,* and *C)* and six interconnecting links (*l.1-l.6*), which is the first partitioning level. At the second partitioning level, there are three individual *SNs* of *A, B,* and *C*. The *SN A* consists of four nodes *(A.a, A.b, A.c,* and *A.d)* and five links *(l.a.1, l.a.2, l.a.3, l.a.4,* and *l.a.5)*. In addition, all *SNs* have a number of border nodes ($N_{borders}$) with connectivity to other *SNs*, for example: *A.b, A.c,* and *A.d* in the case of *SN A, B.a, B.b, B.c,* and *D.d* in the case of *SN B*, and *C.a, C.b,* and *C.d* in the case of *SN C*.

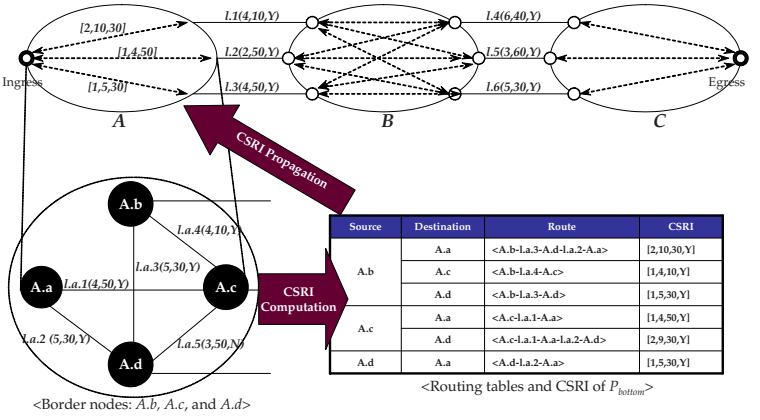
2.2 The Problems of a Hierarchical Network Model in Terms of MPLS Traffic Engineering

The partitioning concept is very useful for deploying a large-scaled MPLS network reflecting the different kinds of administrative policies. However, we cannot provide a globally-optimal route in a hierarchical MPLS network that is deployed based on a partitioning concept. The P_{top} only shows its subordinate network topology. The intermediate partitioning level (P_n) hides P_{bottom} from P_{top}. Therefore, P_{top} finds route with P_n. In addition, P_n finds route with P_{bottom}. For example, if we find a route from $A.a$ to $C.c$ using the shortest-path first routing algorithm, P_{top} selects a route traversing *l.1* and *l.4*, *(A-l1-B-l4-C)*, which can be a reasonable path in terms of P_{top} because the hop counts of all possible routes are the same. Subsequently, each *SN* of *A, B,* and *C* finds an optimal route with their partitioned network topology as shown in Fig. 1 (b). *SN A* selects *(A.a-l.a.2-A.d-l.a.3-A.b)*, *SN B* selects *(B.b)*, and *SN C* selects *(C.b-l.c.3-C.a-l.c.2-C.c)*. As a result of hierarchical routing, the selected route is *(A.a-l.a.2-A.d-l.a.3-A.b-l.1-B.b-l.4-C.b-l.c.3-C.a-l.c.2-C.c)*, which traverses six hops. However, there is a more optimal route than the selected route, which traverses five hops, *(A.a-l.a.1-A.c-l.2-B.a-l.b.s-B.c-l.5-C.a-l.c.2-C.c)*. If P_{top} selects *l.2* and *1.5*, then we can find the most optimal route in a hierarchical environment. In contrast, layering concept gives us an unfavorable side effect whereby the LN_{client} selects an explicit route for LSP configuration with its network topology. The LN_{client} has the highest probability of selecting the Ll , which has a longer transit delay than the others because LN_{client} only has the link attributes of Ll but with the path attributes of LSP_t at LN_{server}.

2.3 A New Hierarchical Network Model

In this paper, we propose a hierarchical network model that highlights the advantages and solves the disadvantages in hierarchical networks. First, we propose the propagation of Condensed Subordinate Route Information (CSRI) to solve problems caused by the partitioning concept. CSRI is a set of summarized routes to be propagated from the subordinate partitioning level to the upper partitioning level.

Each SN computes all possible routes (R) using partitioned network topology. We use the Dijstra algorithm for computation of Rs. There are border nodes (N_b) within the partitioned network topology; for example, $A.b$, $A.c$, and $A.d$ in Fig. 2. If there are n border nodes, the number of SCRI entries can be $n(n-1)/2$. SCRI can be defined as $SCRI = (h, d, b, r)$, where h is hop count, d is accumulated delay, b is available bandwidth, and r is extent or route availability ($r = yes/no$). First, we computed all possible routes between two border nodes: (n_{source}) and ($n_{destination}$). By computing routes between n_{source} and ndestionation, we determined the metrics of R, including hop count ($R(H)$), delay ($R(D)$), bandwidth ($R(B)$), and availability ($R(A)$) as shown in Fig. 3. H is the number of links traversing the computed route R, $R(D)$ is the accumulated delay of links traversing the computed route R, $R(B)$ is the smallest bandwidth



Source	Destination	Route	CSRI
A.b	A.a	<A.b-l.a.3-A.d-l.a.2-A.a>	[2,10,30,Y]
	A.c	<A.b-l.a.4-A.c>	[1,4,10,Y]
	A.d	<A.b-l.a.3-A.d>	[1,5,30,Y]
A.c	A.a	<A.c-l.a.1-A.a>	[1,4,50,Y]
	A.d	<A.c-l.a.1-A.a-l.a.2-A.d>	[2,9,30,Y]
A.d	A.a	<A.d-l.a.2-A.a>	[1,5,30,Y]

Fig. 2. CSRI propagation model

Let N_b is the list of border node within the partitioned network topology
Let $R=\{(n,l)\}$ is the computed route, where n is a node and l is an link.
Let $l=(D,B_{ava},A)$ is the link, where D is a transmission delay of the link, B_{ava} is the available bandwidth of the link, and A is the availability of the link
For $\forall n_{source} \in N_b$ **do** {
  for $\forall n_{destination} \in (N\text{-}n_{source})$ **do** {
    $i = 0$;
    Compute all possible route (R) **between** n_{source} **and** $n_{destination}$ **with the following rules:**
      $R(H)$ **is the number of links traversing the computed** R_i
      $R(D)$ **is the accumulated delay of links traversing the computed** R_i
      $R(B)$ **is the smallest bandwidth of link among the bandwidths of links traversing the computed** R_i
      $R(R)$ **is the reachability of the computed** R_i
    if there is a computed route, then i++;
  }
  // Computation of CSRI
  $CSRI(n_{source},n_{destination})(h,d,b,r) = (min(\sum R(H)), min(\sum R(D)), max(\sum R(B)), \sum R(A) == yes)$;
}

Fig. 3. CSRI computation algorithm

of the link among those traversing the computed route R, and $R(A)$ can be "no" if one of links traversing the computed route R is "no". In order to compute, we used the Dijstra algorithm.

After computing all possible routes between n_{source} and $n_{destination}$, we determined the CSRI metrics according to an administrative policy. There can be one or more possible routes between an identical n_{source} and ndestiation. CSRI metrics can be of the first priority route. Then, how can we select the first priority route among computed routes? In general, we selected aroute having the smallest hop count, smallest accumulated delay, largest available bandwidth, and a accessible route as the first priority route. However, if the service provider wishes to deploy a delay sensitive network, then we must select the route having the least accumulated delay over other metrics as first priority. As we selected the most optimal route among all possible routes between n_{source} and $n_{destination}$, and defined the metrics of the first priority route as CSRI metrics, we dramatically

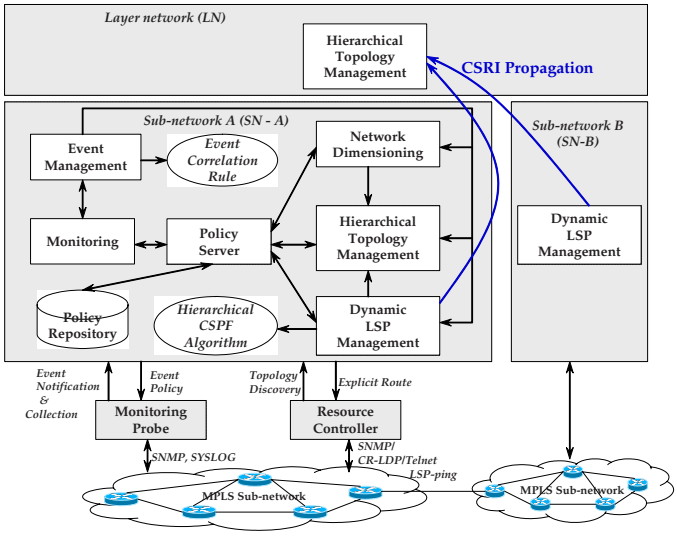


Fig. 4. LSP management architecture

reduced the number of CSRI entries to be propagated to the upper partitioning level. This resulted in topology aggregation or filtering. After selection of CSRI entries, we propagated CSRI to the upper partitioning level as shown in Fig. 2. Propagated CSRI is reflected into the internal topology of the upper partitioning *SN*. Next, we selected the globally-optimal route reflecting CSRI at the N_{top}. If we did not reflect CSRI at N_{top}, the computation of route at N_{top} would result in an unfavorable route. Thus, we provided a globally-optimal route in a hierarchical network model using the concept of CSRI propagation, which makes a network service provider supply traffic-engineered LSP in a hierarchical MPLS network.

3 LSP Management Architecture

In the previous section, we proposed a hierarchical network model to guarantee the provision of globally-optimal traffic engineered LSP in a hierarchical MPLS network. In this section, we propose LSP management architecture from the perspective of service and network management.

There are four broad areas in LSP Management Architectur: network device control, network configuration, event & fault management, and policy management areas. Each sub-network has these four management functions. The device control mediator (DCM) takes the roles of device configuration and monitoring. The resource controller (RC) configures network devices using SNMP, Telnet, and CR-LDP; it also checks the validity of a configured LSP using LSP-Ping.

The monitoring probe (MP) collects myriad events generated from network devices and propagates the events to the event management module.

Policy repository (PR) has rules provisioning LSP under a hierarchical environment. These include rule-based constraints on optimization processes, alarm triggering, and propagation. The network-dimensioning module collects periodic MPLS traffic patterns and analyzes the gathered traffic. According to traffic analysis, the network-dimensioning module configures the hierarchical MPLS network topology as described in section 2.3. It determines the affinity, color, and capacity of all links, reconfigures existing LSPs, and decides protection schemes for each LSP such as 1:1 protection, 1:N protection, N:M protection, or rerouting.

The dynamic LSP management module configures LSPs using the hierarchical CSPF algorithm. This module receives LSP configuration requests from the network-dimensioning module in terms of network planning and the network operator in terms of the on-demand LSP configuration, including MPLS-VPN configuration. In addition, this module maintains numerous provisioned LSPs. In order to support the optimal path provision in a hierarchical MPLS network, the LSP management module calculates CSRI entries and propagates them to the hierarchical topology management module in the upper-layer management system.

The hierarchical topology management module maintains a hierarchical MPLS network topology provisioned and planned by the network-dimensioning module. Network topology is used to calculate an optimal path for LSP provision. In addition, network status is dynamically reflected into the network topology in order to find an optimal path taking into account the current network status. Subsequently, the hierarchical topology management module dynamically changes the status of network topology according to innumerable network events originating in the event-management module. While maintaining hierarchical network topology, the topology management module receives CSRI entries from a LSP management model at the subordinate management system, and reflects received CSRI to the internal cost of sub-networks.

Thus, propagation of CSRI entries between the LSP management model at the subordinate management system and the configuration management model at the upper management system provides a globally-optimal route in a hierarchical network environment.

4 Performance Issues

Our LSP management architecture can provide aglobally-optimal route in hierarchical MPLS network environments. In this paper, we measured the LSP configuration performance in terms of route computation and resource utilization comparing our approach to so-called "CSRI propagation approach" with two other approaches. The two other approaches are as follows: one is to find a route with flat network topology with no hierarchy, a so-called flat topology approach; and the other is to find a route without CSRI propagation in a hierarchical network environment, a so-called "no CSRI propagation approach". Fig. 5

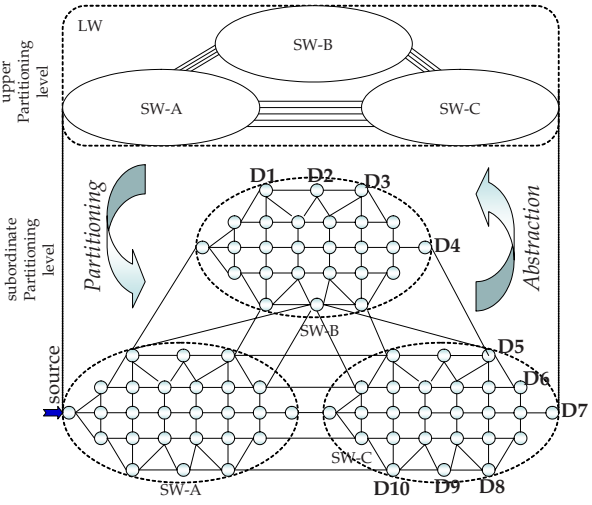


Fig. 5. A simplified test topology

shows a simplified network topology for simulation. There are two partitioning levels. Layer network (LN) consists of three sub-networks: $SW - A$, $SW - B$ and $SW - C$, and ten links connecting the tree sub-networks. In contrast, each sub-network has 26 nodes and 51 links connecting the nodes.

We assume that all links connecting nodes and sub-networks has the same available bandwidth and transit delay. We found a route between the fixed source and ten variant destinations as shown in Fig. 5. We calculated the path at 50 times between the same source and destination. Fig. 6 shows the average performance of 50 attempts. In terms of path computation performance, the "no CSRI propagation approach" shows higher performance than the other two approaches because it finds a route with the highly-abstract network topology of three sub-networks and 13 links. However, our approach showed slight performance degradation compared to the "no CSRI propagation approach". The performance degradation of our approach compared to the "no CSRI propagation approach" stems from CSRI propagation overheads from the subordinate (sub-network) to the upper level (layer network). Conversely, the flat topology approach showed the worst performance because it computed an optimal path with the entire network topology composed of 78 nodes and 166 links. In the case of "no CSRI propagation approach" and our approach, each computed the optimal path in a distributed and concurrent way. That is to say, each sub-network computes its own route with the partitioned network topology composed of 26 nodes and 51 links.

In this simulation, we tried to compute a path for LSP provisioning with 10Mbps between the fixed source and ten variable destinations as shown in Fig. 5.

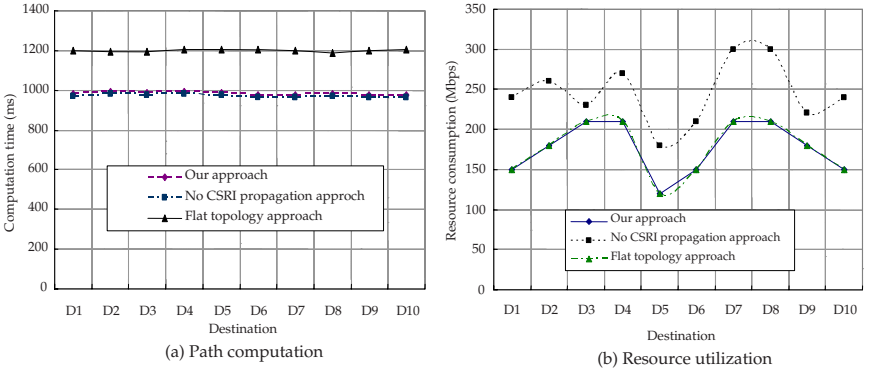


Fig. 6. Performance comparision

From the perspective of network resource utilization, our approach showed the highest performance of all approaches including the "no CSRI propagation approach" and flat topology approach as shown in Fig. 6 (b). By propagating CSRI from the subordinate to the upper level, and reflecting CSRI into the internal cost of node at the upper level, thereby showing the globally-optimal route in a hierarchical network environment. Even though the "no CSRI propagation approach" finds a route with abstracted network topology at the upper level, it cannot guarantee optimal route provision in a hierarchical network environment. Therefore, the resource utilization of the "no CSRI propagation approach" performed the worst among our approach and the flat topology approach as shown in Fig. 6 (b).

The flat topology approach showed nearly the same performance as our approach in terms of network resource utilization; this is because it finds a route with full network topology without abstraction. Taking network resource utilization and path computation speed into account, our approach is the most reasonable compared to the flat topology approach and the "no CSRI propagation approach" in a hierarchical MPLS network environment.

5 Concluding Remarks

In this paper, we proposed a scaleable LSP management architecture that can provide a globally-optimal route under the hierarchical MPLS network. We defined a hierarchical MPLS network model with layering and partitioning concepts. By introducing CSRI propagation from the subordinate partitioning level to the upper partitioning level, we can guarantee a globally-optimal LSP provision under the hierarchical network environment, which was impossible in the hierarchical network environment because of topology abstraction. In addition, this paper proposed an LSP management architecture that is applicable for LSP provision in a hierarchical MPLS network environment.

Through simulation, we compared our scheme with the approaches of flat topology and "no CSRI propagation" under a somewhat complex topology of 78 nodes and 166 links. As a result of this simulation, we proved that our approach is the most reasonable one when compared to the other two when taking path computation speed and resource utilization into account.

References

[1] R. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture, "RFC3031, January 2001.
[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and McManis, "Requirements for Traffic Engineering over MPLS, "RFC2702, September 1999.
[3] S. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC3272, May 2002.
[4] X. Xiao and L. M. Ni, "Internet QoS: A Big Picture," IEEE Network, March/April 1999.
[5] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni, "Traffic Engineering with MPLS in the Internet," IEEE Network, March 2000.
[6] A. Banerjee, J. Darke, J. P. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements," IEEE Communications Magazine, pp.144-150, January 2001.
[7] Gerald R. Ash, "Performance evaluation of QoS-routing methods for IP-based multiservice networks", Computer Communications Vol 26, pp.817-833, 2003.
[8] M. El-Darieby, D. C. Petriu, and J. Rolia, "A Hierarchical Distributed Protocol for MPLS path creation", Proc. of 7th IEEE International Symposium on Computers and Communications, pp.920-926, July 2002.
[9] E. Felstine and R. Cohen, "On the Distribution of Routing Computation in Hierarchical ATM Networks," IEEE Transactions on Networking, Vol. 7. No. 6, December 1999.
[10] B. Awerbuch, Y. Du, and Y.Shavitt, "The Effect of Network Hierarchy Structure on Performance of ATM PNNI Hierarchical Routing," Computer Communications Journal, vol 23, pp.980-986, 2000.
[11] A. Iwata and N. Fujita, "A Hierarchical Multilayer QoS Routing System with Dynamic SLA Management," IEEE Journal on Selected Areas in Communication, Vol. 18, No. 12, pp.2603-2616, December 2000.
[12] Y. Qin, L. Mason, and K. Jia, "Study on a Joint Multiple Layer Restoration Scheme for IP over WDM Networks," IEEE Network, pp.43-48, March/April 2003.

Resource Reconfiguration Scheme Based on Temporal Quorum Status Estimation in Computational Grids

Chan-Hyun Youn[1,2], Byungsang Kim[2], Dong Su Nam[2], Bong-Hwan Lee[3], Eun Bo Shim[4], Gary Clifford[1], and Jennifer Healey[5]

[1] Harvard-MIT Division of Health Science Technology, MIT
Cambridge, MA 02139, USA
{chyoun,gari}@mit.edu
[2] School of Engineering, Information and Communications University
58-4 Hwaam-dong, Yousong-gu, Daejeon 305-732, Korea
{dsnam,bskim}@icu.ac.kr
[3] Dept. of Information and Communications Engineering, Daejeon University
96-3 Yongun-dong, Dong-gu, Daejeon 300-716, Korea
blee@dju.ac.kr
[4] Dept. of Mechanical Engineering, Kwangwon National University
192-1 Hyoja 2-dong, Chunchon 200-701, Kwangwon-do, Korea
ebshim@kangwon.ac.kr
[5] Dept. of Translational Medicine, Harvard Medical School/BIDMC
Boston, MA 02215, USA
jhealey@bidmc.harvard.edu

Abstract. Quality of Service (QoS)-constrained policy has an advantage to guarantee QoS requirements requested by users. Quorum systems can ensure the consistency and availability of replicated data despite the benign failure of data repositories. We propose a Quorum based resource management scheme, which includes a system resource and network resource, both of which can satisfy the requirements of application QoS. We also propose the resource reconfiguration algorithm based on temporal execution time estimation method. Resource reconfiguration performs the reshuffling of the current available resource set for maintaining the quality level of the resources. We evaluate the effectiveness of resource reconfiguration mechanism in a Heart Hemodynamics analysis. Our approach increases the stability of execution environment as well as decreases the completion time compared to the method that does not adopt the proposed reconfiguration scheme.

1 Introduction

Recently, Grid computing has distinguished itself from the conventional distributed computing by focusing on large-scale resource sharing and innovative applications under the environment of widely-connected network. There are a number of architectures known for the management of Grid networks. However, most of them are mainly focused on concrete aspects which do not cover

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 699–707, 2004.

the management of the Grid as a whole. Examples are the Condor-G system [3] and Nimrod-G Grid resource broker [4]. Due to the complexity of the Grid system, and the trend towards increased complexity in both hardware/software and service requirements, the flexible management of the overall Grid system itself is becoming more and more important. Policy-based management (PBM)[5] shows a prominent approach and management architecture over previous traditional network management systems. Policy-based Management can guide the behavior of a network or distributed system through high-level declarative directives that are dynamically introduced, checked for consistency, refined and evaluated, resulting typically in a series of low-level actions [6]. The current PBM architecture has problems in the sense that it can only address a fairly limited set of issues and usually requires human intervention. Grid systems unfortunately suffer the same problems in terms of the scalability and autonomic management. It is time consuming and error-prone for Grid administrator, resource manager or broker to configure their system manually. Furthermore it is extremely hard to configure their local resource while considering other domains in the whole Grid system. In this paper, we propose a resource Quorum model for supporting QoS of resource status. Quorum set is a subset of the resource universe which can be obtained. Quorum set guarantees the reliable resource allocation because it is a set of resource collection that is selected according to the QoS of the resource. Also, we propose resource reconfiguration scheme in order to maintain the integrity of the Quorum set. Resource reconfiguration provides reshuffle of the current resources set and reallocates resources to tasks in order to support QoS. Execution time of an application reflects the current resource status. We use temporal variation of the execution time in order to estimate the resource status. Resource reconfiguration provides a good solution for degraded resources in an unpredictable environment. The remainder of this paper is structured as follows. In Section 2, we suggest the resource Quorum model and resource allocation scheme for reliable scheduling. Section 3 presents the proposed temporal Quorum status estimation method and resource reconfiguration algorithm. Section 4 shows the experimental results based on our reconfiguration algorithm using Heart Hemodynamics analysis. This paper is concluded in Section 5.

2 Resource Quorum Model

To guarantee QoS for user's application, the Grid manager should assign the resources to each application. The Quorum based management system defines the QoS vector to each application from the input profile. We utilize two different resource items: system resource and network resources. Each service requester must specify his QoS requirements for the resource manager in terms of the minimum QoS properties for both the system and network resources. All resources are distinguished from the others and ranked as a group of QoS level in terms of those resource descriptions. We represent the resource descriptions as a system resource vector $\boldsymbol{\theta_s}$ and a network resource vector $\boldsymbol{\theta_n}$.

2.1 Quorum Model for QoS Support

We define the Resource Quorum as $\boldsymbol{Q_R}$, which represents the current status of the resource. Each resource has its resource status vector which is represented by both invariable and variable elements [8]. System resources can take the processor specification or memory size as invariable elements and processor utilization or available memory space as variable elements. Also, network resource would have the link capacity with an invariable elements and end-to-end available bandwidth, delay, data loss rate as variable elements, such that

$$\boldsymbol{Q_R} = \{\langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_{jk} \rangle | i, j, k = 1, \ldots, n\}, \tag{1}$$

where $\boldsymbol{\theta_i}$ denotes the current available resource level of the system resource i and $\boldsymbol{\theta_{jk}}$ represents the current available resource level of the network between system resource j and k. A resource universe $R = \{r_1, \ldots, r_u\}$ assumes a collection of resources that can be taken in the administration domain. A resource, $r = \langle S, N \rangle$, can be represented as undirected graph in the system, S and their communication networks, N. Thus,

$$\boldsymbol{R} = \{r_i, \ldots, r_u\} = \{\langle S_i, N_{ij} \rangle | S_i = r_i, N_{ij} = r_i \times r_j \;\; i \neq j \;\; i, j = 1, \ldots, u\} \tag{2}$$

R is an available resource universe which is a subset of resource universe U, and i and j are elements in the R. S_i is therefore a system resource that represents a computation node i and N_{ij} is a network resource that represents a communication network from system i to system j. An application can be represented by undirected graph with tasks and their communication relation. The application is viewed as composed of the number of m tasks which represent problem size of the application $\boldsymbol{A}$. $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \{\nu^1, \ldots, \nu^m\} = \{\langle \nu^k, e^{kl} \rangle | e^{kl} = \nu^k \times \nu^l, k \neq i, k, l = 1, \ldots, m\}, \tag{3}$$

where m means the number of tasks. ν^k means the vertices that represent each task and e^{kl} means the edge that represents communication between ν^k and ν^l. Thus, l represents all communication peers that related to the ν^k.

If we assume that each task has its QoS requirement issued from SLAs, all resources can be ranked by the performance or be grouped by its attribute in terms of the QoS level. A required QoS level represents the vector of the resource description and has the range between the minimum and maximum requirement. We denote them by $\boldsymbol{q}$ and $\boldsymbol{Q}$, respectively. We define the QoS-Quorum,$\boldsymbol{Q_A}$, which represents the required quality level for the application set $A = \{1, \ldots, m\}$.

$$\boldsymbol{Q_A} = \left\{ \left\langle \left[\boldsymbol{q}_i^k, \boldsymbol{Q}_i^k\right] \left[\boldsymbol{q}_{ij}^{kl}, \boldsymbol{Q}_{ij}^{kl}\right] \right\rangle | i \neq j, i, j = 1, \ldots, \mu \;\; k \neq l, k, l = 1, \ldots, m \right\}, \tag{4}$$

where $\boldsymbol{q}_i^k$ and $\boldsymbol{Q}_i^k$ denote the minimum and maximum QoS level required for task k on the system resource i. $\boldsymbol{q}_{ij}^{kl}$ and $\boldsymbol{Q}_{ij}^{kl}$ represent the minimum and maximum QoS level required for communicating the task k in system resource i and task l in system resource j.

2.2 Available Resource Quorum and Resource Configuration

To achieve the reliability in resource management, we define an available resource Quorum,$\boldsymbol{Q}_{AR}$, which is selected from a resource universe. $\boldsymbol{Q}_{AR}$ satisfies the QoS requirement from SLAs.

$$\boldsymbol{Q_{AR}} = \{\langle S_i, N_{ij}\rangle \subseteq R | \boldsymbol{q}_i^k \leq \boldsymbol{\theta}_i \leq \boldsymbol{Q}_i^k, \boldsymbol{q}_{ij}^{kl} \leq \boldsymbol{\theta}_{ij} \leq \boldsymbol{Q}_{ij}^{kl}\}, \tag{5}$$

where $i, j = 1, \ldots, n' \leq \mu$ and $k, l = 1, \ldots, m$, and $\boldsymbol{Q_{AR}}$ is a set that satisfies a desired minimum QoS level of the application. Then, we define the Resource Configuration Function, $F(A, \boldsymbol{AR})$,as follows:

$$F(A, \boldsymbol{Q_{AR}}) = \{\langle S_i^k, N_{ij}^{kl}\rangle\} = \{\langle V^k, E^{kl}\rangle \xrightarrow{Q} \langle S_i, N_{ij}\rangle\}, \tag{6}$$

where

$$S_i^k = \begin{cases} 1, \text{ if the } \nu^k \in A \text{ and allocated on } S_i \\ \phi, \text{ otherwise,} \end{cases}$$

$$N_{ij}^{kl} = \begin{cases} 1, \text{ if the } e^{kl} \in A \text{ and existed in communication BW of } r_i \text{ and } r_j \\ \phi, \text{ otherwise} \end{cases}$$

Basically, Available Resource Quorum set $\boldsymbol{Q_{AR}}$ has the characteristics to guarantee the minimal QoS requirement. First of all, the minimal QoS constraints created by the SLAs make up two groups of vectors in the QoS Quorum and the Resource Quorum. The QoS Quorum is made for the service class correspondent with one of the QoS services. Simultaneously, the Resource Quorum is determined depending on whether the QoS constraints are satisfied or not.

3 Temporal Quorum Reconfiguration

Resource configuration function is different from general scheduling in terms of QoS support. Resource configuration provides a more reliable scheduling because it assumes that the elements of the available resource Quorum set $\boldsymbol{Q_{AR}}$ satisfy the user's desired quality level. But Quorum status varies as changing of the status of resources which are included in Quorum set. In order to maintain the integrity of QoS, it is necessary to reconfigure the current Quorum set. By monitoring resource status, we can validate the current Quorum status. Execution time of an application reflects the resource status. Now we present the reconfiguration scheme based on variation of execution time of the application.

3.1 Estimation of the Resource Status

Since the execution time of an application is defined as summation of the computation time on the system, $\hat{S}_i^k(\theta_i)$ and communication time, $\hat{N}_{ij}^{kl}(\theta_{ij})$, we can estimate the current resource utilization by each application activity. If an application represents its previous execution progress, we can predict the temporal variation of the resource utilization on target application. Equation 7 shows the

sensitivity of the estimates of the execution time, $Z(\bullet)$, according to resource Quorum.

$$\hat{S_i^k}(\theta_i) = \frac{dS_i^k(\theta_i)}{dZ(S_i^k(\theta_i))} \triangle Z(S_i^k(\theta_i)), \quad \hat{N_{ij}^{kl}}(\theta_{ij}) = \frac{dN_{ij}^{kl}(\theta_{ij})}{dZ(N_{ij}^{kl}(\theta_{ij}))} \triangle Z(N_{ij}^{kl}(\theta_{ij})) \quad (7)$$

Based on resource status which is obtained by Equation 7, we define the utility functions, such as a system utility function and a network utility function. Utility functions of the resource provide the current system and network performance compared to the previous status. Previously, we denoted the minimum QoS of each task required for computation and communication as $\boldsymbol{q}_i^k(\theta_i)$ and network, $\boldsymbol{q}_{ij}^{kl}(\theta_{ij})$, respectively. If a current estimates of system resource is $\hat{S_i^k}(\theta_i)$, then the utility function is represented by

$$\mu_i^k = \frac{\hat{s}_i^k(\theta_i) - q_i^k(\theta_i)}{q_i^k(\theta_i)}. \quad (8)$$

If $\mu_i^k < 0$, we assume that the system does not meet the QoS level. Similarly, if a current estimates of network status is $\hat{N_{ij}^{kl}}(\theta_{ij})$, the utility function is represented by

$$\mu_{ij}^{kl} = \frac{\hat{s}_{ij}^{kl}(\theta_{ij}) - q_{ij}^{kl}(\theta_{ij})}{q_{ij}^{kl}(\theta_{ij})} \quad (9)$$

Furthermore if $\mu_{ij}^{kl} < 0$, we recognize that the network does not guarantee the QoS requirements. Status of the current resource configuration can be presented as the accumulated value of the utility function. Therefore, on the current time instant T_c, the temporal status of the resource configuration,$sRC(A, \boldsymbol{Q_{AR}}, T_c)$, is defined as follows.

$$sRC(A, \boldsymbol{Q_{AR}}, T_c) = \left\{ \langle \mu_i^k, \mu_{ij}^{kl} \rangle \right\}, \quad (10)$$

where

$$\mu_i^k = \begin{cases} \sum_{t=0}^{T_c} \mu_i^k(T_c), & \text{if the } e^{kl} \in A \text{ and existed in communicaion BW of } r_i \text{ and } r_j \\ \phi, & \text{otherwise} \end{cases}$$

$$\mu_{ij}^{kl} = \begin{cases} \sum_{t=0}^{T_c} \mu_{ij}^{kl}(T_c), & \text{if the } e^{kl} \in A \text{ and existed in communicaion BW of } r_i \text{ and } r_j \\ \phi, & \text{otherwise} \end{cases}$$

The utility functions, μ_i^k and μ_{ij}^{kl}, are accumulated QoS reward values that are system and network resources. If the estimate of utility function is smaller than 0, it means that the resource does not match the desired QoS level. We should then identify the triggering condition for the resource reconfiguration.

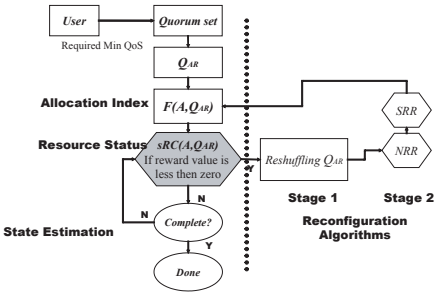


Fig. 1. Operation procedure of the proposed a resource Quorum reconfiguration scheme

3.2 Two-Stage Resource Reconfiguration Algorithm

The reconfiguration steps consist of System Resource Reconfiguration (SRR) when $\mu_i^k < 0$ and Network Resource Reconfiguration (NRR) when $\mu_{ij}^{kl} < 0$, respectively. Once triggered, the resource Quorum management system invokes the two-stage resource Quorum reconfiguration procedure on $sRC(A, \boldsymbol{Q_{AR}}, T_c)$ as shown in Figure 1.

In reconfiguration step, after generating the initial $\boldsymbol{Q_{AR}}$, it should be changed with the time. Before creating alternative configurations, we should obtain a new $\boldsymbol{Q_{AR}}$ by updating itself.

4 Experimental Results

We evaluate the effectiveness of resource reconfiguration mechanism with a Heart Hemodynamics analysis application that is parallel application linked with each task. The parallel applications are connected to each other by the Grids. End-to-end communication quality of service in case of linked chains is more important issue because it has an effect on the entire performance of the application execution. Moreover the communication status has various reasons that cause degradation. In order to point out the communication quality of service, we examine the end-to-end bandwidth between each task. Blood flow in the sac of the Korean Artificial Heart (KAH) is numerically simulated by finite element analysis. A distributed computing algorithm is employed to compute the hemodynamics of the sac using Globus-based MPI programming. Each sub-job of the KAH communicates with its neighbors in order to exchange the message for satisfying the boundary condition. Figure 2 shows the boundary condition of the KAH and the flow chart for simulating the blood sac in the KAH. The program has the iteration characteristics to solve the velocity and pressure at each time frame. The Grid testbed for collaborating among multi-domain environments is comprised of Linux-based Globus platforms for Grid application. The Grid testbed for modeling the KAH is run on three domains (Information and Communications University (ICU), MIT and Hanyang Univeristy (HYU))

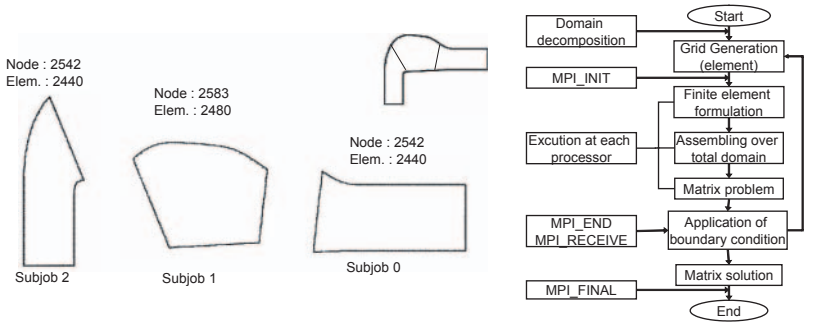
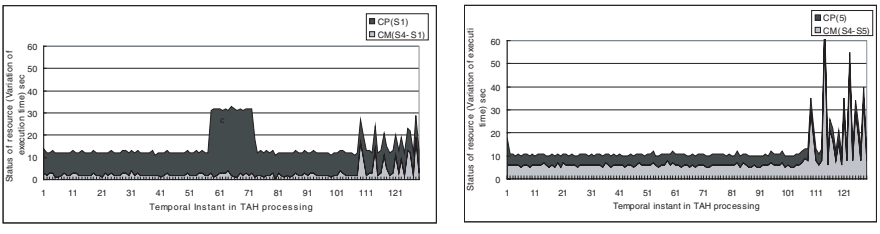


Fig. 2. Computing model for the KAH



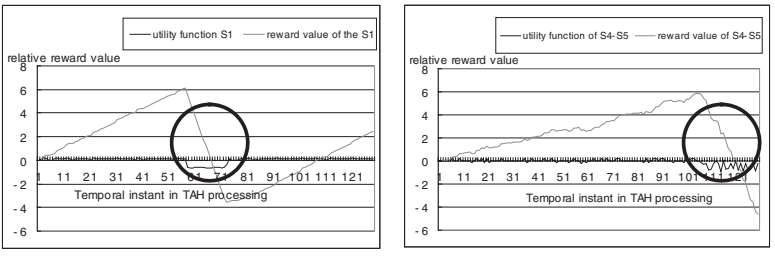
(a) Execution time of system S1 (b) Execution time of system S5

Fig. 3. Performance comparison of systems S1 and S5

and five systems (S1, S2, S3, S4, and S5 with available bandwidth of 0.1 Mbps). They are connected to the Korea Research and Education Network (KOREN) and Science Technology And Research Transit Access Point (STAR-TAP).

We conducted an experiment with offered load at system S1 at the time instant T2 (time instant 50) and network S4-S5 at the time instant T3 (time instant 100). T2 and T3 are the points of the time instant 50 and time instant 100 in Figure 3, respectively. Figure 3 shows the execution time of system S1 and S5. In Figure 3 (a), system S1 increases the computation time on the time instant 50 to 75. Also, system S5 increases the communication time irregularly and rapidly between system S4 and S5 between time instant 105 to 125 shown in Figure 3 (b). The performance was evaluated according to each computation time (CP) and communication time (CM).

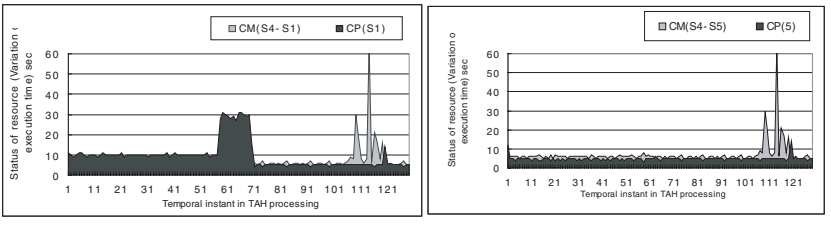
We calculate the reward value for the offered workload situation. Figure 4 shows the characteristics of the utility function and reward value of the system S1 and S5. The reward value of the system S1 has a negative value around at time instance 70 as depicted in Figure 4 (a). Also, the reward value of the systems S4-S5 has negative value at time-instance 115 as shown in Figure 4 (b). The zero point of the reward value of the resource is the reconfiguration triggering point. We take two reconfiguration points at time instance 70 and time instance 115 based on reward value from Figure 4. Let's consider the recon-



(a) Utility function and relative reward value at system S1

(b) Utility function and relative reward value at system S4-S5

Fig. 4. Characteristics comparison of the utility function and the reward values in target systems



(a) Execution time of system S1 after committing the resource reconfiguration

(b) Execution time of system S5 after committing the resource reconfiguration

Fig. 5. Performance comparison of the proposed reconfiguration scheme

figuration policies that are generated by our reconfiguration algorithm. Initial topology has the system S1, S4, S5. After the reconfiguration of system S1 at T2, the system S1 replaced with S3. Also, on the reconfiguration of network S4-S5, the system S4 replaced with S1. The network topology has been changed by the reconfiguration of the system. Figure 5 illustrates the execution time after committing the resource reconfiguration. Figure 5(a) shows the execution time of the system S1 when committing reconfiguration at T2, whereas Figure 5(b) shows the execution time of the system S5 when committing reconfiguration at T3. Figure 5 shows that proposed reconfiguration scheme increases the stability of the execution time compared to Figure 3.

Heart Hemodynamics Analysis Application shows that the effectiveness of system and network reconfiguration of the parallel application. From the experimental results, we can see that the proposed reconfiguration algorithm provides more stability during the execution time. In addition, the proposed algorithm decreases the completion time of the application.

5 Conclusions

Computational Grids are focused primarily on high-performance distributed computing. In a wide area Grid environment, it is most important to guarantee the user desired resources. Reliable resource allocation should be maintained on the temporal and spatial change. Quorum based resource modeling and resource configuration scheme provides more reliable resource scheduling. The proposed resource reconfiguration algorithm has two phases. One is the status estimation using temporal deviation. The other is resource reconfiguration based on estimated status. After triggering based on estimation, the reconfiguration algorithm tries to optimize the current system and network resource. Our approach provides both increase in the stability of the execution environment and decrease in the completion time compared to the methods that do not adopt the resource reconfiguration mechanism.

Acknowledgements

This work was supported in part by the Ministry of Information and Communication of Korea.

References

[1] Foster, I. and Kesselman, C. (eds.). *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, 1999.
[2] Foster, I. and Kesselman, C. The Anatomy of the Grid:Enabling Scalable Virtual Organizations. *Intl J. Supercomputer Applications*, 2001.
[3] Frey, J., Foster, I., Livny, M., Tannenbaum, T. and Tuecke, S. Condor-G: A Computation Management Agent for Multi-Institutional Grids. University of Wisconsin Madison,2001. 700
[4] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic Models for Resource Management and Scheduling in Grid Computing.*Journal of Concurrency and Computation.*Wiley Press, 2002. 700
[5] K.Yang, A. Galis, C. Todd. A Policy-based Active Grid Management Architecture. *In Proceedings of the 10th IEEE International Conference on Networks (ICOIN02)*, pages 243-248, IEEE Press, 2002. 700
[6] P. Flegkas, P. Trimintzios, G. Pavlou, A. Liotta. Design and Implementation of a Policy-based Resource Management Architecture. *In Proceedings of the IEEE/IFIP Integrated Management Symposium (IM'2003)*, pages 215-229, Colorado Springs, USA, 2003. 700
[7] Franken, L. J. N. and Haverkort, B. R. The performability manager.*IEEE Network*, 1994.
[8] A.Leff, J. T.Rayfield, and D. M.Dias. Service-Level Agreements and Commercial Grids. pages 44-5, *IEEE Internet Computing*, 2003. 701
[9] I. Cardei, S. Varadarajan, M. Pavan, M. Cardei, and M. Min. Resource Management for Ad-hoc wireless networks with cluster organization. *Journal of Cluster Computing in the Internet*, 2004.
[10] Christos G, Cassandras. *Discrete Event Systems, Modeling and Performance Analysis*, IRWIN Press, 1993.

Double-Link Failure Recovery in WDM Optical Torus Networks

Eunseuk Oh[1], Hongsik Choi[2], and Jong-Seok Kim[3]

[1] Department of Computer Science
Texas A&M University, College Station, TX 77843-3112, USA
eunseuko@cs.tamu.edu
[2] Department of Computer Science, Virginia Commonwealth University
Richmond, VA 23284-3068, USA
hongsik@vcu.edu
[3] Department of Computer Science, Sunchon National University
Sunchon, Chonnam, 540-742, Korea
rockhee@sunchon.ac.kr

Abstract. Most existing link protection methods have been focused on single-link failures. Recently, the double-link failure model has also been considered. In this model, any two links may fail in an arbitrary order. However, an existing model for protecting the entire network against double-link failures does not always provide a 100% recovery when the link identification is not required for pre-computing backup paths. In this paper, we consider double-link failure models that provide 100% recovery in WDM torus optical networks. Three link protection methods are presented. The first method requires failed link identification, whereas the last two methods do not. Other performance measures such as hop-length and backup capacity are also discussed. Finally, we briefly address how our method is possibly extended to mesh optical networks.

1 Introduction

Due to WDM (wavelength division multiplexing) technology, which is the current favorite multiplexing technology, optical networks provide high-speed and huge bandwidth network facilities. One of the key issues on high-speed optical networks is the network survivability that deals with a mechanism to protect resources against the node or link failures. Node failures occur due to the failure of equipment such as switches. Link failures occur due to the failure of optical links such as fiber cuts. Such a link failure typically causes all fibers in the bundle to be cut, and thus result in the failure of hundreds of channels. Such failures disrupt the optical network service, and may lead to data losses on the order of several gigabits. Therefore, elaborate mechanisms to restore the traffic upon detecting failures are critical.

Main approaches to recover from a link failure are to provide an alternate path. When a failure occurs, all traffic on the failed link are switched to backup

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 708–717, 2004.

paths. Most practical studies for link protection assume single link failures. Recently, the usefulness and necessity of studying double-link failures were presented [1, 2, 4]. The relationship between double-link failures and the performance measures of protection methods was presented in [4]. Based on their quantitative measure of a networks' recovery ability, link protection schemes using pre-assigned capacity were provided in [1, 2]. One disadvantage of their scheme is that it does not always achieve a 100% recovery from double-link failures when the link identification is not required for pre-computing backup paths. Also, it is not known whether backup paths can be pre-computed for every link in a mesh under their requirement: for arbitrary two links in a mesh network, they can not be contained each other's backup path concurrently. In this paper, we consider double-link protection methods that provide 100% recovery in torus networks while keeping our mind on other performance measures such as hop-length and backup capacity.

As one of popular logical topologies of low dimensions such as ring and mesh, the torus has been studied for optical communication networks [3, 6, 7]. The torus is an interesting topology for the metropolitan area networks, and has an unique topological flexibility. First of all, the torus consists of link-disjoint rings. Thus, it has a merit that most link protection algorithms developed on ring-based topologies can be easily applied. Next, the torus is mesh-based topology. That is, the torus is a mesh with wrap-around links. Due to wrap-around links, the torus networks overcome the topological irregularity between boundary nodes and non-boundary nodes of mesh networks. Taking advantage of such a topological property of torus networks, our method provides a possible direction for developing double-link protection methods in mesh networks. Throughout the paper, we only consider a mesh (distinguished from other generalized mesh-type topologies) as the topology obtained by removing wrap-around link from the torus.

The rest of the paper is organized as follows. In section 2, we introduce the basic definitions and notations. In section 3, we present some possible approaches for recovery from double-link failures. In section 4, we provide two link protection schemes against double-link failures. Both schemes require one backup path for each link, but the lengths of backup paths provided by them have different performance aspects. In the end of section, we briefly address how our method can be used for mesh networks. In section 5, we conclude the paper.

2 Preliminaries

A WDM optical network is represented by a graph $G = (N, L)$, where N is the set of logical nodes and optical switches connected the logical nodes, and L is the set of optical links. Each link consists of a pair of directional links - one in each direction. A 2D torus T is a graph with nodes whose label is identified by two coordinates (i, j) and links connecting them, where the lower-most West node is labeled as (0,0). For a node (i, j), it has four neighbors $(i - 1, j)$, $(i + 1, j)$,

$i, j-1)$, and $(i, j+1)$, where "+" and "-" operations on indices are performed as modulo arithmetic.

An X-ring R_x (resp. Y-ring R_y) in the 2D torus is the ring in which nodes have the same value of coordinate y (resp. x). $R_x(e)$ (resp. $R_y(e)$) denotes the X-ring (resp. Y-ring) with a link e. For a link $e =< (i,j),(i,j+1) >$, a semi X-ring of e, $\tilde{R}_x(e)$ is the ring consisting of nodes (i,j), $(i+1,j)$, and nodes in $R_x(h)-h$, where $h =< (i,j+1),(i+1,j+1) >$. Also, for a link $e =< (i,j),(i+1,j) >$, a semi Y-ring of e, $\tilde{R}_y(e)$ is the ring consisting of nodes $(i+1,j)$, $(i+1,j+1)$, and nodes in $R_y(g)-g$, where $g =< (i,j),(i,j+1) >$. Two rings are called *link-disjoint* if they do not share a link. For two links $e =< (i,j),(i+1,j) >$ and $e' =< (i,j'),(i+1,j') >$, $j \neq j'$, two X-rings $R_x(e)$ and $R_x(e')$ are link-disjoint. Also, for two links $e =< (i,j),(i,j+1) >$ and $e' =< (i,j'),(i,j'+1) >$, $j \neq j'$, two semi X-rings $\tilde{R}_x(e)$ and $\tilde{R}_x(e')$ are link-disjoint. Similarly, for two link $e =< (i,j),(i,j+1) >$ and $e' =< (i',j)(i',j+1) >$, $i \neq i'$, two Y-rings $R_y(e)$ and $R_y(e')$ are link-disjoint. Also, for two links $e =< (i,j),(i+1,j) >$ and $e' =< (i',j),(i'+1,j) >$, $i \neq i'$, two semi X-rings $\tilde{R}_x(e)$ and $\tilde{R}_y(e')$ are link-disjoint. Rings and semi-rings in a torus network are shown in Fig 1.

The performance measures to evaluate our schemes are restorability and hop-length. Restorability is the fraction of the number of double-link failures that can be tolerated. Since the torus has $2|N|$ links, $2|N|(2|N|-1)$ link failures can happen. If all possible link failures can be recovered by a scheme, then we say that scheme provides 100% restorability. Hop-length is the number of links used in rerouting for all possible failures. Shorter hop-length indicates better performance because the limitations on the length of feasible backup paths. In addition, all our schemes require backup capacity to be reserved for protection. The protection capacity for each link is no more than 200% of link working capacity.

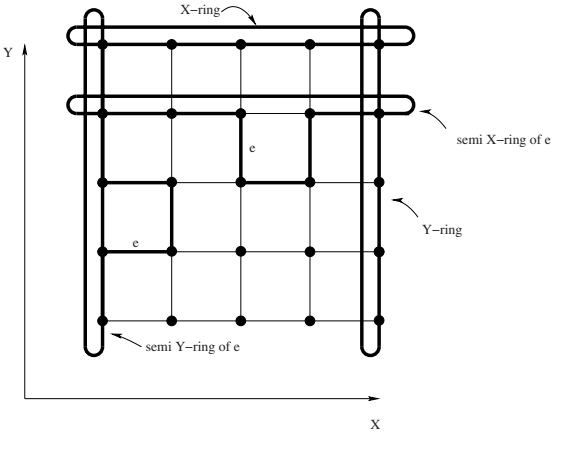


Fig. 1. Rings and semi-rings in the torus network

3 Double-Link Failure Recovery

The 2D torus consists of link-disjoint X-rings and Y-rings. Thus, the 2D torus T naturally contains the simple protection mechanisms of the ring when a single link fails: For a ring R of T, when a link e fails, other part of the ring $R-\{e\}$ works as the primary backup path. In the rest of this section, we discuss protection schemes against double-link failures, where two random links fail simultaneously or one link fails after the other fail.

3.1 Recovery with Link Identification

For each link e in a torus T, $R_x(e)-\{e\}$ is used as the primary backup path $p_1(e)$, and $\tilde{R}_y(e)-\{e\}$ is used as the secondary backup path $p_2(e)$. When a link e fails, the traffic on e reroutes along the primary backup path $p_1(e)$ using pre-assigned capacity. If two links e and f fail and they are in disjoint rings, then $p_1(e)$ and $p_1(f)$ are used for rerouting. If two links e and f fail and they are in the same ring, then $p_2(e)$ and $p_2(f)$ are used for rerouting. To detect whether failed links are in the same ring or not, link identification is required. The protection capacity needed on each link is equal to the working capacity because backup paths do not share a link. However, this scheme requires signaling to inform the failure to all other nodes in the network. In addition, the protection capacity must be pre-allocated on both backup paths $p_1(e)$ and $p_2(e)$.

3.2 Recovery Without Link Identification

For each link e in a torus T, a single backup path $p(e)$ is computed. When two links e and f fail, there are two possible cases depending on whether the traffic on e is rerouted along the backup path $p(e)$(resp. $p(f)$)by going through f(resp. e) or not. Suppose the link e fails first. If f is not contained in $p(e)$, then the traffic on e is rerouted along $p(e)$ as long as f does not fail yet. When f fails, the working traffic on f can be rerouted along $p(f)$ if the traffic on f is rerouted without going through the link e. In this case, the traffic rerouted from e is also switched to $p(f)$. This scheme does not require signaling to inform other nodes about link failures, and failed link identification. The detection of a failed link is detected by the end-nodes of that link, However, this scheme requires the condition that any two arbitrary links cannot be contained in each other's backup path concurrently. This condition leads to the following problem:

> *Maximum Arbitrary Double-Link Protection Problem*(MADPP) [2]: For each link e in a given graph G, find a backup path $p(e)$ such that $F = \{\{e, f\}|e \in p(f) \text{ and } f \in p(e)\}$ has the minimum cardinality, where f is an arbitrary link in G.

A link pair in F is said to be *prohibited*. It is not known whether for every link in any given 3-connected graph, backup paths can be pre-computed such

that $F = \phi$. Note that if the graph is not 3-connected, then there exist link pairs that cannot be tolerated by any algorithm. Since F is exactly the set of link pairs whose failures cannot be tolerated by our scheme, our goal is to find the protection scheme that extracts $F = \phi$, which guarantees 100% restorability from double-link failures. In the following section, we provide maximum arbitrary double-link protection schemes that for each link e in a given torus T, finds a backup path $p(e)$ such that $F = \phi$.

4 The Maximum Arbitrary Double-Link Protection Schemes

In this section, we provide two link protection methods to recover from double-link failures. Readers may not be confused with dynamic restoration approaches that do not reserve capacity. We focus only on schemes that reroute around the failed links along pre-computed backup paths by using pre-assigned capacity. Our first scheme utilizes semi-rings, while our second scheme constructs blocks whose sizes are smaller than semi-rings.

Semi-ring Protection Method (SPM): This scheme simply uses a semi-ring to reroute the traffic on each failed link: for each link e in $R_x(e)$ (resp. $R_y(e)$), a single backup path $\tilde{R}_y(e) - \{e\}$ (resp. $\tilde{R}_x(e) - \{e\}$) is pre-computed.

Theorem 1. *SPM achieves 100 % recovery from double-link failures in a torus network T by providing the backup paths of hop-length bounded by $2\sqrt{N} + 1$.*

Proof. Suppose the link $e =< (i,j),(i+1,j) >$ fails first. We only need to show that for every link $f \in p(e)$, the backup path $p(f)$ does not contain e. Let us assume that e is in X-ring. Then the backup path $p(e)$ consists of nodes $(i+1,j)$ and $(i+1,j+1)$ and nodes in $R_y(h) - \{h\}$, where $h =< (i,j),(i,j+1) >$.

Case 1. $f =< (i,j+1),(i+1,j+1) >$.

The backup path $p(f)$ consists of nodes $u = (i+1,j+1)$ and $v = (i+1,j+2)$, and nodes in $R_y(g) - \{g\}$, where $g =< (i,j+1),(i,j+2) >$. Neither u or v can not be an end node of e. Also, $R_y(g)$ does not contain e because the value of X-coordinate of all nodes in $R_y(g)$ is i. Thus, the backup path $p(f)$ does not contain e. Note that backup paths $p(e) - \{f\}$ and $p(f)$ are not necessarily link disjoint. When f fails, the working traffic on f is rerouted along $p(f)$ of hop-length $\sqrt{N}+1$. If the traffic on e is rerouted from the end-node $(i+1,j)$, then it is again rerouted along the path $\{h_1\} \cup p(f) - \{h_2\}$, where $h_1 =< (i+1,j),(i+1,j+1) >$ and $h_2 =< (i,j),(i,j+1) >$. Specifically, the traffic rerouted from one end-node $(i+1,j)$ of e reaches one end-node of f, and then arrives the other end-node (i,j) of e before it arrives the other end-node of f. Thus, the hop-length for rerouting is bounded by $\sqrt{N}+1$. Also, if the traffic on e is rerouted from the end-node (i,j), then it is rerouted along the path $p(e) - \{f\} \cup p(f)$ whose hop-length is bounded by $2\sqrt{N}+1$.

Case 2. $f =< (i+1,j),(i+1,j+1) >$.

The backup path $p(f)$ consists of nodes $u = (i+1,j)$ and $v = (i+2,j)$, and nodes in $R_x(g) - g$, where $g =< (i+1,j+1),(i+2,j+1) >$. Both nodes u and v can not be the end-nodes of e. Also, $R_x(g)$ does not contain e because the value of Y-coordinate of all nodes in $R_x(g)$ is $j+1$. Thus, the backup path $p(f)$ does not contain e. When f fails, the working traffic on f is rerouted along $p(f)$ of hop-length $\sqrt{N}+1$. Also, the rerouted traffic from e going through f is again rerouted along the path $p(e) - \{f\} \cup p(f)$. Thus, the hop-length for rerouting is bounded by $2\sqrt{N}+1$.

Case 3. f is contained in $R_y(h) - h$, where $h =< (i,j),(i,j+1) >$.

Suppose $f =< (i,x),(i,x+1) >$, $x \neq j$. Then the backup path $p(f)$ consists of nodes $u = (i,x)$ and $v = (i+1,x)$, and nodes in $R_x(g) - g$, where $g =< (i,x+1),(i+1,x+1) >$. Since $x \neq j$, both nodes u and v can not be the end-nodes of e. The value of Y-coordinate of all nodes in $R_x(g)$ is $x+1$. If $j \neq x+1$, then $R_x(g)$ does not contain e. If $j = x+1$, then we have $e = g$. Since $p(f)$ does not contain g, so does not e. Thus, the backup path $p(f)$ does not contain e. When f fails, the rerouted traffic on f is rerouted from e going through f is rerouted along the path $p(e) - \{f\} \cup p(f)$. Thus, the hop-length for rerouting is bounded by $2\sqrt{N}+1$.

So far, we assume that e is in X-ring. If e is in Y-ring, then similarly we can show that the theorem holds. Note that the torus network is 4-connected. Thus, the failures of any two links can not disconnect the network. Therefore, SPM provides 100% recovery from double-link failures thorough the backup paths whose hop-lengths are bounded by $2\sqrt{N}+1$. Fig. 2 illustrates a backup path pre-computed by SPM. □

The backup paths $p(e)$ and $p(f)$ are not necessarily link-disjoint. If $p(e)$ and $p(f)$ share a link, then the protection capacity reserved on that common link must be twice the link working capacity. We showed that SPM provides backup paths of the maximum hop-length $2\sqrt{N}+1$. The disadvantage of SPM is that the hop-length increases as the size of network increases. To overcome

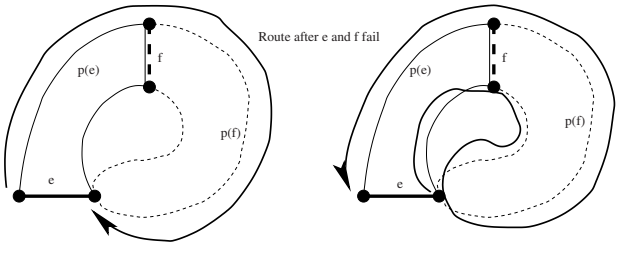


Fig. 2. Backup paths by SPM

this disadvantage, we provide another scheme that provides backup paths whose hop-length is bounded by a constant.

Block-model Protection Method (BPM): BPM is similar to SPM, and uses two different small blocks to reduce the maximum hop-length. Let $e_1 =< (i,j),(i+1,j) >$ be a link in X-ring, and $e_2 =< (i,j),(i,j+1) >$ be a link in Y-ring. Then, their corresponding blocks b_1 and b_2 are shown below.

link	block
e_1	$[(i+1,j)(i+1,j+1)(i,j+1)(i-1,j+1)(i-1,j)(i,j)]$
e_2	$[(i,j)(i+1,j)(i+1,j+1)(i,j+1)]$

We only consider links on boundaries of blocks. Then for a link e_i, $i = 1$ or 2, $b_i - \{e_i\}$ is used as the backup path $p(e_i)$.

Theorem 2. *For a given torus network T, the minimum hop-length of the backup paths to achieve 100% restorability from double-link failures is at least 7.*

Proof. The smallest size of ring in torus T is 4. Thus, we can pre-compute a backup path for each link whose hop-length is 3. Suppose a link f contained in the backup path $p(e)$ fails after a link e failed. Then the working traffic on e is rerouted on the backup path $p(e) - \{f\} \cup p(f)$ whose hop-length is 5. We claim that the optimal hop-length of the backup paths can not be 5.

By way of contradiction, let us assume that the optimal hop-length of the backup paths to achieve 100% restorability is 5. Then every backup paths must have the hop-length 3. For a link $e_1 =< (i,j),(i+1,j) >$, let $[(i+1,j)(i+1,j+1)(i,j+1)(i,j)]$ be its block. Let $e_2 =< (i+1,j),(i+1,j+1) >$, $e_3 =< (i+1,j+1),(i,j+1) >$, and $e_4 =< (i,j),(i,j+1) >$. Suppose e_1 fails first. Then, for each link e_i, $i = 2, 3$, and 4, its block must be $b_2 = [(i+1,j)(i+2,j)(i+2,j+1)(i+1,j+1)]$, $b_3 = [(i+1,j+1)(i+1,j+2)(i,j+2)(i,j+1)]$, and $b_4 = [(i,j+1)(i-1,j+1)(i-1,j)(i,j)]$, respectively. Otherwise, e_1 is contained in $p(e_i)$, e_i, $i \neq 1$, and e_i is contained in $p(e_1)$. Again, let $e_5 =< (i+1,j+1),(i+2,j+1) >$. Then its block must be $b_5 = [(i+2,j+1)(i+2,j+2)(i+1,j+2)(i+1,j+1)]$. Otherwise, e_2 is contained in $p(e_5)$ and e_5 is contained in $p(e_2)$. Now, let $e_6 =< (i+1,j+1),(i+1,j+2 >$. Then its block must be either $[(i+1,j+2)(i,j+2)(i,j+1)(i+1,j+1)]$ or $[(i+1,j+1)(i+2,j+1)(i+2,j+2)(i+1,j+2)]$. In this case, either e_3 is contained in $p(e_6)$ and e_6 is contained in $p(e_3)$, or e_5 is contained in $p(e_6)$ and e_6 is contained in $p(e_5)$. It contradicts our assumption. Links in above discussion are shown in Fig 3.

Thus, the hop-length of backup paths of some link must be at least 5. It shows that the minimum hop-length to achieve 100% restorability in the torus network is at least 7. □

Theorem 3. *BPM achieves 100% recovery from double-link failures in a torus network T by providing the backup paths of near optimal hop-length.*

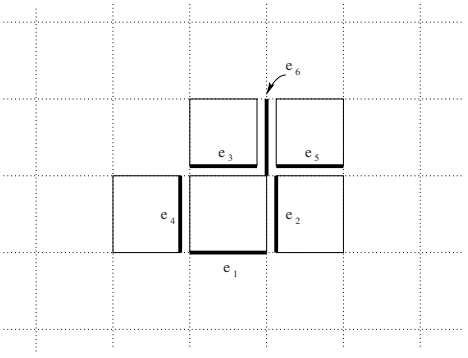


Fig. 3. A counterexample against the optimal hop-length 5

Proof. Suppose the link e_1 fails first. Then each link f_i, $i \neq 1$, in $p(e_1)$, we need to show that the backup path $p(f_i)$ does not contain e_1. Note that the hop-length of $p(e_1)$ is 5. Consider a sequence of labels $l_1 \ldots l_i \ldots l_5$, $l_i \in \{1, 2\}$. A label $l_i = 1$(resp. $l_i = 2$) represents that for a link f_i with the label 1(resp. 2), $p(f_i)$ has the same block shape as $p(e_1)$(resp. $p(e_2)$), where we traverse links of the backup path in counterclockwise order.

Case 1 A link e_1 fails first.

If e_1 fails first, then we have the sequence of labels $\{2, 1, 1, 2, 1\}$. Suppose the link f_1 fails next. The link $f_1 =< (i+1, j), (i+1, j+1) >$ has the label 2. Thus, its block consists of nodes $[(i + 1, j)(i + 2, j)(i + 2, j + 1)(i + 1, j + 1)]$. In this block, both end-nodes of e_1, (i, j) and $(i + 1, j)$ are not contained. It shows that the backup path $p(f_1)$ does not contain the link e_1. Thus, the traffic on e_1 is rerouted along the path $p(e_1) - \{f_1\} \cup p(f_1)$ whose hop-length is 7.

Similarly, we can show that for all other cases, the theorem holds: for each link f_i, $2 \leq i \leq 5$, the backup path $p(f_i)$ does not contain the link e_1. Now, the working traffic on f_i is rerouted along $p(f_i)$. Also, the rerouted traffic from e_1 is rerouted again along the path $p(e_1) - \{f_i\} \cup p(f_i)$ whose hop-length is bounded by 9. Note that some link is on both backup paths $p(e_1)$ and $p(f_i)$. In this case twice the link working capacity is reserved for the protection of that link.

Case 2 A link e_2 fails first.

If e_2 fails first, then we have the sequence of labels $\{1, 2, 1\}$. Suppose the link $f_1 =< (i, j), (i + 1, j) >$ fails next. Then the link f_1 has the label 1, and its block consists of nodes $[(i+1, j)(i+1, j+1)(i, j+1)(i-1, j+1)(i-1, j)(i, j)]$. In this block, the link $e_2 =< (i, j), (i, j+1) >$ is not contained. It shows that the backup path $p(f_1)$ does not contain the link e_2. In this case, the working traffic on f_1 is rerouted along $p(f_1)$ whose hop-length is 5. Also, the rerouted traffic from e_2 is again rerouted along the path $p(e_2) - \{f_1\} \cup p(f_1)$ whose hop-length is 7.

Similarly, we can show that for all other cases, the theorem holds: for each link f_i, $i = 2, 3$, the backup path $p(f_i)$ does not contain the link e_2. Since the

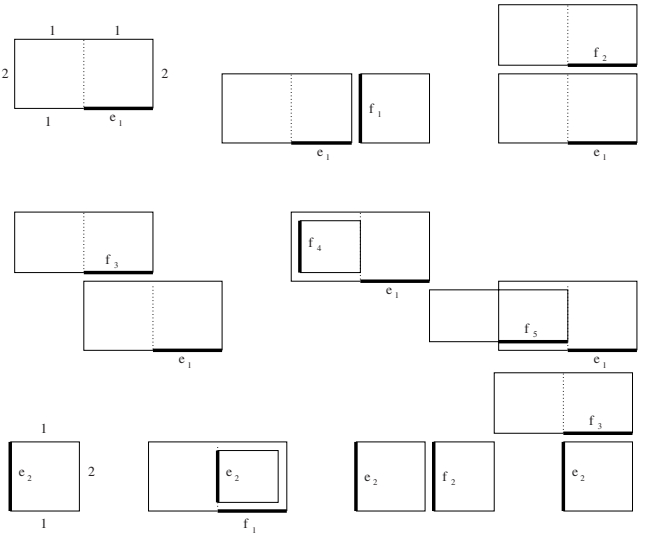


Fig. 4. Backup paths by BPM

hop-length for rerouting is maximum when the path is $p(e_2) - \{f_i\} \cup p(f_i)$, we conclude that the hop-length is bounded by 7. The backup paths pre-computed by BPM are shown in Fig 4.

From the above discussion, we showed that BPM provides 100 % recovery from double-link failures through the backup paths whose hop-lengths are bounded by 9, which is the optimal hop-length + 2. □

Extension to Mesh Optical Networks : Mesh is one of popular topologies for parallel and distributed computing systems, and communication networks because it has small node degree, and provides the flexibility in routing connections. Protection in mesh networks can be efficient, whereas it is more complex because the multiple routes can be used for recovery. Also, the natural evolution of network topologies leads to mesh-type topologies. Different to a torus, the nodes at four corners of mesh have two neighbors, and other nodes on the boundary have three neighbors. The rest of the nodes in a mesh are the same as torus. Such a topological property of mesh networks provide an interesting view of our results: Our results can be viewed as a survivable ring embedding in mesh networks to obtain the minimum cardinality of the set of prohibited link pairs. Moreover, in an infinite mesh which is often considered in literatures for presentational convenience (See [8] for example), our BPM can be directly applied. Note that if, in a finite mesh, any two neighboring links at a corner fail, then such failures can not be tolerated by any algorithms.

5 Concluding Remarks

We provided link protection schemes that in torus optical networks, achieves 100% recovery from double-link failures by providing the backup paths of near optical hop-length. Most our discussion strictly follows the topological property of torus networks. Thus, our approach may not directly apply to practical networks such as ARPANET, NJ LATA LATAX networks. However, one of our main goals is to explore the theoretic aspects of MADPP, which is conjectured as NP-hard in [1, 2]. We argue that our results provide a possible evidence that can settle the conjecture on MADPP. However, we expect that developing an algorithm can give the exact solution or the exact minimum cardinality of prohibited link pairs for general mesh-type topologies will be hard though they have a very simple structure. Further investigation of MADPP on general mesh-type topologies will be carried out for future study, along with performance comparisons with other schemes.

References

[1] H. Choi, S. Subramaniam, and H.-A. Choi, "Loopback Recovery form Double-Link Failures in WDM Optical Mesh Networks," *IEEE/ACM Trans. Networking*, to appear. 709, 717

[2] H. Choi, S. Subramaniam, and H.-A. Choi, "On Double-Link Failure Recovery in WDM Optical Networks," *Proc. INFOCOM*, 2002. 709, 711, 717

[3] Q. -P. Gu and S. Peng, "Multihop All-to-All Broadcast on WDM Optical Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 14, no. 5, pp. 477-486, 2003. 709

[4] S. S. Lumetta, M. Medard, and Y.-C. Tseng, "Capacity Versus Robustness: A Tradeoff for Link Restoration in Mesh Networks," *IEEE/OSA J. Lightwave Tech.*, vol. 18, no. 12, pp. 1765-1775, 2000. 709

[5] E. Modiano and A. Narula-Tam, "Survivable Routing of Logical Topologies in WDM Networks," *Proc. INFOCOM*, pp. 348-357, 2001.

[6] R. Ramaswami and K. N. Sivarajan, "Optical Networks A Practical Perspective," Morgan Kaufmann, 1988. 709

[7] T. Stern and K. Bala, "Multiwavelength Optical Networks," Addison-Wesley, 1999. 709

[8] Y. Yang, and J. Wang, "Pipelined All-to-All Broadcast in All-port Meshes and Tori," *IEEE Trans. Computers*, vol. 50, no. 10, 2001. 716

Multi-Wavelength-Minimum Interference Path Routing Algorithm for Establishing Optimal Optical-LSPs in OVPN

Jong-Gyu Hwang[1], Kamil Ratajczak[2], Hyun-Jin Lee[3], Young-Bu Kim[3], Sung-Woon Kim[2], and Yong-Jin Park[1]

[1] Korea Railroad Research Institute,
360-1 Woulam-dong, Uiwang-city, Kyonggi-do, 437-050, Korea
Hanyang University,
17 Haengdang-Dong, Seongdong-Gu, Seoul, 133-791, Korea
jghwang@krri.re.kr
park@hyuee.hanyang.ac.kr
[2] Pukyong National University,
599-1 Daeyeon 3-Dong Nam-Gu, Busan, 608-737, Korea
kamart7@yahoo.com
kimsu@pknu.ac.kr
[3] Electronics and Telecommunications Research Institute,
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
{petrus,ybkim}@etri.re.kr

Abstract. A "virtual private network (VPN) over Internet" has the benefit of being cost-effective and flexible. Given the increasing demand for high bandwidth Internet and the demand for QoS assurances in a "VPN over Internet", IP/generalized multi-protocol label switching (GMPLS) based on a control plane combined with a high-bandwidth, dense-wavelength division multiplexing (DWDM) optical network is seen as a very favorable approach for realizing the future "optical VPN (OVPN) over IP/GMPLS over DWDM". In this paper, we suggest a new routing algorithm for establishing optimal optical-label switched paths (O-LSPs) in OVPN, called the Multi-Wavelength-Minimum Interference Path Routing (MW-MIPR), to provide more improved performance for connection blocking probability with consideration for potential future network's congestion status. The proposed algorithm improves wavelength utiliza-tion by choosing route that does not interfere too much with many potential future connection requests. Simulation results show that proposed MW-MIPR algorithm achieves more enhanced blocking probability than dynamic routing (DR) that yields the best performance among previous routing and wavelength assignment (RWA) algorithms.

1 Introduction

VPN is an enterprise network based on a shared public network infrastructure but providing the same security, management, and throughput policies as applied in a private network. This shared infrastructure can leverage a service

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 718–728, 2004.

provider's IP, Frame Relay, or ATM backbone network and may or may not utilize the public Internet. The primary advantages of "VPN over Internet" are cost-effectiveness and flexibility while coping with the exponential growth of Internet. However, the current disadvantages are the lack of sufficient QoS and provision of adequate transmission capacity for high bandwidth services. For resolving these problems, OVPNs over the next generation optical Internet (NGOI) have been suggested [1, 2, 3].

Keeping in mind that IETF and ITU-T are standardizing IP/GMPLS over DWDM as a solution for the NGOI, DWDM optical network technology will be used as the NGOI backbone and GMPLS [4] will be used as control protocols for transferring data over IP. Therefore, an OVPN over IP/GMPLS over DWDM is considered as a major trend for next generation VPNs supporting various real-time multimedia services [5].

In a wavelength-routed OVPN backbone network, end users (customer sites) communicate with one another via all-optical DWDM channels, which are referred to lightpaths[6]. Given connection requests, the problem of setting up lightpath by routing and assigning wavelength for each connection so that no two lightpaths on a given link share the same wavelength is called the RWA problem.

The RWA problem is embossed very important as a key role in improving the global efficiency for capacity utilization in DWDM network providing multi-gigabit rate per wavelength, however it is a combinational problem known to be NP-complete because routing and wavelength assignment problems are tightly liked together [7]. Since it was more difficult to work out RWA as coupled problem, this problem has been approximately divided into two sub-problems: routing and wavelength assignment. In previous studies, the routing scheme is recognized as more significant factor on the performance of solution for the RWA problem than the wavelength assignment scheme [8, 9]. Among approaches for the routing problem, DR yields the best performance because DR approaches determine route by considering network's status at the time of connection request [10], on the other hand static routing approaches such as fixed routing (FR) and fixed alternate routing (FAR) set up a connection request on fixed paths without acquiring the information of current network status [11].

In this paper, we propose a new dynamic routing method for establishing optimal O-LSPs in OVPN. This algorithm chooses a route that does not interfere too much with many potential future connection requests and call it the MW-MIPR algorithm. Simulation results show that proposed MW-MIPR algorithm achieves more enhanced blocking probability than DR that yields the best performance among previous RWA algorithms.

In Section 2, an architecture and functional procedure of an OVPN over IP/GMPLS over DWDM is presented. In Section 3, an O-LSP establishment scheme and a new dynamic routing algorithm to solve the RWA problem in the process of optimal O-LSP establishment. Simulation results are described in Section 4 and conclusions are presented in Section 5.

2 Architecture of OVPNs

The suggested OVPN structure is composed of customer sites in the electric control domain and the DWDM-based backbone network in the optical control domain as shown in Fig. 1, respectively. The external customer site is an IP network based on differentiated services (DiffServ) [12, 13]. It aggregates IP packets, which have the same destination client edge nodes (CE) address at the ingress CE to reduce network complexity and to make operation simple. The internal OVPN backbone network is a DWDM network based on GMPLS. It consists of the provider edge nodes (PE) and the provider core nodes (P), and it forwards data traffic from the customer sites without electronic-optic-electronic (E-O-E) conversions. There is a traffic policy server (TP server) for supporting O-LSP connections among customer sites. It negotiates service level agreement (SLA) parameters and sets an optical path according to the negotiated parameters. In this way, it can manage the entire network to support the service that satisfies the SLA through the optical path between end users.

In order to transmit user data transparently through the OVPN optical backbone network, the protocol layer structure should look like that in Fig. 2. The OVPN based on DiffServ suggested in this paper reduces network complexity (1) by gathering IP traffic flows that have the same destination CE address, and (2) by directly mapping the requested service class to the corresponding optical channel calculated by the suggested MW-MIPR algorithm. In the electrical-optical/optical-electrical (E-O/O-E) interface layer, IP packets from the higher layers are sorted according to destination CE address. They are given proper GMPLS labels as to the level of modeled classes. This E-O/O-E interface layer preserves the quality of the optical signals with the bit error rate (BER), electrical signal-to-noise ratio (el.SNR) and optical SNR (OSNR) for guaranteeing end-to-end QoS at the levels of the various classes. The functions are performed by the TP server and the optical resource management agent (ORMA). Furthermore, this layer also guarantees end-to-end QoS at the level of the OCh wavelength by transmitting IP packets transparently through the optical channels.

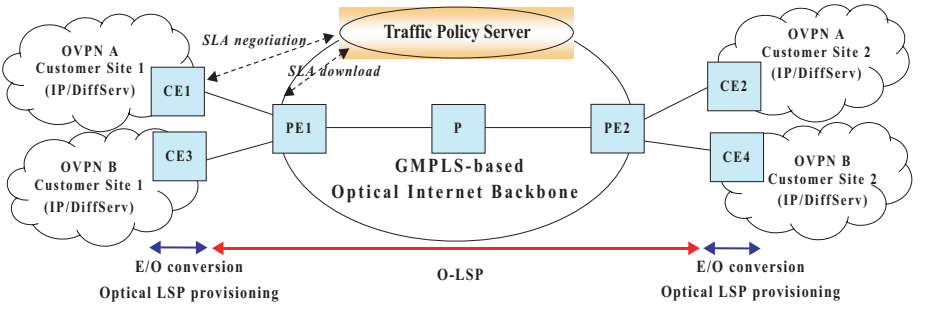


Fig. 1. OVPN model

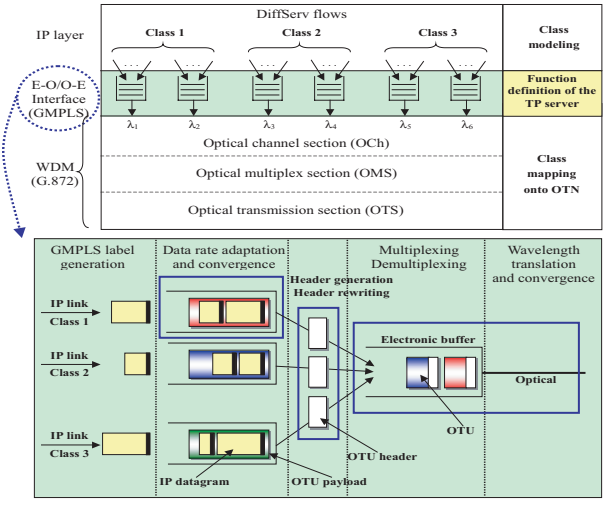


Fig. 2. Differentiated IP service in CE

3 O-LSP Establishment Scheme and MW-MIPR Algorithm

3.1 SLA and O-LSP Establishment

In order to support differentiated optical service through the OVPN backbone network, an implementation of the SLA negotiation procedure between the customer site and the TP server is needed. Fig. 3 depicts the SLA negotiation procedure and the functional blocks in the OVPN node.

First, a policy agent of the CE sends a SLA request that specifies the source and destination IP addresses, the customer port identifier (CPI) and provider port identifier (PPI), the aggregated IP flow information, bandwidth, and QoS parameters. When the TP server receives this request, it verifies the pre-negotiated traffic contract with the OVPN service provider. If it satisfies the traffic contract, then the TP server downloads the SLA parameters onto the policy agent in the appropriate ingress PE (PE1 in Fig. 3) to request a SLA allowance decision, which in turn establishes an O-LSP using the resource reservation protocol with traffic engineering extensions (RSVP-TE+) [14] or the constraint-based routed label distribution protocol with extensions (CR-LDP+) [15].

The policy agent conveys the parameters to the GMPLS signaling agent so that it can establish the GMPLS O-LSP from the ingress PE to the egress PE and can reserve resources along the path. When the GMPLS signaling agent receives a trigger for setting up an O-LSP, it asks the routing agent which uses OSPF extensions in support of GMPLS (OSPF-TE+) [16] or IS-IS extensions in support of GMPLS (IS-IS-TE+) [17] to find the best path to that egress PE router. The address of this egress PE is resolved by using the multiprotocol

extensions of the BGP-4 (MP-BGP) [18] reachability information. At each transit node, where the performance guaranteed path is calculated in the routing agent, the requested bandwidth and specific parameters of class in the message are examined by the call admission control agent (CAC) and the ORMA to see whether or not it is possible to establish the O-LSP. Then it sends the result to the TP server. As soon as the TP server gets the result, it informs the policy agent of the CE that the SLA negotiation had been completed. After SLA negotiation between the customer site and the OVPN backbone network, the GMPLS signaling procedure is started along the performance guaranteed path specified by the routing algorithm (it is fully explained in the next subsection).

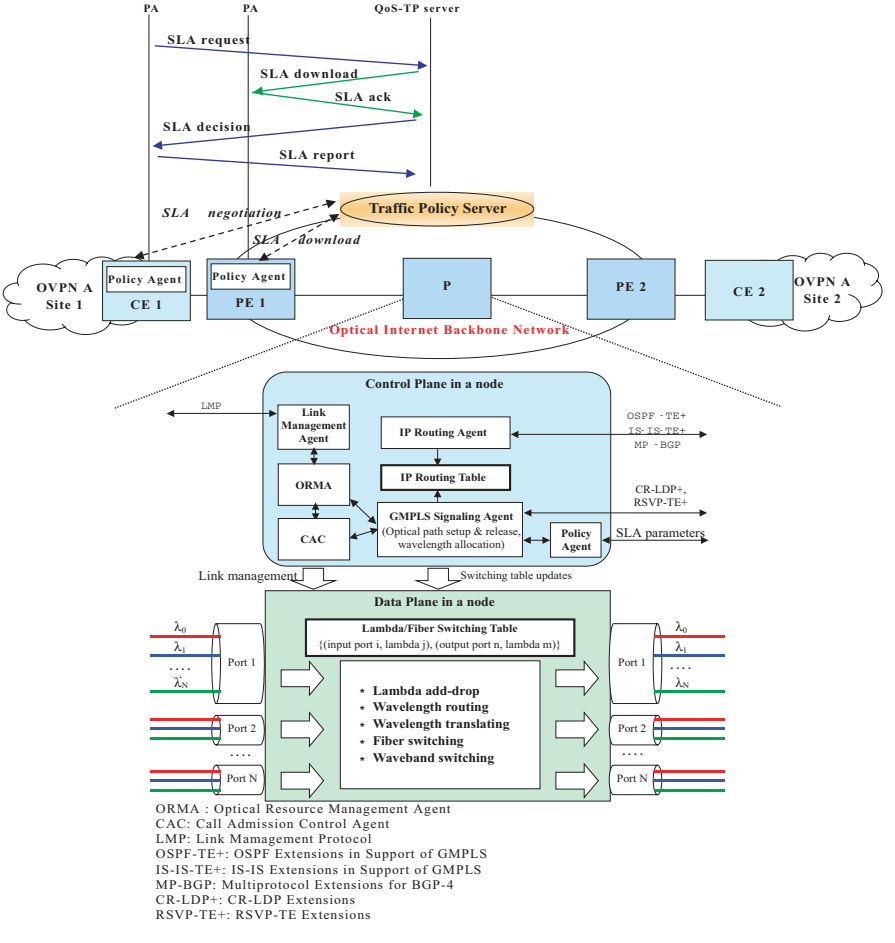


Fig. 3. SLA negotiation procedure and functional blocks in an OVPN node

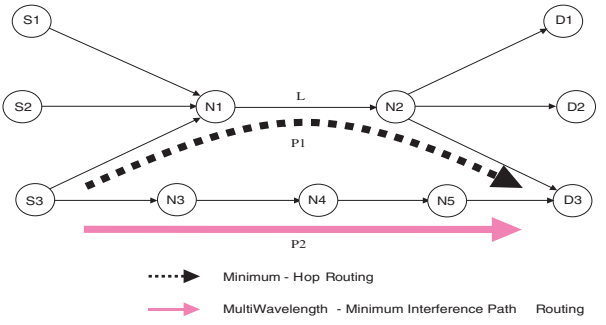


Fig. 4. MultiWavelength-Minimum Interference Path Routing (MW-MIPR)

3.2 MW-MIPR Algorithm for the Routing Agent

In this paper, we propose MW-MIPR algorithm as a new dynamic routing algorithm with consideration for potential block possibility of future traffic demands in OVPN backbone networks based on DWDM with full wavelength conversion (WC) capability and single-fiber. Our work is inspired by previous proposed MIR algorithm estimated as good schemes in terms of traffic engineering because of achieving more improvement in resources utilization as well as in rerouting performance upon a link or node failure than previous proposed routing algorithms in MPLS network [19, 20, 21, 22].

Proposed MW-MIPR algorithm chooses a route that does minimize interference for many potential future connection requests by avoiding congestion links. For an example as shown in Fig. 4, MW-MIPR is to pick route P2 for connection between (S3, D3) pair that has a minimum affect for other connection requests (S1, D1) as well as (S2, D2) even though the path is longer than P1 with a congested link L. Before the description of MW-MIPR algorithm, we define some notations commonly used in this algorithm as follows.

$G(N, L, W)$: The given network, where N is the set of nodes, L is the set of links and W is the set of wavelengths per link. In this graph, W is same for each link l that belongs to L i.e., $\forall\ l \in L$.

M: Set of potential source-destination node pairs that can request connection in future. Let (s, d) denote a generic element of this set.

p_{sd}: The minimum hop lightpath between a (s, d) pair, where $\forall\ (s, d) \in M$.

π_{sd}: Set of links over the minimum hop path p_{sd}.

$R(l)$: The number of currently available wavelengths on a link l, where $\forall\ l \in L$.

Λ_{sd}: The union set of available wavelengths on each link l, where $\forall\ l \in \pi_{sd}$.

F_{sd}: The set of available wavelengths on bottleneck link that has the smallest residual wavelengths among all links within π_{sd} i.e., $\forall\ l \in$

π_{sd} (if all nodes in the network have wavelength-continuity constraint, then F_{sd} is equal to Λ_{sd}).
Ω_{sd}: Set of wavelengths assigned to the minimum hop path p_{sd}.
C_{sd}: Set of critical links for a (s,d) pair, where $\forall\ (s,d) \in M$(links which belong to π_{sd} of a (s,d) pair and are shared on the minimum hop paths of other node pairs at the same time).
α_{sd}: The weight for a (s,d) pair, where $\forall\ (s,d) \in M$.
Δ: A threshold value of available wavelengths on a link(20% – 30% of W).

Based on above notations, MW-MIPR algorithm is assumed that demands arrive one at a time and the current connection request is between a and b nodes, where $(a,b) \in M$.

In DWDM network, the wavelength-continuity constraint can be eliminated if a wavelength converter is at a node [23]. Especially, in the network made of nodes with full WC capability from any wavelength to any other one, a wavelength can be easily assigned if a residual wavelength is on links over selected route [24, 25]. Based on notations such as C_{sd} and Δ, we determine links with congestion possibility for a potential future demand between (s,d) pair in the network with full WC capability as Equation (1), where $\forall\ (s,d) \in M \backslash (a,b)$ and $\forall\ l \in L$, and call them CL_WC_{sd}.

$$CL\_WC_{sd} : (l \in C_{sd}) \cap (R(l) < \Delta), \forall\ (s,d) \in M \backslash (a,b),\ \forall\ l \in L \tag{1}$$

MW-MIPR algorithm gives appropriate weights to each link based on amount of available wavelengths on a link l where $\forall\ l \in L$, so that the current request does not interfere too much with potential future demands. The link weights are estimated by the following procedures. First, let $\partial F_{sd}/\partial R(l)$ indicates the change of available wavelengths on the bottleneck link l for the potential connection request between a (s,d) pair when the residual wavelengths of link l are changed incrementally. With respect to the residual wavelength of the link, the weight $w(l)$ of a link l is set to

$$w(l) = \sum_{(s,d) \in M \backslash (a,b)} \alpha_{sd}(\partial F_{sd}/\partial R(l)), \quad \forall l \in L \tag{2}$$

Equation (2) determines the weight of each link for all (s,d) pairs in the set M except the current request when setting up a connection between the (a,b) pair i.e., $\forall\ (s,d) \in M \backslash (a,b)$, but computing weights of all links is very hard, where $\forall\ l \in L$. To solve this problem, we consider more restricted link than other links for routing with Equation (3) if a link belongs to the set of congestion links for certain (s,d) pair i.e., $\forall\ l \in\ CL\_WC_{sd}$.

$$\begin{cases} \partial F_{sd}/\partial R(l) = 1 & [\ if\ (s,d) : l \in CL\_WC_{sd}\ ] \\ \partial F_{sd}/\partial R(l) = 0 & [\ otherwise\ ] \end{cases} \tag{3}$$

Therefore computing the link weights is simplified as shown in Equation (4). And then if the value of $\alpha_{sd} = 1$ for all (s,d) pairs, $w(l)$ will represents the number of source-destination pairs for which link l is critical.

$$w(l) = \sum_{(s,d):l \in CL\_WC_{sd}} \alpha_{sd} \tag{4}$$

Based on above formulations, the formal description of the MW-MIPR algorithm in the network with full WC capability is given as fallows.

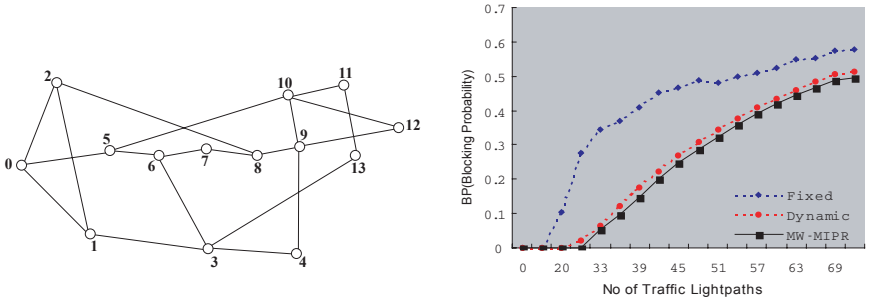
MW-MIPR$(L, M, w(l), \alpha_{sd}, R(l), C_{sd}, CL_WC_{sd})$
(1) If connection is requested between a~node pair (a, b) then {
(2) For each link l, where $\forall\ l \in L$ {
(3) link weight $w(l) = 0$
(4) If $R(l) < \Delta$ then {
(5) For each node pair (s, d), where $\forall\ (s, d) \in M \backslash (a, b)$ {
(6) node pair weight α_{sd}
(7) If $l \in C_{sd}$ then {
(8) $CL\_WC_{sd} := CL\_WC_{sd} \cup l$
(9) $w(l) := w(l) + \alpha_{sd}$ } } } }
(10) Remove a~link l from L with $R(l) = 0$ }
(11) Choose the minimum hop path with the smallest $w(l)$ using the Dijkstra's algorithm

Once the weight of each link l where $\forall\ l \in L$ is determined, MW-MIPR routes the current traffic between (a, b) pair along the path with the smallest $w(l)$. If there is a tie, then min-hop path routing will be used to break the tie.

4 Simulation Results

In this section, simulations are carried out to evaluate the performance of MW-MIPR algorithm. To prove the efficiency of MW-MIPR algorithm proposed in section 3, we analyze the blocking performance of MW-MIPR in case of network with or without wavelength converters and compare with that of the existing FR and DR algorithm via simulations. The network topology of OVPN backbone used in simulations is NSFnet that is consisted of 14 nodes, 20 links and a single fiber as shown in Fig. 5(a). This topology is currently used for WDM network model in USA and adopted in most of papers relation to WDM networks. We choose 5 pairs as the set of potential source-destination node pairs M for our simulations.

We compare proposed MW-MIPR to the existing routing (FR and DR) algorithms in network with full WC capability. Generally, the restriction imposed by the wavelength continuity constraint can be removed by the use of wavelength converter. Wavelength converters thus play an important role in enhancing the resource utilization and reducing the overall call blocking probability of the network. Fig. 5(b) shows the comparison of FR, DR and MW-MIPR algorithm in network with full WC capability. Proposed MW-MIPR algorithm performs better than all others. And then, proposed MW-MIPR algorithm has the lower blocking probability than DR (improved by about 15 %.) because of selecting the minimum interference path with potential future setup requests.



(a) 14-node NSFnet (b) Comparision of FR, DR, MW-MIPR

Fig. 5. Simulation for performance of MW-MIPR algorithm

5 Conclusion

In this paper, we suggest a new routing algorithm for establishing optimal O-LSPs in OVPN to provide more improved performance for connection blocking probability with consideration for potential future network's congestion status. The proposed algorithm improves wavelength utilization by choosing route that does not interfere too much with many potential future connection requests. In the simulated results, proposed MW-MIPR algorithm achieves more improvement in blocking probability than previous routing algorithms used in DWDM network regardless of case that wavelength converter at node is present or not. In future research, it is needed to study MW-MIPR algorithm based on sparse wavelength conversion with regard for the impact of the location or the number of wavelength converters. And it is also required to envisage specific functional extensions and interoperation among many control protocols (MP-BGP, OSPF-TE+/IS-IS-TE+, RSVP-TE+/CR-LDP+, LMP) in an OVPN environment that guarantees the optimal O-LSP establishment.

Acknowledgement

This work was supported by grant No. (R01-2003-000-10526-0) from Korea Science & Engineering Foundation.

References

[1] H. Ould-Brahim et al.: Generalized Provider-provisioned Port-based VPNs using BGP and GMPLS Toolkit, draft-ouldbrahim-ppvpn-gvpn-bgpgmpls-03.txt, IETF Internet Draft, (Mar. 2003) 719

[2] Tomonori Takeda: Layer 1 Virtual Private Network Generic Requirements and Architectures, ITU-T Draft Rec. Y.l1vpnsdr, (Nov. 2002) 719

[3] Y. Qin et al.: Architecture and analysis for providing virtual private networks with QoS over optical WDM networks, Optical Network Magazine, vol. 2, no. 2, (April 2001), pp. 57–65 719
[4] Eric Mannie: Generalized Multi-Protocol Label Switching (GMPLS) Architecture, draft-ietf-ccamp-gmpls-architecture-07.txt, IETF Internet Draft, (May 2003) 719
[5] Mi-Ra Yoon et al.: Optical-LSP Establishment and a QoS Maintenance Scheme Based on Differentiated Optical QoS Classes in OVPNs, Photonic Network Communications, to be published 719
[6] I. Chlamtac et al.: Lightpath Communications: An Approach to High-Bandwidth Optical WAN's, IEEE Transactions on Communications, vol. 40, no. 7, (July 1992) 719
[7] J. S. Choi et al.: Classification of Routing and Wavelength Assignment Schemes in DWDM Networks, OPNET'00, (Jan. 2000), pp. 1109–1115 719
[8] H. Zang et al.: A Review of Routing and Wavelength Assignment Approaches for Wavelength Routed Optical WDM Networks, Optical Networks Magazine, vol. 1, no. 1, (Jan. 2000), pp. 47–60 719
[9] S. Ramamurthy et al.: Fixed-Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks, IEEE GLOBECOM' 98, vol. 4, (Nov. 1998) 719
[10] J. S. Kim and D. C. Lee: Dynamic Routing and Wavelength Assignment Algorithms for Multifiber WDM Networks with Many Wavelengths, ECUMN 2002 (April 2002), pp. 180–186 719
[11] L. Li et al.: Dynamic Wavelength Routing Using Congestion and Neighborhood Information, IEEE/ACM Trans. Networking, vol. 7, no. 5, (Oct. 1999), pp. 779–786 719
[12] S. Blake et al.: An Architecture for Differentiated Services, IETF RFC 2475, (Dec. 1998) 720
[13] Jigesh K. Patel, Sung U. Kim and David H. Su: QoS Recovery Schemes Based on Differentiated MPLS Services in All-Optical Transport Next Generation Internet, Photonic Network Communications, vol. 4, no. 1, (Jan. 2002), pp. 5–18 720
[14] L. Berger: GMPLS Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions, IETF RFC 3473, (Jan. 2003) 721
[15] P. Ashwood-Smith and L. Berger: GMPLS Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions, IETF RFC 3472, (Jan. 2003) 721
[16] K. Kompella, Y. Rekhter: OSPF Extensions in Support of Generalized MPLS, draft-ietf-ccamp-ospf-gmpls-extensions-09.txt, IETF Internet Draft, (Dec. 2002) 721
[17] K. Kompella, Y. Rekhter: IS-IS Extensions in Support of Generalized MPLS, draft-ietf-isis-gmpls-extensions-16.txt, IETF Internet Draft, (Dec. 2002) 721
[18] T. Bates et al.: Multiprotocol Extensions for BGP4, IETF RFC2858, (June 2000) 722
[19] K. Kar et al.: Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications, IEEE JSAC, vol. 18, no. 12, (Dec. 2000), pp. 2566–2579 723
[20] K. Kar et al.: MPLS traffic engineering using enhanced minimum interference routing: an approach based on lexicographic max-flow, IEEE IWQOS'00 (June 2000), pp. 105–114 21 723
[21] D. Bauer: Minimum-interference routing based on flow maximization, Electronics Letters, vol. 38, no. 8, (Apr. 2002), pp. 364–365 723

[22] S. Suri et al.: Profile-Based Routing: A New Framework for MPLS Traffic Engineering, Washington University Computer Science Technical Report WUCS-00-21, (July 2000) 723
[23] M. Frey et al.: Wavelength Conversion and Call Connection Probability in WDM networks, IEEE Transactions on Communications, vol. 49, no. 10, (Oct. 2001) 724
[24] C. B. Ahmed et al.: QoS Routing with Wavelength Conversion and Call Admission Connection in DWDM Networks, IEEE ICCNMC'01 (Oct. 2001), pp. 61–66 724
[25] Jing Fang et al.: Performance Analysis of WDM Optical Networks with Wavelength Usage Constraint, Photonic Network Communications, vol. 5, no. 2, (Mar. 2003), pp. 137–146 724

The Effect of Burst Assembly on Performance of Optical Burst Switched Networks

JungYul Choi[1], Hai Le Vu[2], Craig W Cameron[2], Moshe Zukerman[2], and Minho Kang[1]

[1] Optical Internet Research Center, Information and Communications University
P.O. Box 77, Yusong, Daejon, Korea
{passjay,mhkang}@icu.ac.kr
[2] Centre for Ultra-Broadband Information Networks, Department of Electrical and Electronic Engineering, The University of Melbourne, VIC 3010, Australia
{h.vu,c.cameron,m.zukerman}@ee.mu.oz.au

Abstract. We evaluate and compare the performance of timer-based and threshold-based assembly algorithms in Optical Burst Switching networks. Results including burst blocking probability, mean packet delay and link utilization at the ingress node are presented from both simulations and two theoretical models. The results are obtained for the full range of input traffic load so they can provide guidelines for design and dimensioning links to meet desired Quality of Service levels.

1 Introduction

Optical Burst Switching (OBS) has recently been proposed as a future high-speed switching technology for Internet Protocol (IP) networks that may be able to efficiently utilize extremely high capacity links without the need for data buffering or optical-electronic conversions at intermediate nodes [8]. Packets arriving at an OBS ingress node that are destined for the same egress OBS node and belong to the same Quality of Service (QoS) class are aggregated and sent in discrete bursts, at times determined by the burst assembly policy. At intermediate nodes, the data within the optical signal is not processed but instead, the whole burst is transparently switched according to directives contained within a control packet preceding the burst. At the egress node, the burst is subsequently de-aggregated and forwarded electronically. Unlike classical circuit switching, contention between bursts may cause blocking and make it consequent loss within the network. Jointly minimizing blocking probability and maximizing throughput is the main goal of OBS research. The aim of this paper is to explore the impact of burst assembly algorithms on this optimization objective.

Recently, several burst assembly algorithms have been introduced: timer-based [4], threshold-based [7] and hybrid [6][9][11]. Recent research into burst assembly algorithms has focused mainly on its effect on the Long Range Dependence of the traffic. The Long Range Dependence of self-similar Internet traffic was initially shown to be reduced by applying burst assembly at an ingress

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 729–739, 2004.

node [4]. However, it was later demonstrated that the Long Range Dependence was unaffected by burst assembly [10]. Further work showed potential link-utilization improvements by leveraging the Long Range Dependence of input traffic in choosing adaptive values of the timer and the threshold. The interaction between burst assembly and Transmission Control Protocol (TCP) has also been studied. It was suggested that the TCP performance is more sensitive to the timer value than the threshold value and that the assembly period should equal the TCP window size [2][3]. In [5], increasing burst size was shown to result in higher TCP throughput but only at low blocking probabilities.

However, as far as the authors are aware, the performance impact of different assembly algorithms from packet level input to output bursts has not yet been explicitly explored. In this paper, we compare diverse burst assembly algorithms and study their effect on blocking probability, delay, and utilization with simulations compared with two analytical models. While the impact of the timer value and threshold value on delay is reasonably straight forward, consequent network utilization and blocking probability is more complex and is the subject of the remainder of the paper.

2 Burst Assembly

Burst assembly is a mechanism to aggregate incoming input traffic to create a suitable sized burst for transmission through the optical network. This mechanism can be modelled as a queuing system with a separate queue for each egress node/QoS pair and a shared output link. Two key parameters determine how a data burst is aggregated: the maximum waiting time (timer value) and the minimum size (threshold value) of the burst. Based on these two parameters, burst assembly algorithms can be categorized as *timer-based* [4] and *threshold-based burst assembly* [7] or *hybrid* [10][11], a combination of the two.

2.1 Timer-Based

In timer-based assembly, a timer is started at the initialization of system and immediately after the previous burst is sent. At the expiration of this timer, the burst assembler generates a burst containing all the packets in the buffer at that point. Under low input offered load, this scheme guarantees the minimum delay for burst assembly. However, under high input offered load, it may generate bursts that are quite large, perhaps unnecessarily increasing delay.

2.2 Threshold-Based

In threshold-based burst assembly, a burst is generated when a number of packets in a buffer arrives at a threshold value. In terms of delay, the performance of this scheme is completely the opposite to the timer-based scheme described above. Under low input offered load, this scheme may need to wait for a long period of time until the buffer threshold is reached. However, under high input offered load, the threshold will be reached quickly, minimizing delay.

2.3 Hybrid

A hybrid scheme applies algorithms concurrently [10][11]. Both a timer and a threshold parameter are set and the burst sent at the earlier of the timer expired and threshold reached events. The two values in a burst assembly can also be varied according to the input traffic load. This extension is called *an adaptive hybrid burst assembly* [6].

3 Single Node Model

In order to make queuing networks amenable to analysis, assumptions and approximations must be employed. A popular simplified traffic model is to assume that the bursts follow a Poisson process, leading to the classical Erlang queuing system. However, in many cases, the number of input sources generating bursts, which contend for a group of wavelength channels at the output of an OXC, may be small relative to the number of output wavelengths [12]. In this case, the Poisson model overestimates the loss probability. Another simple model is the Engset loss system which has a limited number of input sources [1]. The corresponding Engset loss formula is

$$B = \frac{\binom{M-1}{K}\rho^k}{\sum_{i=0}^{K}\binom{M-1}{i}\rho^i} \tag{1}$$

where B is the blocking probability, M is the number of input links, K is the number of output links, and ρ is the average intensity of the free input links. However, as discussed in [12], the Engset model may not model OBS networks accurately.

In this paper, we consider a model of a single OXC with a finite number of sources where each generates an on/off input stream (burst). Fig. 1 shows a model of an ingress node with limited number of sources and with one burst assembler for each source. Input traffic packets arrive at the ingress node and are queued with packets destined for the same egress node. The generated data burst are scheduled in the scheduler in order to forward them to the output links. Input traffic is modelled as with λ_p and μ_p. Note that these parameters correspond to the arriving IP packet process. After an ingress node performs burst assembly, the output traffic (the generated bursts) from the burst assembler in each input link is modelled as an input source toward the scheduler with burst arrival rate λ and service rate μ. Therefore, the on and off periods of each source have means of $1/\mu$ and $1/\lambda$, respectively. Let $\rho = \lambda/\mu$.

Using the model from [12], we consider a two dimensional Markov chain assuming exponential on and off source, the output of burst assembler, in order to model the OBS networks. There are three types of customers: (1) busy (bursts that are being transmitted), (2) free (empty input link), and (3) blocked (bursts that being dumped). The sum of the three types is always M, thus the number

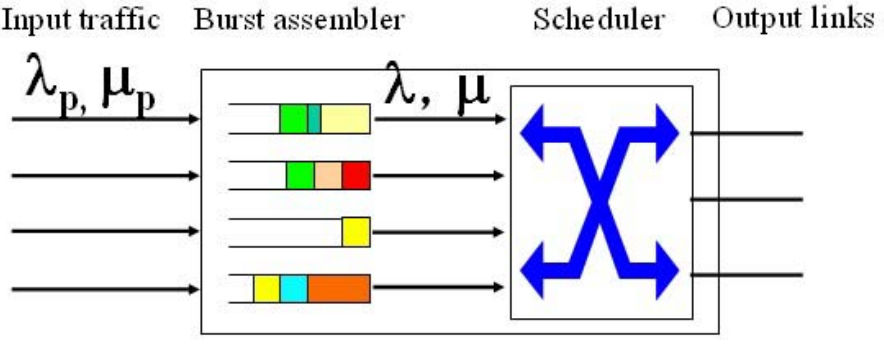


Fig. 1. A Model of an Ingress Node with Burst Assembler

of the free customers is always M minus the other two types. Accordingly, let $\pi_{i,j}$ be the steady state probability where $i(0 \leq i \leq K)$ is the number of busy customers and $j(0 \leq j \leq M-K)$ is the number of frozen customers (sources who transmit blocked bursts). We have the following steady state equations: For $i = 0, 1, 2, ..., K-1$ we have

$$[(M-i-j)\lambda + (i+j)\mu]\pi_{i,j} = (M-i+1-j)\lambda\pi_{i-1,j} + (j+1)\mu\pi_{i,j+1} + (i+1)\mu\pi_{i+1,j}. \quad (2)$$

and

$$[(M-K-j)\lambda + (K+j)\mu]\pi_{K,j} = (M-K+1-j)\lambda\pi_{K-1,j} + (j+1)\mu\pi_{K,j+1} + (M-K+1-j)\lambda\pi_{K,j-1}. \quad (3)$$

For brevity, in (2) and (3) $\pi_{i,j}$ values out of the range $0 \leq i \leq K$ and $0 \leq j \leq M-K$ take the value zero.
Then we also have the normalization equation:

$$\sum_{i=0}^{K}\sum_{j=0}^{M-K}\pi_{i,j} = 1. \quad (4)$$

Since the number of frozen customers cannot be more than M-K, as a customer cannot become frozen if there are less than K busy customer, the offered load is given by

$$T_o = \sum_{i=0}^{K}\sum_{j=0}^{M-K}(M-1-j)\rho\pi_{i,j}, \quad (5)$$

the carried load is given by

$$T_c = \sum_{i=0}^{K} \sum_{j=0}^{M-K} i\pi_{i,j}, \tag{6}$$

and the blocking probability is obtained by

$$B = \frac{T_c - T_o}{T_o}. \tag{7}$$

3.1 Timer and Threshold-Based Models

The above model is burst assembly algorithm invariant and therefore is insufficient for analyzing possible differences between timer and threshold-based algorithms. In both of these two cases, there is a deterministic element not present in the probabilistic Engset model: threshold-based has a fixed burst size but random time interval, timer-based has a random burst size but fixed time interval.

Several simulations were run to evaluate the performance impact of the timer and threshold-based algorithms with the following parameter settings: Number of input links = 6, Number of output links = 3, Capacity of input and output links = 1Gbps. The packet arrivals were modelled by a Poisson process and had exponentially distributed sizes with mean = 1Kbyte. The scheduling algorithm LAUC (Lastest Available Unscheduled Channel) [8] was used to place the bursts on the output links. The simulations were run 10 times, with the number of packets in each simulations ranging from one to eight million. A range of sizes was used to ensure that simulations with longer timers and larger thresholds had similar number of bursts and therefore similar levels of accuracy. 95 percent confidence intervals, based on the Student-t distribution, are shown in the results if large enough to be visible.

As the packet sizes are assumed independent, the variance of the sum of k packet sizes is equal to k times the variance of the size of a single packet. By the central limit theorem, as k increases, the resulting queue size distribution increasingly becomes Gaussian-like with a standard deviation of the order of $\sqrt{k}$. Hence, for large k, the threshold based model can be approximated by a gaussian with small standard deviation and a mean equal to the threshold value. It was shown in [12] that models using Gaussian on and off times give slightly higher blocking probabilities than exponential, hence it was expected that packet-based simulations would also give higher blocking probabilities than the model described in Section 3.

Note that the simulations were run on only a single node. An extension to a network of nodes is currently in progress and will be the subject of a future paper.

4 Results

The two main results are presented in Fig. 2 and Fig. 5. The upper thick line is the blocking probability obtained by the Engset loss formula using (1). The

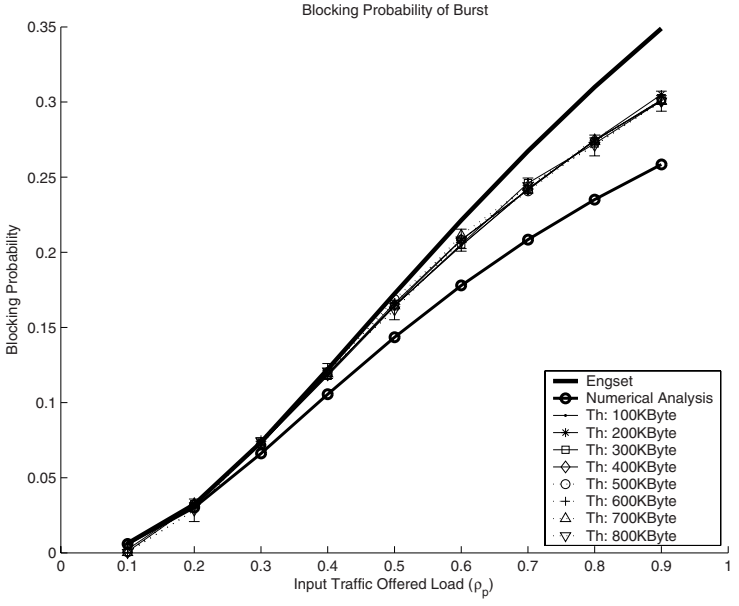


Fig. 2. Blocking Probability of Block: Threshold-based

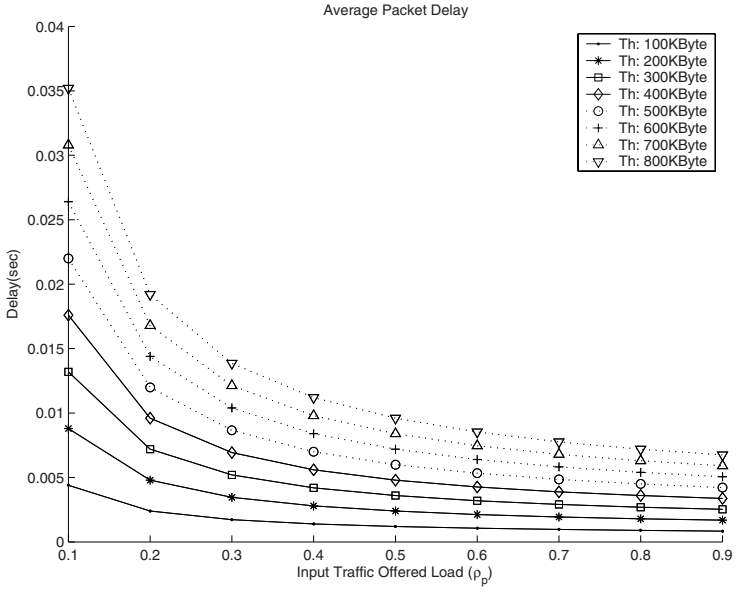


Fig. 3. Average Packet Delay: Threshold-based

lower thick dotted line is the blocking probability from the numerical analysis using (7). In general, the Engset overestimated the blocking probability and the numerical analysis underestimated the blocking probability. This is consistent with results obtained in [12] for the case where the on and off periods are Gaussian distributed. We present utilization results in Fig. 4 and Fig. 7 for completeness.

4.1 Threshold-Based

Fig. 2 shows the burst blocking probability according to threshold values of 100Kbytes to 800Kbytes. Over that range, the blocking probability was threshold size invariant. This is explained by the fact that for threshold-based systems, the parameter of interest (burst size) is deterministic, therefore scaling the system does not change the statistics of the output burst process for a constant load. For example, if we double the threshold and define the time unit as the burst transmission time, the number of burst served per time unit is still one, and the number of burst arrivals per time unit is as before. Thus, we have exactly the same system with exactly the same blocking probability. Fig. 3 shows the average packet delay. As expected from the discussion in Section 2.2, large threshold sizes and low load yield long average delays while small threshold sizes and high load yield short average delays. Fig. 4 shows link utilization that implies output link capacity occupancy of data burst. Under low load of input traffic link utilization yields low efficiency while it yields high efficiency under high load even though there are high blocking probability.

4.2 Timer-Based

Fig. 5 shows the blocking probability of the bursts according to timer values of 1 ms to 8 ms. Unlike the threshold-based method, changing the timer value had a substantial effect on the blocking probability: for low load, longer timers give lower blocking probabilities but for high load, longer timers give higher blocking probabilities. This suggests care must be taken to choose appropriate parameters that match the expected load into the network if a timer-based method is to be employed successfully. In contrast to threshold-based systems, in this case burst sizes are random variables, therefore scaling the system does change the statistics as the variance and higher order moments do not scale linearly, unlike the mean which was the only parameter of interest in the threshold-based system. The blocking probability of Engset loss formula shows intermediate values between that of larger timer value and that of smaller timer value. Result of numerical analysis still presents the lower bound of blocking. Fig. 6 shows the average size of the bursts as a function of the load. As expected from the discussion in Section 2.1, low load yields smaller burst sizes while high load yields larger burst sizes. The burst sizes also increase approximately linearly with timer length. Fig. 7 shows link utilization that smaller timers give high utilization over all range of offered load except only low load. It implies that smaller size of

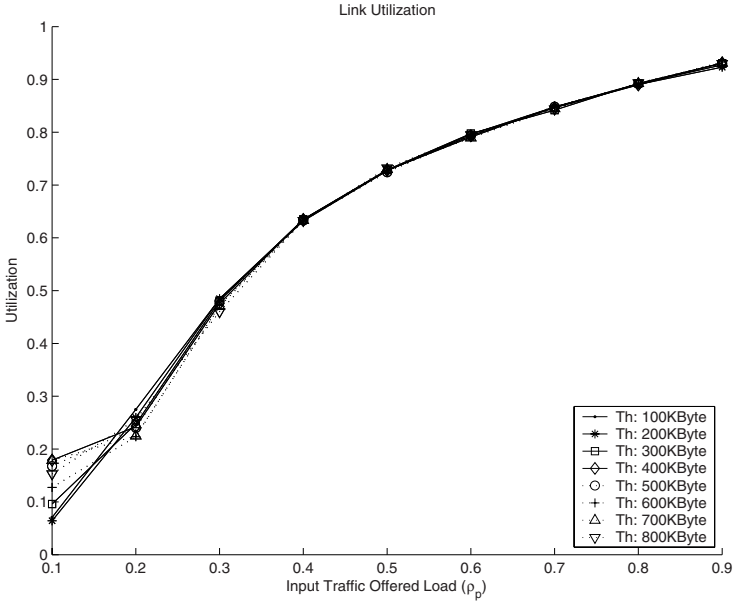


Fig. 4. Link Utilization: Threshold-based

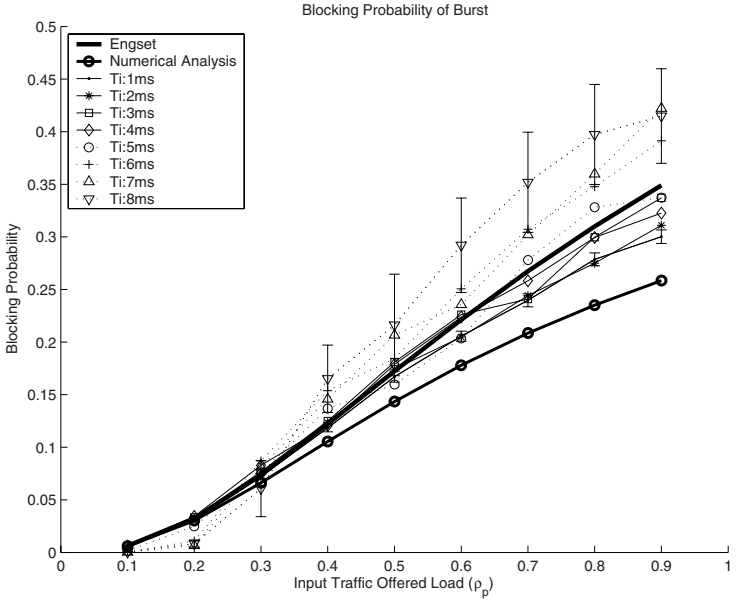


Fig. 5. Blocking Probability of Burst: Timer-based

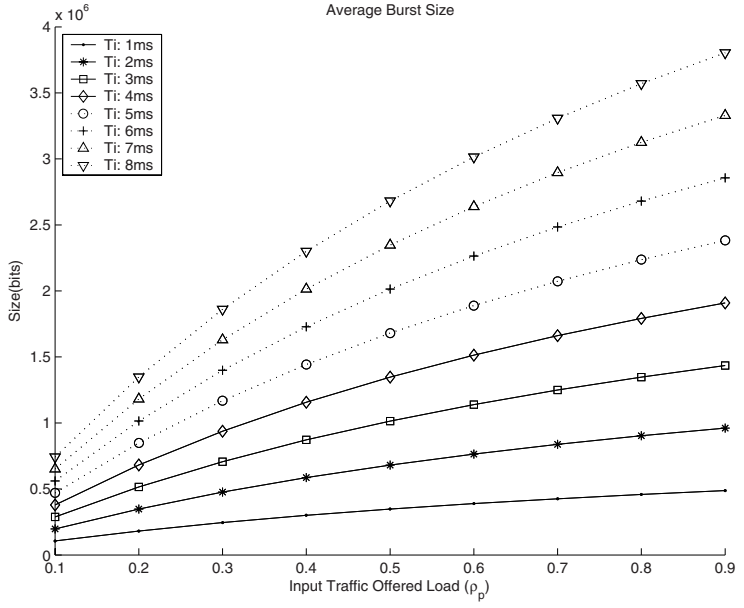


Fig. 6. Average Burst Size: Timer-based

burst is efficiently transmitted through the OBS networks in terms of blocking probability and link utilization.

5 Conclusion

In this paper, we studied the performance of timer and threshold-based burst assembly algorithms in OBS networks and compared the simulation results with two theoretical models: the classical Engset model and an analytical blocking model. The blocking probability was found to be threshold value invariant. But for low load, longer timers yielded lower blocking probabilities and for high load, longer timers yielded higher blocking probabilities. Finally, smaller size of burst shows mostly better performance in terms of link utilization and blocking probability over all range of load except very low load for timer-based burst assembler, as well as for threshold-based even through its effect is very slight.

Acknowledgements

This work was supported in part by the KOSEF-OIRC project, Samsung project, and the Australian Research Council.

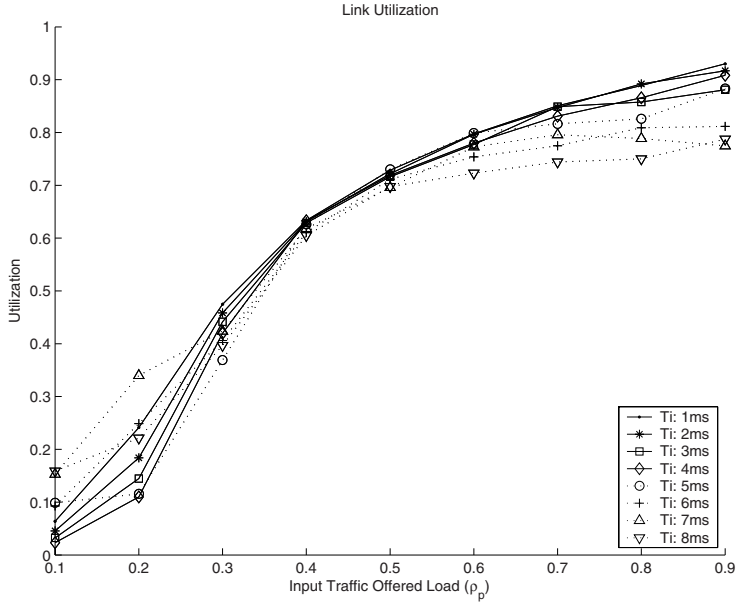


Fig. 7. Link Utilization: Timer-based

References

[1] H. Akimaru and K. Kawashima, *Teletraffic-Theory and Application*, ISBN 3-540-19805-9, 2nd ed., Springer, London 1999. 731

[2] X. Cao, J. Li, Y. Chen and C. Qiao, Assembling TCP/IP Packets in Optical Burst Switched Networks, *IEEE Globecom 2002*. 730

[3] A. Detti and M. Listanti, Impact of Segments Aggregation on TCP Reno Flow in Optical Burst Switching Networks, *Proc. IEEE Infocom 2002*. 730

[4] A. Ge, F. Callegati and L. S. Tamil , On Optical Burst Switching and Self-Similar Traffic, *IEEE Communications Letters*, vol. 4, no. 3, March 2000. 729, 730

[5] S. Gowda, R. K. Shenai, K. M. Sivalingam and H. C. Cankaya, Performance Evaluation of TCP over Optical Burst-Switched (OBS) WDM Networks, *Proc. IEEE ICC 2003*. 730

[6] S. Oh, H. Hong and M. Kang, A Data Burst Assembly Algorithm in Optical Burst Switching Networks, *ETRI Journal*, vol. 24, August 2002, pp. 311-322. 729, 731

[7] V. M. Vokkarane, K. Haridoss and J. P. Jue, Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks, *Proc. Opticom 2002*. 729, 730

[8] Y. Xiong, M. Vandenhoute and H. C. Cankaya, Control Architecture in Optical Burst-Switched WDM Networks, *Journal of Selected Areas in Communications*, vol. 18, no. 10, October 2000. 729, 733

[9] J. White, M. Zukerman and H. L. Vu, A Framework for Optical Burst Switching Network Design, *IEEE Comm. Letters*, vol. 6, no. 6, June 2002. 729

[10] X. Yu, Y. Chen and C. Qiao, Performance Evaluation of Optical Burst Switching with Assembled Burst Traffic Input, *Proc. IEEE Globecom 2002*. 730, 731

[11] M. C. Yuang, J. Shil and P. L. Tien, QoS Burstification for Optical Burst Switched WDM Networks, *Proc. OFC 2002.* 729, 730, 731

[12] M. Zukerman, E. Wong, Z. Rosberg, G. Lee and H. L. Vu, On Teletraffic Applications to OBS, *IEEE Comm. Letters*, vol. 8, no. 2, February 2004. 731, 733, 735

Optical Hybrid Switching – Combined Optical Burst Switching and Optical Circuit Switching

Gyu Myoung Lee[1], Jun Kyun Choi[1], Bartek Wydrowski[2], Moshe Zukerman[2], and Chul-Hee Kang[3]

[1] Information and Communications University (ICU)
103-6, Munji-dong, Youseong-ku, Daejeon,Korea
{gmlee,jkchoi}@icu.ac.kr
[2] ARC Special Research Centre for Ultra-Broadband Information Networks
EEE Department, The University of Melbourne
Victoria 3010, Australia
{b.wydrowski, m.zukerman}@ee.mu.oz.au
[3] Korea University
1,5-ka, Anam-dong, Sungbuk-ku, Seoul, Korea
chkang@widecomm.korea.ac.kr

Abstract. In this paper, we propose a new optical hybrid switching technique which combined Optical Burst Switching (OBS) and Optical Circuit Switching (OCS) using flow classification. In particular, this switching technique classifies incoming IP traffic flows into short-lived and long-lived flows for Quality of Service (QoS) provisioning according to traffic characteristics. The aim is to maximize network utilization while satisfying user's QoS requirements. We model the system as a single server queue in a Markovian environment. The burst generation process is assumed to follow a two-state Markov Modulated Poisson Process (MMPP), and the service rate fluctuates based on the number of concurrent OCS sessions. Results for the delay and OBS burst assembly time are derived.

1 Introduction

Optical network technologies are evolving rapidly in terms of multiplexing bandwidth and control capability. There has been considerable attention given to IP over optical networks to combine the optical and the electronic worlds by network service providers, telecommunications equipment vendors, and standards organizations.

From the optical switching technology point of view, Optical Burst Switching (OBS) technology has been emerging to utilize resources and transport data more efficiently than the existing circuit switching [1]-[4]. OBS is accepted as an alternative switching technology due to the limitation of optical devices that do not support buffering. Another option is to use the so called hybrid switching which combines Optical Circuit Switching (OCS) and OBS [5].

In this paper, we consider a combined OCS and OBS system and propose an analytical performance model of an OCS/OBS switch. We propose a new optical

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 740–749, 2004.

hybrid switching system using a flow classification technique. This technique classifies incoming IP traffic flows into short-lived and long-lived flows for QoS provisioning according to traffic characteristics in an optical hybrid switching environment. Short-lived flows are composed of a few packets and long-lived traffic typically indicate delay-sensitive real-time streams that are better suited for circuit (or wavelength) switching. The aim is to maximize network utilization while satisfying users' QoS requirements.

The remainder of the paper is organized as follows. In Section 2, we propose a new switching scheme in optical networks and introduce QoS provisioning mechanism in this system. Then, in Section 3, we describe an analytical model. Finally in Section 4, we give numerical results of the technique.

2 Optical Hybrid Switching Scheme

We propose a new hybrid switching technique using flow classification in optical edge router shown in Figure 1. For QoS provisioning according to traffic characteristics, an incoming IP traffic flows are classified into short-lived and long-lived flows. The specific classification mechanism uses the existing adaptive flow classification [6]. For short-lived traffic flows, we use OBS to achieve better bandwidth utilization because it allows statistical sharing of each wavelength among bursts that may otherwise consume several wavelengths. So, these flows are performed per class burst assembling process and then data burst is created. On the other hand, for long-lived traffic flows such as video streaming, we consider the aggregation of these flows into aggregated flows for optical circuit/wavelength switching. Flow aggregator performs traffic aggregation according to flow characteristics. These aggregate flows require buffering and scheduling because flows are grouped together subject to specific constraints such as QoS class and destination.

In the case of short-lived traffic flows, data burst is created in burst assembler module which has a separate buffer per class and generates Burst Control Protocol (BCP) packet. And then the scheduling and the class-based resource reservation function are simultaneously performed. The scheduler performs the class-based priority queuing. In resource reservation, higher priority bursts are assigned longer offsets than lower priority bursts using BCP [7]. Finally, after electro-optical conversion, data burst cut through intermediate nodes without being buffered. This increases the utilization of network through OBS for short-lived flows.

In the case of long-lived traffic flows, we assume that these flows have the highest priority and a great influence on network performance. These flows are aggregated in flow aggregator and then the QoS and resource constraints of aggregated flows which are related to traffic parameters and available wavelengths are checked. The admitted traffic flows are allocated the requested resource through static Routing and Wavelength Assignment (RWA) [8] which is executed off-line with average traffic demands and pre-determined shortcut route. Finally, shortcut circuit is established and after electro-optical conversion, these flows are transmitted.

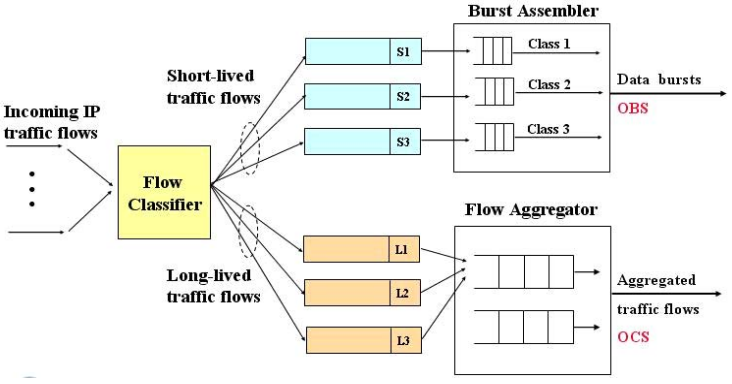


Fig. 1. Optical hybrid switching using flow classification in optical edge router

In the above proposed mechanism, the aim is to maximize network utilization while satisfying user's QoS requirements in a hybrid switching environment taking advantage of both OBS and circuit switching technologies.

3 Numerical Analysis

For node analysis of this system, the queueing model is based on the assumption that the service capacity is made of an integer number of identical units. In principle, this could represent a bottlenecked link made of many wavelengths. The load on this bottlenecked link is made on OCS as well as OBS traffic. The entire service capacity remaining from the OCS usage is used to serve the entire OBS queued traffic in accordance with a single combined server queue (SSQ) model. The bursts in our SSQ are those competing for service in our bottlenecked link. Note that bursts in buffers which are not served by our bottlenecked link are not considered as part of the traffic here.

3.1 The Short-Lived Traffic Generation Model for OBS

The generation process of OBS bursts is modeled using a two state Markov Modulated Poisson Process (MMPP). The MMPP process captures the bursty nature of short-lived traffic. A two state MMPP is an alternating Markovian process with two burst generation states, where the generation process in generation state m is a Poisson process with rate:

$$\lambda_m, \ m = 0, 1.$$

The sojourn time in each generation state is exponentially distributed with the mean sojourn time in generation state 0 and 1 being r_0^{-1} and r_1^{-1} respectively. The values of the mean and the variance of MMPP traffic are given in [9].

The queueing process of the OBS bursts can be represented as an SSQ process whereby the bandwidth available to OBS bursts in this SSQ is dependent upon the number of OCS sessions active on the hybrid switching system, and it fluctuates in accordance with OCS traffic loading. We are assuming here that the order of service of the various bursts at different location does not affect the average burst delay.

3.2 The Long-Lived Traffic Generation Model for OCS

A long-lived traffic stream is allocated a channel for the duration of the connection. The long-lived traffic stream sends packets at a constant rate for duration of the connection. Admission of a long-lived traffic stream decreases the available capacity for the traffic stream and the completion of a long-lived traffic stream increases the capacity available for the traffic stream.

The admission and completion of the long-lived traffic stream OCS connections is modeled as an M/M/k/k process, where k is the maximum number of connections supported by the system. The long-lived traffic connection arrival process is Poisson with parameter λ_c and the connection holding time is assumed exponential with mean $1/\lambda_c$.

3.3 Queueing Analysis

We have modeled optical hybrid switching as a queue in a Markovian environment, and we follow Neuts' analysis of such a queueing model as described in [10] pages 254-264. The infinitesimal generator is first introduced here to describe the system, then, we can calculate our qeueueing problem using Neuts' solution.

We will assume that the total link capacity C is divided into c units of capacity s, $s = C/c$, where s is the capacity used by a single long-lived OCS connection. Of the c units of capacity, d units are reserved exclusively for short-lived OBS bursts. Therefore only $c - d$ capacity units are available for long-lived OCS connections. The state of the system under consideration is denoted by the three dimensional vector (i, j, m) where i is the number of short-lived OBS bursts in the queue (including the one in service), j is the number of capacity units available for bursts, m is the arrival state (m takes the values 0 and 1). The OBS burst service rate is always equal to $j\mu$, where μ is the service rate provided by one capacity unit, and $j = d, d+1, d+2, ..., c$. The OBS burst arrival rate is, $m = 0, 1$. All the possible state transitions are presented in Figure 2. Hence, Figure 2 defines the infinitesimal generator matrix G.

The sum of the entries in each row of G is 0. Let be the probability of being in state (i, j, m), the vector is:

$$(\hat{h}_{0,d,0}, \hat{h}_{0,d,1}, \hat{h}_{0,d+1,0}, \hat{h}_{0,d+1,1}, ...,$$
$$\hat{h}_{0,c,1}, \hat{h}_{1,d,0}, \hat{h}_{1,d,1}, \hat{h}_{1,d+1,0}, \hat{h}_{1,d+1,1}, ...,$$
$$\hat{h}_{1,c,1}, \hat{h}_{2,d,0}, \hat{h}_{2,d,1}, \hat{h}_{2,d+1,0}, \hat{h}_{2,d+1,1}, ...)$$

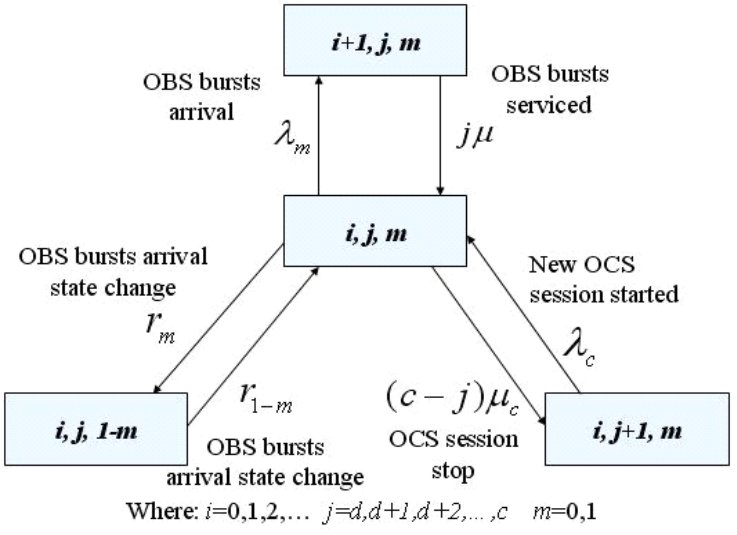


Fig. 2. State transition diagram

The state transition balance equation is then

$$\hat{h}G = 0$$

The steady state queue size distribution x_i is related to $\hat{h}$ as such

$$x_i = \sum_m \sum_j \hat{h}_{i,j,m}$$

We can obtain x_i by another approach, using Neuts' analysis.

The mean queue size is computed using the stationary probability vector x. The mean OBS burst delay is found using Little's law. Burst delay is the time from the generation of the burst to the time the last bit of burst is sent. The mean delay is thus obtained as follows:

$$mean\ OBS\ burst\ arrival\ rate\ = \frac{\lambda_0 r_1 + \lambda_1 r_0}{r_0 + r_1},$$

$$mean\ queue\ size\ = \sum_{i=1}^{i=\infty} x_i \cdot i,$$

$$mean\ OBS\ burst\ delay\ = \frac{mean\ queue\ size}{mean\ burst\ arrival\ rate}.$$

3.4 End-to-End Network Performance Analysis for OBS

We present an analysis of end-to-end delay for OBS using one-way reservation scheme such as JIT(Just-In-Time), JET(Just-Enough-Time). We assume that fiber link capacity (C) is 10Gbps and burst size (L_b) is variable. Packet arrival rate is λ_p (packets/sec) and packet length (L_p) is variable (e.g., 40bytes $\sim$ 1500bytes). The end-to-end delay (T_{total}) is obtained as follows:

$$\begin{aligned} T_{total} &= 2t_u + t_{ba} + t_{off} + n(t_t + t_p) + (n-1)t_c + t_{bd}, \\ t_{off} &\geq nt_{setup} + t_{proc}, \\ T_{total} &\leq 100\text{ms}. \end{aligned}$$

The offset time(t_{off}) depends on the number of hops (n) and setup and processing time (t_{setup}, t_{proc}) at OBS node. The end-to-end delay must be satisfied with delay constraint (100ms) which is recommended at ITU-T Y.1541 [11]. Table 1 shows the parameters and values used in this analysis.

Table 1. Parameters and values used in analysis

Parameter	Value	Description
b	1Gbps	Bit rate of OBS switching system processor
t_u	10ms	The overall delay from an end user to OBS switch
t_{off}	variable	Offset time
t_{setup}	$5\mu s$	Processing time of set up message
t_{conf}	$5\mu s$	OBS switch configuration time
t_c	L_b/b	Cut-through switching over time at each OBS switch
t_p	0.1ms	Propagation delay on a fiber link
t_t	L_b/C	Transmission delay at each OBS switch
t_{ba}	$L_b/(L_p\lambda_p)$	Burst assembly time at ingress OBS switch
t_{bd}	L_b/b	Burst disassembly time at egress OBS switch

4 Numerical Results

The analysis was performed for an optical hybrid switching system of particular parameters. There are 120 available capacity units of link. Out of the 120 available, 10 capacity units are reserved exclusively for OBS bursts only. Each link is 10Gbps. The mean duration of an OCS session is 3 minutes. The OCS load is chosen so that the capacity available to the OCS connection has a utilization of 30%. Table 2 lists all of the parameters used. These parameters were used to obtain a result using analytical approach.

Table 2. Model parameters

Parameter	Value	Meaning
λ_0	variable	OBS state 0 burst generation rate
λ_1	$\lambda_1 = 5\lambda_0$	OBS state 1 busrt generation rate
r_0	0.00001	State 0 to 1 transition rate
r_1	0.000001	State 1 to 0 transition rate
μ	1/1000	Mean OBS burst service rate per unit of capacity
c	120	Total capacity units of link
d	10	Capacity units reserved for OBS bursts only
μ_c	1/180000	1 / (OCS connection hold time)
λ_c	$0.3(c-d)\mu_c$	OCS connection establishment rate

In Figure 3, we present the result for the mean delay versus utilization for OBS bursts. The mean delay is rapidly increased for high utilization (over 0.7). Thus, this result indicates that in order to operate an optical hybrid system with reasonable burst delays the utilization must be kept below 70%. Similarly, Figure 4 shows the queue size probability distribution for OBS bursts when utilization is 0.8.

In Figure 5 and Figure 6, we present the result for end-to-end delay and burst assembly time. Figure 5 shows the end-to-end delay versus the burst size for different load when hop distance is 10. To guarantee end-to-end delay bound (100msec), burst size depending offered load is limited. In Figure 6, we fix offset

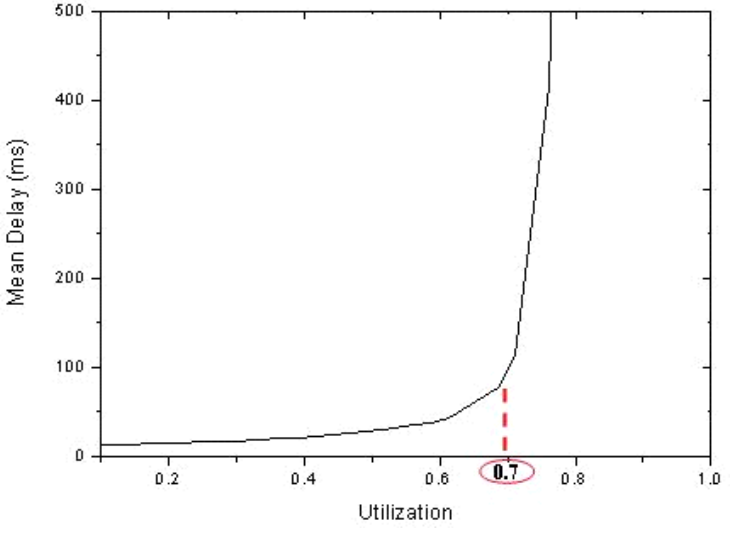


Fig. 3. Mean delay versus utilization for OBS bursts

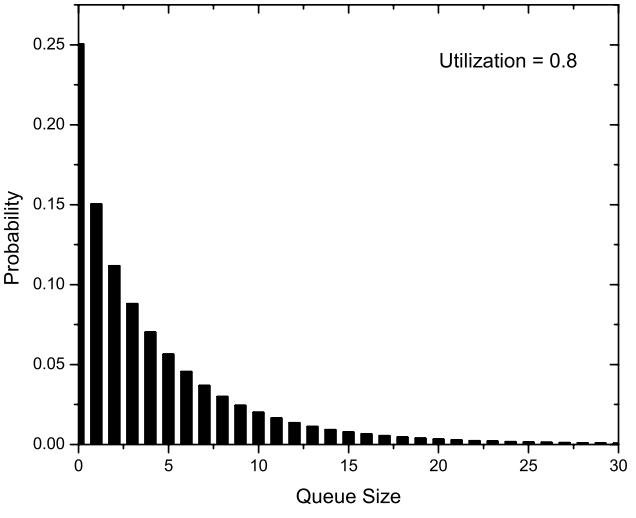


Fig. 4. Queue size probability for OBS bursts

time to 70 μs and hop distance to 10. We present burst assembly time as a function of the offered load. The reader should realize that the burst size is a potent influence on the delay performance from Figure 5 and Figure 6.

5 Conclusions and Future Research

In this paper, we have proposed a new optical hybrid switching system which combines OBS and OCS. We also presented a simple analytical model of this system. We have used MMPP to model the bursty nature of OBS data traffic. This paper has provided an analytical tool for provisioning capacity in an optical hybrid system to increase network utilization subject to meeting QoS requirements. This proposed architecture can give rise to interesting performance evaluation research of the interaction of the different elements such as hybrid switching, traffic classification, queueing and loss networks, routing, connection admission control, each of which, by itself, has been a topic for much research.

Acknowledgements

This research was supported by the Korean Science and Engineering Foundation (KOSEF) through Optical Internet Reserarch Center Project and by Institue of Information Technology Assessment (IITA) through University IT Research Center Project.

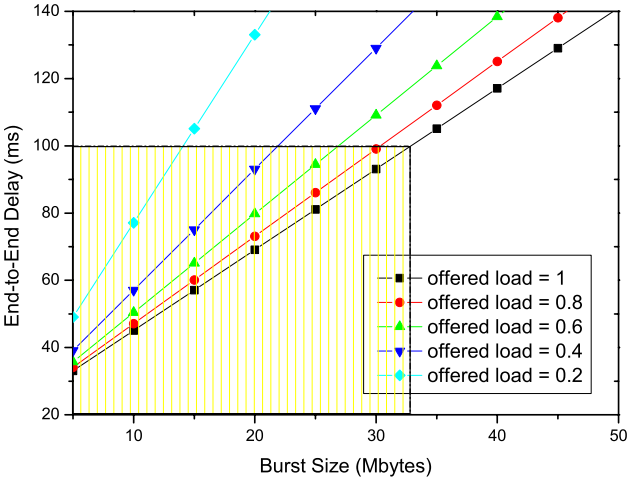


Fig. 5. End-to-end delay versus the burst size for different offered load ($n = 10$)

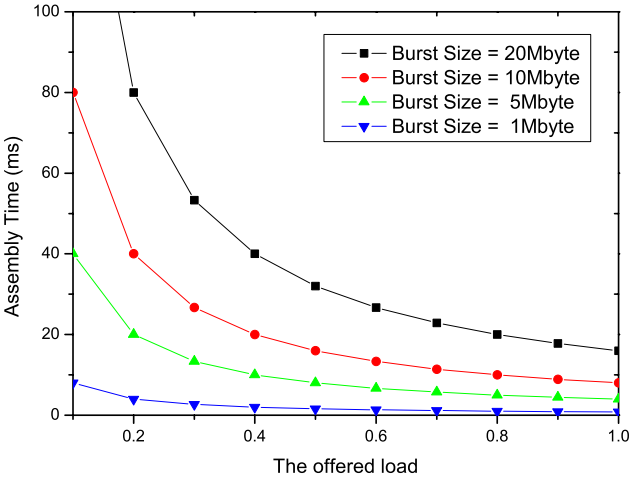


Fig. 6. Burst assembly time versus the offered load for different burst size ($n = 10$, fixed offset time= $70\mu s$)

References

[1] C. Qiao, M. Yoo.: Choice, and Feature and Issues in Optical Burst Switching. Optical Net. Mag., vol.1, No.2, Apr. 2000, pp.36-44. 740

[2] C. Qiao.: Labeled Optical Burst Switching for IP over WDM Integration. IEEE Comm. Mag., Sept. 2000, pp.104 114.

[3] Yijun Xiong, Marc Vandenhoute, Hakki C. Cankaya.: Control Architecture in Optical Burst-Switched WDM Networks. IEEE JSAC, Vol.18, No.10, Oct. 2000.

[4] Ilia Baldine, George N. Rouskas, Harry G. Perros, Dan Stevension.: JumpStart: A Just-in-time Signaling Architecture for WDM Burst-Switching Networks. IEEE Comm. Mag., Feb. 2002. 740

[5] Alberto Leon-Garcia.: Photonic burst switching. Whitepaper, Accelight networks, 2001, available at http://www.accelight.com/ 740

[6] Hao Che, San-qi Li, Arthur Lin.: Adaptive resource management for flow-based IP/ATM hybrid switching systems. IEEE/ACM Trans on Networking, vol. 6, no. 5, October 1998. 741

[7] H. L. Vu and M. Zukerman.: Blocking Probability for Priority Classes in Optical Burst Switching Networks. IEEE Communications Letters, vol. 6, no. 5, May 2002, pp. 214-216. 741

[8] Admela Jukan, et.al.: Service-specific resource allocation in WDM networks with quality constraints. IEEE JSAC, pp. 2051-2061, Oct. 2000. 741

[9] H. Heffes and D. M. Lucantoni.: A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. IEEE JSAC, vol. SAC-4, no-6, Sep. 1986. 742

[10] M. F. Neuts.: Matrix-Geometric solutions in stochastic models: an algorithmic approach. The Johns Hopkins University Press, Baltimore, MD, 1981. 743

[11] ITU-T Recommendation Y.1541.: Network performance objectives for IP-based services, May 2002. 745

Performance Assessment of Signaling Protocols in Optical Burst Switching Mesh Networks

Joel J.P.C. Rodrigues and Mário M. Freire

Department of Informatics, University of Beira Interior
Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal
{joel,mario}@di.ubi.pt

Abstract. This paper presents a performance evaluation of just-in-time (JIT), just-enough-time (JET), Jumpstart and Horizon signaling protocols in optical burst switching (OBS) networks with mesh topologies. The analysis is focused on the following topologies: rings, chordal rings, mesh-torus, NSFNET, ARPANET and the European Optical Network. It is shown that choral rings with smallest diameter lead to best network performances. For 16 nodes and 64 data channels, the nodal degree gain due to the increase of nodal degree from two (ring) to three (chordal ring with smallest diameter) is about three orders of magnitude in the last hop of both topologies. Mesh networks with a number of nodes ranging from 14 to 26 nodes have been analyzed and it was shown that chordal ring topologies with smallest diameter are very suitable for this kind of networks. It is also observed that, for the cases under study, the network performance for JIT, JET, JumpStart and Horizon is very close.

1 Introduction

Optical burst switching (OBS) [1]-[4] has been proposed an alternative paradigm to optical packet switching (OPS) in order to overcome the technical limitations of OPS, namely the lack of optical random access memory and the problems with synchronization. OBS is a technical compromise between wavelength routing and optical packet switching, since it does not require optical buffering or packet-level processing and is more efficient than circuit switching if the traffic volume does not require a full wavelength channel. In OBS networks, IP (Internet Protocol) packets are assembled into very large size packets called data bursts. These bursts are transmitted after a burst header packet, with a delay of some offset time. Each burst header packet contains routing and scheduling information and is processed at the electronic level, before the arrival of the corresponding data burst. Several signaling protocols have been proposed for optical burst switching networks. In this paper, we concentrate on just-in-time (JIT) [3], JumpStart [4]-[6], just-enough-time (JET) [1] and Horizon [2] signaling protocols.

A major concern in OBS networks is the contention and burst loss. The two main sources of burst loss are related with the contention on the outgoing data burst channels and on the outgoing control channel. In this paper, we consider bufferless networks and we concentrate on the loss of data bursts in

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 750–759, 2004.

OBS networks with mesh topologies. For comparison purposes, ring topologies are also considered.

The reminder of this paper is organized as follows. In section 2, we describe the model of the OBS network under study, and in section 3 we present a performance analysis of OBS networks with mesh topologies. Main conclusions are presented in section 4.

2 Network Model

We consider OBS networks with the following mesh topologies: chordal rings with number of nodes ranging from 14 up to 26, mesh-torus with 16 and 25 nodes, the NSFNET with 14-node and 21 links [7], the NSFNET with 16 nodes and 25 links [8], the ARPANET with 20 nodes and 32 links [7], [9], and the European Optical Network (EON) with 19 nodes and 37 links [10]. For comparison purposes bi-directional ring topologies are also considered. These topologies have the following nodal degree: ring: 2.0; chordal ring: 3.0; mesh-torus: 4.0; NSFNET with 14-node and 21 links: 3.0; the NSFNET with 16 nodes and 25 links: 3.125; the ARPANET with 20 nodes and 32 links: 3.2; and the EON: 3.895.

Chordal rings are a well-known family of regular degree three topologies proposed by Arden and Lee in early eighties for interconnection of multi-computer systems [11]. A chordal ring is basically a bi-directional ring network, in which each node has an additional bi-directional link, called a chord. The number of nodes in a chordal ring is assumed to be even, and nodes are indexed as 0,1,2,...,N-1 around the N-node ring. It is also assumed that each odd-numbered node i (i=1,3,...,N-1) is connected to a node $(i+w) mod\ N$, where w is the chord length, which is assumed to be positive odd. For a given number of nodes there is an optimal chord length that leads to the smallest network diameter. The network diameter is the largest among all of the shortest path lengths between all pairs of nodes, being the length of a path determined by the number of hops.

In each node of a chordal ring, we have a link to the previous node, a link to the next node and a chord. Here, we assumed that the links to the previous and to the next nodes are replaced by chords. Thus, each node has three chords, instead of one. Let $w1$, $w2$, and $w3$ be the corresponding chord lengths, and N the number of nodes. We represented a general degree three topology by D3T($w1$, $w2$, $w3$). We assumed that each odd-numbered node i (i=1, 3, ..., N-1) is connected to the nodes $(i+w1) mod\ N$, $(i+w2) mod\ N$, and $(i+w3) mod\ N$, where the chord lengths, $w1$, $w2$, and $w3$ are assumed to be positive odd, with $w1 \leq N-1$, $w2 \leq N-1$, and $w3 \leq N-1$, and $w_i \neq w_j, \forall i \neq j$ and $1 \leq i,j \leq 3$. In this notation, a chordal ring with chord length w is simply represented by D3T(1,N-1,$w3$).

Now, we introduce a general topology for a given nodal degree. We assume that instead of a topology with nodal degree of 3, we have a topology with a nodal degree of n, where n is a positive integer, and instead of having 3 chords we have n chords. We also assume that each odd-numbered node i (i=1,3,...,N-1) is connected to the nodes $(i+w) mod\ N$, $(i+w2) mod\ N$, ..., $(i+wn) mod\ N$,

where the chord lengths, $w1$, $w2$, ..., wn are assumed to be positive odd, with $w1 \leq N-1, w2 \leq N-1, ..., wn \leq N-1$, and $w_i \neq w_j, \forall i \neq j$ and $1 \leq i,j \leq n$. Now, we introduce a new notation: a general degree n topology is represented by DnT($w1$, $w2$,...,wn). In this new notation, a chordal ring family with chord length w is represented by D3T(1,N-1,w). In this new notation, a chordal ring family with a chord length of $w3$ is represented by D3T(1,N-1,$w3$) and a bi-directional ring is represented by D2T(1,N-1).

We assume that each node of the OBS network supports F+1 wavelength channels per unidirectional link. One wavelength is used for signaling (carries setup messages) and the other F wavelengths carry data bursts. Each OBS node consists of two main components [12]: i) a signaling engine, which implements the OBS signaling protocol and related forwarding and control functions; and ii) an optical cross-connect (OXC), which performs the switching of bursts from input to output. It is assumed that each OXC consists of non-blocking space-division switch fabric, with full conversion capability, but without optical buffers. It is assumed that each OBS node requires [12]: i) an amount of time, T_{OXC}, to configure the switch fabric of the OXC in order to set up a connection from an input port to an output port, and requires ii) an amount of time, $T_{setup}(X)$ to process the setup message for the signaling protocol X, where X can be JIT, JET, horizon, and JumpStart. It is also considered the offset value of a burst under reservation scheme X, $T_{offset}(X)$, which depends, among other factors, on the signaling protocol, the number of nodes the burst has already traversed, and if the offset value is used for service differentiation. In this study, it is assumed that [12]: $T_{OXC} = 10\mu s$, $T_{setup}(JIT) = 12.5\mu s$, $T_{setup}(JET) = 50\mu s$, $T_{setup}(Horizon) = 25\mu s$, $T_{setup}(JumpSart) = 12.5\mu s$, the mean burst size, $1/\mu$, was set to $50\mu s$, and the burst arrival rate λ, is such that $\lambda/\mu = 32$.

3 Performance Assessment

In this section, we present a performance assessment of JIT, JET, Horizon, and Jump-Start signaling protocols in OBS networks with mesh topologies. The performance assessment is based on the burst blocking probability obtained by simulation.

In chordal ring topologies, different chord lengths can lead to different network diameters, and, therefore, to a different number of hops. One interesting result that we found is concerned with the diameters of the D3T($w1$,$w2$,$w3$) families, for which $w2$=($w1$+2)$mod\ N$ or $w2$=($w1$-2)$mod\ N$. Each family of this kind, i.e. D3T($w1$,($w1$+2)$mod\ N$, $w3$) or D3T($w1$, ($w1$-2)$mod\ N$, $w3$), with $1 \leq w1 \leq 19$ and $w1 \neq w2 \neq w3$, has a diameter which is a shifted version (with respect to $w3$) of the diameter of the chordal ring family (D3T(1, N-1, $w3$)). For this reason, we concentrate the analysis on chordal ring networks, i. e., D3T(1, 19, $w3$).

Fig. 1 shows the burst blocking probability in the last hop of ring, chordal rings, mesh-torus and NSFNET networks, all with 16 nodes. As may be seen in Fig. 1, when enough network resources are available (F=64), the chordal ring

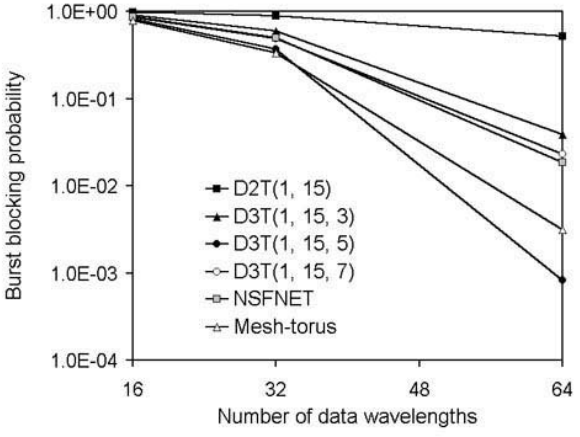


Fig. 1. Burst blocking probability, as a function of the number of data wavelengths per link (F), in last hop of ring, chordal rings, NSFNET and mesh-torus networks with N=16 nodes and for JIT protocol

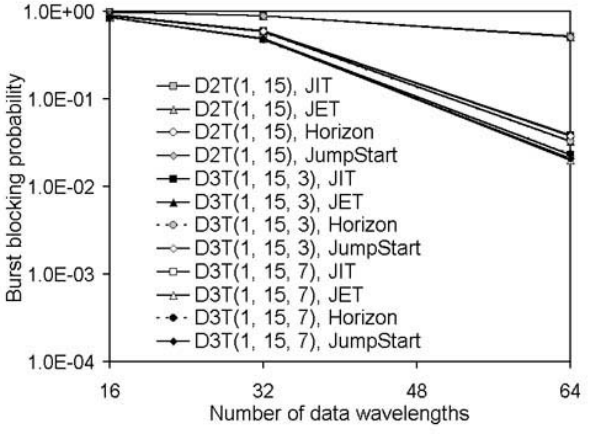


Fig. 2. Burst blocking probability, as a function of the number of data wavelengths per link (F), in the last hop of D2T(1, 15), D3T(1, 15, 3), and D3T(1, 15, 7) for JIT, JET, Horizon, and JumpStart signaling protocols; N=16

network with chord length of $w3$=5 clearly have better performance. This figure also shows that the performance of the NSFNET is very close to the performance of chordal rings with chord length of w3=3 or w3=7. This results reveals the importance of the way links are connected in the network, since chordal rings and NSFNET have similar nodal degrees and therefore a similar number of network links. Also interesting is the fact that chordal rings with w3=5 have better performance than mesh-torus networks, which have a nodal degree of 4, i. e., more

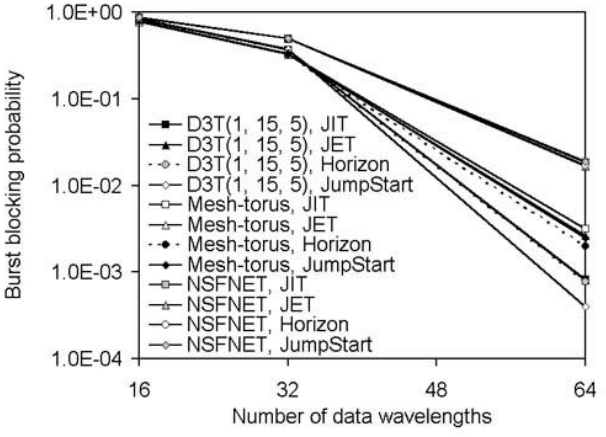


Fig. 3. Burst blocking probability, as a function of the number of data wavelengths per link (F), in the last hop of D3T(1, 15, 5), NSFNET, and mesh-torus for JIT, JET, Horizon, and JumpStart signaling protocols

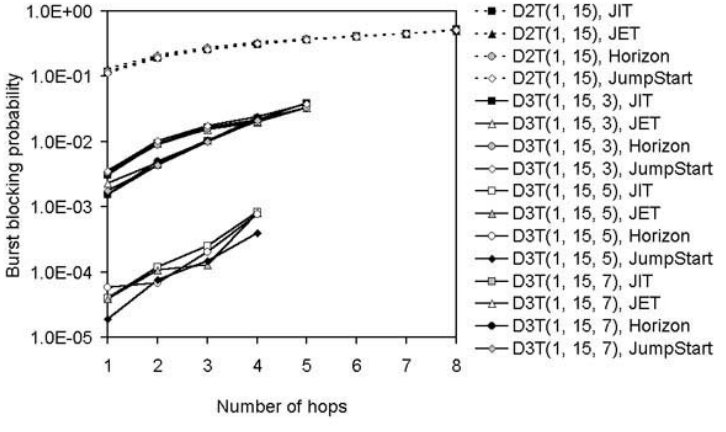


Fig. 4. Burst blocking probability, as a function of the number of hopes, for rings and chordal rings networks using JIT, JET, Horizon, and JumpStart signaling protocols; F=64

25/100 of network links. We have also observed that the best performance of chordal ring network is obtained for the smallest network diameter. Results presented in Fig. 1 were obtained for the JIT signaling protocol. Similar results have been obtained for JET, Horizon and JumpStart, as may be seen in Fig. 2 and 3. Fig. 2 shows the burst blocking probability in the last hop of D2T(1, 15), D3T(1, 15, 3), and D3T(1, 15, 7) for JIT, JET, Horizon, and JumpStart and Fig. 3 shows the burst blocking probability in the last hop of D3T(1, 15, 5), NSFNET, and

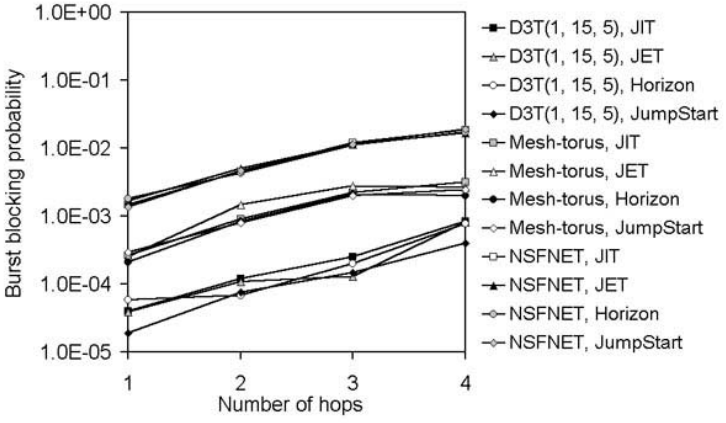


Fig. 5. Burst blocking probability, as a function of the number of hopes, D3T(1,15,5), mesh-torus, NSFNET networks using JIT, JET, Horizon, and JumpStart signaling protocols; F=64; N=16

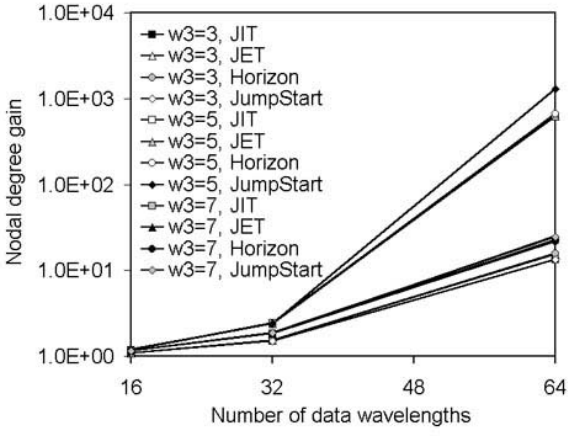


Fig. 6. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to 3 (D3T(1, 15, $w3$)), in the last hop of each topology, for JIT, JET, Horizon, and JumpStart signaling protocols; N=16

mesh-torus for JIT, JET, Horizon, and JumpStart signaling protocols. As may be seen in these figures, the performance of the four signaling protocols in the OBS mesh networks under study is very close. Fig. 4 and Fig. 5 confirm these results. These figures show the burst blocking probability as a function of the number of hopes. Since the burst blocking probability is a major issue in OBS networks, clearly ring topologies are the worst choice for these network due to very high blocking probabilities and, surprisingly, chordal rings with smallest di-

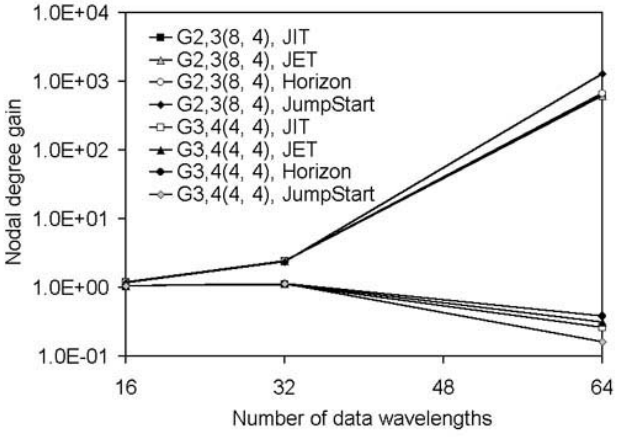


Fig. 7. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to 3 (D3T(1, 15, $w3$)), and due to the increase of the nodal degree from 3 (D3T(1, 15, $w3$)) to 4 (mesh-torus) in the last hop of each topology, for JIT, JET, Horizon, and JumpStart signaling protocols; N=16

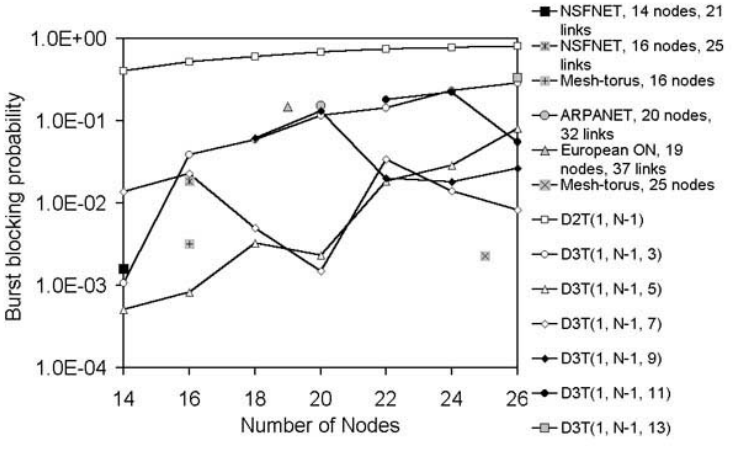


Fig. 8. Burst blocking probability, as a function of the number of nodes (N), in the last hop of rings, chordal rings, NSFNET, ARPANET, European Optical Network, and mesh-torus, for JIT protocol; F=64

ameter have a good performance with burst blocking probabilities ranging from $10^{-3} - 10^{-5}$, depending on the number of hops.

In order to quantify the benefits due to the increase of nodal degree, we introduce the nodal degree gain, $G_{(n-1),n}(i,j)$, defined as:

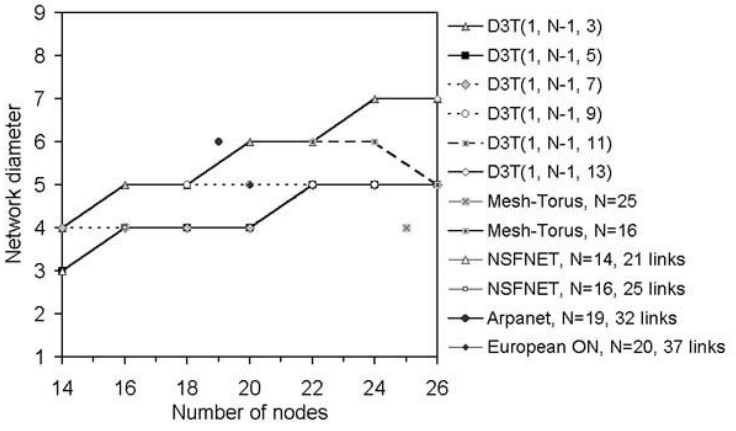


Fig. 9. Network diameter, as a function of the number of nodes (N), for chordal rings, NSFNET, ARPANET, European Optical Network, and mesh-torus networks

$$G_{n-1,n}(i,j) = \frac{P_i(n-1)}{P_j(n)} \tag{1}$$

where $P_i(n-1)$ is the burst blocking probability in the i-th hop of a degree (n-1) topology and $P_j(n)$ is the burst blocking probability in the j-th hop of a degree n topology, for the same network conditions (same number of data wavelengths per link, same number of nodes, etc), and for the same signaling protocol.

Fig. 6 shows the nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to 3 (D3T(1, 15, $w3$)), in the last hop of each topology, for JIT, JET, Horizon, and JumpStart signaling protocols. Fig. 7 shows the nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to 3 (D3T(1, 15, $w3$)), and due to the increase of the nodal degree from 3 (D3T(1, 15, $w3$)) to 4 (mesh-torus) in the last hop of each topology, for JIT, JET, Horizon, and JumpStart signaling protocols. As may be seen, the increase of the nodal degree from 2 (rings) to 3 (chordal rings with $w3$=5) can lead to nodal degree gains of about 3 orders of magnitude, where as the increase of the nodal degree from 3 (chordal rings with $w3$=5) to 4 (mesh-torus) can lead to a performance degradation. Once again, the results obtained for the four signaling protocols under study are very close.

Fig. 8 shows the burst blocking probability, as a function of the number of nodes (N), in the last hop of rings, chordal rings, NSFNET, ARPANET, European Optical Network, and mesh-torus, for JIT protocol. Fig. 9 shows the corresponding network diameters (except for rings). As may be seen, the D3T(1, N-1, 5) has a very good performance, except for N=26, because this topology leads to a smallest diameter in the whole range from 14 to 26 nodes, as may be seen in Fig. 9. When the number of nodes is larger, D3T(1, N-1, 7) has

better performance than D3T(1, N-1,5). However, D3T(1, N-1, 7) has worse performance than D3T(1, N-1,5) when the number of nodes is smaller. This results have been obtained for JIT. Similar results have been obtained for JET, Horizon and Jumpstart, which are not displayed due to space limitations.

4 Conclusions

We have presented a performance assessment of JIT, JET, JumpStart and, and Horizon signaling protocols in optical burst switching (OBS) networks with the following topologies: rings, chordal rings, mesh-torus, NSFNET, ARPANET and the European Optical Network. It is shown that chordal rings with smallest diameter lead to best network performances. For 16 nodes and 64 data channels per link, the nodal degree gain due to the increase of nodal degree from 2 to 3 (chordal ring with smallest diameter) is about three orders of magnitude in the last hop. Those topologies have also been analyzed for a number of nodes ranging from 14 to 26 nodes. It is shown that the D3T(1, N-1, 5) and D3T(1, N-1, 7) have very good performance, being D3T(1, N-1, 5) better for a smaller number of nodes and being D3T(1, N-1, 7) better for a larger number of nodes. In all of these cases, it was observed that the network performance for JIT, JET, JumpStart, and Horizon is very close.

Acknowledgements

Part of this work has been supported by the Group of Networks and Multimedia, Institute of Telecommunications-Covilhã Lab, Portugal, and by the Euro-NGI (Design and Engineering of the Next Generation Internet, Towards convergent multi-service networks) Network of Excellence, Sixth Framework Programme, Information Society Technologies IST.

References

[1] Qiao, C., Yoo, M.: Optical burst switching (OBS)-A New Paradigm for an Optical Internet. Journal of High Speed Networks, Vol. **8**, No. 1 (1999) 69-84.
[2] Turner, J. S.: Terabit Burst Switching. Journal of High Speed Networks, Vol. **8**, No. 1 (1999) 3-16.
[3] Wei, J. Y., McFarland, R. I.: Just-in-time signaling for WDM optical burst switching net-works. In Journal of Lightwave Technology, Vol. **18**, No. 12 (2000) 2019-2037.
[4] Baldine, I., Rouskas, G. N., Perros, H. G., Stevenson, D.: JumpStart: A just-in-time signaling architecture for WDM burst-switched networks. In IEEE Communications Magazine, Vol. **40**, No. 2 (2002) 82-89.
[5] Zaim, A. H., Baldine, I., Cassada, M., Rouskas, G. N., Perros, H. G., Stevenson, D.: The JumpStart Just-In-Time Signaling Protocol: A Formal Description Using EFSM. In Optical Engineering, Vol. **42**, No. 2, February (2003) 568-585.

[6] Baldine, I., Rouskas, G. N., Perros, H. G., Stevenson, D.: Signaling Support for Multicast and QoS within the JumpStart WDM Burst Switching Architecture. In Optical Networks, Vol. **4**, No. 6, November/December (2003)
[7] Sridharan, M., Salapaka, M. V., and, Somani, A. K.: A Practical Approach to Operating Survivable WDM Networks. IEEE Journal on Selected Areas in Communications, Vol. **20**, No. 1, (2002) 34-46.
[8] Ramesh, S., Rouskas, G. N., and Perros, H. G.: Computing Blocking Probabilities in Multi-class Wavelength-Routing Networks With Multicast Calls. IEEE Journal on Selected Areas in Communications, Vol. **20**, No. 1, (2002) 89-96.
[9] Nayak, T. K., and Sivarajan, K. N.: A New Approach to Dimensioning Optical Networks. IEEE Journal on Selected Areas in Communications, Vol. **20**, No. 1, (2002) 134-148.
[10] O'Mahony, M. J.: Results from the COST 239 Project: Ultra-high Capacity Optical Trans-mission Networks. Proc. 22nd European Conf. on Optical Communication (ECOC), Oslo, Norway, Vol. 2, (1996) 2.11-2.18.
[11] Arden, B. W., Lee, H.: Analysis of Chordal Ring Networks. IEEE Transactions on Computers, Vol. **C-30**, No. 4 (1981) 291-295.
[12] Teng, J., Rouskas, G. N.: A Comparison of the JIT, JET, and Horizon Wavelength Reserva-tion Schemes on A Single OBS Node. First International Workshop on Optical Burst Switching, Dallas Texas, USA, October 16 (2003).

Part IV

Next Generation Internet Architecture

A Network Processor-Based Fault-Tolerance Architecture for Critical Network Equipments*

Nen-Fu Huang[1,2,3], Ying-Tsuen Chen[1], Yi-Chung Chen[1], Chia-Nan Kao[2], and Joe Chiou[3]

[1] Department of Computer Science, National Tsing Hua University
Hsinchu, Taiwan, R.O.C
[2] Institute of Communication Engineering, National Tsing Hua University
Taiwan, ROC
[3] Broadweb Corp., Hsin-Chu Industrial Science Park
Hsin-Chu, Taiwan, ROC
nfhuang@cs.nthu.edu.tw

Abstract. Businesses and individuals often suffer from significant amount of damage as a result of network failures, and that is why network fault tolerant mechanism is important for network design and management. The most common failure is happen to the network equipments. Moreover, network equipments located at the entrance of a network play an important role in the availability and reliability of the internal network. Therefore, we design and implement a fault-tolerant system especially for the network equipments located at the entrance of a network. On the situation that no redundant device exists, the fault-tolerant system could bypass the forwarding path to survive the network connections. We adopt the Intel IXDP1200 Network Processor as development platform to implement the proposed system.

1 Introduction

Providing any kind of network service and applications is based on the premise that the network is zero faults or the fault is tolerable. Network faults may cause existing connections to break and new arrival connections establishment to fail, and then the network services rely on the faulting network will be stopped. Thus, network fault tolerance has always been a vital issue in network design and management.

Most researches related to network fault-tolerance focus on finding the failure link or failure network node (host or equipment) under certain network topology and trying to find the recovery strategy or alternative healthy path for traffics [1]. Typically those solutions need the cooperation of some kind of network management protocol to gather current network status to determine whether and where the fault occurs.

* This work was partially supported by the MOE Program for Promoting Academic Excellence of Universities under Grant 89-E-FA04-1-4.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 763–772, 2004.

We proposed a fault-tolerant system (FTS) architecture which is capable of actively monitoring and detecting current status of network equipments using a lightweight method. When fault is detected, the system will interchange the physical connection between failure equipment and redundant backup equipments very fast without management overhead. Even in the situation that network equipment fails but without any redundant equipment available, the system will still keep the network connection alive by the intelligent switching mechanism within the system. Besides, the proposed system could support more than one type network equipments simultaneously while providing fault-tolerant capability for each type separately. The word type means the different specific functionality of corresponding network equipment.

In our system design, all traffics that traverse through the network equipments will also traverse transparently through our system to obtain fault-tolerance. In order not to let the fault-tolerance system become the performance bottleneck of overall network, fast packet forwarding process is required. Therefore, we adopt network processor as an essential part of our system concept and our implementation platform. Network processor is a programmable processor. Unlike the general purpose processor do almost anything, network processors aim at processing network packets rapidly and satisfying the balance of flexibility and performance.

2 Related Works

2.1 Network Fault Tolerance

Fault management is one of the most important aspects of network management. Network fault tolerance indicates the ability to against the failure of network components or links. It may be a hard issue because of lacking the full knowledge of topology information to identify where the failure point is, and which component or link is failed. Thus providing a fault resilient network is more difficult than before as the result of the dynamic nature and heterogeneous of network in the present days.

We can find that fault detection and fault recovery are the most time critical parts in fault tolerable network system. In order to reduce the duration of inability of system and improve the system availability and reliability, we should try to minimize the time spent on fault detection and recovery.

2.2 Related Definitions

Some definitions related to fault management are given in [6]:

Fault Detection: is the process to discover the fault in a specified network area. Fault detection time dominates the duration of fail-over. Thus many researches are working out to find efficient detection techniques while minimizing the impacts on network performance.

Fault Recovery: is the identification and selection of an alternative route or component which will serve to reconnect the source to the destination [6].

Typically, recovery mechanism is involved with the redundant components or architecture as prerequisite.

MTBF (Mean Time between Failures): is the average expected time between failures of network equipment, assuming the equipment goes through repeated periods of failure and repair.

MTTR (Mean Time to Repair): is the average expected time to restore the fault. Therefore, MTTR includes the times for failure detection, fault diagnosis, fault isolation, the actual repair, and any software synchronization time needed to restore the entire service. Under the situation without redundancy, high availability is achieved by increasing the MTBF and decreasing the MTTR.

2.3 Failure Types

Network failure has been classified into hard failures and soft failures by many works e.g., [5,7]. Hard failures are characterized by the inability to deliver packets, while soft failures are characterized by a partial loss of bandwidth, increase of packet delay, equipment performance degradation, etc. Hard failures generally cause the network throughput down to zero thus is the most serious failure type among those two types. But hard failures are more easily to be discovered than soft failures. On the other hand, soft failures are less well-defined. Possible reasons for soft failures are inappropriate use of the network, temporary congestion causing delay transmission, failed host hardware, failure of higher level protocols, mischievous users, and network attack like DoS or DDoS. Soft faults are hard to be defined and detected. Especially for those faults have transient property, i.e. faults that are not continuously happening. The FTS is capable of deal with hard failures for network equipments more efficiently without human resources.

2.4 Failure Detection and Recovery

In network fault tolerance research domain, many works have been done with fault detection and recovery techniques especially on the focus of soft faults. A classical review [4] explores network fault detection models and algorithms. "Fault feature vector" as the basis of fault signature matching is defined in [5] to detect soft faults. Meanwhile, many fault reasons are studied in [5] to select the proper parameters for fault feature vector. [2] and [3] provided an open-solution-based fault tolerant Ethernet (OFTE) for processor control networks. OFTE requires no change of vendor hardware and software, and it is transparent to control applications. OFTE claimed that it can perform less than 1ms end-to-end LAN swapping time and less than 2-sec failover time. A layered model is introduced in [9] to enhance the level of automation in fault isolation and recovery. [9] also made a comprehensive survey about the dependencies at various network layers in the aspects of network function, services, and protocols respectively. [10] is a remarkable research about fault recovery where a fault recovery model is built to evaluate the rerouting mechanisms (i.e. route selections and

establishments) in different fault scenarios when identifying link faults. Bejerano and Rastogi [1] extended the probe-based link delay monitoring technique to detect network link failures.

2.5 Network Processors

In NP based network equipment, network processor unit (NPU) is usually composed of multiple custom built or RISC processor units to deal with packet forwarding, scheduling, and classification at wire speed in data plane. While there is still a general purpose processing unit handling table management, traffic management, and configuration interface etc. sometimes, NPU cooperates with co-processors to accelerate certain specific packet processing purpose like high-layer packet classification, encryptions and decryptions, etc.

NP-based solution has been considered to be the major next generation networking equipment solution and increasingly works and researches are devoted to provide advanced function on network processor to meet high bandwidth and quality of service requirement. Also, Network Processing Forum (NPF) plays an important role in promoting network processor industrial advancement and standardizing the programming interfaces for related software API of network processors and the physical and message layer of interconnection between Traffic Manager and switching fabric.

3 Fault Tolerant System Concept

3.1 Main Idea and Operation

The main idea is to design a system capable of providing fault-tolerance for network equipments on critical link while minimizing the impacts on transmission performance. We focus on providing the link survivability when network equipment fault comes up. Thus, the basic requirement is, the system should have the ability to switch to the corresponding redundant equipment if available, within the minimum link downtime. On the other hand, if no redundant equipments exist, the system should be able to bypass the failure device. The above two criteria should be done automatically to reduce the mean time to recovery (MTTR) which is an important index to network availability discussed in Section 2.

The proposed *Fault Tolerance System (FTS)*, as in Figure 1, is a hardware-based configurable switching platform which consist of multiple 10/100 Fast Ethernet ports. The number of port can be variant relying on the requirement and scale of network equipments need fault-tolerance. *Inbound Port* serves as the port connecting to the intranet while *Outbound Port* serves as the port connecting to the outside network. All traffic from and to the intranet will pass through the Inbound and Outbound ports. Another two ports serve as *Mirror Ports* for monitoring traffic from *any* two other ports. *Monitor Stations (MS)* are connected to mirror ports for analysis purpose. MS must have the capability to capture packets from mirror ports in promiscuous mode and analyze abnormal

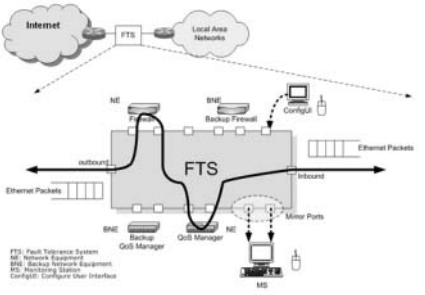


Fig. 1. Proposed Fault Tolerance System (FTS)

behavior from the gathered packets. Except for the above four ports, the other ports are used to connect the network equipments which need fault-tolerance. We define the connected network equipments with specified functionality as *a type of NE* (network equipment). The term *type* means the specified functionality of the NE, the same type NE will equip with the same functionality. FTS could support different type of NEs simultaneously while each type NE could have redundant as *Backup NE (BNE)*. A *Configure User Interface (ConfigUI)* is connected to FTS via console RS232 port or via the Ethernet 10/100BaseT cable. ConfigUI is responsible for configuring the default forwarding sequence and ports need mirroring within the FTS. The forwarding sequence constitutes a *path* for the traffic to traverse through the specified NEs connected to FTS. In FTS, any packet header or payload is not modified but just completely forwarded between NEs and FTS, i.e. FTS is *transparent* to the traffics.

3.2 Major System Functions

The detailed functions of FTS are described as follows:

3.3 Fault Detection

FTS periodically detects the status of connected NEs and BNEs by a simple polling strategy. The assumption for fault detection in FTS is that each NE owns TCP/IP protocol stack and can perform ICMP reply action. This assumption to general L2/L3/L4 or higher layer network equipments is apparently reasonable. Upon polling startup, FTS sends ICMP echo messages to all NEs and BNEs. If the ICMP reply was received from certain NE before the ICMP reply time-out. FTS expect that NE is healthy. Otherwise, the NE without replying ICMP message is identified as a failed NE. when the failure NE is detected, FTS will seek for the same type BNE. If BNE exist, then FTS will perform *Traffic Migration*. If not, FTS will perform *Failure Bypass*. The polling duration is a design constraint in order to satisfy the system performance and the fault detection efficiency at the same time. Too short duration will result in the heavy load of

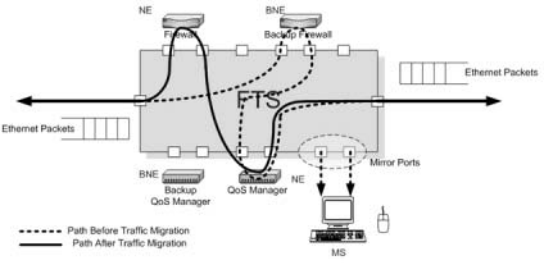


Fig. 2. Traffic Migration

system, too long duration will make the fault detection can not detect the fault as soon as possible.

3.4 Fault Recovery

FTS has two faults recovery strategies to handle the failure network equipments

Traffic Migration: when fault detection component find an available BNE to replace the position of failure type NE, FTS need to switch from the failure NE to BNE and thus re-construct a new forwarding path for traffic. The path reconstruction action is what we called Traffic Migration. After rebuilding the new path, traffic originally pass through the failed NE will now pass through the selected BNE. Traffic Migration time is supposed to be very short so that the downtime of connections can be minimized. Figure 2 illustrate the migration scenario.

Failure Bypass: when fault detection component can not find the corresponding BNE to replace the specified failure type NE, FTS will execute Failure Bypass strategy to save the broken connection. One thing FTS need to do is to rebuild a new forwarding path while ignore the failure type NE. Ignoring the failure NE may cause another damage for client users: imagine that, when an IDS network equipment fails and the Failure Bypass strategy is executed, now the network inside FTS is not protected by IDS thus may intruded by hackers or attackers. But comparing to keeping the maximum connection uptime, bypass is supposed to be the most instant and efficient solution. In the meanwhile, FTS will send an alarm message to notify ConfigUI that there is a failure NE without the BNE support and need to be repaired as soon as possible. Figure 3 illustrate the scenario for Failure Bypass.

3.5 Mirroring Traffic

FTS has reserved two ports as mirror ports. Through ConfigUI, system manager could specify which two ports are mirrored by FTS. At the MS terminal, manager could analyze and compare the traffics from the two mirror ports to find out the abnormal behavior. Besides, due to the transparency nature of FTS, mirror ports

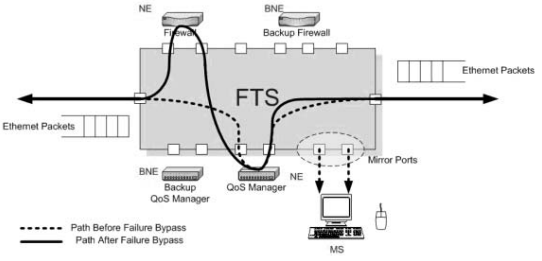


Fig. 3. Failure Bypass

could be used to discover what the actions NE does to traffic. It's beneficial for testing the ability and performance of NEs connected to FTS in a dramatically convenience way. For example, an IDS (Intrusion Detection System) is protected by FTS. In order to discover what kind of packets is filtered by IDS, We could mirror two ports which are connected to the inbound and outbound ports of IDS respectively.

3.6 Features and Advantages

The proposed FTS have the following features and advantages that traditional fault-tolerance system doesn't have:

Simple: FTS itself is simple to configure, and the overhead for overall network is very slight.

No Changes about Hardware and Software of NE: there is no need to change or add any hardware and software onto existing network equipments. Besides, in real operation, FTS is totally transparent to each NE.

Flexibility: FTS support fault-tolerance for multiple network devices simultaneously. Through the ConfigUI, managers could setup the same type devices as primary NE and redundant BNE. Managers need not to really change the physical connections if the fault-tolerance policy has been changed.

Topology Simplification: compare to typical fault-tolerant system topology, FTS simplify the cabling complexity to achieve the same goals. Only one physical cable need to be constructed between internet and intranet to support redundant NE in FTS while two cables are needed for primary connection and redundant connection in typical network fault-tolerant architectures.

Scalability: FTS could be cascaded to support more network equipments for increasing demand of fault-tolerance scale. When cascading FTSs, the configuration for each FTS is independent, i.e. no synchronization is required between the cascaded FTSs.

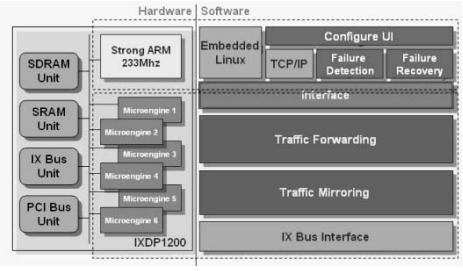


Fig. 4. System Block Diagram

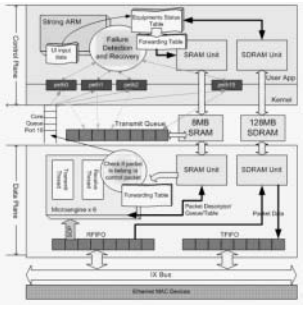


Fig. 5. System Architecture

4 Design and Implementation

Figure 4 is the system block diagram designed for providing failure detection and recovery on network processor for the network equipments on critical paths. We adopt the 16 ports layer 3 forwarding reference design project (L3fwd16 project) as our basic programming development platform.

Figure 5 illustrate the FTS system architecture combined with data plane and control plane.

5 Performance Evaluation

5.1 Evaluation and Test Procedure

Figure 6 illustrates the environment setup for evaluating and verifying the functionality of proposed system. In this environment, two Intrusion Detection Systems (IDS) are used to represent the NE served by FTS. One IDS is configured as primary NE and the other is configured as Backup NE. Moreover, two PC equipped with Ethereal Sniffer software are used to capture and analysis packets from the two Mirror Ports. The test procedure is as follows:

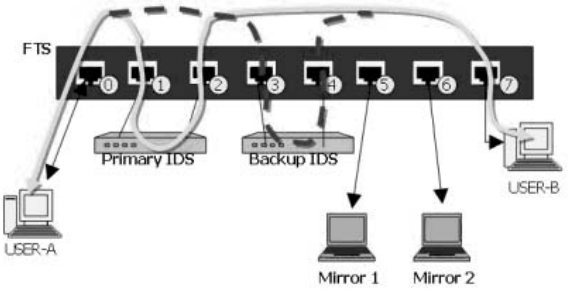


Fig. 6. Testing Environment

1. At the initial state, the packet flow is traversed through primary NE only.
2. USER-A performs Smurf attack [8] to USER-B. Meanwhile, USER-A sends continuously but not intrusive UDP traffic to USER-B to simulate normal traffics.
3. The launched attack will be intercepted by primary IDS while normal traffics are passed through FTS, which can be monitored by mirroring traffics from the two ports of IDS,
4. Turn off the primary IDS to simulate the fault occurrence of NE,
5. Then FTS will detect the failure NE and perform "Traffic Migration" to Backup IDS in short time,
6. USER-A still performing Smurf attack to USER-B,
7. Attack is intercepted by Backup IDS, and FTS still forward normal traffics through Backup IDS from USER-A to USER-B
8. Turn off Backup IDS and then FTS perform "Failure Bypass" procedure,
9. No IDS is available now, so the overall traffic including Smurf attack packets will be forwarded through FTS, and USER-B is subjected to attack now.

6 Conclusions

We explored the network fault tolerance issues aimed at the point of view of network equipment. Issues including the failure type, fault detection and recovery methods were discussed. Then we proposed a novel network fault-tolerant system architecture concentrating on providing fault-tolerance for network equipments located at the critical paths (e.g., the entrance of certain network). Our implementation of the proposed concept is based on IXDP1200 network processor platform and the real testing cases show our implementation is feasible in today's heterogeneous network environment. The result of performance evaluation demonstrates that the system could detect failures of network equipments in a very short time and the failure recovery is efficiently and automatically thus reduce the network downtime in a significant effect. Our approach also possesses several advantages described in section 3.

References

[1] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," In *Proceedings of IEEE INFOCOM'2003*, San Francisco, California, USA, Apr. 2003.

[2] J. Huang, S. Song, L. Li, P. Kappler, R. Freimark, J. Gustin, and T. Kozlik, "An open solution to fault-tolerant Ethernet: design, prototyping, and evaluation," In *IEEE International Performance, Computing and Communications Conference (IPCCC '99)*, pp. 461-468, Feb.10-12, 1999.

[3] S. Song, J. Huang, P. Kappler, R. Freimark, and T. Kozlik, "Fault-tolerant Ethernet middleware for IP-based process control networks," In *Proceedings of 25th Annual IEEE Conference on Local Computer Networks, 2000(LCN 2000)*, pp. 116-125, 2000.

[4] A. A. Lazar, Weiguo Wang and R. H. Deng, "Models and algorithms for network fault detection and identification: a review," *Singapore ICCS/ISITA '92. 'Communications on the Move'*, vol. 3, pp. 999-1003, Nov. 16-20, 1992.

[5] F. Feather, D. Siewiorek and R. Maxion, "Fault detection in an Ethernet network using anomaly signature matching," *ACM SIGCOMM'93*, Ithaca, N. Y., USA, Sep. 1993.

[6] D. D. Clark, "Fault isolation and recovery," RFC 816, Jul. 1982.

[7] Jun Li and C. Manikopoulos, "Investigation of the performance of GAFT, a novel network anomaly fault detection system," In *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02)*, Tampa, Florida, U. S. A., Nov, 2002.

[8] Smurf IP Denial-of-Service Attacks, http://www.cert.org/advisories/CA-1998-01.html

[9] R. Gopal, "Layered model for supporting fault isolation and recovery," *7th IEEE/IFIP Network Operations and Management Symposium (NOMS 2000)*, Honolulu, Hawaii, U. S. A., Apr. 10 - 14, 2000.

[10] A. Banerjea, "Fault recovery for guaranteed performance communications connections," *IEEE/ACM Transactions on Networking*, vol.7, no.5, pp. 653-668, Oct,1999.

A Mean-Field Theory of Cellular Automata Model for Distributed Packet Networks

Maoke Chen[1], Tao He[1], and Xing Li[2]

[1] Dept. of Electronic Engineering, Tsinghua Univ.
Beijing 100084 P R China
[2] China Education and Research Network (CERNET)
Beijing 100084 P R China

Abstract. A Mean-Field theory is presented and applied to a Cellular Automata model of distributed packet-switched networks. It is proved that, under a certain set of assumptions, the critical input traffic is inversely proportional to the free packet delay of the model. The applicability of Mean-Field theory in queue length estimation is also investigated. Results of theoretical derivations are compared with simulation samples to demonstrate the availability of the Mean-Field approach.

1 Introduction

Modelling computer networks is important for people to understand the network behaviors, especially those related to the critical phenomena. Assumptions were made in accordance with network topology in these models. Some of them were based on graph theory [1], while others were given as the topology was built on regular lattices [2, 3, 4, 5]. Choosing what kind of models depends upon the application background.

This paper focuses on architectures of distributed packet-switched networks. The model presented in this paper is extended from Fukś' work [3]. Actually, the same topology was studied even in the pre-Internet history, within the reports of RAND on distributed networks authored by Paul Baran and others [6, 7]. However, only survivability of the lattices was studied then. We step forward this effort into dynamic behavior of packet-switched Cellular Automata over the lattices.

On the other hand, previous works on Cellular Automata for data networks have discovered in simulations that critical traffic behavior is related to free packet delay in the networks [5, 3]. By the help of Mean-Field Theory technique, we step forward this discovery to an approximated analytical theorem for the extended model, where the critical traffic is inversely proportional to the free delay of the network provided the conditions of Mean-Field Theory are fulfilled. The most basic idea in the approximation relies on simplifying the system with an identical open Jackson network. However, the application of the Mean-Field Theory could not be exaggerated. In estimating queue length of the model, the Mean-Field approach is not accurate.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 773–783, 2004.

The rest parts of the paper are organized as such: section 2 describes the model and its parameters. Section 3 applies Mean-Field Theory to the model, with a certain set of approximations. Section 4 briefly discusses queue length estimated by the Mean-Field Theory, with comparison to simulations. Finally, we summarize the works with emphasizing its significance to analysis and design of distributed network architectures.

2 Model Definitions

Cellular Automaton is a mathematical model for physical systems containing large amount of simple, identical and locally interacting units [8]. Any Cellular Automata could be defined with a 4-tuple of lattice space, neighborhood, state set, and rule of state-transition [9]. The Cellular Automata models for distributed packet-switched networks (briefly "the model" or "our model", later through the paper) are also defined as such.

2.1 Lattice

The model are defined on d-dimensional Euclidean lattice space. Originally the lattice is boundless and extended to infinity. In digital simulation, however, the lattice is often truncated in a certain d-dimensional hypercube, say L as its width. Because a distributed network has no geometrical center, the truncated lattice should be thought as periodical, i.e. the coordinate values with same remainder modulo L are identical. Therefore the lattice of the model is denoted with

$$\mathcal{L}^d \triangleq \mathbb{Z}^d \cap [0, L)^d$$

And the bases of the lattice space are denoted with $\mathbf{e}_i, i = 1, 2, \cdot, d$.

2.2 Neighborhood and Metric

A neighborhood is a mapping from the lattice to its power set, $A : \mathcal{L}^d \mapsto P(\mathcal{L}^d)$. For the purpose of routing packets among the sites in the model, metrics are defined with the neighborhood as well. For example, von Neumann neighborhood and the periodic Taxicab metric [1] are defined by:

$$A(\mathbf{x}) = \bigcup_{i=1}^{d} \{\mathbf{x} + \mathbf{e}_i, x - \mathbf{e}_i\}, \quad \forall \mathbf{x} \in \mathcal{L}_d$$

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{L}{2} - \left| |x_i - y_i| - \frac{L}{2} \right|$$

[1] Or Manhattan metric.

While Moore neighborhood and the periodic Moore metric are defined as follows:

$$A(\mathbf{x}) = \left(\bigcup_{y_1=-1,0,1} \cdots \bigcup_{y_d=-1,0,1} \left\{ \mathbf{x} + \sum_{i=1}^{d} y_i \mathbf{e}_i \right\} \right) \setminus \{\mathbf{x}\}$$

$$D(\mathbf{x}, \mathbf{y}) = \max_i \left\{ \frac{L}{2} - \left| |x_i - y_i| - \frac{L}{2} \right| \right\}$$

The Taxicab and Moore metrics are identical in 1-dimensional cases.

2.3 State and Transition

The state of a site is a first-in-first-out queue of packets, each of which contains at least the information of its destination. Each site $\mathbf{x}$'s queue length at time k, denoted as $q(\mathbf{x}, k)$, is updated by both packet input and packet forwarding processes. The routing rule described here is as same as what Fukś introduced [3].

At each discrete time k, one packet enters the network through any site independently with an identical probability λ, while more than one packets do with probability $o(\lambda)$. Destination of a new packet is randomly selected among all possible sites with the same probability.

At each moment, a site serves the first packet in its queue, forwarding it to one of its neighbors properly selected under the routing rule. The service time is a constant of unity. Two criteria are applied to route selection. First, the next-hop should be selected from the neighbors, nearest to the destination in the term of given metric. The neighbor set of a site $\mathbf{x}$ nearest to destination $\mathbf{z}$ is

$$B(\mathbf{z}; \mathbf{x}) = \{\mathbf{y} \in A(\mathbf{x}) : D(\mathbf{z}, \mathbf{y}) \to \min\}$$

Second, the next-hop should be selected from the neighbors with minimum queue length within the neighbors nearest to destination.

$$C(\mathbf{z}; \mathbf{x}, k) = \{\mathbf{y} \in B(\mathbf{z}; \mathbf{x}) : q(\mathbf{y}, k) \to \min\}$$

Finally, if the minimum-queue nearest-to-destination neighbor set $C(\mathbf{z}; \mathbf{x}, k)$ contains more than one coordinates, then anyone is selected as the next-hop, randomly with the same probability. If this one is current destination $\mathbf{z}$, then the packet is not queued anymore: it leaves the system.

3 Mean-Field Theory

Any queueing system has a critical traffic as the upper bound of the input traffic, so that the system converges into a stable state instead of going far from stability. This critical traffic is just the service rate μ in a single queueing-service system, while it is different for networks.

Ohira has shown that the critical traffic is related to the free delay of the network [5] and Fukś has uncovered that, in two-dimensional von Neumann Cellular Automata model, the sufficient and necessary condition for stability of the

model is $\lambda < 1/\bar{\tau}_0$, or equivalently, $\lambda_c = 1/\bar{\tau}_0$, where $\bar{\tau}_0$ is the average delay of a free packet without being queued anywhere. These previous results are observed in simulations. Now we prove that this law is an analytical result under certain assumptions and is available for whatever dimensionality and neighborhoods rather than only for two-dimensional von Neumann cases.

3.1 Approximation to Open Jackson Network

The model defined in the previous section has the property of open queueing networks, i.e. any packet enters the system from outside and finally leaves once it arrives at the destination. This inspires utilizing well-known conclusions of open Jackson network.

Open Jackson network is of Markovian queueing network, i.e. packet arrival is a Poisson process and the service time of each site conforms to exponential distribution. The Jackson theorem presents the condition of stability as well as the queue length distribution in stable state [10, 11]. In the Jackson theorem, a parameter $\sigma(\mathbf{x})$, called as "site traffic", is defined as the traffic observed at site $\mathbf{x}$ in the system and we have the traffic equilibrium equations

$$\sigma(\mathbf{x}) = \lambda(\mathbf{x}) + \sum_{\mathbf{y}} \sigma(\mathbf{y}) r_{\mathbf{yx}}, \quad \forall \mathbf{x} \tag{1}$$

where $r_{\mathbf{yx}}$ is the forwarding probability from site $\mathbf{y}$ to $\mathbf{x}$ and $\sum_{\mathbf{x}} r_{\mathbf{yx}} = 1 - r_{\mathbf{y},\infty}$ where $r_{\mathbf{y},\infty}$ represents the probability of leaving.

Our model contains all the sites over a lattice space, each of which is a discrete-time M/D/1 queueing system. In order to utilize Jackson's approach, the model is approximated with a continuous-time open Jackson network where input traffic at each site is identical, $\lambda(\mathbf{x}) = \lambda$; and the constant service time is replaced by a exponential distribution with its mean value of $1/\mu = 1$; furthermore, details in route selection is ignored and approximated with a simple, time-invariant, and destination-free probability, $r_{\mathbf{yx}}$.

3.2 Mean-Field Theory for Critical Traffic

Observing simulations of queue growth processes in the model, one can easily summarized that site traffic $\sigma(\mathbf{x})$ seems to be a constant without difference referring to the coordinates. This implies that a Mean-Field Theory can be developed in order to derive the critical traffic law of the open Jackson network for the model.

Mean-Field Theory is an approximate technique widely used in statistical physics, which treats the order-parameter as spatially constant [12]. In our model, the Mean-Field approximation aims at an identical parameter σ such that

$$\sigma(\mathbf{x}) = \sigma, \quad \forall \mathbf{x} \in \mathcal{L}^d \tag{2}$$

A sufficient condition for Equation (2) consists of three Mean-Field Theory assumptions.

1. Isotropy Lattice space, either infinite or periodic, is isotropic in geometry. Therefore, it is reasonable to assume that, at any moment, any site forwards the first packet with a same probability to its neighbors. That is

$$r_{\mathbf{yx}} = r_{\mathbf{y}} = \frac{1}{|A(\mathbf{y})|}\left(1 - r_{\mathbf{y},\infty}\right), \quad \forall \mathbf{x} \in A(\mathbf{y}), \quad \forall \mathbf{y} \in \mathcal{L}^d \tag{3}$$

2. Homogeneity Further, it is assumed that the packet departure probability is same among all the sites.

$$r_{\mathbf{y},\infty} = r_\infty, \quad \forall \mathbf{y} \in \mathcal{L}^d \tag{4}$$

3. Spatial and temporal ergodicity It is assumed that the queue length process $q(\mathbf{x}, k)$ is spatially ergodic, i.e.

$$E\{q(\mathbf{x}, k)\} = \lim_{L\to\infty} \frac{1}{|\mathcal{L}^d|} \sum_{\mathbf{x}\in\mathcal{L}^d} q(\mathbf{x}, k), a.s. \tag{5}$$

And furthermore, it is also temporally ergodic as long as the stable state $q(\mathbf{x}) \triangleq \lim_{k\to\infty} q(\mathbf{x}, k)$ is achieved. Later on, we denote $\bar{q}$ for either the ensemble, or temporal, or spatial average (in the stable state if it exists) of queue length processes on sites over the lattice.

From (3) and (4), note that $\lambda(\mathbf{x})$ are identical to λ, the traffic equations (1) is simplified to

$$\sigma(\mathbf{x}) = \lambda + \left[\frac{1}{A}(1 - r_\infty)\right] \sum_{\mathbf{y}} \sigma(\mathbf{y}), \quad \forall \mathbf{x} \in \mathcal{L}^d \tag{6}$$

For linear equations (6) are symmetric to permutations of $\{\sigma(\mathbf{x})\}$, it is definite that they have a unique solution which is satisfying (2).

Then equation (6) is reduced to a single equation referring to σ, and finally it is solved that $\sigma = \lambda / r_\infty$.

The value of r_∞ has not been determined yet. We'd like to present current result first and then derive r_∞ by applying the well-known Little's Law to the mean value of packet lifetime.

With the help of the assumptions above, and the Jackson Theorem, we have

Theorem 1. *Under the assumptions of Mean-Field Thoery, the queueing network of the model converges to stable state if and only if*

$$\rho \triangleq \frac{\sigma}{\mu} = \frac{\lambda}{r_\infty} < 1 \tag{7}$$

And in the stable state, queue length on each site, $q(\mathbf{x})$, conforms to an identical geometric distribution:

$$P(q(\mathbf{x}) = n) = (1 - \rho)\rho^n, \quad \forall n \geq 0, \quad \forall \mathbf{x} \in \mathcal{L}^d \tag{8}$$

And the mathematical expectation of stable queue length is:

$$\bar{q} = \frac{\rho}{1-\rho}, \quad \forall \mathbf{x} \in \mathcal{L}^d \tag{9}$$

Now from the Little's Law of arbitrary queueing system, it holds in the stable state that

$$\bar{q} = \lambda \bar{\tau} \tag{10}$$

where $\bar{\tau}$ is the mathematical expectation of packet lifetime in stable state. When all the sites have identical queue length, lifetime of a packet is independent upon the path that it pass through, and is equal to the free delay $\bar{\tau}_0$ plus the total time of being queued. Note that the packet must be queued $\bar{\tau}_0$ times, and apply the third assumption to regard queue length of each site on a packet path is constant and equal to the average $\bar{q}$, then we have

$$\bar{\tau} = \bar{\tau}_0 + \bar{\tau}_0 \bar{q} \tag{11}$$

Apply (11) to (10), we obtain the following equation

$$\bar{q} = \lambda(\bar{\tau}_0 + \bar{\tau}_0 \bar{q}), \quad \text{and} \quad \bar{q} = \frac{\lambda \bar{\tau}_0}{1 - \lambda \bar{\tau}_0} \tag{12}$$

Recall the formulae (7), (9) in the Theorem 1 and compare them to the equation (12), we have the following theorem, which represents the law of critical traffic as a function of the free delay in the model[2].

Corollary 1. *In the Mean-Field Theory of the model, the packet's leaving probability and the site traffic are respectively*

$$r_\infty = \frac{1}{\bar{\tau}_0} \tag{13}$$

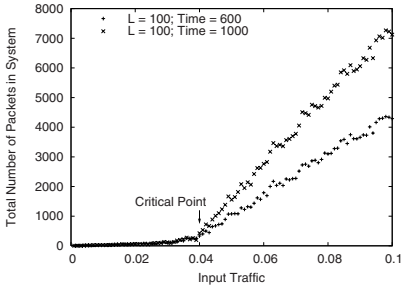
$$\textit{and} \quad \sigma = \lambda \bar{\tau}_0 \tag{14}$$

And the stability condition is equivalent to $\lambda < 1/\bar{\tau}_0$*, or, equally to say, the critical input traffic is* $\lambda_c = 1/\bar{\tau}_0$*.*

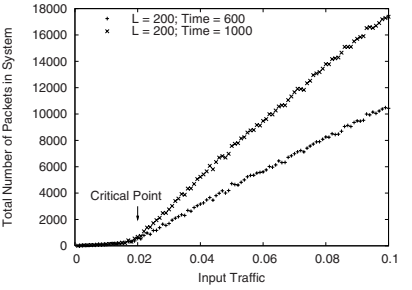
The theorem, esp. the formula (14), shows that the free delay of a network does significantly impact on the critical traffic. Actually, the delay linearly *amplifies* input traffic to site traffic.

It is emphasized that, in the Mean-Field Theory demonstrated above, there is not any requirement to lattice dimensionality, nor its neighborhood type. The analytical result of Corollary 1 is universally available, provided the Mean-Field Theory assumptions are conforming to physical properties of the model.

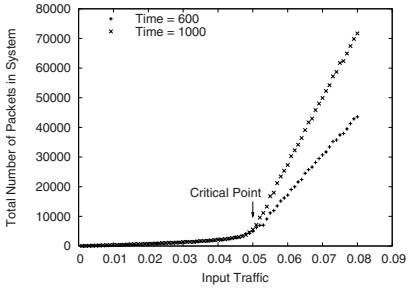
[2] The third assumption plays an important role here. Note that in formula (9), $\bar{q}$ is, in fact, the ensemble average; while in Little's Law (12), $\bar{q}$ is actually a time average. Without the ergodicity assumption, the two formula could not be combined.



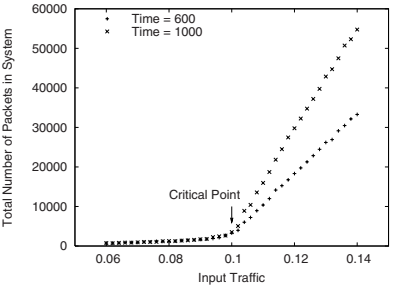
(a) $d = 1$, $L = 100$, $\bar{\tau}_0 = 25$, $\lambda_c = 0.04$



(b) $d = 1$, $L = 200$, $\bar{\tau}_0 = 50$, $\lambda_c = 0.02$



(c) von Neumann lattice, $d = 2$, $L = 40$, $\bar{\tau}_0 = 20$, $\lambda_c = 0.05$



(d) Moore lattice, $d = 2$, $L = 30$, $\bar{\tau}_0 = \frac{1801}{180} \simeq 10.0$, $\lambda_c = 0.10$

Fig. 1. Simulation samples for showing the critical behavior. With whatever dimensionality and whatever neighborhood, the model's critical traffic λ_c is almost equal to $1/\bar{\tau}_0$.

3.3 Simulation Samples

As the upper bound of input traffic for existence of stable state, the critical behavior in an instance of the model may be demonstrated by its queue length processes or the total number of packets in the entire system, say $Q(k) \triangleq \sum_{\mathbf{x}} q(\mathbf{x}, k)$. If the input traffic exceeds the critical point, then $Q(k)$ will increase to infinity, in stead of getting stable. These phenomena are shown in Figure 1, where several sample cases with variety of parameters or topology characteristics are provided. Figure 1(c) is similar to the case that [3] has provided. However, we have extended the results on critical traffic to far more general environments.

4 Queue Length Estimation

As any other Mean-Field approaches to statistical-physical systems, the Mean-Field Theory presented here can *not* be treated as an accurate quantitative result, though it almost accurately predicts the critical behavior of the model.

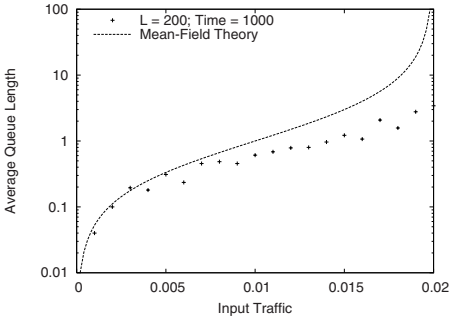


Fig. 2. Comparison simulation to Mean-Field Theory on a one-dimensional sample with $L = 200, k = 1000$, for stable state queue length. The Mean-Field Fluid approximation overestimates the average queue length.

4.1 Queue Length of Stable State

Theorem 1 has given the mean value for the queue length on any site in the stable state, if it exists. However, the result overestimates it a little.

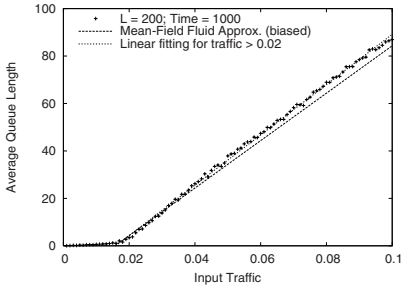
The overestimation may originated from approximating the model to an open Jackson network. Each site in the model is defined as an M/D/1 queueing system, while it becomes to M/M/1 in the Jackson network. It has been proved that, for Markovian routing schemes, an M/D/1 queueing network has less average queueing delay (or, equivalently, less queue length) in stable state than its M/M/1 counterpart [13].

4.2 Fluid Approximation of the Mean-Field Theory

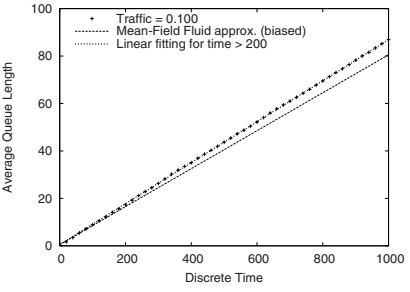
On the other hand, when the value of input traffic exceeds the critical point, it is presented in the simulations that queue length of a site approximately grows as a linear function to both time and input traffic. To explain this phenomenon, we combine the assumptions of the Mean-Field Theory and (13) with the Fluid approximation in queueing theory [14].

The Fluid approximation, based on the law of large numbers, replaces discontinuous stochastic arrival and departure processes with continuous deterministic versions. Let $\overline{\alpha(\mathbf{x}, t)}$ and $\overline{\delta(\mathbf{x}, t)}$ represents, respectively, the two continuous processes for packet arrival and departure happening at site $\mathbf{x} \in \mathcal{L}^d$ in our model. Then for the queue length approximation $\overline{q(\mathbf{x}, t)}$, we have

$$\frac{\partial \overline{q(\mathbf{x}, t)}}{\partial t} = \overline{\alpha(\mathbf{x}, t)} - \overline{\delta(\mathbf{x}, t)}, \quad \forall \mathbf{x} \in \mathcal{L}^d \tag{15}$$



(a) Fixed time ($t = 1000$). The Mean-Field Fluid approximation shows a line with slope $t = 1000$; while the linear fitting is $\overline{q(\lambda)}\Big|_{t=1000} = 1061.31\lambda - 16.86$



(b) Fixed traffic ($\lambda = 0.10$). The Mean-Field Fluid approximation shows a line with slope $(\lambda - \lambda_c) = 0.08$; while the linear fitting is $\overline{q(t)}\Big|_{\lambda=0.10} = 0.0863t + 0.528$

Fig. 3. Comparison the simulation result to Mean-Field Fluid approximation on a one-dimensional sample with $L = 200$, for queue length growth far from stability. The Mean-Field Fluid approximation is near to but a little underestimates the queue growth rate.

Arrivals are resulted by both traffic input and forwarding events, and the departure rate is constant $\mu = 1$. Therefore, we have the Fluid version of (1).

$$\overline{\alpha(\mathbf{x},t)} = \overline{\alpha(\mathbf{x},0)} + \int_0^t \lambda dy + \sum_{\mathbf{y}\in A(\mathbf{x})} r_{\mathbf{yx}}\overline{\delta(\mathbf{y},t)}, \tag{16}$$

$$\overline{\delta(\mathbf{x},t)} = \overline{\delta(\mathbf{x},0)} + \int_0^t dy = \overline{\delta(\mathbf{x},0)} + t, \quad \forall \mathbf{x} \in \mathcal{L}^d \tag{17}$$

With (13), i.e. $r_{\mathbf{yx}} = (1 - 1/\bar{\tau}_0)/A$, another corollary is obtained.

Corollary 2. *In the Mean-Field Theory of the model, approximated with Fluid model, queue length at any site is growing linearly if the input traffic exceeds the critical value, and the growth rate is*

$$\frac{d\overline{q}}{dt} = \lambda - \frac{1}{\bar{\tau}_0} = \lambda - \lambda_c, \quad \forall \mathbf{x} \in \mathcal{L}, t \to \infty \tag{18}$$

The result is compared to simulations with either time fixed and input traffic variant, or vise-versa (Figure 3). Because the integral of (18) contains an arbitrary constant, we add the intercept of the linear fitting (as a bias) to the Mean-Field Fluid approximation, in order to focus the comparison on the growth rate. The Mean-Field Fluid approximation underestimates the rate of queue growth because Fluid approaches replace the stochastic processes with deterministic (D/D/1) systems [14]. The larger the input traffic is, the more the estimated rate is close to the reality.

5 Summary

This paper develops a modelling method for studying distributed network behaviors. It promotes the previous works of simulation to an approximated analytical result, where the law of critical traffic is proved under a certain set of Mean-Field assumptions. Therefore, one may improve the critical behavior of a network by minimizing the free delay of its topology equivalently. Analyzing non-isotropic, non-homogeneous and non-ergodic models, which are more close to real systems, are of the successive work after this paper.

On the other hand, the Mean-Field Theory is not able to accurately estimate average queue length in a network. Generally speaking, the farther the input traffic is away from the the critical point, the better the Mean-Field theory estimates queue length behavior. One should take a deep sight at the local details, where the fluctuations are ignored by the Mean-Field Theory.

The work is a part of research efforts on next-generation distributed network architectures. The critical traffic law presented here implies that free packet delay determines the rate of packet departure and the critical traffic behavior. This is especially important for packet-switched overlay networks. Without fixed infrastructure, an overlay's (e.g. IPv6 over IPv4) topology might be so badly deployed that a packet is considerably more delayed than that when it were delivered in the physical network. This might make the physical network filled up with deadweight. Detailed study on interaction between overlay and physical networks is in progress.

References

[1] Tretyakov, A., Takayasu, H., Takayasu, M.: Phase transition in a computer network model. Physica A **253** (1998) 315–322 773
[2] Deane, J., Smythe, C., Jefferies, D.: Self-similarity in a deterministic model of data transfer. Journal of Electronics **80** (1996) 677–691 773
[3] Fukś, H., Lawniczak, A. T.: Performance of data networks with random links. Mathematics and Computers in Simulation **51** (1999) 101–117 773, 775, 779
[4] Liu, F., Ren, Y., Shan, X. M.: A simple cellular automata model for packet transport in internet. Acta Physica Sinica **51** (2002) 1175–1180 773
[5] Ohira, T., Sawatari, R.: Phase transition in a computer network traffic model. Physics Review E **58** (1998) 193–195 773, 775
[6] Baran, P.: History, alternative approaches and comparisons. In: On Distributed Communications. Volume V. RAND RM-3097-PR (1964) 773
[7] Baran, P.: Introduction to distributed communications network. In: On Distributed Communications. Volume I. RAND RM-3420-PR (1964) 773
[8] Wolfram, S.: Universality and complexity in cellular automata. Physica D **10** (1984) 1–35 774
[9] Weimar, J. R.: Cellular Automata for Reactive Systems. Phd dissertation, Université Libre de Bruxelles, Belgium (1995) 774
[10] Kleinrock, L.: Queueing Systems. Volume I: Theory. Wiley Interscience (1976) 776

[11] Sheng, Y. Z.: Queueing Theory and its Applications to Computer Communication (in Chinese). Beijing University of Post-Telecommunication Press (1998) 776
[12] Chaikin, P. M., Lubensky, T. C.: Principles of Condensed Matter Physics. Cambridge University Press (1997) 776
[13] Harchol-Balter, M.: Network Analysis without Exponential Assumptions. Phd dissertation, University of California at Berkeley (1996) 780
[14] Kleinrock, L.: Queueing Systems. Volume II: Computer Applications. Jon Wiley & Sons (1976) 780, 781

Constructing an Overlay Using a Shared Object Set for Streaming Services on a P2P Network

Hyunjoo Kim and Heon Y. Yeom

School of Computer Science and Engineering, Seoul National University
Seoul, 151-742, Korea
{hjkim,yeom}@dcslab.snu.ac.kr

Abstract. In an on-demand streaming service on a P2P network, a client peer can receive parts of a video object from different peers instead of from just one server. For this, the client peer must find enough peers that have the required object. In this paper, we propose a method of constructing an overlay to search for objects efficiently for the on-demand streaming services on a pure P2P network. The proposed overlay is composed of groups, and all peers in a group have common objects. We compared our method with Gnutella by simulation. The results show that our method reduces traffic overheads, hop counts, and the number of messages, at the cost of join/leave overhead.

1 Introduction

Video streaming services are limited in client/server architectures because the number of clients that a server can support and the outbound bandwidth of a server are limited. In CoopNet [1], a client having the required object can provide a service to another client in place of a server. However, CoopNet is based on the client/server architecture, and many authors have tried to address this problem using many ordinary nodes as servers in a P2P network.

There are two kinds of streaming service. In live media services, one source broadcasts video data to many clients, whereas in an on-demand service, stored video data in a storage system are published.

The main approach to improving the quality of live media services is application-layer multicast trees, which have been reported in Narada [2], SpreadIt [3], NICE [4], and ZIGZAG [5]. In these trees, a source becomes a root and clients become tree nodes. On the other hand, in on-demand services, many source peers can provide a streaming service to a client peer because they transmit a stored video object from their own storage systems [6, 7]. In [8] and [9], an overlay network is composed of multicast groups with clients, and the same video object is served to the group members.

Super-peers [10] are installed in Gnutella to manage peers and find objects efficiently. However, because super-peers are intended to be servers rather than normal nodes, they absorb the installation and management overheads, and the overheads increase as the number of nodes increases.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 784–792, 2004.

In this paper, we focus on on-demand streaming services from multiple server peers in a pure P2P network. We propose an overlay network that is composed of groups, with one peer in each group selected as a directory server for the management of its group, while the other peers in the group provide streaming services. These roles are determined dynamically and autonomously when peers join the network by comparing their shared objects.

2 Overlay Scheme

In our proposed scheme, all peers in a group have as many common shared objects as possible. A group consists of a *Leader Peer (LP)*, which is the representative of the group, and some *Member Peers (MP)*. A peer becomes an LP if it has some unique objects not in the current network, otherwise it becomes an MP of one or more groups sharing the most common objects with it.

All requests are forwarded to the network through LPs. Because LPs serve as directory servers, when they receive a request they forward it depending on conditions. However, LPs do not provide streaming services to client peers, as the MPs do this. When an LP receives a request, it forwards the request to other LPs only if it cannot be satisfied within the group. A request is flooded only to LPs, not to all peers in the network. The LPs forward the request to any of their own members that have the requested object.

Each LP manages the location information of other LPs for flooding requests. In addition, each LP manages the location information of its MPs for forwarding requests.

2.1 Joining Process

We assume that a join peer knows the location of a peer that manages the process of entering the network (entry peer) and that only LPs can be entry peers. A join peer sends a join request with <*peer id, shared object set*> to the entry peer. When the LP (entry peer) receives the request, it compares its own shared object set with that of the join peer. If the LP has all the objects that the join peer shares, it accepts the join peer as a member. Otherwise, the LP forwards the join request to other LPs to find a more appropriate LP for the join peer. Each LP compares its shared object set after receiving the request and then replies to the join peer with the common shared object set and a join flag. The join peer gathers the object sets and checks whether it has any unique object not in the current network. If it has such objects, it becomes a new LP. If not, it determines which LP is the best to join and then becomes an MP to that LP. The join peer can be a member of one or more groups and it selects LPs to join by the number of common objects. If the join peer has all the objects of any LP, the join peer replaces the LP and makes it its MP.

Join flags represent the relations between shared object sets and are classified into the following (ShrObjSet(p) means the shared object set of a peer p):

Table 1. Join flags example

	shared object set				
LP	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3
join peer	1, 2	2, 3, 4	1, 2, 3, 4	1, 2, 3	4
join flag	ALL	PART	SUB	EQUAL	NONE

ALL: {ShrObjSet(join peer) $\subset$ ShrObjSet(LP)}
PART: {ShrObjSet(join peer) $\cap$ ShrObjSet(LP)$\neq \emptyset$} and {ShrObjSet(join peer) - ShrObjSet(LP) $\neq \emptyset$} and {ShrObjSet(LP) - ShrObjSet(join peer) $\neq \emptyset$}
SUB: {ShrObjSet(LP) $\subset$ ShrObjSet(join peer)}
EQUAL: {ShrObjSet(LP) == ShrObjSet(join peer)}
NONE: {ShrObjSet(join peer) $\cap$ ShrObjSet(LP) == $\emptyset$}

Table 1 shows examples of join flags.

Figure 1-(1) shows an example of a join process. A join peer has a shared object set {3,4} and (1) sends a join request to an entry peer, LP4. LP4 compares its shared object set with that of the join peer's. LP4 does not have all objects of the join peer, hence, (2) it forwards the join request to the other LPs, that is, LP1, LP2 and LP3. The LPs receiving the request compare their own shared object sets with that of the join peer and then (3) reply to the join peer with the common object sets and join flags. The replies are <{3}, PART> for LP1, <{4}, PART> for LP2, <$\emptyset$, NONE> for LP3, and <{3}, PART> for LP4.

A join peer decides which group to join based on the replies from the LPs. If some LPs send EQUAL or ALL for a join flag, the join peer can be an MP of all of those LPs.

If all LPs reply with the NONE flag, it means that no LP has objects in common with the join peer. Hence, the join peer becomes a new LP and makes a new group. In the new group, there are unique objects not in the other groups, so service requests for those unique objects can be served from only this group. Later, if another peer joins the network and has those objects plus some new unique objects, it will become a new LP and those objects of the previous LP can be published in this group too.

If some LPs reply with NONE, they have no common objects with the join peer and if some reply with PART, they have some common objects. However, the join peer does not know whether or not it can be an LP, because it cannot know whether or not it has unique objects in the current network. Hence, the join peer gathers the replies from the LPs and checks if it has any unique objects. If so, it becomes an LP, otherwise it becomes an MP of one or more of the LPs that sent PART flags. The join peer can connect to only the LP that has the most objects in common with it, or to some LPs that send PART flags. If a join peer connects to more LPs, it will receive more hits on requested objects as a client. *Connection rate* is the number of actual connections divided by the number of

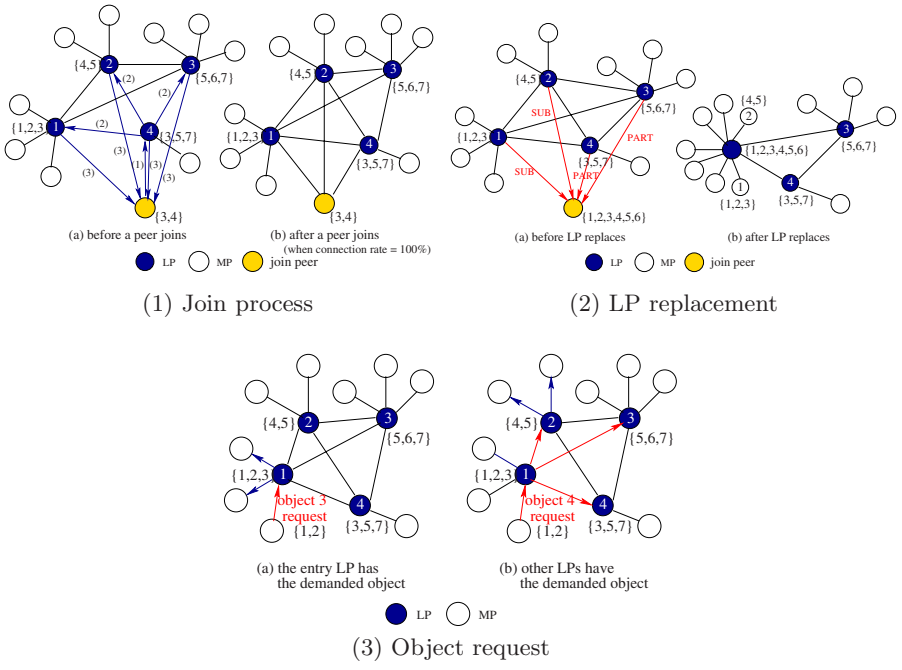


Fig. 1. Overlay scheme example

all possible connections between a join peer and LPs, and indicates the number of LPs with which a join peer is connected.

For example, assume that a join peer has a shared object set {2, 3, 4, 5 } and LP1, LP2, and LP3 have {1, 2, 3, 4}, {4, 5, 6}, and {3, 6, 7} respectively. LP1, LP2, and LP3 send the replies <{2, 3, 4}, PART>, <{4, 5}, PART>, and <{3}, PART> flags to the join peer. The join peer can connect to all the LPs (LP1, LP2, LP3) or some of them, or only one. If the join peer connects to only one LP, it must be LP1, because LP1 has the most common shared objects. The connection rate is a percentage value of the number of common objects divided by the maximum number of common objects. In this example, the maximum number of common objects is three (by LP1). Hence, if the join peer connects to LP1, the connection rate will be 0%, for LP1 and LP2 it will be 50%, and for all LPs it will be 100%. In Fig. 1-(1), the join peer connects to LP1, LP2, and LP4, which send PART join flags, for a connection rate of 100%.

If one or more LPs send SUB join flags, it means that the join peer has all the objects of those LPs. Hence, the join peer replaces those LPs and they and their MPs become members of the new LP. At all times, a new LP has some unique objects not previously in the network. Figure 1-(2) shows the process by which some LPs are replaced by a join peer. The join peer has a shared object set {1, 2, 3, 4, 5, 6}. When the join peer sends a join request, LP1, LP2, LP3, and LP4 return <{1, 2, 3}, SUB>, <{4, 5}, SUB>, <{5, 6}, PART>, and

<{3, 5}, PART>, respectively. The join peer becomes a new LP controlling the group members of LP1 and LP2 because the join peer has all the objects of LP1 and LP2.

2.2 Search Process

All requests are forwarded through LPs. Both LPs and MPs can request services. An LP requests object services from other LPs when it does not have the objects in its group. The request of an MP can be satisfied within its own group when its LP has the requested object. If the LP has the requested object, then some members also have the object. Hence, an MP makes a request to its LP first. If the LP has the object, the LP forwards the request to its other MPs. Otherwise, the LP forwards the request to the other LPs.

Figure 1-(3)(a) shows an example in which a request is satisfied within a group. When an MP requests object 3 from its leader LP1, LP1 checks whether it has the requested object. It does, so it forwards the request to its other MPs. The MPs in the group that have the requested object reply to the client and can be server peers. Then the client MP requests different segments of the object from each server peer using the assignment methods in [7]. If the client does not find enough server peers, LP1 forwards the request to the other LPs to search for more server peers.

When an MP requests an object from its LP but the LP does not have that object, the LP forwards the request to the other LPs. Those LPs that have the requested object forward the request to their own MPs. In Fig.1-(3)(b), a client peer requests object 4 from its leader LP1. LP1 forwards the request to the other LPs because it does not have that object. Among those LPs, LP2 has object 4 and can provide the service. Hence, only LP2 forwards the request to its MPs and the MPs having the object reply to the client. The client selects the server peers from those MPs.

2.3 Leaving Process

When an MP leaves the P2P network, it sends a "leave" message to the LP of its group and the LP deletes the MP's information from the MP list. When an LP leaves the P2P network, a member of its group replaces it and the new LP notifies the other group members of its takeover. The MPs of the group change their information about the LP. Without takeover, all the MPs in a group would be disconnected from the network by the departure of their LP. To address this problem, the LP selects a candidate LP to replace it.

An LP selects the candidate LP that has the most of the common shared objects. The LP knows the shared object sets of all its MPs because peers send their shared object sets when they join the network. Hence, an LP updates the information about possible candidate LPs whenever a new join peer has more common shared objects than the current candidate LP. However, LPs do not manage the detailed information about the shared object sets of MPs.

Table 2. Simulation parameters

Parameter	Value
Number of peers	5000
Number of total objects in P2P network	1000
Max. number of objects of a peer	50–800
Interval between requests by a peer	86400 sec. (exponential distr.)
Lifetime of peers	1800–14400 sec. (exponential distr.)
Idle time between connections	21600–345600 sec. (exponential distr.)
Connection rate	0–40%
Number of neighbors in Gnutella	10
Time-to-live (TTL) in Gnutella	7
Simulation time	864000 sec. (10 days)

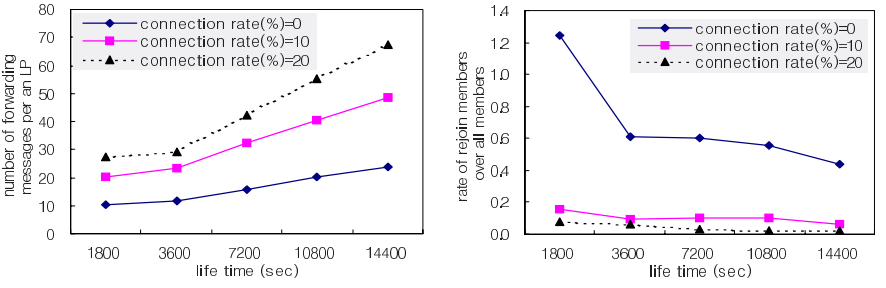
When an LP leaves the network, it sends a "take over" message to its candidate LP. The candidate LP takes over and then notifies the other LPs and its MPs of the change of LP. Then the MPs replace their current LP with the new LP. After the change of LP, some MPs may not be members of the group any longer. For example, assume that the current LP has the shared object set {1, 2, 3, 4, 5}, its candidate LP has {3, 4, 5}, and one of its MPs has {1, 2}. The candidate LP and the MP can be in the same group with the current LP because they have the common object sets {3, 4, 5} and {1, 2}, respectively. However, after the candidate LP takes over, the MP cannot be in the same group because it has no common objects with the new LP. To address this case, the candidate LP sends a state-change message to its MPs after takeover. All MPs compare their own object sets with the LP's and inappropriate MPs in the group rejoin the network.

3 Simulation

In this section, we present and discuss simulation results. The parameters used for the simulation are shown in Table 2. We used CSim for the simulation. Peers connect to (join) the network and request services or provide them during their lifetimes and then disconnect from the network. All peers repeat this processes during the simulation time.

When the connection rate is large, a join peer becomes the member of more groups. Hence, the number of MPs of an LP increases when the connection rate increases. Similarly, the hit rate and the number of redundant forwarding messages also increase.

It is not necessary for a client to find all the peers having the requested object for an on-demand streaming service. In the proposed scheme, a client can find the appropriate number of peers for a service by adjusting the connection rate. Unnecessary messages for a search can be reduced when the connection rate is small. In Fig. 2, the performance is compared as the lifetime varies for 0%, 10%, and 20% connection rates. Figure 2-(a) shows that the forwarding overhead for an LP is small for small connection rates because the group size is small. However, Fig.2-(b) shows that a 0% connection rate can cause problems,



(a) (b)

Fig. 2. The effects of connection rate

as all MPs in a group must rejoin the network when the group's LP leaves the network. This makes the rejoin overhead high. Hence, the selection of the proper connection rate can improve the performance by reducing overheads for leaving nodes.

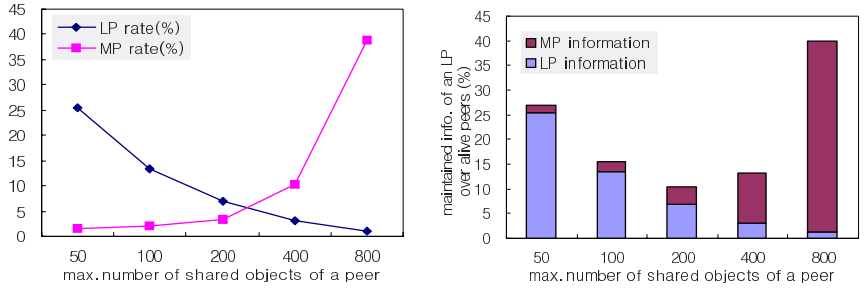
In Fig. 3-(a) the LP rate is the number of LPs divided by the number of live nodes and the MP rate is the number of members in a group divided by the number of live nodes. If peers have very large stable storages and they can share enormous numbers of objects, the probability that a peer has unique objects in the network becomes low. Hence, the number of LPs decreases and the number of MPs increases. If we assume the size of an object is 600 MB, the realistic number of objects of a peer is about 100–200 and the LP rate is about 5–15%. Figure 3-(a) also shows that each LP manages less than 5% of the live peers as MPs when the maximum number of shared objects is less than 200.

Figure 3-(b) shows the amount of information an LP must maintain. An LP manages the location of all LPs and its MPs. As the maximum number of shared objects in each peer increases, each LP maintains less LP information and more MP information. In this figure, the number of LPs and MPs is balanced when the maximum number of shared objects is 200. Even though we cannot control the number of LPs and MPs, there is a balanced value depending on the maximum number of shared objects.

We compared the performance with Gnutella. We do not show the results because of the space limitation, but we found that the proposed scheme reduces traffic overhead, hop counts, and the number of messages compared to Gnutella, at the cost of overhead as peers join and leave the network.

4 Conclusion

In this paper, we proposed an overlay scheme using shared objects for an on-demand streaming service in a pure P2P network. The proposed scheme is composed of groups and all peers in a group have common objects. A leader peer



(a) (b)

Fig. 3. The overheads of the proposed overlay

(LP) in a group serves as a directory server while the member peers (MPs) provide on-demand streaming services. The role of peers is decided autonomously when a peer joins the network, and service requests are managed by some peers without any server installation and management cost. The environment for the on-demand streaming service can be tuned by careful selection of the connection rate.

References

[1] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai, "Distributing streaming media content using cooperative networking," in ACM/IEEE NOSSDAV, 2002. 784

[2] Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang, "A case for end system multicast," in ACM SIGMETRICS, 2000. 784

[3] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming Live Media over a Peer-to-Peer Network," Stanford Database Group Technical Report (2001-30), Aug. 2001. 784

[4] S. Banerjee, Bobby Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in ACM SIGCOMM, 2002. 784

[5] Duc A. Tran, Kien A. Hua, and Tai Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in IEEE INFOCOM, 2003 784

[6] W. T. Leung and J. Y. B. Lee, "A server-less architecture for building scalable, reliable and cost-effective video-on-demand systems," in Internet2 Workshop on Collaborative Computing in Higher Education: Peer-to-Peer and Beyond, 2002 784

[7] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava, "On Peer-to-Peer Media Streaming," in IEEE ICDCS, 2002. 784, 788

[8] Kien A. Hua, Duc A. Tran, and Roy Villafane, "Overlay multicast for video on demand on the internet," in ACM Symposium on Applied Computing, 2003. 784

[9] Duc A. Tran, Kien A. Hua, and Simon Sheu, "A new caching architecture for efficient video services on the internet," in IEEE Symposium on Applications and the Internet, 2003. 784

[10] B. Yang and H. G. Molina, "Designing a super-peer network," in IEEE ICDE, 2002. 784

Cost-Effective Design of GMPLS Networks with Sparse Multi-granularity Optical Cross-Connect

Dae-Gun Kim[1], Myungmoon Lee[2], Jun Kyun Choi[3], Jinwoo Park[4], and Chul-Hee Kang[4]

[1] KT, Korea University, 206 Jungja-dong, Bundang-gu
Sungnam City, Kyonggi-do, 463-711, Korea
dkim@kt.co.kr
[2] Suwon Science College, P.O.BOX 57, Suwon Post Office
San 9-10, Botong-ri, Jeongnam-myun, Hawsung-si, Kyunggi-do, Korea
mmlee@ssc.ac.kr
[3] Information and Communication, University (ICU)
58-4, Hwaam-dong, Yuseong-gu, Daejeon, 305-732, Korea
jkchoi@icu.ac.kr
[4] Electronic and Computer Engineering, Korea University
1,5-ka, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
{jwpark,chkang}@korea.ac.kr

Abstract. In this paper, we propose the efficient placement of MG-OXC (multi-granularity optical cross-connect) nodes in GMPLS networks to reduce all the network cost. When only a limited number of nodes are allowed to have the capability of waveband switching, it is shown that selection based on max-traffic performs better than Random and Nodal-degree schemes. Performance of the sparse placement of max-traffic scheme can be achieved very close to that of all placement of MG-OXC.

1 Introduction

Generalized Multi-Protocol Label Switching (GMPLS) is being developed in the Internet Engineering Task Force (IETF) [1]. In GMPLS with ordinary-OXC networks, pass-through traffic dominates over add-drop traffic in large-scale networks. Since many lightpaths conveying path-through traffic might have the same route, the size of the optical switch matrix would be largely reduced if they were dealt with as a single channel. Fig.1 shows the node architecture for networks employing MG-OXC where the direct waveband add/drop ports in waveband crossconnect (WBXC) are added to the ordinary-OXC architecture described in [2]. The switching units of FXC, WBXC, and WXC are fiber, waveband, wavelength, respectively. Note that wavelength con-version is allowed only at the wavelength crossconnect because of the significant technical difficulty of waveband conversion. In all previous research on waveband switching in optical networks, it is assumed hat every network node has waveband switching capability, which may not be practical or cost-effective in a nationwide optical

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 792–799, 2004.

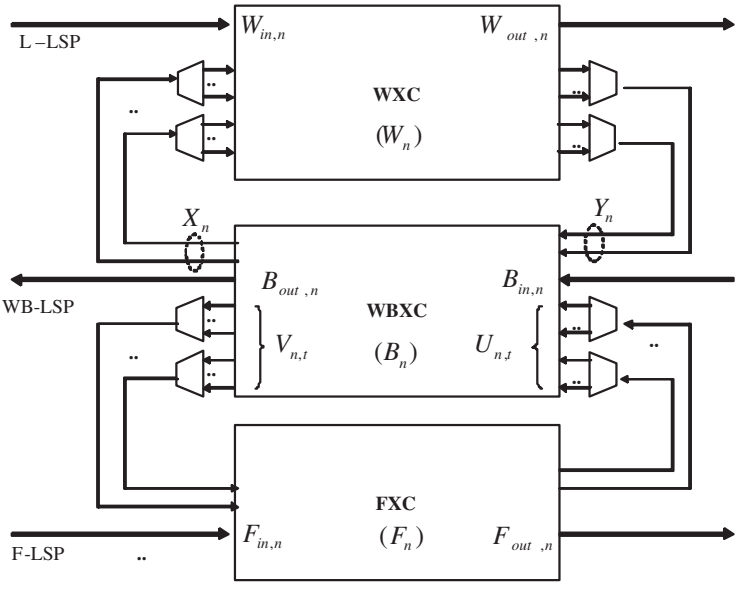


Fig. 1. The proposed architecture of MG-OXC [2]

backbone network [3,4]. In this paper, we propose a selection method for sparse MG-OXC nodes in GMPLS networks. We also propose a heuristic design procedure as a practical method to general large scale networks (not presented in this paper because of space limitation). By applying the proposed design method to ARPA network, the benefits of the sparse MG-OXC placement are discussed with respect to the traffic load.

2 Design of GMPLS Networks with Sparse MG-OXC Nodes

The characteristics of three different cost functions for selecting MG-OXC nodes, namely nodal-degree selection, maximum flow selection, and random selection. Note that some ideas on these node selection schemes are borrowed sparse-wavelength-converter-placement studies [5].

Nodal-Degree Selection: In this scheme, the first M nodes which have the maxi-mum nodal degree are picked to be MG-OXC nodes. If several nodes have same nodal degree and only some of them can be chosen, random selection is used to break any ties.

Max-Traffic Selection: For a given node ν , the total amount of traffic which may pass-through the node is computed, assuming that each traffic request is routed physical network topology using a k-shortest path routing algorithm. The M nodes which have maximum amount of pass-through traffic flow

can be selected as MG-OXC nodes. Instead of routing the traffic requests between a node pair (s, d)using a single shortest path route, it may be also possible to compute alternate paths between (s, d) and bifurcate the traffic among these K alternative paths.

Random Selection: In this scheme, M nodes are randomly selected to be MG-OXC nodes.

One can also design other schemes to select the MG-OXC nodes. To minimize the size of MG-OXC with the minimum number of wavelengths and to select the MG-OXC with maximum traffic, the heuristic design method explained below consists of three stages: routing, selecting the MG-OXC nodes, and waveband assignment in MG-OXC node

2.1 Routing of Lightpaths

To maximize the number of complete waveband paths, it is clear that the light-paths with the same destination should have the same route. Therefore, we make all light-paths in the network satisfy the optimality principle. That is, if node y is on the optimal path from node x to node z, then the optimal path from node y to node z follows the same route on that part from x to z. As an optimal path, we use *k-shortest path* because the size of the MG-OXC is proportional to the number of hops of a path. Based on optimality principle, we can construct auxiliary graphs for each destination node. However, several auxiliary graphs may be found for one destination node be-cause multiple shortest paths may exist. In order to choose an auxiliary graphs leading to minimum number of wavelengths, we use least loaded routing. This procedure is summarized as follows:

1. Find the *k-shortest* paths for each lightpath request.
2. List all lightpath requests in descending order of minimum hop path length.
3. Decide the route of each lightpath request on the list.
4. If a lightpath request has several shortest paths, select a path with the least link loading on the route.

2.2 Selecting the MG-OXC with Max-Traffic

After selecting the route of each lightpath, we determine which waveband each light-path would be grouped into. To achieve a large reduction gain of the size of MG-OXC, it is clear that the lights path with many common links should be grouped into a wave-band. We use the following simple lightpath grouping rules.

1. Classify each lightpath into classes according to the destination node, and list all lightpaths in each class in descending order of the number of hops.
2. Select a lightpath on the top of the list in a class and find $W_B - 1$ lightpaths which have the most common links to form a waveband group. Then assign a waveband number to the group of lightpaths and remove them from the list. Reapply this procedure to the remaining lightpsths in the class.
3. Repeat step 2. for the other classes.

4. Once the wavebands and the routes for all lightpaths are determined, calculate the size of MG-OXC and list all MG-OXC in descending order of maximum size of MG-OXC.
5. Select the MG-OXC nodes, M on the list of MG-OXC.

2.3 Wavelength Assignment in a Waveband of MG-OXC Node

Each lightpath in a waveband path should have a wavelength for all links of its route because the wavelength conversion is not allowed in WBXC. We propose following new wavelength assignment rule for the lightpaths in a waveband of MG-OXC nodes.

1. Select a group of MG-OXC nodes, which has the Max-Traffic, and do the following.
2. Choose a node which has the Max-Traffic. If several candidates exist, select the waveband path is the longest.
3. If w is set as the lowest wavelength in each waveband path on the list, lightpaths in this waveband path may have one of the wavelengths from w to $w+W_B-1$. Then solve the following binary linear programming, where W_B is wavelength granularity:

$$\textit{Maximize} \qquad \sum_{p=1}^{W_B} \sum_{k=w}^{W_B+k} \alpha_{p,k}\beta_{p,k} \tag{1}$$

$$\textit{Subject to} \qquad \sum_{k=w}^{W_B+k} \alpha_{p,k}\beta_{p,k} \leq 1, \qquad \forall p = 1, \ldots, W_B \tag{2}$$

$$\sum_{p=1}^{W_B} \alpha_{p,k}\beta_{p,k} \leq 1, \qquad \forall k = w, \ldots, W_B + w \tag{3}$$

   where $\beta_{p,k}$ is a binary variable that becomes 1 when pth lightpath in a complete waveband path is assigned wavelength k, and $\alpha_{p,k}$ is binary constant that becomes 1 when wavelength k is available for pth lightpath in a complete waveband path. The value of objective function is equal to W_B if every lightpath in a complete waveband path is successfully assigned a wavelength. If not, increase w by W_B and solve again the above binary linear programming until the objective value is equal to W_B as shown Fig.2.
4. If any MG-OXC nodes in the group are not assigned the waveband, then go to step 2). Otherwise, go to step 5.
5. If all MG-OXC nodes are assigned the waveband, this algorithm finishes. Otherwise, go to step 1.

The above problem formulation is quite tractable because the number of constraints and the number of variables are $2W_B$ and W_B^2 , respectively, and independent of network size.

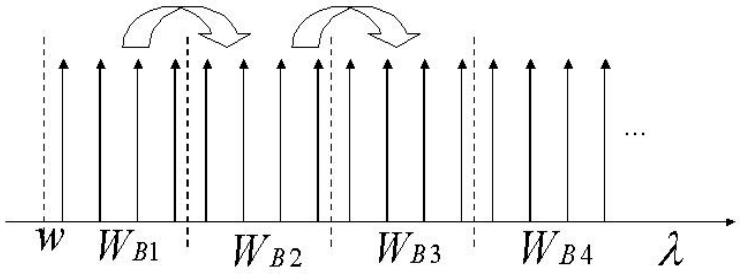


Fig. 2. An example of wavelength assignment in a complete waveband. W_{B1} and W_{B2} are not selected if every lightpath in a complete waveband path is failed when wavelengths are assigned to every lightpath in wavelength granularity W_B . However, W_{B3} is selected because every lightpath in a complete waveband path is successfully assigned a wavelength

3 Numerical Results of Heuristic Design

We calculate six parameters of MG-OXC networks that Mwxc, Mwbxc, SumMax, Twxc, Twbxc, and SumTotal are the maximum WXC size, the maximum WBXC size, the sum of Mwxc and Mwbxc, the sum of WXC sizes in all nodes, the sum of WBXC sizes in all nodes, and the total sum of Twxc and Twbxc, respectively. The ARPA network with N=20 and L=30 was chosen as a test network [2]. Fig.3 shows the example of MG-OXC selection when $W_B = 8, M = 14$. In this case, node 10 has largest waveband number, node 19 has has least waveband number. Fig.4 shows the number of port of MG-OXC with respect to the various wavelength granularity values. For uniform demand pattern of $D_p = 2$, we can achieve a maximum reduction gain of more than 50% for SumTotal at $W_B = 6$ (SumTotal of MG-OXC and ordinary OXC are 4492 and 8676 ports, respectively). Therefore, it can be concluded that an optimal wavelength granularity value leading to maximum reduction gain may exist. However such an optimal wavelength granularity value may depend on network topology, traffic demand, and traffic pattern. Nonetheless, the reduction of MG-OXC size can be still be expected with non-optimal wavelength granularity values. Another advantage is that the maximum size of WXC and WBXC in MG-OXC network is less than half of the maximum size of ordinary OXC (that is 619 ports). On the contrary, disadvan-tage of MG-OXC is the increase in number of required wavelength at overall wave-length granularity values. Also, the number of port in MG-OXC is saturated to any values as the increase of wavelength granularity values. From Fig.4, we can see the relation of waveband and traffic demand per node. In less traffic load, smaller waveband needs less number of ports. However, in more traffic load, more wavelength granularity needs less number of ports. In Fig.5, we observe that the Max-Traffic selection can achieve much better performance than Random and Nodal-Degree selection. This validates the importance of problem of MG-OXC placement. When there are five MG-OXCs, Nodal-Degree and Max-traffic selection result in the same placement scheme, which have better

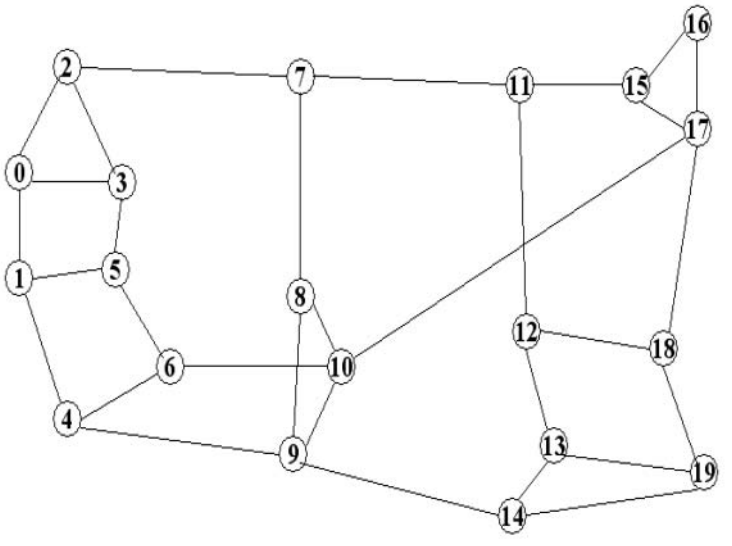


Fig. 3. An Example of MG-OXC placement when $W_B = 8, M = 14$

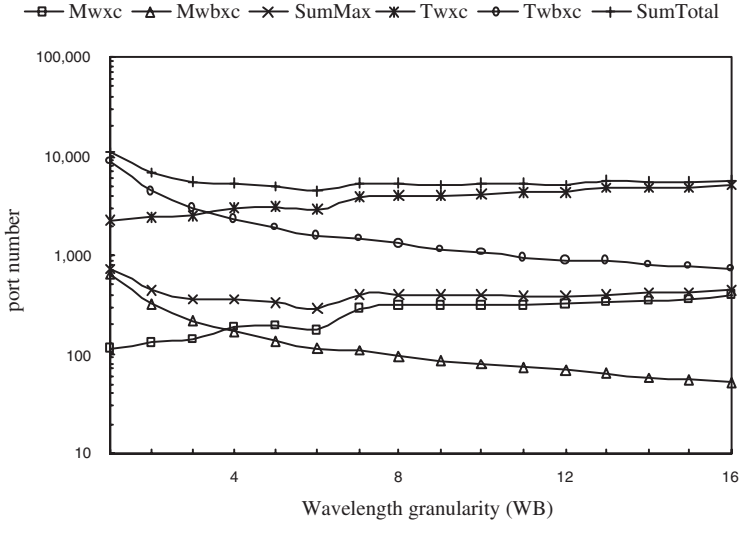


Fig. 4. The effect of wavelength granularity for uniform demand pattern of $D_p = 2$

performance than Random. When there are ten MG-OXCs, we observe that with the number of waveband switching increased, the blocking probability is increased because of wave-length shortage. From the above simulation analysis, we conclude that sparse MG-OXC placement improves blocking probability performance significantly in mesh networks if the MG-OXCs are placed appro-

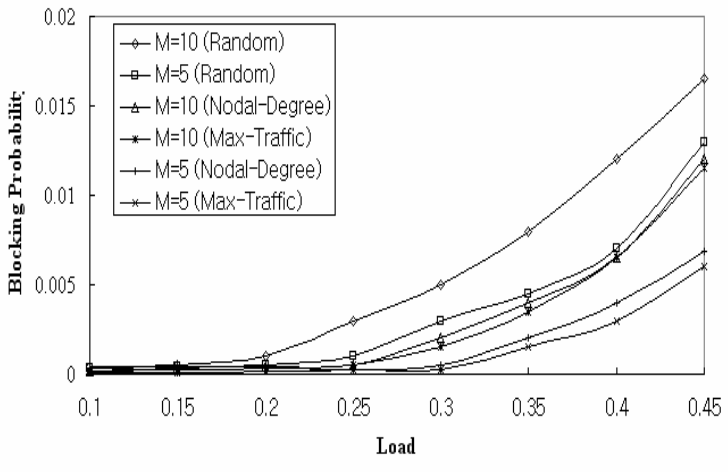


Fig. 5. Blocking probability vs. traffic load in the 20-node ARPA network

priately. The proposed Max-traffic selection for MG-OXC can achieve better performance than Random and Nodal-Degree selection.

4 Conclusion

In this paper, we propose cost-effective design of multi-granularity optical cross-connect (MG-OXC) which significantly reduces the number of used ports and hence the cost of GMPLS network. The proposed selection method of MG-OXC is that the nodes with maximum of traffic flow are chosen. Performance analysis shows that the proposed design method of MG-OXC is more cost-effective than that of Random and Nodal-degree.

Acknowledgements

This research was supported by the Korean Science and Engineering Foundation (KOSEF) through Optical Internet Reserarch Center Project and by Institue of Information Technology Assessment (IITA) through University IT Research Center Project.

References

[1] E.Mannie, et al.: Generalized Multi-Protocol Label Switching Architecture. Internet draft, ¡draft-ietf-ccamp-gmpls-architecture-07.txt¿, work in progress,(2003)
[2] M. Lee, et al.: Design of Hierarchical Crossconnect WDM Networks Employing a Two-Stage Multiplexing Scheme of Waveband and Wavelength. IEEE Journal of Selected Areas in Communications, Vol. 20, No. 1 (2002) 166–171
[3] P. H. Ho, H. T. Mouftah.: Routing and Wavelength Assignment With Multigranularity Traf-fic in Optical Networks. IEEE Journal of Lightwave Technology, Vol. 20, No. 8 (2002) 1292-1303
[4] X. Cao, V.Anand, and C. Qiao : Waveband Switching in Optical Networks. IEEE Commu-nication Magazine, Vol. 41, No. 4 (2003) 105-112
[5] A. S. Arora, et al.:Converter Placement in Wavelength Routing Mesh Topologies. Proc. IEEE00, (2000) 1282-1288

Experiments with SCTP Multi-path Access for Single-Homed Hosts

Norihisa Matsumoto and Yuki Moritani

Network Laboratories, NTT DoCoMo, Inc.
3-5 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-8536 Japan
{n-matumo,moritani}@netlab.nttdocomo.co.jp

Abstract. The stream control transmission protocol (SCTP) is a transport protocol that is robust against network failure. However, it cannot select routes for packet rerouting unless the receiver is a multi-homed host. We propose a method that allows single-homed hosts to support SCTP multi-path access; it allows hosts in a local network to select routes to single-homed severs on the Internet. In this method, a single-homed client constitutes a virtual multi-homed client with a Multi-path gateway (MPGW) and communicates with a single-homed server over multiple routes. Our experiments show that a virtual multi-homed client can communicate with a normal SCTP server and that the method establishes connections that experience the same network failure to recover at different times, whereas IP dynamic routing recovers them at the same time.

1 Introduction

When two hosts want to communicate over a network, there may be several topological routes between them. For any application, using several or all of these routes at once can be an effective way of enhancing robustness against network failure. Moreover, careful route selection can yield close to the desired communication quality.

To select the routes used, the host needs one or more selection criteria. The quality of communication that one application desires may differ from that needed by another. Therefore, route selection should be performed per flow on the transport layer, like a TCP connection.

The stream control transmission protocol (SCTP) [1] is a transport protocol that permits route selection, but only when the partner is a multi-homed host. We propose a method that allows SCTP multi-path access to a single-homed host in order to enable hosts that have a single IP address to select routes by SCTP. Sect. 2 outlines SCTP route selection. Next, we introduce our proposal and experiments in Sect. 3 and Sect. 4. The proposal is compared to existing technology in Sect. 5. Our conclusions complete this paper.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 800–809, 2004.

2 SCTP Multi-homing

SCTP is a connection-oriented transport protocol that offers, like TCP, congestion control and retransmission control, and also the feature of multi-homing; a multi-homed host can uses his multiple IP addresses to establish a connection. If the peer is a multi-homed host, or, has multiple IP addresses, a host can indirectly set a route to transfer packets by selecting one of the peer's addresses as the destination address of the packet. The selection policy is described below. A "path" in SCTP terminology means an end-to-end route for communication between two hosts.

1. Original packets are sent on the so-called the primary path.
2. Retransmitted packets are sent on another path.
3. If N retransmission time-outs (RTO) occur in succession on the primary path, the host judges the primary path to be unreachable and thereafter sends all packets over another path. Protocol parameter Path.Max.Retrans corresponds to N.
4. Hosts periodically send a HEARTBEAT message on each path and confirm the receipt of HEARTBEAT-ACK message to check reachability. If the primary path becomes reachable, original packets are sent over it again.

SCTP offers a guarantee of reachability by means of selecting routes based on the transport layer condition; we note that this guarantee is for just the connection. SCTP can be expected to guarantee other characteristics as requested by applications through the addition of a sophisticated route selection policy. However, SCTP has a restriction in that a host can only use as many routes as there are IP addresses of the peer. There are many hosts on the Internet that have just a single IP address, and so when hosts communicate with them, only a single route can be used.

We have developed a method of SCTP multi-path access that enables single-homed hosts to use SCTP route selection.

3 Proposed Method

3.1 Assumption

We assume a local network that has multiple routes to the Internet (See Fig. 1). SOHO (Small Office/ Home Office), corporations, and ISPs (Internet Service Providers) often adopt this form to keep reliable connectivity to the Internet.

In Fig. 1, a local network is located in the global domain and connected to the Internet through a transit network TNW1. All packets destined for the local network from the Internet are routed to TNW1.

The local network also connects to network TNW2 through gateway G2, but routes for IP addresses in the local network are not informed to the Internet through TNW2. For example, we can sign up for an ADSL residential service provided by an ISP and get an only global IP address Agc assigned to G2. G2

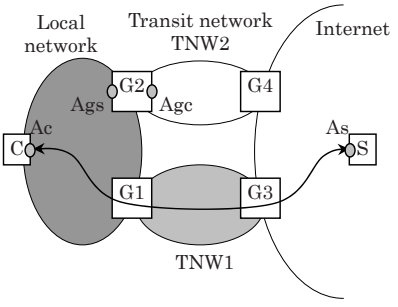


Fig. 1. A local network connected to the Internet by two routes

uses IP address Agc to receive packets routed over the Internet and IP address Ags for packets routed within the local network.

Client C in the local network has a single address and accesses an arbitrary server S on the Internet. S has a single address and a standard SCTP/IP suite.

3.2 Concept—Virtual Multi-homing

A host using SCTP can change the route only by selecting an alternative destination IP address, so SCTP is useless if the peer has a single IP address. Our solution that the host sends packets to the peer via a network node that is not located on the currently active route; the network node transfers the received packets to the peer. In Fig. 1, C and S send packets to the IP address of G2, and G2 transfers received packets to S and C, respectively. This allows C and S to communicate via the route through TNW2, which normally would not be recognized by the network as a route between C and S. We call this network node the multi-path gateway (MPGW).

To establish virtual multi-homing without any server modification, a client notifies his own address and his MPGW's address to a server as if he had two addresses. In Fig. 2, S recognizes Ac and Agc as addresses of C. C treats As and

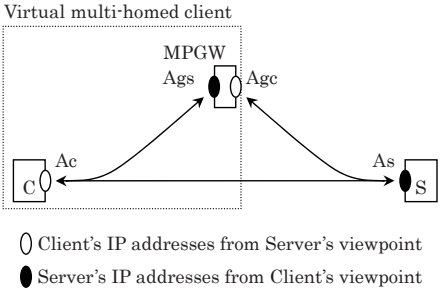


Fig. 2. A virtual multi-homed client has two IP addresses, Agc and Ac

Ags as the addresses of S. Consequently, the server can communicate with the client by normal SCTP multi-homing.

3.3 Signaling Flow

Fig. 3 shows an example of the signaling flow among client, MPGW and server. An "association" in SCTP terminology means a connection. We use the word below.

First, the client knows his MPGW's IP address, Ags, in advance, by means of, for example, manual configuration or address resolution based on DNS.

When a client wants to establish an association with a server, the client notifies the association information—client's IP address and SCTP port, and server's IP address and SCTP port—to the MPGW. The MPGW and the client negotiate to ensure that client's port Pc is unique among all clients using the same MPGW, so that the MPGW can distinguish packet associations. After the negotiation, the MPGW registers the association information in an internal table, and passes IP address Agc to the client to allow it to establish virtual multi-homing.

The client starts the procedure to establish an association. INIT message has a source transport address Ac:Pc and a destination transport address As:Ps, and includes Ac and Agc as client's IP addresses in the payload. The server receives INIT message and recognizes the client as being a multi-homed host. Although the server sends the INIT-ACK message whose payload includes only server's IP address As, the client treats Ags as the server's address, as well as As.

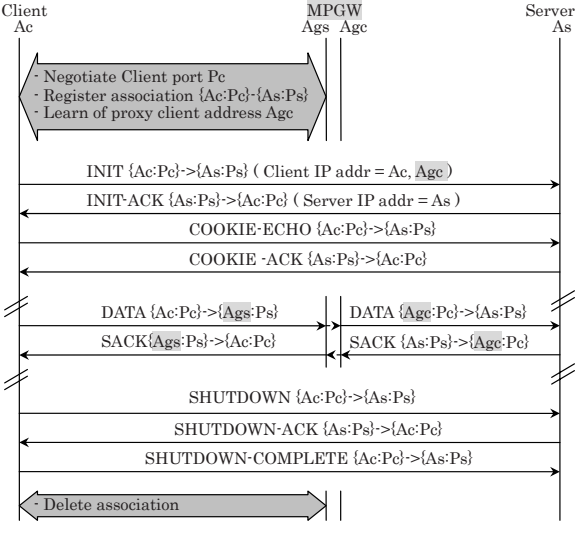


Fig. 3. Signaling flow

This procedure ends with COOKIE-ECHO and COOKIE-ACK messages sent between the client and the server.

When the client sends packets on the route through the MPGW, the packets' destination IP address is Ags. Upon receipt of packets, the MPGW rewrites the packets' source address and destination address with Agc and As, respectively, and sends the packets to the server. Similarly, when the server sends packets on the route through the MPGW, the server sends them to Agc and the MPGW transfers them to Ac.

To close the association, the client orders the MPGW to delete association information following the normal SCTP shutdown procedure.

We note that the procedure in Fig.3 is intended to keep the route via MPGW in the regular manner, so client port negotiation is done prior to establishment of SCTP association. As another consequence, establishing SCTP association prior to registering association information to the MPGW may allow the transfer of application data to start as early as in standard SCTP; however, the MPGW might refuse the use of any route via it because of inadequate client port uniqueness. The order shown suits applications that give more importance to short delay in starting data transfer than to reliability by using multiple routes.

4 Experiments and Results

We tested our method to confirm two points. One is that it can communicate with normal SCTP servers without any problem. The other is that the method enables transport connections through the same networks to select their own level of reachability guarantee, or more specifically, to select their own threshold which determines the timing of recovery from communication failure. SCTP possesses a basic advantage in that it can recover communication from network failure by selecting routes based on monitoring the condition of one transport connection independently of the others. It is important the virtual multi-homing technology retains this advantage.

We realized the signaling flow from INIT message to SHUTDOWN-COMPLETE message on an experimental network and evaluated our method.

4.1 Implementation of Client and MPGW

We used "SCTP reference implementation Ver.4.0.5" (the appendix to [3]) as the client and server software. We also added some functions to the client. Our client can manage the IP address Ags as a server's address and notify IP address Agc as a client's address to servers in the INIT message.

We implemented MPGW as a daemon on Linux. Association information is provided to MPGW as a text file. Below is an example of this text file and the rules for MPGW packet transfer.

```
###   association table (static)
###   client_port     client_address      server_address
assoc   1000             110.1.1.110         120.1.1.120
assoc   2000             110.1.1.111         120.1.1.121
```

Rule 1: If the destination address of received packet is Ags, the source port is a client port. Search the association table for the source port and identify the association. Rewrite the destination address and the source address of the packet with the server's IP address and Agc respectively, then send out.

Rule 2: If the destination address of received packet is Agc, the destination port is a client port. Search the association table for the destination port and identify the association. Rewrite the destination address and the source address of the packet with the client's IP address and Ags respectively, then send out.

4.2 Experimental Setup

Fig. 4 shows the experimental network. Clients, a server, Linux routers and a packet monitor were built on PCs. Network1 represented a local network while Network2 represented an external network.

OSPF ran among Linux routers in Network 1 and BGP-4 ran between Network 1 and 2. We set the parameters of the routing protocols to assign higher priority to the lower route than the upper route between Network1 and 2. If the lower route failed, the communication between Network1 and 2 was established on the upper route under the control of IP dynamic routing. Routing daemons were Gated.

Considering the real Internet, we added 50ms delay in Network2 by NIST Net[2]. We attached a packet monitor to the L2 link between a server and Linux router 5 and observed packets that the server sent out and received.

10Mbps hubs were placed between Network1 and 2, but other links were 100Mbps. In these experiments, we triggered network failure by disconnecting the Ether cable from the 10Mbps hub.

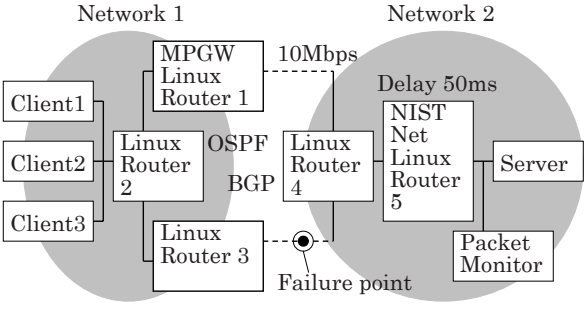


Fig. 4. Experimental Setup

We located the MPGW daemon on Linux router 1. To allow the proposed method to be compared to dynamic routing, we made an environment in which the lower route failed and communication was recovered on the upper route, regardless of which method was used.

In these experiments, the server sent a 8MB data set to the clients by FTP/TCP or a specific test application on SCTP. Receiver window size and sender window size in SCTP and TCP were both 32kB.

4.3 Results and Analysis

Impact of Network Failure on Data Transfer First, we compared our method to IP dynamic routing in terms of communication recovery. We observed data transfer to client1 by TCP, SCTP, and our method; the network failed 15sec after starting the transfer.

Fig. 5 shows the TCP sequence number of data sent by the server. Network failure occurred at $t_{failure}$, the route between client1 and the server was recovered by IP dynamic routing at t_{update}, and data transfer was resumed at t_{resume}. Regarding t_{update}, it reflects the later value among the time Gated on router 2 issued a system call to renew the routing table and the time Gated on router 4. That is, more than a few seconds was needed for the renewal to be seen as valid in the Linux kernel. TCP first backed off and started retransmission after $t_{failure}$; continuous transmission was recommenced after t_{update}. t_{update} depended on the time BGP-4 used to identify network failure. Below is the relation among KeepAlive timer $T_{KeepAlive}$, Hold Time timer $T_{HoldTime}$ and t_{update}, the first and second are parameters of BGP-4.

$$T_{HoldTime} - T_{KeepAlive} < t_{update} - t_{failure} \leq T_{HoldTime} \ . \tag{1}$$

We used the default values of Gated: 60sec for $T_{KeepAlive}$ and 180sec for $T_{HoldTime}$. Therefore,

$$134.9 < t_{update} \leq 194.9 \ . \tag{2}$$

Fig. 6 shows the case of SCTP. In SCTP, a sequence number is assigned to each data "chunk", which makes the Y axis differ from that of Fig. 5, although the same amount of data was transmitted. Like Fig. 5, data transmission was recovered by just IP dynamic routing because endhosts knew only the route indicated by the networks. SCTP backed off and started retransmission after $t_{failure}$; continuous transmission was resumed after t_{update}. The difference from TCP was that with SCTP, the interval between t_{resume} and the retransmission right before t_{resume} was shorter. That means that retransmission back off was not the trigger to resume continuous transmission. SCTP can detect the recovery of a route by HEARTBEAT and HEARTBEAT-ACK messages, which allowed continuous transmission to resume earlier than the trigger of the first retransmission after t_{update}.

Figure 7 shows the case of our method. At t_{resume}, the number of successive RTOs on the primary path, or the lower route, exceeded the value of Path.Max.Retrans. Therefore, continuous transmission resumed independently of t_{update}.

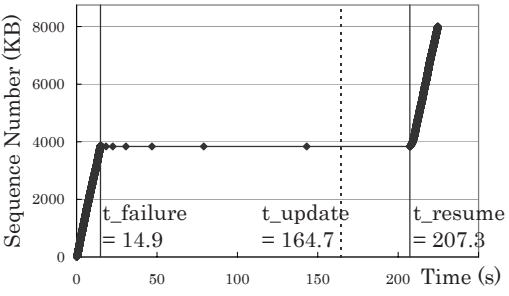


Fig. 5. TCP data transfer recovery from failure by dynamic routing

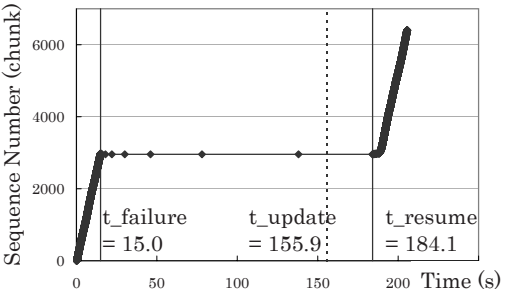


Fig. 6. SCTP data transfer recovery from failure by dynamic routing

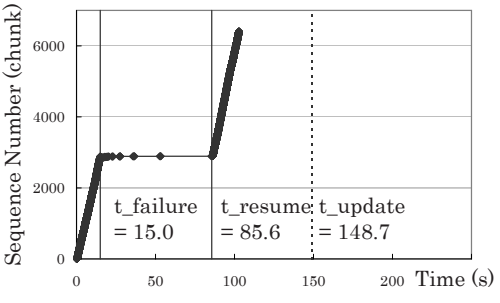


Fig. 7. SCTP data transfer recovery from failure by proposed method

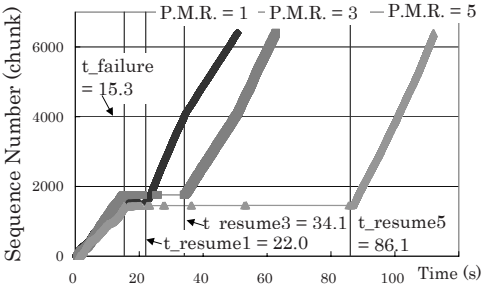


Fig. 8. Recovery from failure with different value of Path.Max.Retrans

Different Values of Path.Max.Retrans Next, we tested whether connections through the same networks could recover from the same network failure at different times. Concretely, clients 1, 2 and 3 established associations to the server and the server started to send the same 8MB data set to each client. 15sec after the commencing, we triggered a network failure. Values of Path.Max.Retrans of the associations for client1, 2 and 3 were 1, 3, and 5, respectively.

Fig. 8 shows the results. The default value of Path.Max.Retrans is 5, which was used in Fig. 7. The periods from $t_{failure}$ to $t_{resume1}$, $t_{resume3}$ and $t_{resume5}$ are approximately 7sec, 19sec, and 71sec respectively, which shows that our method can recover associations from a network failure at individual timing values.

On the experimental network, the lower and the upper route in each association didn't experience any RTO except for the periods in which the network failed. If both routes have sufficiently small RTO occurrence probabilities, Path.Max.Retrans should be set at the smaller value. If both routes have high RTO occurrence probabilities in the normal condition, Path.Max.Retrans should be set according to the maximum number of successive RTOs on the route with smallest packet loss.

5 Related Work

This section compares our proposal to other technologies that allow the endhost to select routes.

First, IP source routing can control the routing of a TCP connection. It yields larger packet delay than our proposal because every router always analyzes the IP header option, which controls source routing, even if the router is not intended to perform any source routing function. IP source routing is also used to establish malicious attacks against servers on the Internet. These attacks are often countered by denying source routing access. This means that IP source routing is impractical. Regarding implementation, although changing routes after the establishment of a TCP connection is required by [4] with "SHOULD" condition, we found in our experiments that Windows, Linux, and FreeBSD do not support this function well. Moreover, to detect route failure, TCP needs to know the route that each packet should follow, which requires the addition of some SCTP-like function to TCP.

Migrate TCP[5] can change a route by means of changing the endhost's IP address. This technology was proposed to enable a mobile host to hold a TCP connection even if the IP address of the host was changed due to his movement. A host communicating by TCP performs an address change procedure with TCP Migrate option when a new IP address is assigned by networks. This procedure is based on the 3-way handshake by SYN packets. This imposes excessive overheads if the change should be performed packet by packet. The endhost recognizes only one route at one time, and so cannot know the condition of the other routes. In contrast, SCTP monitors all routes by HEARTBEAT and HEARTBEAT-ACK messages and measures RTT (round-trip-time). SCTP can consider the RTT when selecting routes to recover the connection.

6 Conclusion and Future Work

This paper proposed the method of SCTP multi-path access for single-homed hosts, and showed the experiments conducted to confirm its performance. Our method allows that multi-homed network providers, including users who manage their home networks can provide hosts in their networks with robust communication to the Internet by the multi-path gateways placed only on their own boundaries. A single-homed SCTP client in the network notifies its MPGW's IP address as another client IP address to the server; this realizes a virtual multi-homed client. In experiments, a single-homed client communicated with a single-homed, normal SCTP server over multiple routes. We also verified that SCTP associations can recover from the same network failure at different times under the control of an SCTP protocol parameter. This means that our method retains the feature of SCTP, offering a proper guarantee level of reachability considering the transport layer condition of routes which belong to a specific set that each application employs.

We intend to clarify the best timing for changing routes considering the characteristics of each route. Another research goal is to set the guarantee level according to the QoS requests issued by the applications and users by means of implementing a more sophisticated route selecting policy.

Our method allows that hosts in the moving network with multiple gateways to fixed networks hold their data connections by seamless handover, and that mobile hosts communicating with each other by mobile SCTP[6] hold the connection however they change their IP addresses simultaneously. We will study our method considering various applications.

References

[1] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, IETF, October 2000. 800

[2] "NIST Net Home Page", http://www-x.antd.nist.gov/nistnet/, NIST. 805

[3] R. Stewart and Q. Xie, "Stream Control Transmission Protocol (SCTP): A Reference Guide", Addison Wesley, October 2001. 804

[4] R. Braden, "Requirements for Internet Hosts – Communication Layers", RFC1122, IETF, October 1989. 808

[5] A. Snoeren and H. Balakrishnan, "An End-to-End Approach to Host Mobility", 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00), August 2000. 808

[6] Seok Joo Koh, Mee Jeong Lee, Maximilian Riegel, Mary Li Ma and Michael Tuexen, "Architecture of Mobile SCTP for IP Mobility Support", draft-sjkoh-sctp-mobility-02.txt, IETF, June 2003. 809

A Route Optimization Mechanism Using an Extension Header in the IPv6 Multihoming Environment*

Ji-Young Huh[1], Eun-Young Park[1], Dong-Hun Lee[1], Jae-Hwoon Lee[1], and Yong-Jin Kim[2]

[1] Dept. of Information and Communication Engineering, Dongguk University
26, 3 Pil-dong, Chung-gu Seoul, 100-715 Korea
{jiyoung,eypark,windong,jaehwoon}@dongguk.edu
[2] Modacom, Jinsuk Building 5F
1536-7, Socho-Dong, Socho-Gu, Seoul, 137-073 Korea
Cap@Modacom.co.kr

Abstract. A multihomed enterprise or AS (Autonomous System) improves reliability and performance by acquiring its Internet connectivity from more than one ISP (Internet Service Provider). When connectivity through one of the ISPs fails, its connectivity to the Internet can be continued by using tunneling mechanism through Non-direct EBGP (Exterior Border Gateway Protocol). However, this mechanism causes the problem that makes the communication route non-optimal. In this paper, we propose a route optimization mechanism by using an extension header in the IPv6 mutihoming environment.

1 Introduction

As the Internet becomes more important, an enterprise or AS (Autonomous System) needs to improve reliability and performance of its Internet connectivity. Multihoming, which allows an AS to acquire its Internet connectivity from more than one ISP (Internet Service Provider), is suggested for this purpose. Maintaining connectivity via more than one ISP makes connectivity to the Internet more reliable by preserving the Internet connectivity via other ISPs when connectivity through one of the ISPs fails. In addition to providing more reliable connectivity, the multihoming mechanism improves performance by distributing load among multiple connections.

A multihomed site connected to a set of ISPs is allocated with several address prefixes by each of these ISPs, and each host within the multi-homed site is allocated with an IPv6 address made from one of several address prefixes. The site exit router connected to each ISP provides an Internet connectivity for the hosts that use its address prefix. Each site exit router and the ISP router directly connected to it transmit the routing information to each other using EBGP

* This work was supported by University IT Research Center project.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 810–816, 2004.

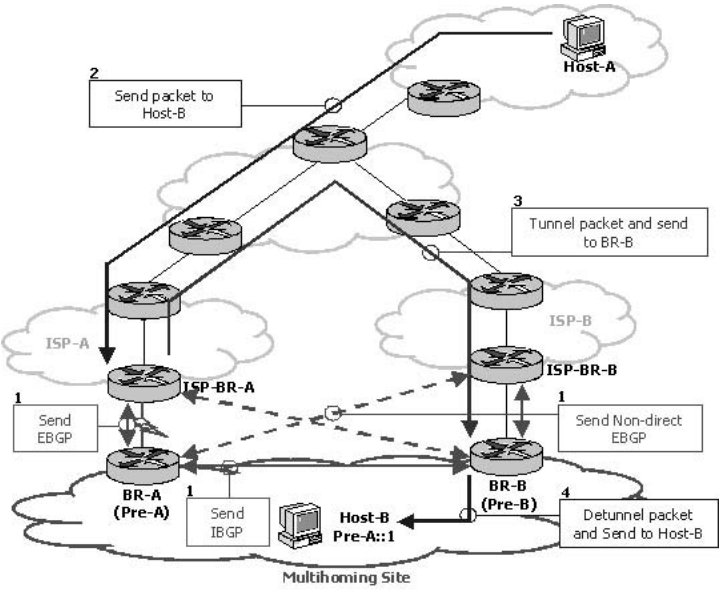


Fig. 1. Tunneling mechanism using Non-direct EBGP

(Exterior Border Gateway Protocol), and site exit routers transmit the routing information to each other using IBGP (Interior Border Gateway Protocol). Each site exit router and the ISP router non-directly connected to it also transmit the routing information to each other using Non-direct EBGP so that the Internet connectivity can be continued via the other ISP even if connectivity between the site exit router and the ISP router directly connected to it fails [1][2].

Figure 1 shows the tunneling mechanism using Non-direct EBGP. The Host-B within the multihomed site usually acquires its Internet connectivity from ISP-A because it uses the address prefix, Pre-A, that ISP-A has allocated to it. But if the connectivity between ISP-BR-A as a border router of ISP-A and BR-A as a border router of the multihomed site fails, ISP-A could not provide normal Internet service for Host-B any more. Therefore ISP-BR-A lets the other ISP provide the Internet connectivity for Host-B by using the tunneling mechanism through Non-direct EBGP. That is, when ISP-BR-A receives a packet destined to Host-B from Host-A, it tunnels the packet to BR-B. The tunneled packet is transmitted to BR-B via ISP-BR-B. And BR-B decapsulates and transmits the packet to Host-B. As above, when one of ISPs fails, IPv6 multihoming could improve reliability by using the tunneling mechanism through Non-direct EBGP, but cause the problem that makes routes non-optimal.

SERDB (Site Exit Router Database) mechanism is suggested to solve the problem related to IPv6 multihoming. In this mechanism, SERDB stores the list of multiple site exit router addresses and unreachable site exit router addresses of each multihomed site. If the ISP border router receives a packet destined to

the multihomed site when connectivity between one of ISPs and the multihomed site fails, it discards the packet and sends an ICMPv6 destination unreachable message toward the source node including failed network address of the multihomed site. Then, the border router at the source's site intercepts the ICMPv6 message and queries the address of a reachable site exit border router of the multihomed site to the SERDB. The SERDB replies with the reachable site exit router address. After this, if the border router of the source site receives a packet destined to multihomed site, it transmits the packet to the multihomed site using the routing extension header to cause the packet to be routed via the reachable site exit router [3]. This mechanism can solve the non-optimal route problem caused by the tunneling mechanism using Non-direct EBGP, but also has the following problems. The SERDB mechanism must maintain SERDB that stores the information about all multihomed sites and have the border router at source's site intercept the ICMP message. Besides, the packet sent to the unreachable site exit router of the multihomed site can be discarded before the border router at the source site receives ICMPv6 message.

In this paper, we propose the mechanism that can provide route optimization without packet losses using an extension header as well as maintain the Internet connectivity in IPv6 multihoming, even though connectivity between one of ISPs and a multihomed site fails.

The rest of this paper is organized as follows. Section 2 describes the operation of the route optimization mechanism using an IPv6 extension header. Section 3 evaluates the performance of the proposed mechanism through simulation. Finally Section 4 concludes this paper.

2 The Route Optimization Mechanism Using an IPv6 Extension Header

In the proposed mechanism, a new Hop-by-Hop Option named Multihomed Binding Option is defined. When connectivity between one of ISPs and multihomed site fails, the ISP transmits packet including Multihoming Binding Option toward the source. The packet allows intermediate routers between a source and a destination to know the loss of connectivity between one of ISPs and the multihomed site. After that, if intermediate routers receive packets destined to a multihomed site via an unreachable site exit router, they tunnel the packets directly to a reachable site exit router.

Figure 2 shows the operation of the proposed route optimization mechanism. The multihoming site is connected to the Internet through two ISPs, ISP-A and ISP-B. And Host-B within the multihoming site is usually connected to the Internet through ISP-A because it uses the address prefix, Pre-A, that ISP-A has allocated to it.

Suppose that Host-A transmits a packet to Host-B when the connectivity between ISP-BR-A and BR-A has failed. At first, the packet is transmitted to ISP-BR-A. Then ISP-BR-A tunnels it to BR-B using Non-direct EBGP. At the same time, it transmits a packet including Multihoming Binding Option toward

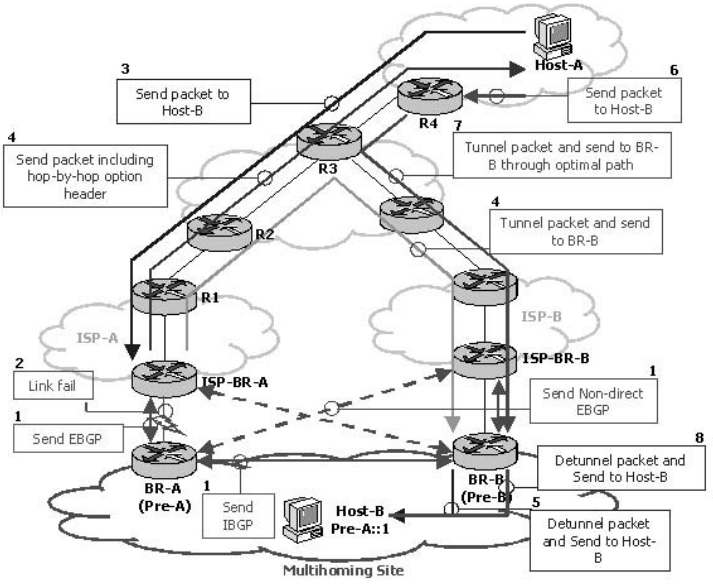


Fig. 2. The operation of the route optimization mechanism using IPv6 extension header

the source node, Host-A. The packet is examined by all intermediate routers between Host-A and Host-B because Multihomed Binding Option is a Hop-by-Hop Option. When R1, the first intermediate router, receives the packet, it decides whether it can recognize the option by checking the Option Type field. If R1 cannot recognize Multihomed Binding Option, it ignores the option and transmits the packet toward Host-A. If R1 can recognize the option, it stores the information of Multihomed Binding Option and transmits the packet toward Host-A. After that, R1 establishes the tunnel with BR-B and transmits packets toward the multihomed site to BR-B by using the tunnel. R2 and R3 also perform the same process as R1 does. As each intermediate router tunnels packets to BR-B directly, the route that the packets go through becomes more optimal. When the border router of source site, R4, receives the packet including Multihomed Binding Option and stores the information, the router establishes a tunnel with BR-B that makes the route most optimal.

Each intermediate router sends the Binding Request message to ISP-BR-A to check whether the connectivity is still in failure before the lifetime of the binding information expires. If the connectivity between ISP-BR-A and BR-A is up, ISP-BR-A ignores the Binding Request message. Then, the intermediate router that doesn't receive the Bind Reply message deletes the binding information for Pre-A as soon as Lifetime expires. On the contrary, if the connectivity is still in failure, ISP-BR-A transmits Binding Reply message to the source router of Binding Request message. The intermediate router that receives the Bind Reply message updates Lifetime for the binding information.

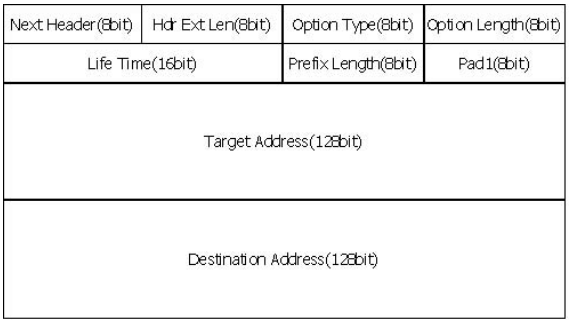


Fig. 3. Multihomed Binding Option format

As shown in Figure 3, four fields are defined in the proposed IPv6 Multihomed Binding Option. The lifetime field indicates the expiration time of the binding information. The Destination Address and the Prefix Length fields indicate the address prefix allocated by the ISP whose connectivity has failed with site exit router. Lastly, the Target Address field indicates the destination address of the packet tunneled by each intermediate router.

Each intermediate router having Multihomed Binding Option for the unreachable site exit router decides whether received packets are toward the unreachable site exit router using the Destination Address and the Prefix Length fields. And then it decides where to tunnel those packets using the Target Address field.

3 Performance Evaluation

In this section, the performance of the proposed mechanism is evaluated through simulation using the Network Simulator $ns-2$ [3]. The network topology and the parameters of the simulation are shown in figure 4. In this simulation model, Host-A transmits traffic to Host-B via ISP-A. When the connectivity between BR-A and ISP-A fails, ISP-A tunnels received packets to BR-B and transmits the packet including Multihomed Binding Option toward Host-A. After SR1 and SR2 receive the binding information, they tunnel the packets toward BR-A to BR-B directly.

Figure 5 shows the transmission delay of the tunneling mechanism through Non-direct EBGP (ND-EBGP) and the proposed Route Optimization Mechanism using Multihomed Binding Option. For this case, two failures have been generated. During the failure period, the transmission delay of the ND-EBGP mechanism is kept very high because the packets transmitted by Host-A continue to be tunneled to BR-B by ISP-A. On the other hand, the proposed mechanism incurs the same transmission delay as the ND-EBGP mechanism when the first packet is tunneled by ISP-A. However, the transmission delay decreases as packets are directly tunneled by SR1 after Multihomed Binding Option is transmitted to SR1. Finally the transmission delay becomes very low as packets are

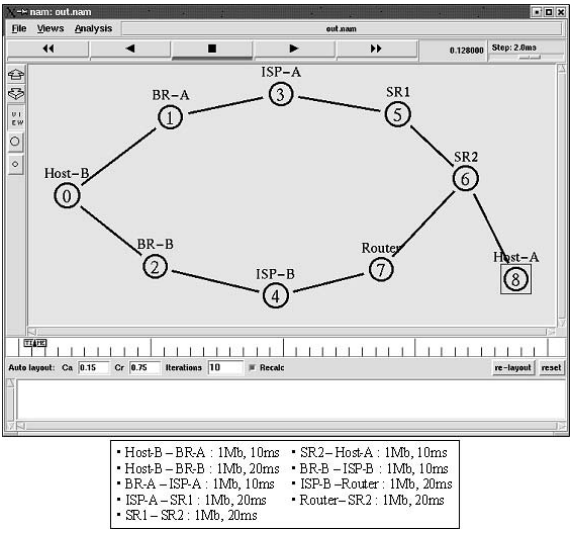


Fig. 4. Simulation Network Topology

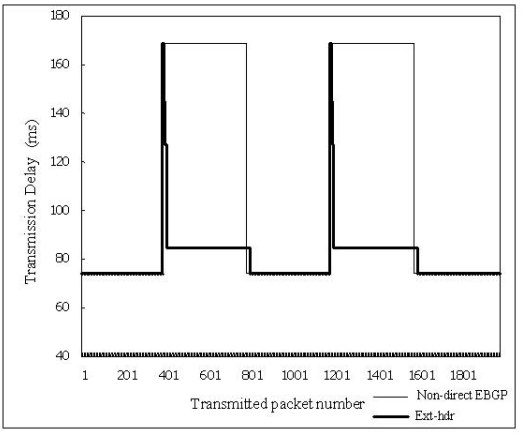


Fig. 5. The comparison of transmission delay

transmitted through the optimal route after SR2 receives Multihomed Binding Option.

Figure 6 shows the average transmission delay for the number of failures in the ND-EBGP mechanism and the proposed mechanism. The failure recovery time is exponentially distributed. As shown in Figure 6, the average transmission delay of the ND-EBGP mechanism increases rapidly as the number of failures increases. On the other hand, we can observe that the average transmission delay of the proposed mechanism increases slowly as the number of failures increases.

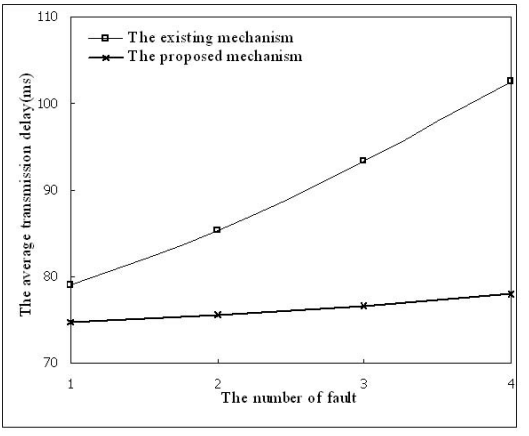


Fig. 6. The average transmission delay for the number of failures

4 Conclusions

The multihoming mechanism allows the multihomed site to improve reliability and performance of its Internet connectivity by maintaining connectivity from more than one ISP. However, the mechanism causes the problem that makes the transmission route non-optimal when connectivity between one of ISPs and the multihomed site is failed.

In this paper, we proposed the mechanism that optimizes the transmission route using Multihomed Binding Option. The performance is analyzed and compared for the ND-EBGP mechanism and the proposed mechanism by using the $ns-2$. From the simulation results, we can observe that the proposed mechanism improves the average transmission delay performance.

References

[1] T. Bates and Y Rekhter, "Scalable support for multihomed multi-provider connection", IETF RFC 2260, Jan. 1998.
[2] J. Hagino and H. Synder, "IPv6 multihoming support at site exit routers", IETF RFC 3178, Oct. 2001.
[3] K. I. Kim, C. M. Park, T. I. Kim and S. H. Kim, "Novel scheme for efficient and scalable multihoming support in IPv6", *Proc. IEEE ICCS*, pp.656-660, 1998.
[4] LBL, Xerox PARC, UCB, USC/ISI, VINIT Project, The Network Simulator ns-2, www.isi.edu/nsnam/ns.

A Novel TCP-Friendly Congestion Control with Virtual Reno and Slack Term

Yuan-Cheng Lai and I-Fang Chen

Department of Information Management
National Taiwan University of Science and Technology
laiyc@cs.ntust.edu.tw
chenifang@giga.net.tw

Abstract. In this paper, a novel TCP-friendly congestion control with Virtual Reno and Slack Term, VRST for short, was proposed. The design objective of VRST is to maintain a smoother sending rate and to be friendly to TCP. VRST maintains a virtual Reno to mimic the TCP Reno behavior. To avoid sawtooth-like rate variation in TCP Reno, a limit on window adaptation is imposed on VRST to prevent drastically reducing or increasing congestion window. Owing to the discrepancy in window adaptation between VRST and virtual Reno, the window compensation is provided. This mechanism first keep track of the difference in window between VRST and virtual Reno, then this difference should be compensated or paid back at some future time, so the aim of TCP-friendliness can be achieved. From the simulation results, in most situations, VRST dose get the almost same throughput with TCP and have a smaller variation in throughput.

1 Introduction

Even with the rapid growth of Internet users and applications, the Internet to date still works stably without network collapse. This stability has been primarily due to the congestion control mechanism of TCP, which is still the most popular protocol used in the Internet. Nevertheless, over recent years, the dramatically increasing popularity of the real time multimedia applications, such as IP-telephone, video-on-demand (VOD), RealAudio, built on top of UDP are growing in bandwidth usage on the Internet. There are three primary reasons why they don't choose TCP as their transport layer: 1) Owing to the design principle in TCP that favors reliability over timeliness, it is likely to make no sense if TCP sends obsolete data that would no longer be useful to the receiving application. 2) TCP's congestion control will reduce the sending rate by half in response to a single packet drop [1]. This seems to be unnecessarily severe to most multimedia applications, as it will suddenly reduce the user-perceived quality [2]. 3) A user using TCP, which employs congestion control, usually gets less bandwidth than a user using UDP.

Since UDP does not employ any congestion control mechanism like TCP, which will regulate the amount of data in the network by maintaining a window

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 817–826, 2004.

in the sender side. Thus, the increase of deploying these UDP traffic not only brings a severe congestion situation in the Internet but also results in the unfair treatment towards TCP flows. The extreme unfairness may ultimately cause the starvation for TCP traffic [3] and even lead to "congestion collapse", which is a situation where, although the network links are being heavily utilized, very little useful work is being done [4].

In order to address these problems mentioned above, a new term named "TCP-friendly" protocol has been proposed. TCP-friendly protocol means that the throughput of a non-TCP flow should be roughly the same as that of a TCP flow under conditions with the same round-trip time and packet loss [4]. Furthermore, in order to cater to the features of multimedia applications, the TCP-friendly flow commonly has two requirements. One is its sending rate or the variation of congestion window should have fewer oscillations than that of TCP. The other is it remains to make its throughput as close to TCP as possible even under premise that its long-term throughput should be less than or equal to TCP.

In this paper, we proposed a novel TCP-friendly congestion control, which uses the techniques, namely, Virtual Reno and Slack Term, VRST for short.

2 Related Works

Transmission Control Protocol (TCP) is the most popular transport protocol in the Internet. It provides a connection oriented, reliable, byte stream service and, above all, congestion control mechanism, which is regarded as the single dominant reason for the stability of the Internet [5]. Since the first specification of TCP appeared in RFC761, which was finally standardized in RFC793, its version has over time been improved in several ways. The successive versions of TCP contain Tahoe, Reno, SACK, New Reno, and latest D-SACK. We briefly introduce TCP Reno congestion control mechanism because the implementation of our proposed algorithm, VRST, modified directly from TCP Reno.

2.1 TCP Reno

TCP uses a sliding-window protocol for end-to-end congestion control. Each TCP sender maintains a congestion window, *cwnd*, to control the maximum amount of data in a round trip time (RTT) that it can send without overloading the network. Three significant components of TCP involve slow start, retransmission timeout, and AIMD (Additive Increase Multiplicative Decrease), which is the basis of TCP congestion control. The spirit of AIMD is that the sender raises its offering load by a constant if no congestion happens; or the sender reduces its offering load to a fraction of the current load in order to alleviate the congestion situation. The working scheme of TCP Reno includes five phases, where three phases of congestion avoidance, fast transmission, and fast recovery, together construct the spirit of AIMD.

2.2 TCP-Friendly Protocol

To reach TCP-friendliness, it perhaps needs to know the TCP throughput in a certain network situation. Padhye et al. [6] derived the equation for TCP throughput in terms of packet loss rate and round trip time. His analysis considered fast retransmit, fast recovery, and retransmission timeout. The equation is shown as follows,

$$T(S, p, RTT, t_{RTO}) = \frac{S}{RTT\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1 + 32p^2)} \tag{1}$$

where S is the packet size, p is the packet loss rate, RTT is the round trip time and t_{RTO} is the TCP retransmission timeout value.

If ignoring the effect of retransmission timeout, Equation (1) can be simplified in the following,

$$T(S, p, RTT) = \frac{1.5\sqrt{\frac{2}{3}}S}{RTT\sqrt{p}} \tag{2}$$

Equation (2) provides a simple but essential criteria for validating whether the new invented protocol is TCP-friendliness or not, namely $T \propto \frac{1}{\sqrt{p}}$.

In recent years, several TCP-friendly protocols have been proposed. Some directly apply to equation (1), and others are based on the $T \propto \frac{1}{\sqrt{p}}$ relation. These proposed TCP-friendly protocols are further categorized into two classes: window-based and rate-based algorithms. The former includes GAIMD [7] and IIAD [8] while the latter includes TEAR [9] and TFRC [10].

3 Virtual Reno and Slack Term

The objective of VRST is to provide the smoother transmission rate, and in the meanwhile, to pursue TCP-friendliness property. The former is achieved by limiting the degree of the window adjustment, while the later is achieved by incorporating a compensation mechanism. Besides, this mechanism requires VRST to maintain a virtual Reno to mimic the Reno behavior. The architecture of VRST is illustrated in Figure 1, and in the following subsections, we will dwell on each component of VRST.

3.1 Slack Term

VRST limits its degree of the window adjustment to avoid reducing or increasing congestion window drastically. Owing to the discrepancy in window adaptation rule, VRST will eventually lead to different number of packets being sent compared to virtual Reno. Hence, VRST records this difference in window between VRST and virtual Reno, and this difference will be compensated by increasing the window or will be paid back by decreasing the window at future time.

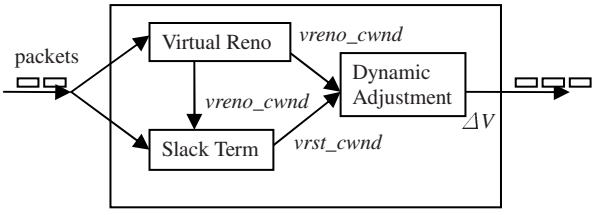


Fig. 1. Architecture of VRST

VRST uses variable C to record this difference and ΔV as maximum changeable window each time, and variable *vrst_cwnd* and *vreno_cwnd* as the window size of VRST and virtual Reno respectively. For instance, assuming that both *vrst_cwnd* and *vreno_cwnd* are equal to w, and after one RTT, *vreno_cwnd* is equal to w+1, while *vrst_cwnd* is equal to $w + \Delta V$ $(0 < \Delta V < 1)$. Reno will send additional one packet than VRST, so we add this one packet into C to reflect this difference. Obviously, VRST gets less bandwidth than Reno if C is positive; accordingly, VRST should progressively increase its window as compensation until having found the number of delivered packets is the same as or similar to that of Reno. In contrast, if C is negative, VRST then gradually decreases its window to avoid getting more throughput than Reno. Slack Term algorithm is exhibited as follows.

Algorithm 1: Slack Term

1. [*This step initializes C to 0, vrst_cwnd to 1, and* ΔV *to Vw*]
$C \leftarrow 0$, *vrst_cwnd* $\leftarrow 1$, $\Delta V \leftarrow Vw$

2. [*This step adjusts VRST's window per RTT*]

2.1 [*VRST calculates window difference, and adds it into C*]
$C \leftarrow C +$((int (*vreno_cwnd*)–int (*vrst_cwnd*))

2.2 [*VRST determines to increase or decreaseits window size according to C*]
if $C > 0$ then *vrst_cwnd* $\leftarrow$ *vrst_cwnd* $+\Delta V$
$C \leftarrow C–\Delta V$
else if $C < 0$ then
vrst_cwnd $\leftarrow$ *vrst_cwnd*$–\Delta V$
$C \leftarrow C +\Delta V$

The choice of the Vw mentioned above, in fact, involves the tradeoffs between smoothing the transmission rate and quickly reacting to the network status.

3.2 Virtual Reno

The virtual Reno agent also plays an essential role in VRST because it provides the estimation of TCP Reno's window. Clearly, if the window of virtual Reno is determined incorrectly, the compensation mechanism responsible to fairness requirement could fail. An example is introduced to explain why we need to maintain a virtual Reno. Assume that the congestion window of VRST and

virtual Reno are $2w+1$ and w respectively. After one RTT, VRST's window is $(2w+1) \pm \Delta V$ (the sign of positive or negative according to C) and the window of virtual Reno is supposed to be $w+1$. The window of virtual Reno, however, is found $w+2$ instead of $w+1$. That is because first w ACKs lead the window of virtual Reno to become $w+1$, where each incoming ACK contributes 1/cwnd. Next, sequent $w+1$ ACKs lead the window of virtual Reno to become $w+2$. Clearly, this is different from the increase part of AIMD that the congestion window adds only by one after one RTT.

A straightforward way to overcome this situation is to synchronize the time of window adjustment of virtual Reno and VRST, because the interval between two window adjustments in VRST is about RTT. However, this way may not mimic TCP well, because the basis of the window increase in real TCP is according to each receiving ACK rather than according to the expiration of a RTT. We, therefore, adopt an alternative way, which is based on each receiving ACK. Two variables, *vreno_cwnd* and *vreno_ssthresh*, are created here, whose functionalities are identical to the variables *cwnd* and *ssthresh* used in TCP Reno, respectively. With such two variables, the window increase in virtual Reno totally depends on the current phase. For example, in case virtual Reno is in congestion avoidance, *vreno_cwnd* should increase by $1/vrst\_cwnd$ so that *vreno_cwnd* will increase by one in each RTT, and in case virtual Reno is in slow start *vreno_cwnd* should increase by *vreno_cwnd / vrst_cwnd* so that *vreno_cwnd* will double in each RTT. It is sufficiently clear that what we do is just to make virtual Reno behave more like real TCP Reno.

As for the decrease part of window adjustment in virtual Reno, the procedure is also identical to TCP Reno. Therefore, we should determine which of following congestion events virtual Reno belongs to: retransmission timeout, three duplicate ACKs, or ECN echo, and, in turn, virtual Reno will do appropriate actions in response to the congestion event it belongs to. The algorithm 2 states the window increase in virtual Reno, while algorithm 3 stated the window decrease in virtual Reno.

3.3 Dynamic Adjustment

After introducing the basic concept of Slack Term in algorithm 1, we describe an advanced version of Slack Term in this section, in which ΔV is a variable rather than a constant. Recall that in algorithm 1 VRST makes use of C to guarantee the TCP-friendliness property in the long-term situation; however, VRST does not consider short-term situation, so VRST may not work well when the network environment changes frequently. As a result, at any time the window adjustment in VRST should depend more than just on C solely. The adjustment should also take the difference in window between itself and maintained virtual Reno into consideration, so that VRST can thus gain the abruptly increased bandwidth and can react to the abruptly decreased band-width in the short-term situation.

Algorithm 2: Mimic the increase of congestion window in TCP Reno

1. [*This step determines what state Virtual Reno is currently in, slow start or congestion avoidance*]
if *vreno_cwnd* < *vreno_ssthresh* then /*slow start*/
vreno_cwnd ← *vreno_cwnd* + *vreno_cwnd* / *vrst_cwnd*
else /*congestion avoidance*/
vreno_cwnd ← *vreno_cwnd* + 1 / *vrst_cwnd*
2. [*The flow control mechanism in TCP is also considered*]
if *vreno_cwnd* > receiver advertised window then
vreno_cwnd ← receiver advertised window.

Algorithm 3: Mimic the decrease of congestion window in TCP Reno

1. [*This step determines that a congestion indication comes from which of the following: retransmission timeout, three duplicate ACKs, or ECN echo*]
if the congestion indication comes from timeout then
vreno_ssthresh ← *vreno_cwnd* / 2.
vreno_cwnd ← 1.
else, it must comes from three duplicate ACKs, or ECN echo
vreno_cwnd ← *vreno_cwnd* / 2.
If *vreno_cwnd* < 1 then
vreno_cwnd ← 1.
vreno_ssthresh ← *vreno_cwnd*.
2. if *vreno_ssthresh* < 2
vreno_ssthresh ← 2

Accordingly, each time either the window's increasing or the window's decreasing in VRST acts according to the following rule; that is,$\Delta V = (C\ /\lambda + \mathit{diff})\ *\ Vw$, where *diff* is the value of *vreno_cwnd* – *vrst_cwnd*,λ is a *amortized factor,* which can be thought of as making C zero in a matter of λ times of RTT, and the rest of the notation are identical to the previous definition. The following pseudo code is used in place of step 2.2 in algorithm 1.

$\Delta V \leftarrow (\ C\ /\lambda + \mathit{reno\_cwnd}—\mathit{vrst\_cwnd}\ )*\ Vw$
$\mathit{vrst\_cwnd} \leftarrow \mathit{vrst\_cwnd} + \Delta V$
$C \leftarrow C – \Delta V$

By the above rule,once C is too far from the zero, which means that the total number of packets being sent is significantly different from that of virtual Reno, VRST can hence expand ΔV to rapidly reduce this difference. In addition, the *diff* is used with the intention of keeping as closer to the reaction of TCP as possible, especially in the short-term network situation.

4 Simulation Results

Several simulations were conducted by means of well-known ns-2 simulator to investigate the performance of VRST. The performance compared with some other TCP-friendly schemes is also investigated. In the most researches related

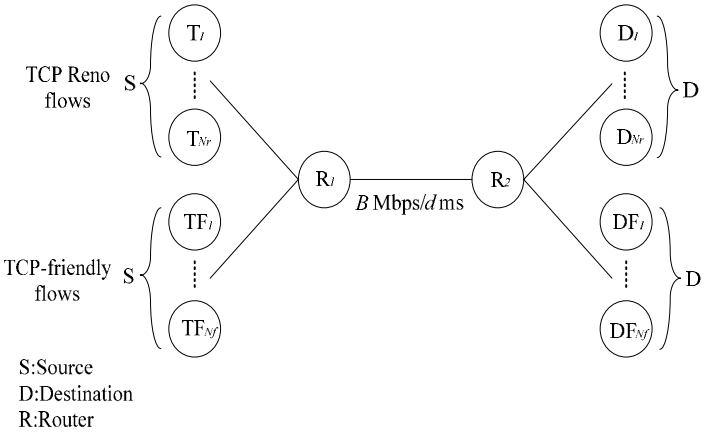


Fig. 2. Network topology

to TCP-friendliness, the commonly used performance metrics are friendliness and smoothness, and both will be discussed in the following subsections.

4.1 Simulation Environment

Our network topology is a single bottleneck link ("dumbbell") with RED (Random Early Discard) router as shown in Figure 2. Let N_r denote the total number of TCP Reno connections and N_f denote the total number of TCP-friendly connections. The bottleneck link has a bandwidth of B Mbps and a propagation delay of d ms. The bottleneck buffer size is set to 2.5 times the bandwidth-delay product, and the parameters of *min_thresh* and *max_thresh* used in RED are set to 0.25 and 1.25 times the bandwidth-delay product, respectively. All senders always have infinite data to send, which are de facto modeled as FTP sessions in ns-2 simulator. Simulation parameters, including network topology and RED queue management, are tabularized in Table 1.

4.2 TCP-Friendliness

In this section, we intend to verify the ability of non-TCP flows to gain similar bandwidth with TCP flows. The *fairness ratio*, F, is defined as the ratio of the mean throughput obtained by TCP-friendly flows ($\overline{T_F}$) to the mean throughput obtained by TCP flows ($\overline{T_R}$); that is,$F = \frac{\overline{T_F}}{\overline{T_R}}$.

As you see from Figure 3, in a wide range of situations, the fairness ratio of VRST approaches one, which means VRST gains the nearly same bandwidth as competing TCP flows. The remaining protocols, including GAIMD, TFRC, SQRT, and IIAD, all have a smaller fair ratio than VRST and GAIMD (0.2).

Table 1. Parameters and default values in network configuration

Parameter	Value
Packet size	1,000 byte
ACK size	40 bytes
B/W of bottleneck link (B)	10 Mbps
Propagation of bottleneck link (d)	20 ms
B/W of side links	100 Mbps
Propagation of side links	2 ms
Receiver max window size	1,000
TCP timer granularity	100 ms
RED buffer size	$2.5 \times \quad B \times \quad d$
RED parameters	min_thresh:$0.25 \times B \times d$ max_thresh:$1.25 \times B \times d$

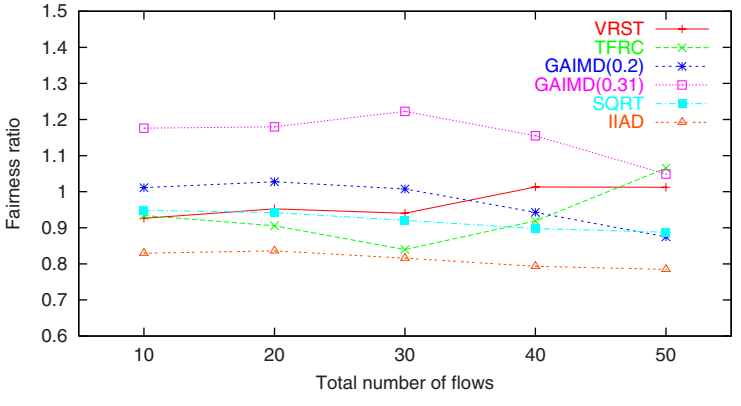


Fig. 3. Fairness ratio

4.3 Smoothness

We again emphasize that the TCP-friendly protocol is used for multimedia applications, so it is necessary to have throughput as smooth as possible. In this subsection, we would like to show a characteristic of smoothness. Before the performance metric of smoothness is formally defined, we first start ten flows, five VRST and five TCP flows, at the same time to show the throughput trace. For clarity of presentation, we present the result with two randomly selected flows for each kind of flow. By observing from Figure 4, the throughput trace of TCP varied frequently and greatly; in contrast, each of VRST flows exhibits relatively smooth behavior with a tight band of throughput variation. Now we turn our attention to define the performance metric of smoothness formally. The coefficient of variance (CoV) of throughput is usually used as an index to quantify the level of rate fluctuation, where CoV is defined as follows. Note that a flow

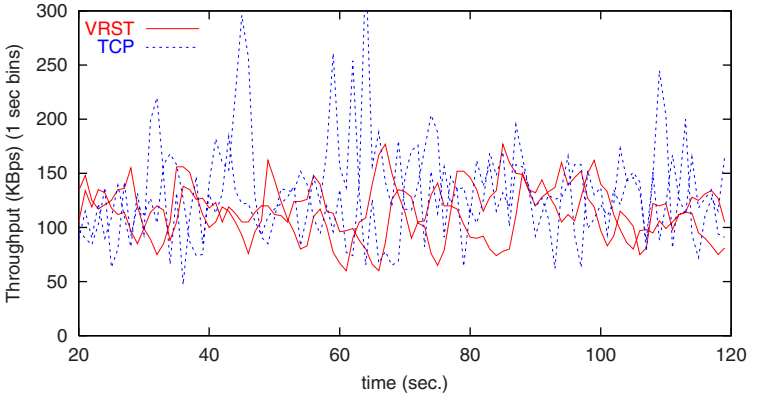


Fig. 4. Throughput traces of 5 VRST flows and 5 TCP flows

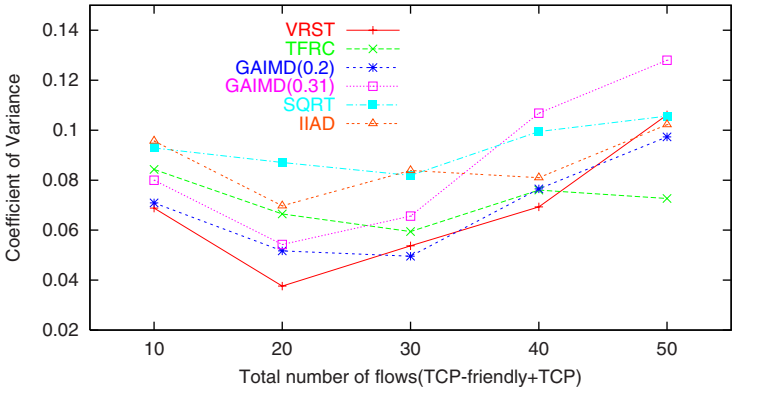


Fig. 5. Smoothness index (CoV of throughput)

with small CoV is smoother than a flow with large CoV.

$$CoV = \frac{s}{\overline{x}},$$

where s is the standard deviation and $\overline{x}$ is the mean. In Figure 5, VRST has a relatively small CoV than other protocols in most cases except the case of TFRC when total number of flows reaches 50. GAIMD (0.2) is smoother than GAIMD (0.31) in most cases. IIAD and SQRT have the almost same CoV.

5 Conclusions

In this paper, we proposed a new TCP-friendly transport protocol, VRST. In our experiments we have shown that VRST is TCP-friendly and with a smaller variation in the sending rate under various network conditions.

References

[1] V. Jacobson, "Congestion avoidance and control," In Proceedings of ACM SIGCOMM '88, August 1988.
[2] W.-T. Tan and A. Zakhor, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," IEEE Trans. on Multimedia, 1, June 1999.
[3] S. Floyd and K. Fall, "Promoting the Use of End-to-end Congestion Control in the Internet," IEEE/ACM Trans. Net., vol. 7, no. 4, Aug. 1999, pp. 458–72.
[4] S. Floyd, "Congestion Control Principles," RFC 2914, Sept. 2000.
[5] B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet," RFC 2309, Apr. 1998.
[6] J. Padhye et al., "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," IEEE/ACM Trans. Net., vol. 8, no. 2, Apr. 2000, pp.133–45.
[7] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control," Proceedings of ICNP 2000, Osaka, Japan, Nov. 2000.
[8] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms," Proceedings of IEEE INFOCOM, Apr. 2001.
[9] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP emulation at receivers - flow control for Multimedia streaming," Tech. report, Dept. of Comp. Sci., NCSU, Apr. 2000.
[10] S. Floyd et al., "Equation-based Congestion Control for Unicast Applications," Proceedings of ACM SIGCOMM, Aug. 2000.

Offset-Time Based Scheduling Algorithm for Burst Control Packet in Optical Burst Switching Networks

Jaegwan Kim, Jinseek Choi, and Minho Kang

Optical Internet Research Center, Information and Communications University
58-4, Hwaam-Dong, Yuseong-gu,Daejeon, 305-732, Korea
{tecmania,jin,mhkang}@icu.ac.kr

Abstract. This paper proposes a novel-scheduling algorithm, which considers offset time for burst control packet in optical burst switching networks. In the proposed algorithm, a burst control packet (BCP) with short residual offset time is served by the scheduler prior to a BCP with long residual offset time. Our proposed scheme can decrease the data loss due to early arrival problem of the data burst in optical burst switching networks. Finally, we evaluate the performance improvement through the simulation.

1 Introduction

The Internet has been the fastest-growing field in the world. This rapid increase of the Internet traffic is driving the demands for high transmission bandwidth and multimedia services [1]. Wavelength Division Multiplexing (WDM) has matured sufficiently to satisfy the demands. Moreover, IP router has the ability to directly access the WDM networks by eliminating the protocol overhead between the IP and the WDM networks. Although a lot of optical packet switching mechanisms have been studied for delivering IP packets for a long time, the difficulty in all-optical synchronization and optical buffering of pure optical packet switching technologies is still in the research stage. Recently, optical burst switching (OBS) was proposed as a new switching paradigm for optical networks requiring more relaxed technology than optical packet switching [2]. It is still being improved as a novel switching technique for the optical layer in optical burst-switched WDM networks.

In an OBS network, packets are assembled into bursts at an ingress router, and then routed via the OBS network and disassembled back into packets at an egress router. An original feature of the OBS is the physical separation of the optical data transport and the electronic control of the switch about data burst, which can facilitate the electronic processing of BCPs at OBS core nodes and provide end-to-end transparent optical paths for transporting the data burst [3]. A data burst may enter into the optical switching. fabric before its control packet has been fully processed due to excessive processing delay of the BCPs. This event is called early arrival. Early arrival problem can result in data burst

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 827–834, 2004.

loss in OBS node. It happens frequently that data burst is dropped as it is closed to the egress. Early arrival problem can be solved by using fixed-length inlet fiber delay lines (FDLs). However, this method occurs the additional delay of data burst owing to FDLs.

In this paper, we present the control plane issue to solve the early arrival problem in optical burst switching WDM networks. In section II, we investigate an issue in terms of early arrival due to excessive BCP delay in the control plane. And, the early arrival creates serious problems such as data loss. In section III, we propose an advanced scheduling algorithm named as Scheduling Considering Offset-Time (SCOT), the focus of this paper for BCP in optical burst switching WDM networks. The proposed algorithm can solve the early arrival problem at the OBS control plane instead of the data plane, where a BCP with short residual offset time is served by the scheduler earlier than a BCP with long residual offset time under a burst environment. We will show performance evaluations via computer simulation in section IV, with our conclusions in section V.

2 Burst Control Packet Delay

In the control packet network, the switch control plane may experience an overload period because of random fluctuations in the burst control packet arrival process. Namely, BCP congestion could occur in a switch control plane, as a result, the process of BCP is delayed. Then, it results in early arrival problem. Especially, most of the data loss by early arrival happens at the egress node or near the egress node, as shown in figure 1.

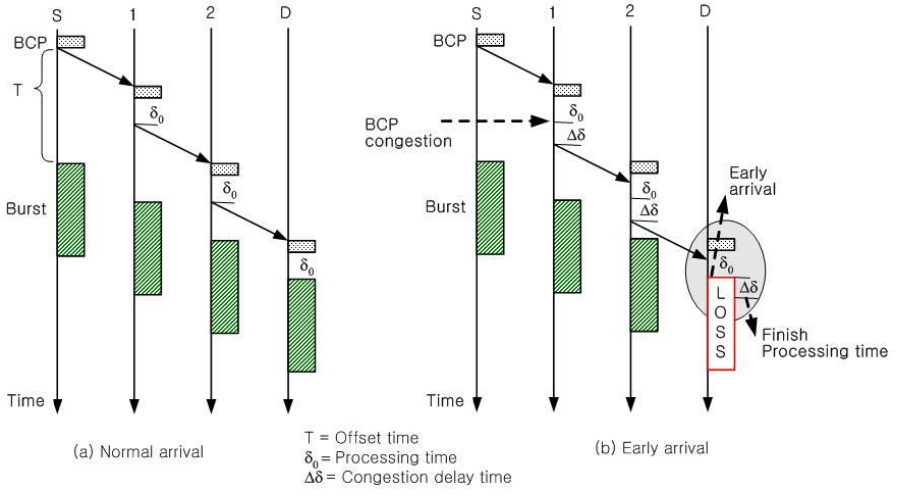


Fig. 1. Early arrival problem under the congestion at control plane

To avoid this event, a conventional solution uses fixed-length inlet fiber delay lines (FDLs), placed at all data channels in order to delay the data burst for a budget time for sufficient BCP processing [4]. That is the solution taken at the data plane to solve early arrival problem. If the fixed-length inlet FDLs are used to hinder early arrival, all of the data burst passing through the OBS router may be delayed. Therefore, a new approach taken at the control plane without delay is proposed.

According to OBS protocols, the BCP is transmitted to the core OBS router earlier than its corresponding data burst. At ingress node, we determine the offset time by the product of the processing time (δ_o) at each intermediate node and the number of hops (H_i, $i = 1, 2, 3, ...$) traversed by the BCP. Therefore, the offset time is longer at the ingress node than at the egress node. As a BCP proceeds toward the egress node, the residual offset time diminishes as much as the processing time is consumed at intermediate nodes.

The proposed algorithm can solve the early arrival problem at the OBS control plane instead of the data plane, where a BCP with short residual offset time is served by the scheduler earlier than a BCP with long residual offset time.

3 Scheduling Considering Offset-Time

A BCP that enters the control channel will experience O/E/O conversion in OBS switch control unit. The control packet contains specific data information that is wavelength identification, burst data length, offset time, etc. Tracking the BCP information, the number of remaining or passed hops is known. As mentioned previously, the processing time is included in the offset time at each OBS node. The relation is considered between the processing time and the queue size, namely, the number of packets in the queue. Let W_q be the waiting time in queue, N_q the queue size, the service time, which is the fixed value depending on the system. The processing time is the waiting time plus the service time,

$$\delta_o = W_q + \sigma \tag{1}$$

Here, the waiting time is the product of the buffer size and the service time,

$$W_q = N_o * \sigma \tag{2}$$

So, the processing time is Burst

$$\delta_o = (N_q + 1) * \sigma \tag{3}$$

Therefore, the buffer size N_q is then given by

$$N_q = \frac{\delta_o - \sigma}{\sigma} \tag{4}$$

Clearly, the buffer size is proportional to the processing time. In order to reduce the burst loss owing to the unduly BCP delay, the buffer size must be increased.

As a result, offset time related to the processing time also has to be extended. Hence, data burst should be delayed in the ingress node as many as the increment buffer size. The SCOT operates as followings. In case BCPs get to the egress node, control packets are first classified into two types which are one residual offset time($\delta_H(1)$) packet and two more than residual offset time($\delta_H(i)$, $i = 2, 3, ...$) packet and then assigned to a queue according to the residual offset time, as shown in figure 2. The residual offset time is $\delta_H(i)$, i = the remaining hop count.

Here,

$$\delta_H(i) = \sum_{j=1}^{i} \delta_o = \sum_{j=1}^{i} \delta = \delta * i \tag{5}$$

In normal condition, each of the queue is served in a round-robin order. However, if BCPs with $\delta_H(1)$ offset time arrive at the buffer overflowed over N_{Th}, the corresponding data burst gets lost in this node since there are not enough time for switch configuration, that is, an early arrival occurs. In this case, the scheduler should serve more BCPs with $\delta_H(1)$. Then, the weight w of queue is defined as :

$$w = \frac{N_{Th} + \Delta N}{N_{Th}} \tag{6}$$

Here, N is increment of $\delta_H(1)$ BCPs. As the scheduler serves $\delta_H(1)$ BCPs earlier than $\delta_H(i)$ BCPs, burst loss problem corresponding to a BCP close to the egress node due to early arrival can be solved This scheduling algorithm has an effect on the same processing time, even though the buffer size at control plane increases by using FDLs at data plane. This SCOT algorithm results in good performance, reducing the loss probability due to early arrival problem.

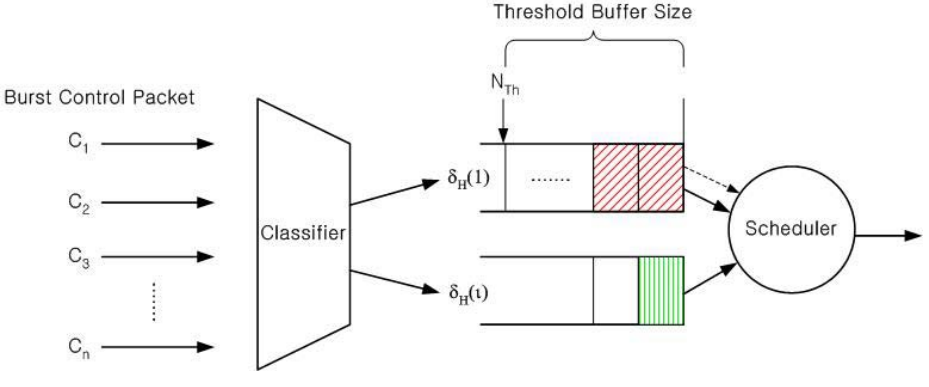


Fig. 2. Example behavior of the total BCP processing delay time(μs) with Pareto distribution (The scheduling considering offset time (SCOT) algorithm

4 Performance Evaluation

To validate the proposed schemes, computer simulations were performed under the following assumptions: The switch was 8 x 8 optical core OBS node. To model the burst BCP traffic, the ON/OFF source model was used, the most popular. We assumed that the burst inter-arrival rate use the Pareto distribution with Hurst parameter H=0.9, affected by the data burst generation. This means that BCP traffic also had burst characteristics because data is burst traffic. If the constant assemble time of data burst generation scheme [5], it was to implement easier than variable assembly. A number of remaining or passed hops were distributed uniformly. We also assumed that control channel had one channel per each fiber/port and the bit rates of 10Gbps per wavelength. The BCP service times for the forwarder, the scheduler, and the switch controller on the burst arrival rate to an OBS router were set to 0.1 μ, 0.1 μ, and 0.025 μ respectively [6]. Early arrival problem happens frequently under the above-mentioned condition when BCP processing time is longer than burst arrival time. These assumptions may appear difficult to implement through the current technique, but, with some pipelining and parallelism, scheduling process can be shortened.

Considering the input traffic processing delay, as mentioned before, the inter-arrival rate is assumed to use the Pareto distribution with Hurst parameter H=0.9. If the constant assemble time in data burst generation scheme which is easier to implement than variable assembly is used at the OBS edge node, the BCP traffic also has burst characteristics. The infinite variance of a Pareto

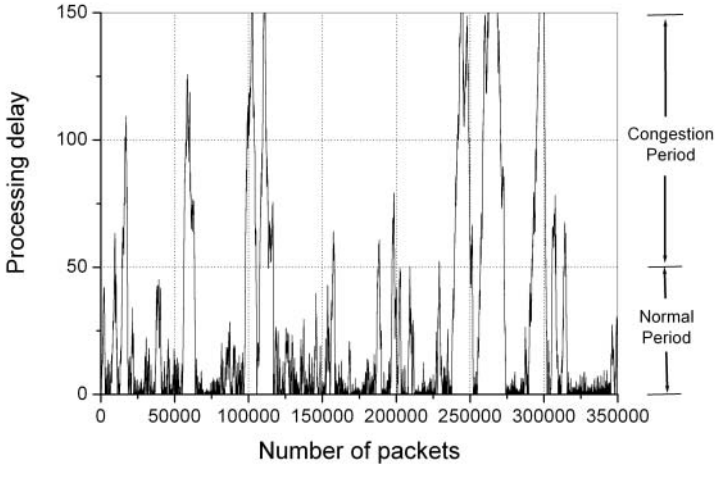


Fig. 3. Example behavior of the total BCP processing delay time(μs) with Pareto distribution (Hurst parameter H=0.9, Offered load=0.8, Control Channel=8, Service time=0.225 μs, Simulation time=100ms)

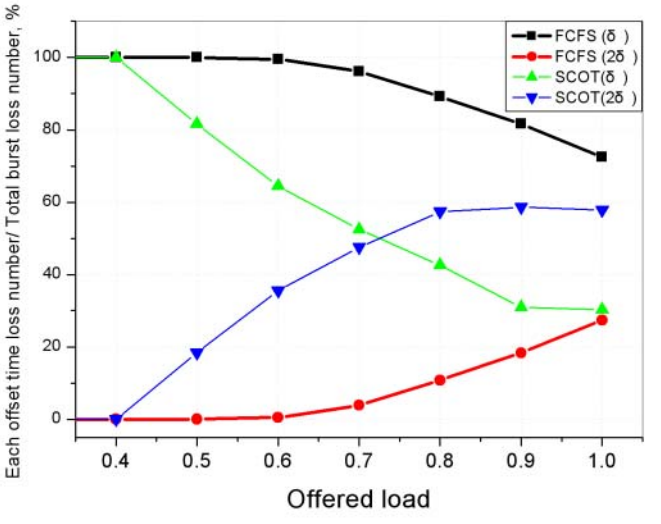


Fig. 4. Ratio of burst loss to total burst loss with different offset time under self-similar traffic (threshold buffer size = 20, Hurst parameter H=0.9)

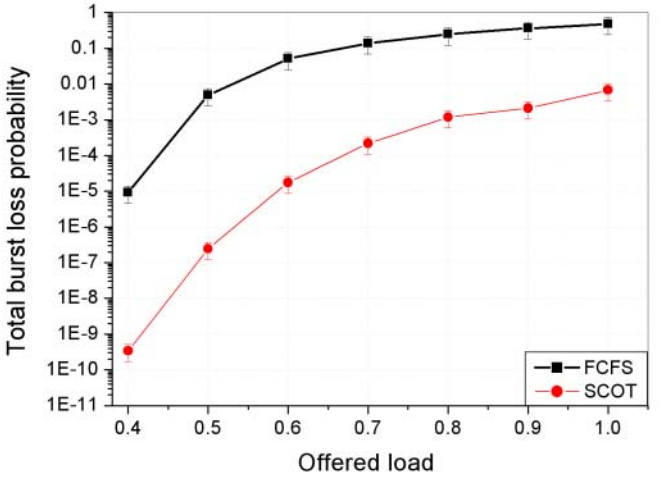


Fig. 5. Total burst loss probability due to early arrival problem under self-similar traffic (N_{Th}= 20(4.5 μs) (Offered load = 0.8)

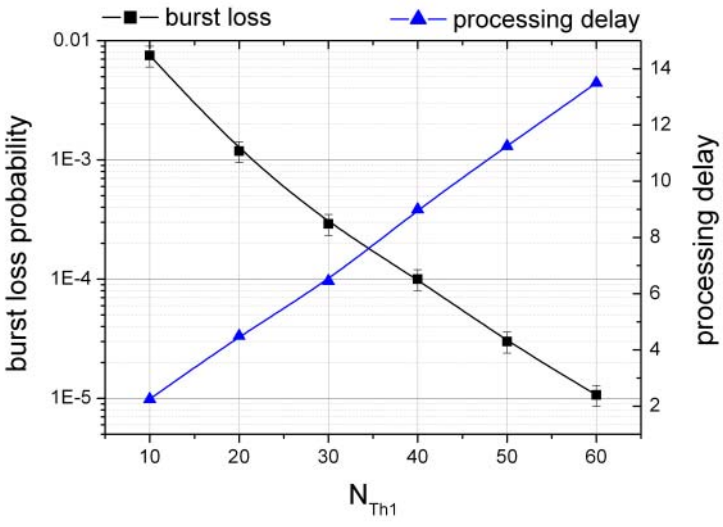


Fig. 6. Threshold buffer size (N_{Th}) versus burst loss probability and processing delay(μs) (Offered load = 0.8)

distribution exhibits extreme variability. Hence, the early arrival occurs during the congestion period as depicted in figure 3.

We compared SCOT algorithm to the general buffering, the First-Come First-Served (FCFS) scheduling algorithm that is commonly used in OBS control plane. The inter-arrival rate of offered load used the Pareto distribution with Hurst parameter H=0.9. The BCP service time was 0.225 μs/burst. Therefore, when offered load is 1, the arrival rate is 4444444 burst/sec. The threshold buffer size(N_{Th}) that was determined by offset time is set to 20 packets. Hence, the processing time of BCP in each OBS node was 4.5 μs in this simulation.

Figure 4 shows the ratio of burst loss with the offset time of $\delta_H(1)$ and $\delta_H(i)$ to total burst loss. We do not observe any lost packet when the input offered load is smaller than 0.3 in total 1012 BCP packets. As shown in figure 4, most of BCPs loss due to early arrival happens to BCPs with short residual offset time in FCFS algorithm. On the other hand, if we apply SCOT algorithm, BCPs loss with only $\delta_H(1)$ offset time decreases. But, BCPs loss with $\delta_H(i)$ offset time increases as the offered load is increased. In order words, the burst loss corresponding to BCP with $\delta_H(1)$ offset time is decreased by trading off BCPs with $\delta_H(i)$ offset time.

Furthermore, as shown in figure 5, total loss probability due to early arrival problem decreases by 2 to 3 orders, compared to FCFS algorithm. The reason is that the SCOT algorithm survives data that is lost due to early arrival in FCFS algorithm.

The threshold buffer size (N_{Th}) must be considered, but it is not easy to decide the threshold value, critical to the dimensioning of the control plane. As shown in figure 6, if N_{Th} is set to large, the burst loss decreases, however, processing delay increases. Therefore, the optimal threshold value must be determined to satisfy two conditions.

5 Conclusion

In this paper, the scheduling algorithm in the control plane to solve the early arrival problem in optical burst-switched WDM networks was presented. In the proposed algorithm, a BCP with short residual offset time is served prior to BCPs with long residual offset time. The priority-scheduling algorithm eliminates the early arrival problem due to the long processing delay of BCP. Therefore, the proposed algorithm can reduce the burst loss ratio at each node and the total loss probability over the networks. From the analysis of the loss probability under the burst traffic through simulations, it was learned that the proposed algorithm could decrease the total loss probability at the control plane without producing excessive delay.

Acknowledgement

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through OIRC project and ETRI.

References

[1] M. Listanti, V. Eranmo: Architecture and technological issues for future optical Internet networks. IEEE Communication Magazine, Sept. (2000) 82-92
[2] C. Qiao, M. Yoo: Optical burst switching (OBS)- a new paradign for an optical Internet. Journal of High Speed Networks, Vol. 8, No. 1, (1999) 68-84
[3] Jonathan S. Turner: Terabit Burst Switching. Journal of High Speed Networks, Dec. (1999)
[4] F. Callegati, H. C. Cankaya, Y. Xiong, M. Vandenhoute: Design Issues of Optical IP Routers for Internet Backbone Applications. IEEE Communication Magazine, Dec. (1999) 124-128
[5] A. Ge, F. Callegati: On Optical Burst Switching and Self-similar Traffic. IEEE Communication Magazine, Vol. 4, No. 3, March (2000)
[6] Y. Xiong, M. Vandenhoute, H. C. Cankaya: Control Architecture in Optical Burst switched WDM Networks. IEEE JSAC, Vol. 18, No.10, Oct. (2000) 1838-1851

Part VI

Security

Fast Classification, Calibration, and Visualization of Network Attacks on Backbone Links

Hyogon Kim[1], Jin-Ho Kim[2], Saewoong Bahk[2], and Inhye Kang[3]

[1] Korea University
[2] Seoul National University
[3] University of Seoul

Abstract. This paper presents a novel approach that can simultaneously detect, classify, calibrate and visualize attack traffic at high speed, in real time. In particular, upon a packet arrival, this approach makes it possible to immediately determine if the packet constitutes an attack and if so, what type of attack it is. In this approach, a flow is defined by a 3-tuple, composed of source address, destination address, and destination port. The core idea starts from the observation that only DoS attack, hostscan and portscan appear as a regular geometric shape in the hyperspace defined by the 3-tuple. Instead of employing complex pattern recognition techniques to identify the regular shapes in the hyperspace, we apply an original algorithm called RADAR that captures the "pivoted movement" in one or more of the 3 coordinates. From the geometric perspective, such movement forms the aforementioned regular pattern along the axis of the pivoted dimension. Through real execution on a Gigabit link, we demonstrate that the algorithm is both fast and precise. Since we need only 3 to 4 memory lookups per packet to detect and classify an attack packet, while simultaneously running 2 copies of the algorithm on a Pentium-4 PC, the algorithm incurred no packet loss over 330Mbps live traffic. Memory requirement is also low - at most 200MB of memory suffices even for Gigabit pipes. Finally, the method is general enough to detect both DoS's and scans, but the focus of the paper is on its capability to identify the latter on backbone links, in the light of recent global worm epidemics.

1 Introduction

Detecting attacks on backbone-speed links, let alone performing attack classification and other more involved tasks, is hard. The formidable speed forbids any algorithm requiring more than a few memory lookups and computation steps per packet, to operate in-line. Traditional anomaly-based approach [1, 2] is obviously not usable in this environment since, first, it requires traffic accumulation to characterize normal traffic, second, it usually requires complex computation. In this paper, we discuss an approach to simultaneously detect, classify, and calibrate attack traffic at backbone speed, in real time. Better yet, it easily lends itself to

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 837–846, 2004.

the visualization of on-going attacks. To be more specific, it has the following desirable properties: a) real-time detection and classification: done in O(1) per-packet processing, immediately upon packet arrival, b) low memory requirement: less than 200MB for gigabit pipes, c) ease of calibration: attack source/victim, duration, intensity, dimensions identified without off-line post-mortem analysis, d) minimal false positives/negatives, e) no requirement for the support from the Internet infrastructure in any form: neither protocol modification, protocol addition, nor coordination between networks/routers, f) simultaneous DoS, hostscan and portscan tracking, and finally, g) immunity from asymmetric routing.

This paper is organized as follows: Section 2 presents our real-time classification method. A novel representation of attacks, their particular signatures, and the implementation of the signature generator are discussed. In Section 3, we show the result of applying the algorithm to a backbone trace, and live network traffic on campus backbone. The paper is concluded in Section 4. Due to the space constraints we omit the discussion on the statistical nature of the method, its analysis, performance evaluation of the scheme in terms of the speed, memory requirement, sensitivity, estimation error, and false positive rate. Interested readers are referred to [3] for these details and related work.

2 Real-Time Attack Classification

On each packet arrival, we want to judge whether it is (highly likely) part of an attack or not. And if indeed it constitutes an attack, we want to classify the type of attack: DoS, hostscan, or portscan . Furthermore, we want to identify who is the victim (DoS), who is the perpetrator and what ports are scanned (hostscan, portscan), and the intensity of the attack. In this section, we discuss our approach to achieve these goals. First, we define a flow to be a 3-tuple $< s, d, p >$, composed of the source address (s), destination address (d), and destination port (p). Our novel idea starts from the observation that only DoS attack, hostscan and portscan appear as a regular geometric entity in the hyperspace defined by the 3-tuple. For instance, source-spoofed DoS packets maintain a fixed destination address, thus appears as a straight line (in case destination port is fixed) parallel to the s axis, or as a rectangle (in case destination port is randomly varied) parallel to the s-p plane. Legitimate flows, on the other hand, appear as random points scattered across the hyperspace. Figure shows the flows observed at 9:35 and 9:36 a.m. in December 14th, 2001 on two trans-pacific T-3 links connecting the U.S. and a Korean Internet Exchange. The three axes are the source IP address, destination IP address, and destination port as used in the flow definition above. (The source and the destination addresses have decimal scale.) Each dot in the 3-dimensional hyperspace represents a single flow (not a packet). Total of 2.22 million packets were mapped to the hyperspace in the figure, where the packets in the same flow fall on the same position. We can easily recognize the regular geometric formations, such as a large rectangle and a leaner rectangle lying parallel to s-axis, lines parallel to d-axis, and numerous vertical lines. These regular formations are (destination port varied) DoS at-

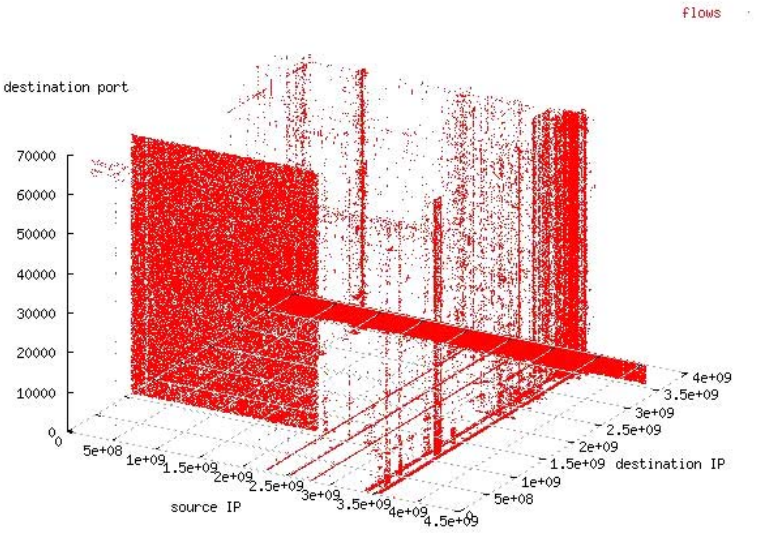


Fig. 1. Flows at around 9:35 a.m., Dec. 14th, 2001

tacks, hostscans, and portscans, respectively. Although far outnumbering them, legitimate flows do not form any regular shape, and are less conspicuous. Instead of employing complex pattern recognition techniques such as 3-dimensional edge detection, we apply an original algorithm that captures the "pivoted movement" in one or more of the 3 coordinates. This is because, from graphical perspective, such movement forms the aforementioned regular pattern along the axis of the pivoted dimension. In hostscan, the source IP address and the destination port are fixed, while the destination IP address pivots on them [5]. In portscan, the destination port pivots on the source and the destination IP address. In source-spoofed DoS, the destination IP address is fixed, while either only the source IP address or both the source IP address and the destination port pivots on it [9].

In order to detect the presence of pivoting in the traffic stream, our scheme first generates a signature for each incoming packet. The signature is simply a tuple consisting of 3 binary values: $< K_s, K_d, K_p >$. The coordinates in the signature one-to-one correspond to the flow coordinates. Each coordinate value in the signature tells us whether the corresponding value in the flow (that the packet in hand belongs to) was seen "recently" or not. (The degree of recentness for different coordinates could vary, and we will deal with it later.) For example, suppose two flows

Arrival time	Flow	Flow ID
t:	$< 3.4.5.6, 5.6.7.8, 90 >$	1
$t+1$:	$< 1.2.3.4, 5.6.7.8, 80 >$	2

pass through the monitor that executes our scheme. For convenience, throughout the paper we will call the monitor RADAR monitor (for Real-time Attack Detection And Report), and the algorithm that it executes, RADAR algorithm. Unless we explicitly mention the algorithm, we refer to the monitor (that includes the algorithm) when we simply say RADAR. RADAR remembers these two flows for a finite time duration L. For the sake of explanation, let us assume for now that the time duration is the same for every coordinate, e.g., $L = 2$. When a packet with source IP = 1.2.3.4, destination IP = 3.4.5.6, destination port = 90 appears at time $t + 2$, RADAR tells that this packet's signature is $< K_s, K_d, K_p >=< 1, 0, 1 >$. This is because source IP address 1.2.3.4 appeared in flow (2) and port 90, in flow (1). But 3.4.5.6 was not used either in (1) or (2) as the destination address, so $K_d =' 0'$. If $L = 1$, flow (1) would have been purged from RADAR at the time of the packet arrival, and the signature would be $< 1, 0, 0 >$. In principle, this per-packet signature determines whether the packet is part of a "pivoted movement", and if so, what type it is. Note that when pivoting occurs, the value of the pivoted coordinate changes constantly from packet to packet within the attack stream. From the perspective of RADAR algorithm, the pivoted coordinate is viewed as persistently presenting recently unobserved values. In Fig. 2, for instance, the pivoted coordinate is the destination address, and each packet presents a new value: 72.142.101.84 → 72.142.101.197 → 197.14.58.120 → .... So RADAR will keep generating $< 1, 0, 1 >$ signatatures for hostscan. This way, RADAR gets to yield the signatures $< 1, 0, 1 >$, $< 1, 1, 0 >$, or $< 0, 1, * >$ rather frequently in the presence of hostscan, portscan, or source-spoofed DoS, respectively. ('*' is wildcard, i.e., '0' or '1'). These signatures are what we call attack signatures, and the corresponding flow goes through further examination. Sometimes legitimate traffic can get attack signatures, and vice versa. Or one attack might be mistaken as another, all due to hapless modification of one or more coordinates in the signature, so some refinement is required in back-end processing (which is much less time-pressed). The accuracy of the proposed algorithm thus depends on how likely these unwanted changes in the signature are, and the analysis of this statistical aspect of our algorithm can be found in [3].

2.1 Attack Signatures

In this section, we explore possible signatures and their semantics. There are attack signatures and the signatures of legitimate traffic, and we start the discussion with the former. Figure 3 exhaustively enumerates all signatures and their conceivable implied attack types. As we described earlier, '0' in a signature means that the monitor has not recently seen the value in the given coordinate. Thus, if a packet belongs to an attack stream, '0' value in a coordinate most probably means that the coordinate is pivoting. The leftmost column is the number of dimensions that are pivoting. The second column is how the attacks might manifest themselves geometrically when the attack is mapped on to the 3-d hyperspace a la Figure 1. An important note here is that the signatures listed in Table I are self-induced. Namely, the values in a signature are what are

Time	Source address	Source port	Destination address	Destination port
.....	.....	.....	.....	.....
09:35:23.955222	x.x.x.x	64218	72.142.101.184	111
09:35:23.958716	x.x.x.x	64232	72.142.101.197	111
09:35:23.965132	x.x.x.x	64310	197.14.58.120	111
09:35:23.965443	x.x.x.x	64311	197.14.58.121	111
09:35:23.966412	x.x.x.x	64316	197.14.58.126	111
09:35:23.974520	x.x.x.x	64322	197.14.58.132	111
09:35:23.976617	x.x.x.x	64331	197.14.58.141	111
09:35:24.091332	x.x.x.x	64424	19.231.216.127	111
09:35:24.093271	x.x.x.x	64423	19.231.216.126	111
09:35:24.093317	x.x.x.x	64422	19.231.216.125	111
.....	.....	.....	.....	.....
09:35:24.104956	x.x.x.x	64438	19.231.216.141	111
09:35:24.105238	x.x.x.x	64437	19.231.216.140	111
09:35:24.106191	x.x.x.x	64433	19.231.216.136	111
09:35:24.107471	x.x.x.x	64429	19.231.216.132	111
09:35:24.125654	x.x.x.x	64466	85.114.173.117	111
09:35:24.126519	x.x.x.x	64464	85.114.173.115	111
.....	.....	.....	.....	.....

Fig. 2. Real-life pivoting example: hostscan

Dim.	Graphical manifestation	Signature	Implied attack
0	Dot	<1, 1, 1>	Single-source-spoofed DoS
1	Straight line	<1, 1, 0>	Portscan
		<1, 0, 1>	Hostscan
		<0, 1, 1>	Source-spoofed DoS (destination port fixed)
2	Rectangle	<1, 0, 0>	Kamikaze
		<0, 1, 0>	Source-spoofed DoS (destination port varied)
		<0, 0, 1>	Distributed hostscan
3	Hexahedron	<0, 0, 0>	Network-directed DoS

Fig. 3. Attack signatures

caused by the corresponding attack itself, but not by others. To wit, these are what an attack would obtain in the absence of any cross (legitimate + other type of attack) traffic. But as we discussed earlier, cross traffic might overlap in one or more coordinates, and these signatures are not always those detected when corresponding attack is under way. For $< 0, 0, 0 >$, one or more coordinates can be flipped to 1 by cross traffic that happens to coincide on IP addresses or

port number. Suppose a flow $< 4.4.4.4, 2.2.2.2, 5555 >$ is initiated after a flow $< 1.1.1.1, 2.2.2.2, 3333 >$ is registered by RADAR. Then the former will receive $< 0, 1, 0 >$ signature, which RADAR recognizes as the port-varied DoS attack. Since the signatures in Table I are before their attack traffic is subject to possible overlap, we call them original signatures. In contrast, if an original signature does get modified by overlap, we call the resulting signature transformed signature. For instance, if the transformation $< 0, 0, 0 > \rightarrow < 1, 1, 1 >$ occurs, where $< 0, 0, 0 >$ is the original signature and $< 1, 1, 1 >$ is the transformed signature. So when RADAR detects an attack signature, it might be a transformed signature, or an original signature kept intact. Most signatures in Table I are fairly straightforward, but there are a few that call for some explanation. First, even if nothing is pivoting (signature $< 1, 1, 1 >$), theoretically it still can constitute an attack. One may use a single, spoofed source IP address and a fixed destination port number in a DoS attack. But it is impractical from the perspective of the attacker. Once the attack is identified as DoS, simply filtering on the single (spoofed) source address leads to the complete elimination of the attack. "Worse" yet, the collateral damage in the filtering process is limited to the spoofed host only (it is denied an access to the victim). Therefore, we assume in this paper that this type of attack is not employed in reality. Second, we assume the distributed hostscan (signature $< 0, 0, 1 >$) will be detected as multiple hostscans (signature $< 1, 0, 1 >$), as it is. Third, the network-directed DoS (signature $< 0, 0, 0 >$) is an attack on the ingress pipe rather than on any particular host in the victim network. The only rationale might be that the attacker wants to evade detection because attack intensity for individual destination IP address contained in the pivoting range is proportionally reduced. But then the attacker is assuming (micro) flow-based detector as its potential opponent, which is lame under the whole gamut of other existing detecting/filtering methods [6, 7]. So in this paper, we also reject this type of attack as dubious. In sum, we reject three among the listed eight as original attack signatures: $< 0, 0, 0 >$, $< 0, 0, 1 >$, and $< 1, 1, 1 >$ (shaded in Table I). Finally, distributed DoS (DDoS) does not appear in Table I. We can consider two cases. If DDoS sources spoof source IP address, they will collectively be detected as a single DoS attack $< 0, 1, * >$. If spoofing is not used, since individual DoS streams look like legitimate flows from our monitor's viewpoint, they will not be detected as attacks. Usually, however, DDoS mobilizes a large DoS network of agent hosts to maximize the impact - e.g., more than 359,000 machines were made an agent by Code-Red version 2 [4] in an attempt to bombard the White House web site. The Sapphire worm infected more than 70,000 hosts [5]. Therefore, when the attack commences, RADAR will begin to see a great many source IP addresses all of a sudden. This will produce a noticeable amount of $< 0, 1, * >$ signature at a fast pace, and draw the attention of RADAR. Provided the intensity exceeds the tolerable threshold, which is low enough to be used on a spoofed DoS attack from a single attacker (see Section V), RADAR will raise an alarm. The remaining five cases are of our interest in the paper. First of all, "Kamikaze" is special. A single source spews packets at a high rate towards random destination hosts at random ports.

Apparently, it cannot be an effective attack, but rather, it seems suicidal. The origin of this type of "attack" is not clear, but it does appear in our traces [3]. One explanation could be a bug in the DoS attack code - pivoting destination address instead of source. But a more plausible theory is that it is the backscatter [8] from the DoS victim towards spoofed attack sources. And in Table I, we list two DoS types, but the distinction is only for the convenience of analysis - it does not bear any practical significance. The signatures of the legitimate traffic can be similarly analyzed, but we omit the discussion due to space constraint. Interested readers can find them in [3].

2.2 Signature Generation

Fig. 4 shows the construction of main filter in the attack monitor. This is what we have called the "front-end" thus far. It is composed of 3 hash tables, and collectively these hash tables generate the signature for each incoming packet. The network/transport packet header is mirrored to the filter, where a single, separate lookup is made against source IP address, destination IP address, and destination port number table, respectively. When a value (address or port) is 'not found', i.e., recently unobserved, it is registered in the corresponding hash table as a new sighting. Any hash function can be used as long as it has good distributional property and can be quickly calculated. Among these two properties, however, the speed weighs more for the front-end. For instance, MD5 and SHA-1 may have good distributional property, but they require too complicated a computation, so they would not fit our environment. Our experience shows that using the least significant 24 bits from the IP address suffices for casual operation. Against the backbone trace we have, it resulted in 1.0072 comparisons on average (most are 0 and 1, where 0 means empty hash bucket), with only a few reaching up to 8 comparisons. For port hash table, the hash function is identify function, i.e., we use the port number as the index itself. This is because there are only 64K port number values. Since the hash lookups are used, the complexity of the main filter can be engineered at O(1). with each entry is the last accessed time t_l. We maintain a moving time window L beyond which registered IP addresses or port numbers age out. Namely, if $t_{now} - L > t_l$, we remove the entry from the corresponding hash table. We call the time window lifetime, and we define two lifetimes as follows:

- L_H $(= L_s = L_d)$: [source/destination] host lifetime
- L_p : destination port lifetime

The reason that we perform a separate lookup for each coordinate is clear. If we maintained each flow entry indexed by $< s, d, p >$ collectively, we would not know which coordinate is responsible for a failed flow lookup. It means that we would not know immediately which coordinate is being pivoted, i.e., what type of attack is being mounted. Then some additional processing would be necessary on these new flows in order to achieve classification. Therefore, for real-time classification, separate hash lookups are essential. Earlier we mentioned the possibility

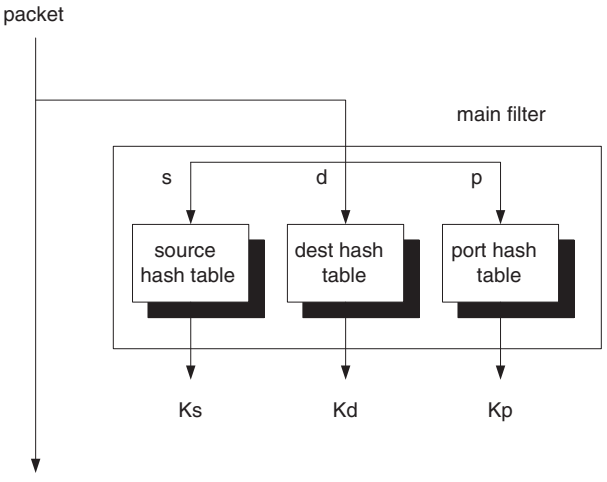


Fig. 4. Signature generation by the main filter

of signature transformation. In particular, when the signature of the first packet belonging to a legitimate flow gets transformed, the packet may be identified as an attack. For $< 1, 1, 1 >$, on the other hand, the cause of misinterpretation is the inadequately set lifetime(s). In case it is set too low, RADAR forgets too fast (i.e., before the flow ends), and returns 0 when it should return 1. Likewise, attack packets can get non-attack or incorrect attack signatures depending on the number and location of the flipped bit(s). So there is always possibility that any coordinate can suffer this unwanted bit flip(s). In [3], we analyze the false positive and false negative probability of the proposed algorithm caused by bit flip(s).

3 Implementation

We implemented a prototype of the RADAR system. Figure 5 shows the result of applying RADAR to the 8-hour trace (Dec. 14th, 2002) of about 612 million packets. It processed the trace in just 2.5 hours on a Pentium-3, 966MHz PC. The figure clearly shows that it successfully extracts attacks. Interested readers can find and compare animations of attacks and their processed results in [3]. We also plugged RADAR to a campus network gateway. The incoming packets were optically tapped from the gateway router on two Gigabit Ethernet interfaces [3]. A Pentium-4 2.4GHz machine with 512MB Rambus memory, Intel PRO/1000MF dual port LAN card, and PCI 2.2 (32bit) bus simultaneously run a separate instance of the RADAR algorithm on each Ethernet port. The total traffic rate was roughly 330Mbps (65Kpps) at the time of the experiments [3]. The most important result is that there was no packet loss at the kernel [3], due to RADAR processing. This is remarkable considering that we simultaneously run 2 instances of the algorithm. The memory requirement of the hash tables in the main filter

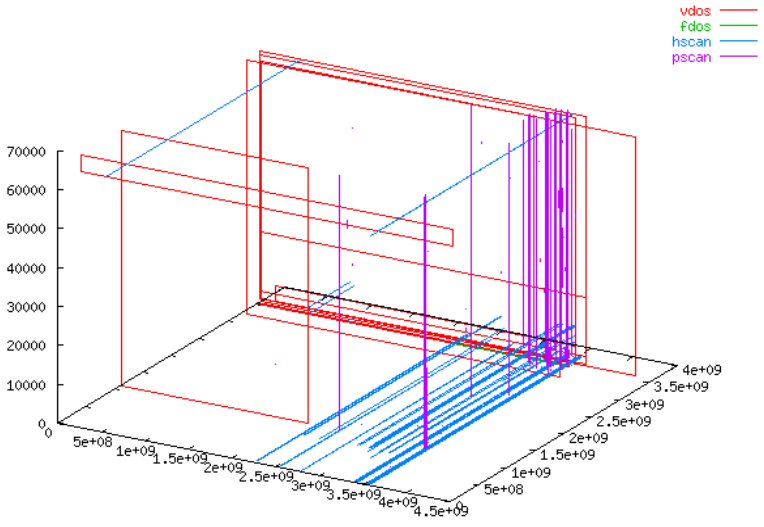


Fig. 5. Graphical output from the post-filter, a real RADAR-processed result of Figure 1

and the post filter [3] is moderate. Assuming we use a 24-bit hash for the source and destination IP tables, we need at least 225 hash buckets whose heads are a pointer (usually 4 octets). This alone is 128MB. Over and above, we need to store each flow in these tables, where a flow has at least 2 IP addresses, 1 port number, and a timestamp. Also each entry needs a pointer to the next entry. So each flow entry requires at least 17B. Assuming there are 1 million flows being tracked simultaneously, 34MB should be used. Then 1 million flows in the main filter IP table translates to approximately 10Gbps (OC-192) based on our flow arrival rate constant, since we have by default $L_H = 10s$. Over and above, we have the port table in the main filter. However, there are only 64K entries, thus it adds little to the memory requirement. In the post-filter, we do not have large tables, since concurrent attacks must be only handful. We do not expect to see, say 64,000 attacks all simultaneously under way, even it is on a backbone link. Therefore, we use 16-bit hash for all tables. Again, the memory requirement will be insignificant, most likely less than 2MB. In sum, more than half of the memory of RADAR is used to construct the IP tables in the main filter. If memory is a critical resource, we could use 23-bit hash, halving the requirement, and then 22-bit hash and so forth.

4 Conclusion

This paper proposes a novel approach that determines for each arriving packet if it constitutes an attack, and if so, what type of attack it is, on a high-speed link, in real time. The approach is based on a simple observation that only network attacks such as DoS and scans manifest themselves as a regular geometric

entity in a 3-dimensional hyperspace whose dimensions are source IP address, destination IP address, and destination port number. Instead of employing complex pattern recognition algorithms to detect such regular patterns, we propose a novel algorithm, RADAR, that captures the "pivoting" behavior which directly translates to the forming of abovementioned regular geometry in the 3-d hyperspace. RADAR algorithm requires only a few memory lookups per packet, yet the classification error is minimal. This algorithm pans out only suspicious packets matching the pivoting behavior, so buys enough time for a more sophisticated back-end processing which removes the false positives from the suspicious packets. We analyze the performance of RADAR algorithm in terms of speed, sensitivity, relative error, and false positive rate. The simulation and real implementation experiments demonstrate that the algorithm indeed performs up to our expectation on high-speed links, and that it could be a useful building block for an early warning and reaction framework against fast global attacks of the future.

References

[1] R. B. Blazek et al., "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods," IEEE Systems, Man, and Cybernetics Information Assurance Workshop, June 2001. 837

[2] C. C. Zhou, "Using Hidden Markov Model in Anomaly Intrusion Detection," http://tennis.ecs.umass.edu/ czou/research/HMM/index.htm. 837

[3] H. Kim, "Fast Classification, Calibration, and Visualization of DoS and Scan Attacks for Backbone Links," Technical Report, June 2003, http://net.korea.ac.kr/papers/RADAR.html. 838, 840, 843, 844, 845

[4] CAIDA, "CAIDA analysis of Code Red," http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml, July 2001. 842

[5] CAIDA, "Analysis of the Sapphire Worm," http://www.caida.org/analysis/security/sapphire/, Jan. 30, 2003. 839, 842

[6] M. Poletto, "Practical Approaches to Dealing with DDoS Attacks," NANOG presentaion, May 2001. http://www.nanog.org/mtg-0105/poletto.html. 842

[7] Ratul Manajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker, "Controlling High Bandwidth Aggregates in the Network," ACM CCR, V.32 N.3, July 2002. 842

[8] David Moore, Geoffrey Voelker, and Stefan Savage, "Inferring Internet Denial-of-Service Activity," in proceedings of the 2001 USENIX Security Symposium. 843

[9] K. Houle and J. Weaver, "Trends in Denial of Service Attack Technology," CERT Coordination Center, Oct. 2001. 839

On Layered VPN Architecture for Enabling User-Based Multiply Associated VPNs

Yoshihiro Hara[1], Hiroyuki Ohsaki[1], Makoto Imase[1], Yoshitake Tajima[2], Masahiro Maruyoshi[2], and Junichi Murayama[2]

[1] Graduate School of Information Science and Technology, Osaka University
1–3 Machikaneyama–cho, Toyonaka–shi, Osaka, 560–8531 Japan
[2] NTT Information Sharing Platform Laboratories, NTT Corporation
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan

Abstract. In our previous work, we have proposed a new VPN architecture for enabling user-based multiply associated VPNs [1]. Almost all existing VPN technologies assume that users never simultaneously access more than a single VPN. Thus, for realizing a new VPN service allowing users to simultaneously join multiple VPNs, several fundamental mechanisms, such as dynamically changing user's VPN association status according to the user's request and authorizing user's access to a group of VPNs, are required. In this paper, we propose a layered VPN architecture for realizing user-based multiply associated VPN. Our layered VPN architecture consists of three network levels such as PNL (Physical Network Level), LNL (Logical Network Level), and UNL (User Network Level). First, we discuss and classify functions required for each network level. We then present several approaches for implementing each network level using existing layer 2, 3, and 4 networking technologies, and quantitatively evaluate their advantages and disadvantages from several viewpoints including scalability and transmission speed.

1 Introduction

With recent advancements in network technology, various social activities such as commerce and trade, politics, labor, and other functions are relying more on network communications. In the near future, this may form virtual organizations within the network. We call these virtual organizations "cyber-societies." A "person" in cyber-society needs to establish secure communication and associate with multiple virtual organizations. We believe these virtual organizations can be realized through Virtual Private Networks (VPNs) as network services.

As current technologies for VPN services, there are Provider Provisioned VPN (PPVPN) [2, 3, 4] and extranets [5, 6]. However, PPVPN simply builds a VPN between customers' LAN sites. Also, extranets are difficult to manage and transmission performance is degraded when hosts attempt to connect to a lot of VPNs. These problems with existing VPN technology prevent users from associating themselves simultaneously to multiple VPNs at a user level.

To address this problem, we are considering a new VPN architecture that would allow users to simultaneously associate with multiple VPNs [1]. We call

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 847–856, 2004.

this new VPN "Multiply-Associated VPN (MAVPN)". This paper has two goals. First, we will show that by using a layered model for MAVPN's architecture, MAVPN can be easily realized by integrating existing layer-based technologies. Next, from various perspectives, we will evaluate the advantages and disadvantages of integrating MAVPN into layers 2, 3 and 4 of the network layer model.

2 Layered MAVPN Architecture

In this section we will explain our proposed layered MAVPN architecture. With existing PPVPN, due to its site-to-site VPN tunnel connection method, it is not possible for users to make their own VPN connections with other users. Also, with existing extranet technology it is not possible to simultaneously make a number of VPN connections. To address these problems, we have proposed the MAVPN architecture.

To realize MAVPN, the following three main processes are required. First, provide a base network. Then, build various VPNs on top of the base network. Finally, provide VPN control functions so users can access multiple VPNs securely and simultaneously.

The actual implementation of these processes appears to be complex and difficult. However, by building these three processes on existing network technology through layers, we believe it implementation will be relatively simple.

For example, since existing PPVPN constructs a logical network over a base network, it is expected that an additional layer for VPN control functions can be easily provided.

Below, we discuss the required features for each of three layered network levels (physical network level, logical network level, user network level). A definition of terms for each network level is shown in Tab. 1.

First, we will discuss the Physical Network Level (PNL). The PNL provides the network that serves as the foundation for building a VPN. Figure 1 is a graphical representation of the PNL. As shown in Fig. 1, nodes such as routers, switches and hosts are connected by links.

Next, we will discuss the Logical Network Level (LNL). In the LNL, a VPN is formed on top of the network provided by the PNL. With PPVPN, the site is the basic unit in forming a VPN. With the MAVPN LNL, however, we introduce the concept of host entity as the basic unit. This entity could be host users, user-level applications or server programs. In this paper, the term "entity" is defined for intended application in VPN services targeted to Application Service Providers (ASP) or for multi-OS usage. Figure 2 graphically represents the LNL. As shown in Fig. 2, an entity-based VPN is created through specifying the entity as the basic unit for authentication. It should be noted that such an entity-based VPN can be implemented using several existing VPN technologies.

Finally, we will discuss the User Network Level (UNL). The UNL controls access from the entity to each VPN when the entity simultaneously connects to, or is multiply associated with, multiple VPNs. Specifically, it provides the controls which let the entity transparently connect to and use multiple VPNs,

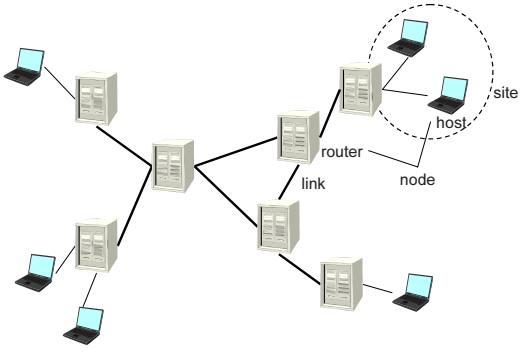


Fig. 1. Physical Network Level (PNL) in MAVPN architecture

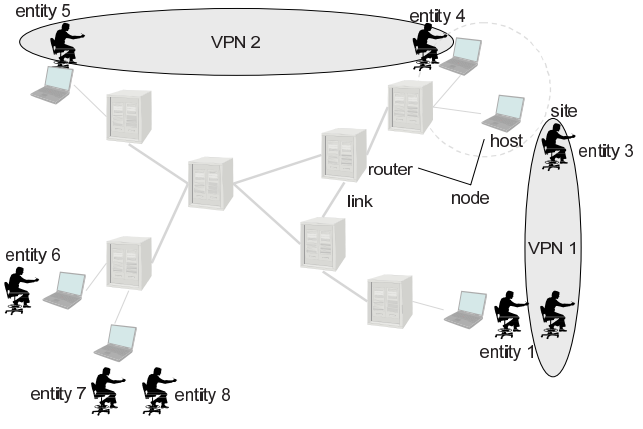


Fig. 2. Logical Network Level (UNL) in MAVPN architecture

as well as preventing unauthorized access across any other associated VPNs. Figure 3 is a graphic representation of the UNL. Note that the UNL utilizes some information stored in the packet for associating an entity with the corresponding VPN. Such association function can be performed at routers or other network devices.

3 Three Typical MAVPN Architecture

Because current wide-area connection services are commonly provided through network layer 2 or layer 3, we consider the PNL to be realized through network layer 2 or 3. Likewise, we consider the UNL to be realized through Layer 3 or Layer 4. Therefore, in this paper, we will discuss the following three MAVPN architecture types which are based on the three layered network levels.

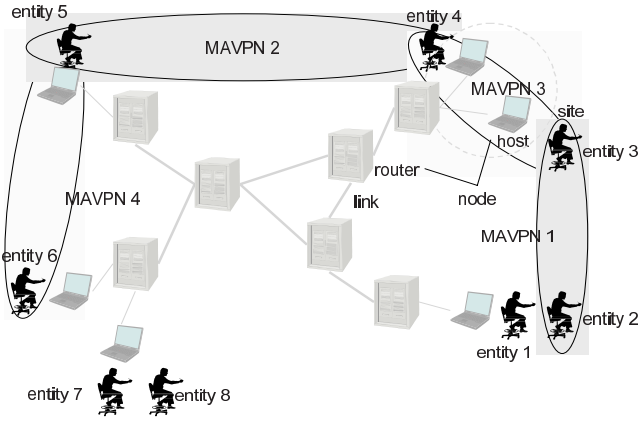


Fig. 3. User Network Level (UNL) in MAVPN architecture

Table 1. Definition of terms

Terms for Physical Network Level	
Host	Terminal, PC
Node	Devices like Hostsíñrouters, and switches
Link	Physical line between nodes
Terms for Logical Network Level	
Entity	Users, user-level applications, and server programs on hosts
VPN	Virtual closed network consist of entities
User for Logical Network Level	
Multiple Association	A single entity simultaneously connects to multiple VPNs

3.1 Architecture 2-3-4

Architecture 2-3-4 uses different network layers for each of the physical, logical, and user network levels. Architecture 2-3-4 is explained below.

First, the PNL is realized from information in network layer 2. The layer 2 network could be provided by Ethernet or MPLS [7], for example.

Second, the LNL is realized from information in network layer 3. The layer 3 network could be provided through MPLS-VPN [8], for example.

Next, the UNL is realized from information in network layers 4 and higher. For this method, the LNL, using information in packets from layer 4 or higher, would send packets from the entity to the appropriate multiply-associated VPN.

3.2 Architecture 2-2-3

Architecture 2-2-3 uses network layer 2 information for PNL and LNL, and network layer 3 information for the UNL. Architecture 2-2-3 is explained below.

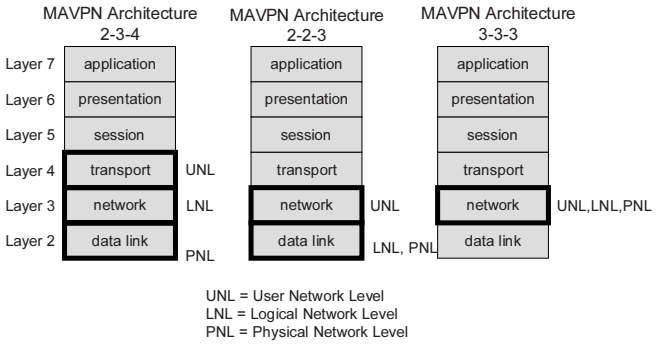


Fig. 4. Three typical MAVPN architectures

First, the PNL is realized from information in network layer 2. The layer 2 network could be provided by Ethernet or MPLS, for example.

Second, the LNL is realized from information in network layer 2. The layer 2 network could be provided by IEEE 802.1Q VLAN [9] or L2TP [10], for example.

Next, the UNL is realized from information in network layer 3. For this method, the LNL, using information in packets from layer 3, would send packets from the entity to the appropriate multiply-associated VPN.

3.3 Architecture 3-3-3

Architecture 3-3-3 uses network layer 3 information for each of the physical, logical and user network levels. Architecture 3-3-3 is explained below.

First, the PNL is realized from information in network layer 3. The layer 3 network could be provided by IP or other protocols, for example.

Second, the LNL is realized from a tunneled layer 3 network. The tunneling used in layer 3 could be provided by IPSec [11] or other protocols, for example.

The UNL is realized from information in network layer 3. For this method, the LNL, using information in packets from layer 3, would send packets from the entity to the appropriate multiply-associated VPN.

For each of these MAVPN architectures, Fig. 4 shows the relation of the three levels (physical, logical, user) and the layers in the OSI reference model.

4 Evaluating MAVPN Architecture from Several Viewpoints

In this paper, from several viewpoints, we will quantitatively evaluate the advantages and disadvantages of these three MAVPN architecture types.

When considering its expected application in the formation of a cyber-society, MAVPN must be able to operate on an extremely large network. For this reason, it is most preferable to have a high degree of scalability in numbers of nodes, VPNs and entities. Therefore, it is necessary to evaluate the scalability of these objects.

Also, in the past few years, the size of content on the Internet has mushroomed. For this reason, it is most preferable that transmission speeds under MAVPN are fast. Therefore, it is necessary to evaluate transmission speed.

Additionally, it is preferable that MAVPN users be able to use various types of network services. Therefore, it is necessary to evaluate the numbers of usable services.

Finally, it is preferable that MAVPN be flexible enough to meet user needs through ease of VPN configuration and VPN connection to entities. Therefore, it is necessary to evaluate entity and VPN manageability.

5 Evaluating MAVPN

5.1 Scalability (Number of Nodes, VPNs and Entities)

We evaluate the three MAVPN Architectures 2-3-4, 2-2-3, and 3-3-3 from the viewpoint of node, VPN and entity scalability.

Node Scalability Scalability of nodes in MAVPN is determined by the scalability of nodes in the PNL. These are considered below for each of the MAVPN architecture types.

- Architecture 2-3-4
  The physical network layer is created through network layer 2. Because of this, node scalability is more negatively impacted than Architecture 3-3-3 which uses network layer 3. For example, a typical layer 2 protocol like Ethernet is more negatively impacted in terms of scalability than a typical layer 3 protocol like IP.
- Architecture 2-2-3
  The physical network layer is created through the network layer 2. For this reason, node scalability is more negatively impacted than Architecture 3-3-3. On the other hand, scalability is comparable to Architecture 2-3-4, which also uses network layer 2 for the PNL.
- Architecture 3-3-3
  The physical network layer is created through the layer 3 network. For this reason, scalability is excellent as compared to Architecture 2-3-4.

Based on the above examination, Architecture 3-3-3 excels most in node scalability.

VPN Scalability VPN scalability in MAVPN is determined by VPN scalability in the LNL. These are considered below for each of the MAVPN architecture types.

- Architecture 2-3-4
  For the LNL, the scalability of number of VPNs is determined by what type of layer 3 network is used. For example, if using MPLS VPN, by stacking MPLS labels, a high degree of scalability is possible.
- Architecture 2-2-3
  For the LNL, the scalability of number of VPNs is determined by what type of layer 2 network is used. For example, if using IEEE 802.1Q tagging VLAN, by stacking tags, a high degree of scalability is achievable.
- Architecture 3-3-3
  For the LNL, the scalability of number of VPNs is determined by what type of layer 3 network is used. For example, if using a common tunneling technology like IPSec, it is necessary to connect a mesh of VPN tunnels between entities. Due to the limitation in the number of tunnels, the VPN scalability is limited. For instance, the maximum number of IPSec tunnels (number of SAs) is restricted to the available memory or system resources of the connected nodes.

Based on the above examination, Architectures 2-3-4 and 2-2-3 excel most in VPN scalability.

Entity Scalability Entity scalability in MAVPN is determined by UNL scalability. These are considered below for each of the MAVPN architecture types.

- Architecture 2-3-4
  Architecture 2-3-4 uses network layer 4 and higher information. In this case, since entity identification is based on network layer 4 or higher information, there is no restriction on the number of entities inherited from the physical or LNL. Therefore, for entity scalability, this architecture excels most when compared with the other MAVPN architecture types.
- Architecture 2-2-3
  Since Architecture 2-2-3 uses network layer 3 information in the UNL, the number of entities is restricted by logical address limitations in network layer 3. For example, if using a typical layer 3 protocol like IPv4, the entity limit is determined by the IP address limit (2^{32}). However, as IPv6 is adopted, this entity limitation is resolved. For this reason, excellent scalability is expected for the near future.
- Architecture 3-3-3
  Architecture 3-3-3, like Architecture 2-2-3, uses network layer 3 information for the UNL and therefore is restricted by layer 3 logical address limitations. However, like Architecture 2-2-3, in the near future the entity limitation is expected to be resolved. For this reason, excellent scalability is expected for the near future.

Based on the above examination, any of the architecture types are expected to enjoy good entity scalability for the future.

5.2 Transmission Speed

We evaluate the three MAVPN Architectures 2-3-4, 2-2-3, and 3-3-3 from the viewpoint of transmission speed. Since the most complex operations are performed at the UNL, it is necessary to consider the transmission speed scalability within this level. These are considered below for each of the architecture types.

- Architecture 2-3-4
  At the UNL, it is necessary to process information from network layer 4 and higher. Therefore, compared to the other two architectures, performance at the UNL is expected to be poor. Therefore, when compared to the other two architecture types, we expect Architecture 2-3-4 performance to suffer the most.
- Archtecture 2-2-3
  At the UNL, it is necessary to process information from network layer 3. Therefore, compared to Architecture 2-3-4, which processes layer 4 and higher data, faster processing speeds in the UNL are expected. Therefore, we expect performance to be better than Architecture 2-3-4.
- Archtecture 3-3-3
  At the UNL it is necessary to process data from network layer 3. Therefore, compared to Architecture 2-3-4, which processes layer 4 and higher data, faster processing speeds in the UNL are expected. Performance is expected to be similar to Architecture 2-2-3 which also processes information from network layer 3.

Based on the above examination, Architectures 2-2-3 and 3-3-3 excel the most in link speed scalability.

Usable Service Scalability We evaluate the three MAVPN Architectures 2-3-4, 2-2-3, and 3-3-3 from the viewpoint of usable service scalability. Since usable services are dependent on the protocols available to the user, it is necessary to consider the protocols used in forming the UNL. These are considered below for each of the architecture types.

- Architecture 2-3-4
  Since the UNL handles information from network layer 4 and higher, compared to MAVPN architectures which handle layer 3 information, there are fewer protocols available to users. Therefore, as compared to other MAVPN architectures, Architecture 2-3-4 suffers from lack of usable protocols.
- Architecture 2-2-3
  Since the UNL handles information from network layer 3, there are more usable protocols available to users than MAVPN Architecture 2-3-4, which handles information from layer 4. Therefore, this architecture excels over Architecture 2-3-4 in the number of available protocols for users.
- Architecture 3-3-3
  In Architecture 3-3-3, the UNL handles information from network layer 3. When compared with Architecture 2-3-4 which handles network layer 4 and

higher information, Architecture 3-3-3 excels in the number of available protocols for users. However, depending on the tunneling technology that Architecture 3-3-3 uses, the number of protocols available to users may be limited. For example, when using a currently common tunneling technology like IPSec, the number of protocols available to users is limited. For this reason, when compared with Architecture 2-2-3 which handles information from network layer 3, the number of usable services available to Architecture 3-3-3 users is worse.

Based on the above examination, in terms of number of services available to users, the suitability of the architectures is ranked from best to worst as: Architecture 2-2-3, 3-3-3, 2-3-4.

5.3 VPN Management

Evaluation of the ease of management of Architectures 2-3-4, 2-2-3, and 3-3-3 will be the topic of a future discussion.

5.4 Overall Evaluation Result

Table 2 shows the overall evaluation result as discussed in the previous sections. With regards to the evaluated criteria, of the three architectures (2-3-4, 2-2-3, 3-3-3), Architecture 2-2-3 excels most overall. Based on these results, we plan to further direct our attention to Architecture 2-2-3, including development of a prototype.

Table 2. Evaluation of each architecture

Viewpoints		2-3-4	2-2-3	3-3-3
Scalability	Number of Nodes	△	△	◯
	Number of VPNs	◯	◯	×
	Number of Entities	◯	△	△
Transmission Speed		×	◯	◯
Usable Service		×	◯	△
Total		×	◯	△

◯: good △: no good ×: bad

6 Conclusion and Topic for Future Discussion

In this paper we have proposed a new VPN architecture which would allow users to be multiply associated with several VPNs simultaneously. First, we proposed a VPN architecture which would allow users to multiply associated with multiple VPNs. We also discussed the required functionality. The VPN architectures proposed in this paper are layered in configuration, with three network levels (PNL,

UNL, and UNL). After determining the functional requirements and evaluating how each representative architecture type's utilized network layer 2, 3, and 4 in realizing each of the network levels, we were able to conclude that that Architecture 2-2-3 was superior. Topics for future discussion include evaluating a wider range of criteria such as manageability, security, reliability, and building an MAVPN prototype to run with existing network technology.

Acknowledgement

This study was performed through Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology of the Japanese Government.

References

[1] Hara, Y., Ohsaki, H., Imase, M., Tajima, Y., Maruyoshi, M., Murayama, J., Matsuda, K.: VPN architecture enabling users to be associated with multiple VPNs. Technical Report of IEICE (IN2003-50) (2003) 47–52 847

[2] Carugi, M., et al.: Service Requirements for Layer 3 Provider Provisioned Virtual Private Networks. Internet Draft `<draft-ietf-l3vpn-requirements-00.txt>` (2003) 847

[3] Nagarajan, A.: Generic Requirements for Provider Provisioned VPN. Internet Draft `<draft-ietf-l3vpn-generic-reqts-01.txt>` (2003) 847

[4] Callon, R., et al.: A Framework for Layer 3 Provider Provisioned Virtual Private Networks. Internet Draft `<draft-ietf-l3vpn-framework-00.txt>` (2003) 847

[5] Hara, H., Murayama, J., Isagai, K., Imaida, I.: IP-VPN Architecture for Policy-Based Networking. IEICE Technical Report IN2000-101 **100** (2000) 39–46 (in Japanese). 847

[6] Miyoshi, J., Imaida, I., Isagai, K., Murayama, J., Kuribayashi, S.: A Mechanism of Policy-Based Service Control in Communication between VPNs. IEICE Technical Report SSE99-171 **99** (2000) 61–66 (in Japanese). 847

[7] Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture. Request for Comments (RFC) 3031 (2001) 850

[8] Rosen, E., Rekhter, Y.: BGP/MPLS VPNs. Request for Comments (RFC) 2547 (1999) 850

[9] IEEE Standards for Local and Metropolitan Area Networks: Virtual bridged local area networks. IEEE Standard 802.1Q-1998 (1998) 851

[10] Townsley, W., et al.: Layer two tunneling protocol "L2TP". Request for Comments (RFC) 2661 (1999) 851

[11] Kent, S., Atkinson, R.: Security architecture for the internet protocol. Request for Comments (RFC) 2401 (1998) 851

SVAM: The Scalable Vulnerability Analysis Model Based on Active Networks*

Young J. Han[1], Jin S. Yang[1], Beom H. Chang[2], and Tai M. Chung[1]

[1] Internet Management Technology Laboratory
Scool of Information and Communication Engineering
SungKyunKwan University
300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do, 440-746, Korea
{yjhan,jsyang,tmchung}@imtl.skku.ac.kr
[2] Network Security Dept., Information Security Research Div.,
Electronics and Telecommunications Research Institute, Korea
bchang@etri.re.kr

Abstract. Active networks are novel approach to providing the flexibility in the network and service by allowing the network infrastructure to be programmable by using active packet, in which a traditional packet is replaced by a program which controls a behavior of nodes in the network. However, active packet may create new vulnerabilities at active node and attacks using those vulnerabilities can spread over network by mobility of active packet easily and rapidly. To be beforehand with preventing these attacks, we need the more scalable model than existing vulnerability analysis model. We present the Scalable Vulnerability Analysis Model which can manage vulnerable nodes quickly and efficiently in active networks. This approach provides good scalability by distributed vulnerability checking mechanism based on policy and fast adaptability by automated deploying mechanism of new vulnerability checking code. This approach will be suitable in large scale active networks.

1 Introduction

With the dramatic development in the network technology, various applications and services based on the network have been increasing. But, several important problems with existing networks were identified: the difficulty of integrating new technologies and standards into the shared network infrastructure, the poor performance due to redundant operations at several protocol layers, and the difficulty in accommodating new services in the existing architectural model [1]. In order to solve these problem, the concept of active networks was introduced from discussions within the broad DARPA (Defense Advanced Research Projects Agency) research community in 1994 and 1995 on the future direction of networking system [1]. Until now, researches regarding active networks have been actively progressing.

* This study was partially supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea(02-PJ3-PG6-EV08-0001)

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 857–866, 2004.

Because intermediate nodes can execute program code within packet and modify content of packets, active networks are called 'active'. The active network technologies present new direction for flexible, fast, and scalable service deployment. On the other hand, security issues about active networks are appearing.

Threats to network infrastructure are intimately tied to the model used for sharing the infrastructure. As it stands, the infrastructure of existing networks is vulnerable to not only various DoS (Denial of Service) attacks but also other various attacks such as traffic analysis. In active networks, the execution of program at intermediate nodes accompanies more security holes relatively to infrastructure of existing networks. Active networks, being more flexible, considerably expand possibilities of the threat [5].

Most threats or attacks start with scanning vulnerabilities of the network or the system. Therefore, those factors of risk could apparently decrease if any preventive action could be performed after checking vulnerabilities. Researches about vulnerability analysis management architecture have been performed and many management systems for vulnerability analysis have been developed in the existing network system, but any researches in the active networks have not been done yet [4].

In this paper, we describe the SVAM (Scalable Vulnerability Analysis Model) as our proposed model which can manage vulnerable nodes actively and efficiently in active networks. Section 2 introduces backgrounds of the SVAM. Section 3 describes the requirement of vulnerability analysis model in active networks, and then describes the framework and procedures of the SVAM in detail. Section 4 specifies the performance evaluation model in the aspect of scalability and proves the improvement of performance of our model. At last, section 5 summarizes our work and discusses future works.

2 Backgrounds

2.1 Active Networks and Their Security Considerations

Active networks allow intermediate routers to perform computations up to the application layer. In addition, users can program the network by injecting their programs into it [7]. These programs travel inside network and are executed at intermediate nodes resulting in the modification of their state and behavior [1]. The packet of active networks can be regarded as programs. We call these packets 'active packets'. The common active packets consist of program and data. We call these intermediate nodes 'active nodes' to distinguish them in existing networks. Figure 1 shows the concept of active networks. The active node consists of NodeOS (Node Operating System), EEs (Execution Environments), and AAs (Active Applications) [6]. NodeOS provides services such as packet scheduling, resource management, and packet classification. In addition, it provides services of physical resource access for EEs that operates on NodeOS. EE handles active packet and offers environment in which evaluates active packet. Four EEs are

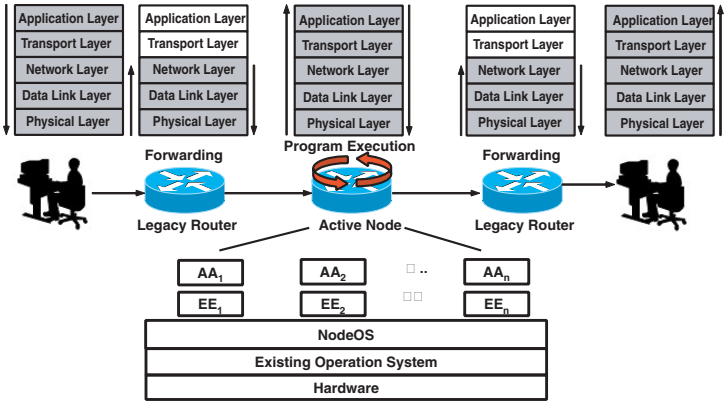


Fig. 1. The concept of active networks

currently operated and studied in ABone (Active Network Backbone): ANTS (Active Node Transfer System), PLAN (Packet Language for Active Network), ASP (Active Signaling Protocol), and CANES (Composable Active Network Elements) [2]. At last, AA is a program which is executed by the VM (Virtual Machine) of the particular EE. AA is implemented as customized service for end user applications, using the programming interface supplied by the EE. AAs are embodied in various forms according to implementation method of active networks.

Execution of spontaneous code on active networks has the advantage of flexibility and scalability, but it has several security considerations for itself. Active packet may misuse active nodes, network resources, and other active packets in various ways. Also, active node may misuse active packets. Some of possible problems that may occur are the following: Damage, Denial of Service, Theft, and Compound attack [7]. Currently, there are several researches to solve security issue by providing active node with integrity, safety, privacy, and availability [8]. But any researches about vulnerability analysis have not been done yet. There is fundamental limitation on perfectly securing active node. So, there still is potential vulnerability in the active node. Therefore, by checking vulnerability of active node in advance, we have to protect active node and entire network from external attack.

2.2 Existing Vulnerability Analysis Models and Their Weaknesses

XVAMs (eXisting Vulnerability Analysis Models) can be classified into host based vulnerability analysis models and network based vulnerability analysis models. In this paper, as comparative model for our proposed model, we mainly deal with remote network based vulnerability analysis model. Network based vulnerability analysis is model that protects components on the network from external intruder by detecting vulnerability of important system like firewall and web server of domain in remote position.

Centralized server-client based architecture that is general in this vulnerability analysis model is shown in Fig. 2. General vulnerability analysis model usually goes through following conception: Policy Establishment → Target Node Setup → Gathering Data → Induction Data → Report [4]. According to this procedure, manager which audits and analyzes vulnerabilities is placed at center. And managed nodes execute auditing by command from the manager and return the result. It is easy to install, implement, backup, renew, and maintain in one node that have major data and information, but single point of failure can be occurred. Other weaknesses of centralized vulnerability analysis model are follows.

Weak Extensibility and Scalability If managed node is added, additional agent needs to be installed by administrator. As the number of managed node increases, more information will be come up to be managed and manipulated. That is why bottle-neck and overhead can be occurred.

Slow Adaptability for New Vulnerability Renewal of the system for newly discovered vulnerability has to be manually done by administrator. It creates the delay to update.

3 SVAM

3.1 The Requirement of Vulnerability Analysis Model in Active Networks

The requirements of vulnerability analysis model based on active networks are following.

Consideration for Large Scale Network Attacks that are made in existing networks damage not only several systems but also whole network. In active networks, because these attacks use mobility of active packet to gain more artificial intelligence, more damages could be added. To be beforehand with preventing these attacks, design of vulnerability analysis model should be considered not just the system itself, but the large scale network.

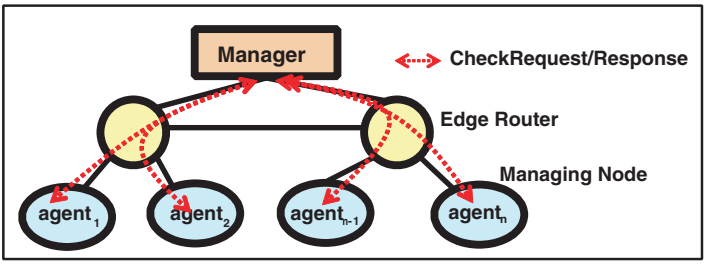


Fig. 2. Centralized vulnerability analysis model in existing networks

Scalable Vulnerability Analysis To don't affect the performance of entire network, the managed node should be easily expanded.

Fast Adaptability for New Vulnerability Because it is expected to have more active attacks in active networks, there should be fast responses for new vulnerabilities.

3.2 The Framework of the SVAM

The SVAM is the model which overcomes flaws of XVAMs but maintains merit of those. Figure 3 shows how the framework of the SVAM is modeled. Managed domain is new concept that is defined to divide large scale network into small managed areas and is a unit that vulnerability analysis policy is applied and scheduled in. The SVA Coordinator is lying on between the SVA Manager and the managed node. The SVA Manager is responsible for establishing and deploying policy; the SVA Coordinator is responsible for checking vulnerabilities. Dotted line *a* in Fig. 3 represents policy-distributing flow in managed domain, straight line represents vulnerability-checking flow in the SVA Coordinator and dotted line *b* represents deploying flow of vulnerability checking code for new vulnerability. The managed node is an end node which is managed by the SVA Manager in managed domain. It can be either an active node or a non-active node. Information of all managed nodes is registered and managed by the SVA Manager. The managed domain includes one SVA Manager, *vc*Capsule repositorys, and SVA Coordinators. The each component works as described below.

SVA Manager The SVA Manager has charge of managing its domain, that is, it manages managed nodes, SVA Coordinators, and the list of their vulnerabilities. It also configures policy for vulnerability analysis for each managed node and distributes the policy among SVA Coordinators that managed nodes are included. These policies will be usually distributed through *vp*Cpasule. Also it receives results of checking vulnerabilities from SVA Coordinators and reports those to administrator.

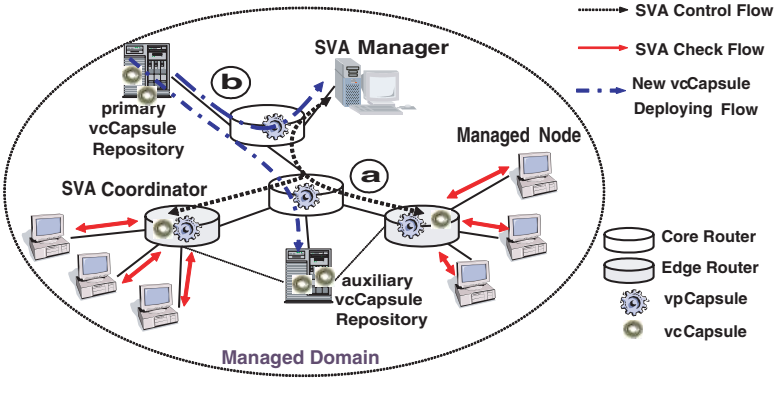


Fig. 3. The framework of the SVAM

SVA Coordinator The SVA Coordinator works on the edge router that is an active node in managed domain. The SVA Coordinator inspects vulnerabilities of managed nodes on subnet that are connected with itself, by policy received from the SVA Manager. When it wishes to check vulnerability for certain managed node, if *vc*Capsule does not exist in capsule cache, it receives *vc*Capsule from nearby the *vc*Capsule Repository.

***vc*Capsule Repository** In managed domain, several *vc*Capsule repositories exist. One of them gets 'primary' permission and the others get 'auxiliary' permission. The *vc*Capsule Repository keeps *vc*Capsule which checks vulnerability and provides the SVA Coordinator with *vc*Capsule. And the *vc*Capsule Repository is responsible for deploying new *vc*Capsule. The first deployment of new *vc*Capsule is done by the primary *vc*Capsule Repository.

The SVAM has three procedures: procedure for distributing policy, procedure for checking vulnerability, and procedure for deploying new *vc*Capsule. Figure 4 shows procedures for distributing policy and checking vulnerability. Process in the gray box shows when *vc*Capsule isn't in the cache, so the SVA Coordinator receives *vc*Capsule from the *vc*Capsule Repository for checking vulnerabilities towards the managed node. In the case of procedure for deploying new *vc*Capsule, when new *vc*Capsule is generated, it will be registered at primary *vc*Capsule Repository. If registration completes at the SVA Manager, the primary *vc*Capsule Repository deploys *vc*Capsule to auxilary Repositories in same domain. The Repository that received new *vc*Capsule registers to its *vc*CapsuleBase and deploys *vc*Capsule to nearest the Repository. By this process, new *vc*Capsule can be spread all over the managed domain. As shown above, workloads for all managed nodes that are concentrated on the SVA Manager, are distributed to the SVA Coordinator on edge router that is the nearest from each managed node. As the number of managed node increases, workload of the SVA Manager slowly increases by the SVA Coordinator that is naturally added. Also, fast deployment of information and checking code about newly discovered vulnerability by the *vc*Capsule Repository can provide enhanced adaptability in networks. The SVAM can solve problems in XVAMs and can expect improvement of network performance.

3.3 Capsules

As we described briefly in before Section, policy-distribution by the SVA Manager or vulnerability-checking by the SVA Coordinator is done by capsule. Capsule used in the SVAM is designed based on ANTS and has two types like following.

***vp*Capsule(vulnerability policy-distributing Capsule)** A policy is set on the basis of vulnerability. Policy is defined when will do some vulnerabilities check to some node. *vp*Capsule is a control capsule that contains policy information for each vulnerability. *vp*Capsule is generated by the SVA Manager and is transmitted to the SVA Coordinator.

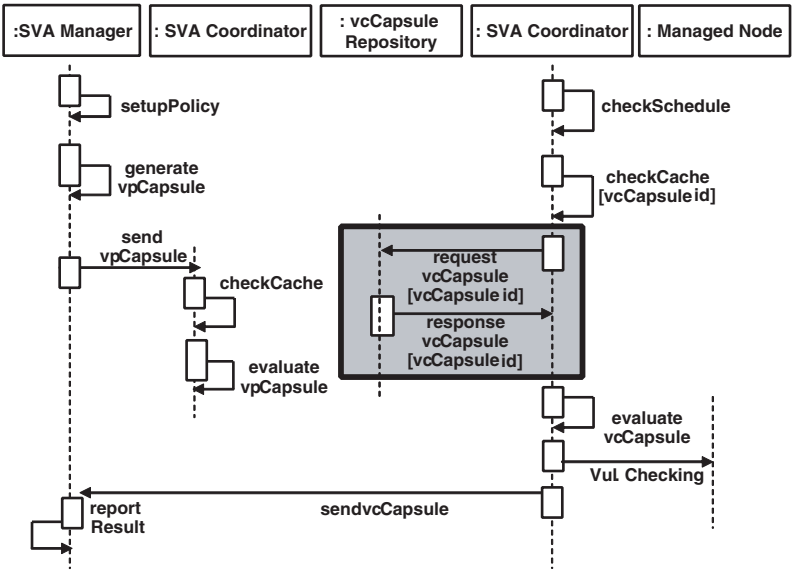


Fig. 4. A sequence diagram of deploying policy and checking vulnerability

***vc*Capsule(vulnerability-checking Capsule)** *vc*Capsule is a checking capsule that checks specific vulnerability at the SVA Coordinator. *vc*Capsule contains code that checks specific vulnerability. A kind of this code is following; port scanning, http vulnerability checking, and so forth.

4 Performance Evaluations

In this Section, we specify the performance evaluation model which evaluates XVAMs and the SVAM in the aspect of scalability. After that, evaluation result is followed.

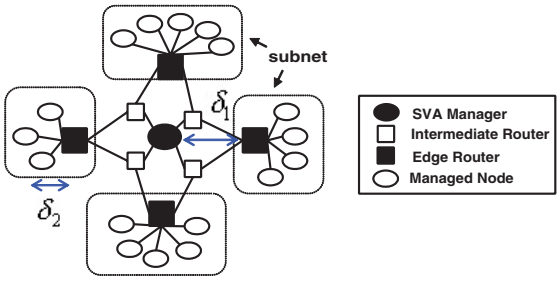


Fig. 5. The network model for evaluations

Table 1. A parameter for modeling and vaule of the parameter

Parameter	Description	Value
N_S	The number of subnet in the managed domain	3
N_N	The maximum number of managed node in one subnet	254
N	Total number of managed node	$N_S \times N_N$
R_i	The number of request of vulnerability checking to ith of managed node	10
L_V	Length of packet for checking vulnerability of OS finger print ($L_V = IP_{Header} + TCP_{Header}$)	40 bytes
$N_{capsule}$	Length of *vp*Capsule and *vc*Capsule ($L_{capsule} = IP_{Header} + UDP_{Header} + ANEP_{Header} + ANTS_{Header} + payload$)	100 bytes
δ_1	Network average delay from SVA Manager to edge router	100 ms
δ_2	Network average delay from edge router to managed node	20 ms
β	Network bandwidth	300 KB/s

Figure 5 shows network model for evaluation. Assumption is posed for modeling is followed.

Assumption 1 Network has equivalent network bandwidth among all nodes.

Assumption 2 $\delta_1 \geq \delta_2$

Assumption 3 The maximum number of managed node in one subnet is 254.

Assumption 4 In the case of the SVAM, *vc*Capsule was deployed by *vc*Capsule deploying mechanism in the SVA Coordinator for evaluation in advance.

Table 1 shows parameter used in the modeling. The parameter is defined for performance evaluations in mobile agent environment which is similar to active networks in the aspect of mobility of code [3].

In the case of XVAMs, for checking one vulnerability, a pair of request and response is occurred. If there is R_i request of checking vulnerability to ith of the managed node, traffic occurs with double of L_V. L_{XVAM} represents total network load which occurs in XVAMs. T_{XVAM} is consumption time on networks which spends to perform network traffic of L_{XVAM} in XVAMs.

$$L_{XVAM} = 2\sum_{n=1}^{N}\sum_{r=1}^{R_n} L_V, \qquad T_{XVAM} = 2\sum_{n=1}^{N}\sum_{r=1}^{R_n}(\delta_1 + \delta_2) + \frac{L_{XVAM}}{\beta} \tag{1}$$

Equation (1) represents L_{XVAM} and T_{XVAM}. Because request and response occurs double of R_i for managed nodes of N number, $\sum_{n=1}^{N}\sum_{r=1}^{R_n}(\delta_1 + \delta_2)$ delay is added to T_{XVAM}.

In the case of the SVAM, it must be separately considered that policy deploying-process from the SVA Manager to the SVA Coordinator and vulnerability-checking process from the SVA Coordinator to each managed node. LP_{SVAM} is a network load and TP_{SVAM} is consumption time for policy-deploying process; LV_{SVAM} is a network load and TV_{SVAM} is a consumption time for

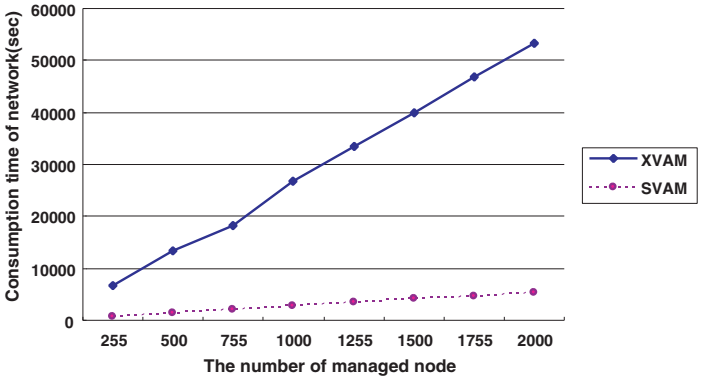


Fig. 6. Comparison of network consumption time regarding the number of managed node

vulnerability-checking process. Equation (2) represents total network load and total consumption time of the SVAM.

$$L_{SVAM} = LP_{SVAM} + LV_{SVAM}, \qquad T_{SVAM} = TP_{SVAM} + LV_{SVAM} \tag{2}$$

Policy is deployed to each SVA Coordinator of each managed node. The SVA Coordinator in every subnet is one. So, LP_{SVAM} and TP_{SVAM} are calculated as (3). Unlike XVAMs, $\sum_{n=1}^{N_S} \delta_1$ delay is added to TP_{SVAM}.

$$LP_{SVAM} = \sum_{n=1}^{N_S} L_{vpCapsule}, \qquad TP_{SVAM} = \sum_{n=1}^{N_S} \delta_1 + \frac{LP_{SVAM}}{\beta} \tag{3}$$

Equation (4) and (5) are used for calculating LV_{SVAM} and TV_{SVAM}. $N_S \left(\sum_{n=0}^{N_N} \sum_{r=1}^{R_N} (2L_V) \right)$ of LV_{SVAM} is workload of checking vulnerability from the SVA Coordinator to managed nodes. $N_S \sum_{r=1}^{R_n} L_{vcCapsule}$ of LV_{SVAM} is workload of returning vulnerability-checking result.

$$LV_{SVAM} = N_S \left(\sum_{n=0}^{N_N} \sum_{r=1}^{R_n} (2L_V) + \sum_{r=1}^{R_n} L_{vcCapsule} \right) \tag{4}$$

$$TV_{SVAM} = N_S \sum_{n=1}^{N_N} \sum_{r=1}^{R_n} \delta_2 + \sum_{n=1}^{N_S} \delta_1 + \frac{LP_{SVAM}}{\beta} \tag{5}$$

Finally, equation (6) and (7) are used for calculating total network load and total consumption time in the SVAM.

$$L_{SVAM} = \sum_{n=1}^{N_S} L_{vpCapsule} + N_S \left(\sum_{n=0}^{N_N} \sum_{r=1}^{R_n} (2L_V) + \sum_{r=1}^{R_n} L_{vcCapsule} \right) \tag{6}$$

$$T_{SVAM} = 2\sum_{n=1}^{N_S}\delta_1 + N_S\sum_{n=1}^{N_N}\sum_{r=1}^{R_n}\delta_2 + \frac{LP_{SVAM} + LV_{SVAM}}{\beta} \quad (7)$$

Comparing network consumption time of XVAMs with that of the SVAM by the number of managed node and vulnerability checking is depicted in Fig. 6. That is result of evaluation applying sample data to performance evaluation model. The sample data is listed *value* column in Table 1. As showed at Fig. 6, consumption time of the SVAM is fewer than that of XVAMs.

5 Conclusion and Future Works

This paper describes the SVAM as vulnerability analysis model which can manage in active networks actively and efficiently. The SVAM makes more adaptive and more scalable than XVAMs by distributing workload of the SVA Manager among SVA Coordinators.

Recently, we are implementing the SVAM in ANTS-based active networks. After completing implementation, we will work performance evaluation by various methods. In addition, because the SVAM is based on ANTS-based active network, it has limitation. In the future work of this study, not only ANTS, but also general SVAM which can support various EEs will be created.

References

[1] D. L. Tennenhouse, et al.: 'A Survey of Active Network Research', IEEE communications magazine, Jan. 1997. 857, 858
[2] D. Wetherall, et al.: 'ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols', IEEE OPENARCH'98 Proc., Apr. 1998. 859
[3] H. C. Kwon, et al.: 'A Performance Evaluation of Mobile Agent for Network Management', The Transaction of the KIPS, Vol.8-C, Num.1, Korea, Feb. 2001. 864
[4] Internet Security System: 'Internet Security Systems, Network and Host-based Vulnerability Assessment', Technical White Paper. 858, 860
[5] J. S. Yang, et al.: 'A Study on Security Vulnerability of Active Network', Proc. of the 19th KIPS Fall Conference, Korea, May 2003. 858
[6] K. Calvert, et al.: 'Architectural Framework for Active Networks', AN Working Group, July 1999. 858
[7] K. Psounis: 'Active Networks: Applications, Security, Safety, and Architectures', IEEE Communications Surveys, First Quarter, 1999. 858, 859
[8] S. L. Murphy: 'Secure Active Network prototypes', Proceedings of the DARPA Active Networks Conference and Exposition (DANCE'02), 2002. 859

On the Security Effect of Abnormal Traffic Controller Deployed in Internet Access Point

Kwangsik Kim[1], Taekyong Nam[1], and Chimoon Han[2]

[1] Information security research division, ETRI, Yusong
P.O. Box 106, Taejon, 305-600, Korea
{kks63453, tynam}@etri.re.kr
[2] Hankuk University of Foreign Studies, Korea
cmhan@hufs.ac.kr

Abstract. ATC (Abnormal traffic controller) is presented as next generation security technology to securely support reliable Internet service in traffic-intended unknown attack. The key concept of the ATC is abnormal traffic monitoring and traffic control technology. When fault factors exist continuously and/or are repeated, abnormal traffic control guarantees service completeness as much as possible. The ATC with control policy on abnormal traffic is superior to the ATC with blocking policy as well as conventional network node, when the ratio of effective traffic to abnormal traffic is higher than 30%. As the proposed ATC can be applied to the edge point of backbone network as well as Internet access point, it guarantees network survivability and provides reliable Internet service.

1 Introduction

Recently the trend of cyber attack is shifted from simple system attack to network attack that makes specific service stopped [1,2]. As an example, Internet worm made Internet service stopped in Jan. 25, 2003. Unknown attack is becoming current trend in global network environment. When unknown attack occurs, network nodes mainly make a decision that input traffic is not malicious but suspicious traffic. If most of abnormal traffic is the traffic made by malicious attack, it is meaningless to serve abnormal traffic. Therefore, we need a mechanism to control abnormal traffic.

In security aspects, representative researches on reliable Internet service are DARPA FTN (Fault tolerant network) [3] and Arbor Inc. Peakflow [4]. When Peakflow measures, aggregates and correlates security data, it depends on traffic analysis of the CISCO netflow. It can be only applied to the environment that CISCO routers exist.

In this paper, ATC (Abnormal traffic controller) is presented as next generation security technology to securely support reliable Internet service. The key concept of the ATC is abnormal traffic monitoring and traffic control technology. When fault factors exist continuously and/or are repeated, abnormal traffic control guarantees service completeness as much as possible.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 867–876, 2004.

This paper is organized as follows: next section presents a general description of the concept of the ATC. The tele-traffic model is presented in section 3 and performance measures in section 4. In section 5 some numerical results are presented that indicate the performance improvement achieved by the ATC scheme while in section 6 conclusions are given.

2 The Proposed Scheme

Recently unknown attack frequently happens and it will be usual in the future. In the case of unknown attack, security devices such as IDS, IPS and security appliance have false-positive decision about ISP ingress traffic from customer network.

Security device has a difficulty whether it makes blocking or passing abnormal traffic, because some of abnormal traffic is effective traffic, where the effective traffic means not a corrupted traffic but a real traffic generated by the Internet user. The proposed ATC as shown in Figure 1 has a security policy of soft firewall function on abnormal traffic to protect effective traffic among abnormal traffic.

Assume that the ATC is located nearby access network nodes such as DSLAM in ADSL technology. The ATC is a kind of front-end-processor and is operated as plug-in type in network node or standalone system. Consider that the ATC monitors the start time and the end time of virtual connection per a session. For session management in security node, Internet traffic can be modeled as Erlang loss formula like voice traffic.

Assume that the Packet Monitor can perform flow based packet monitoring. If a packet is doubted as a corrupted packet, it is considered as abnormal traffic. Giving the lower priority to the abnormal traffic controls abnormal traffic. Control policy can be blocking or rate limiting. By doing so, corrupted packets have a lower survivability and total throughput may be increased.

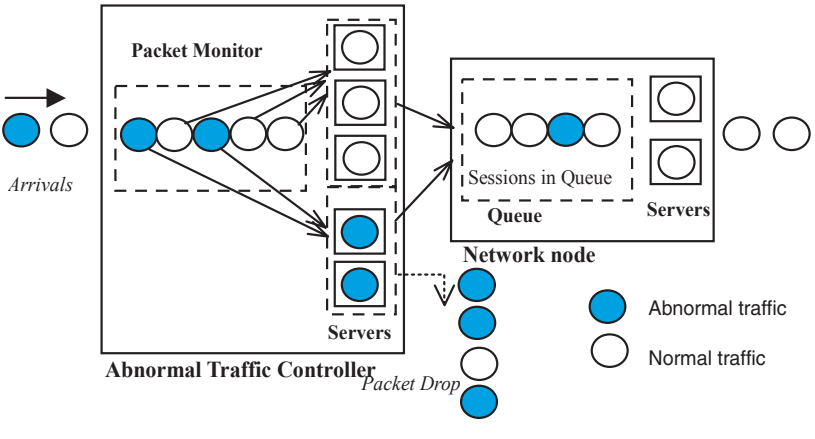


Fig. 1. ATC service model

3 Tele-traffic Model

In this section, the mathematical model of the proposed scheme is proposed. For quality guaranteed mission critical service, it can be usual to use virtual channel service in access network nodes in the future. And the sessions are served by virtual channel connection like telephone channel. In [5], the arrival events of WWW or FTP sessions within a WAN environment can be well modelled by a Poisson process with fixed hourly rates. It is considered that session duration time follows Pareto distribution as well as exponential distribution.

3.1 Arrival Process of a Session Event

Consider the ATC consisting of n channels with an infinite session caller population. If an arriving session finds all n channels busy, it does not enter the system and is lost instead. The arrival of a session is likely to approximate the statistics of a homogeneous Poisson process with fixed hourly rates. The Poisson process can be characterized as a renewal process whose inter-arrival times are exponentially distributed with rate parameter λ such that

$$F(t) = 1 - e^{-\lambda t}. \tag{1}$$

Then, successive new session attempts in the ATC is given by

$$\lambda_s = \lambda(1 - P_b). \tag{2}$$

Where P_b is the probability of new session blocking.

3.2 A Session Size and Session Duration Time

How can the file of a session be described statistically? To be statistical analysis, session size is converted to time coordinate. Assume that basic 1Mbits/s virtual channel is used and overhead of session such as packet header, packet acknowledgement, etc is 50%. Also assume that a session consist of 5 files. Then, Table 1 gives duration time of a session.

We assume that session duration times are independent, exponentially distributed random variables with parameter μ and independent of the session arrival process. For session duration time T_s , exponential distribution can be

Table 1. Session duration time

Size category	File size	Session size(5files)	Session duration time
Small	5 $\sim$ 100kbytes	25 $\sim$ 500kbytes	0.4 $\sim$ 0.8s
Medium	100 $\sim$ 1000kbytes	500 $\sim$ 5000kbytes	8 $\sim$ 80s
Large	1 $\sim$ 5Mbytes	5 $\sim$ 25Mbytes	80 $\sim$ 400s

considered owing to easiness of statistical analysis. That is, session duration time follows exponential distribution with mean $1/\mu$. The PDF is given by

$$f(t) = \mu e^{-\mu t}. \tag{3}$$

Where μ is session service rate.

More realistically, the overall statistics of data files are well approximated by the Pareto distribution [5,6]. In this paper, medium size of a session is considered similar to voice traffic.

3.3 Arrival Process of a Channel Failure Event and Repair Time

Assume that the times to channel failure and repair are exponentially distributed with mean $1/\gamma$ and $1/\tau$, respectively. Also assume that all channels share a single repair facility.

3.4 Abnormal Traffic

There are two factors on abnormal traffic to be considered in global network: the first is the ratio of abnormal traffic to total arrival traffic; and the second is the ratio of effective traffic to abnormal traffic. Define Q_{at} the ratio of abnormal traffic to total arrival traffic and Q_{ea} the ratio of effective traffic to abnormal traffic. To find Q_{at} and Q_{ea} is beyond the scope of this paper. In the following, the proposed ATC scheme is analyzed with the given value of Q_{at} and Q_{ea} to show the effect of the ATC scheme comparing with the conventional scheme. Assume that the inter-arrival time and service time of abnormal traffic session are exponentially distributed with mean $1/\zeta$ and $1/\nu$, respectively

4 Performance Measures

4.1 Erlang Loss Model for Conventional Network Node with Abnormal Traffic

In burst fault environment with malicious attack, performability composes of performance and the affection of abnormal traffic. Ref [7] showed the composite model for the combined performance and availability analysis and the state diagram. By using the composite model of Ref [7], we derive the proposed composite model for the combined abnormal traffic and performance analysis and the state diagram as shown in Figures 2 and 3. In our approach, a top-level abnormal traffic model (Figure 2) is turned into a Markov reward model (MRM), where the reward rates come from a sequence of performance models (Figure 3) and are supplied to the top-level abnormal traffic model.

The abnormal traffic in the Internet may happen as the result of malicious attack. The affection of abnormal traffic in network nodes is similar to availability model, because abnormal traffic consumes buffers. Abnormal traffic consumes

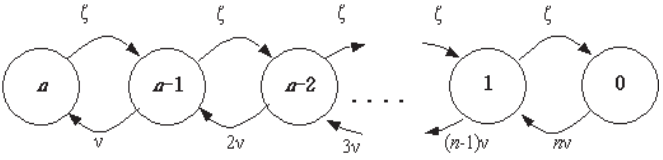


Fig. 2. State diagram for the Erlang loss abnormal traffic model

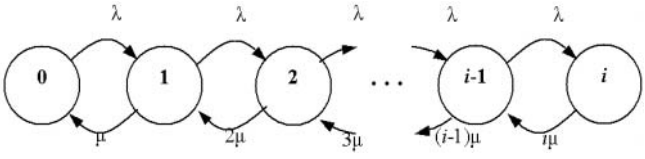


Fig. 3. State diagram for the Erlang loss performance model

system resources by assigning some of resources to serve abnormal traffic. Therefore, available resources for normal traffic are reduced.

Consider the network node with limited number of resources (or servers), n, in the resource pool. Hierarchical decomposition is used to obtain an approximate solution: we first present an upper level abnormal traffic (AT) model that accounts for the possible resource possession and releases. Finally, the two models are combined together and give performability measures of interest. The upper level AT model, as shown in Figure 2, describing the possession and release behavior of the system, is abnormal traffic model. Let $\psi_i (i \in 0, 1, 2, ..., n)$ be the steady-state probability of the CTMC being in state i of upper level abnormal traffic model. We know that

$$\psi_i = i!(\nu/\zeta)^i \psi_0,\ i = 1, 2, ..., n \tag{4}$$

where the steady-state system unavailability occurred by abnormal traffic:

$$U = \psi_0 = [\sum_{i=0}^{n} i!(\nu/\zeta)^i]^{-1} \tag{5}$$

Consider the performance model with the given number i of non-failed channels. The quantity of interest is the blocking probability, that is, the steady-state probability that all buffers are busy, in which case the arriving session is refused service. The performance model of this system is an $M/M/i$ loss system, and the state diagram is shown in Figure 3. In this model, the blocking probability with i channels in the system is given by

$$P_b(i) = \frac{\frac{(\lambda/\mu)^i}{i!}}{\sum_{j=0}^{i} \frac{(\lambda/\mu)^j}{j!}} \tag{6}$$

Attach a reward rate r_i to the state i of the abnormal traffic model as the blocking probability with i channels in the system, that is $r_i = P_b(i), i \gg 1$ and $r_0 = 1$. Then the required total blocking probability in abnormal traffic condition can be computed as the expected reward rate in the steady-state and is given by

$$\hat{Y}_b = \sum_{i=0}^{n} r_i \psi_i = \psi_0 + P_b(n)\psi_n + \sum_{i=1}^{n-1} P_b(i)\psi_i \tag{7}$$

where ψ_i is the steady-state probability in abnormal traffic condition that i non-failed channels are there in the system.

The total loss probability expression above can be seen to consist of three summands: the first part is system unavailability U happened by abnormal traffic, the second part is the session blocking probability due to buffer full weighted by the probability that the system is up and the bracketed part on the right-hand side of Equation (7) is the buffer full probability in each of the degraded states weighted by the probability of the corresponding degraded state in abnormal traffic condition.

4.2 Erlang Loss Model for the ATC with Controlling Abnormal Traffic

The Erlang loss formula in Section 4.1 cannot be applied to the ATC with the abnormal traffic control function. In this section, we discuss a two level hierarchical performability model for the ATC with abnormal traffic control function. In burst fault environment with malicious attacks, performability composes of performance, availability and the affection of abnormal traffic control. Then, the composite model for the combined ATC and performance analysis and the state diagram is shown in Figure 4. The abnormal traffic in the Internet may happen as the result of malicious attack. The effect of abnormal traffic control in the ATC is similar to performance model rather than availability model, because some of abnormal traffic is survived as normal traffic. Abnormal traffic control consumes system resources by assigning some of resources to serve abnormal traffic. Therefore, available resources for normal traffic are reduced. To include abnormal traffic model, performance model in Figure 3 can be modified as in Figure 4. Here, the reservation scheme, that is, giving a priority to normal traffic is used. r is the number of reserved channels and n is total number of channels.

For the simplicity, we consider that the service rates of both normal and abnormal traffic are the same. Consider the performance model with the given number i of non-failed channels. The quantity of interest is the blocking probability, that is, the steady-state probability that all buffers are busy, in which case the arriving session is refused service. Note that in this performance model, the assumption is that blocked sessions are lost (not re-attempted). The performance model of this system is an $M/M/i$ loss system, and the state diagram is shown in Figure 4. The blocking probability with i channels in the system with ATC function is given by

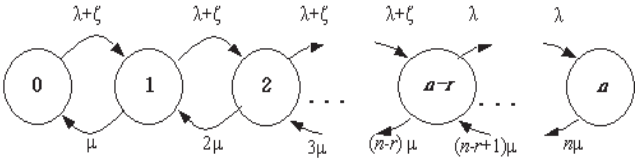


Fig. 4. State diagram for the Erlang loss composite model (normal + abnormal traffic)

$$S_b(i) = \begin{cases} \dfrac{\frac{((\lambda+\zeta)/\mu)^i}{i!}}{\sum_{j=0}^{i} \frac{((\lambda+\zeta)/\mu)^j}{j!}}, \ i = 0, 1, 2, ..., n-r \\ \dfrac{\frac{((\lambda+\zeta)/\mu)^{n-r}(\lambda/\mu)^{i-n+r}}{i!}}{\sum_{j=0}^{n-r} \frac{((\lambda+\zeta)/\mu)^j}{j!} + \sum_{j=n-r+1}^{i} \frac{((\lambda+\zeta)/\mu)^{n-r}(\lambda/\mu)^{j-n+r}}{j!}}, \ i = n-r+1, ..., n \end{cases} \tag{8}$$

In this model, the steady-state probability π_i for the number i of non-failed channels in the system [7] is given by

$$\pi_i = \frac{(\tau/\gamma)^i \pi_0}{i!}, \ i = 1, 2, ..., n \tag{9}$$

where the steady-state system unavailability is derived by

$$U = \pi_0 = [\sum_{i=0}^{n} \frac{(\tau/\gamma)^i}{i!}]^{-1} \tag{10}$$

Attach a reward rate r_i to the state i of the availability model as the blocking probability with i channels in the system, that is $r_i = S_b(i)$, $i \geq 1$ and $r_0 = 1$. Then the required total blocking probability $\hat{W}_{nb}$ of normal traffic can be computed as the expected reward rate in the steady-state and is given by

$$\hat{W}_{nb} = \sum_{i=0}^{n} r_i \pi_i = \pi_0 + S_b(n)\pi_n + \sum_{i=1}^{n-1} S_b(i)\pi_i \tag{11}$$

The total loss probability of normal traffic can be seen to consist of three summands: the first part is system unavailability U by failure-repair; the second part is the session blocking probability due to buffer full weighted by the probability that the system is up, where buffer is used by both normal traffic and abnormal traffic; and the bracketed part on the right-hand side of Equation (11) is the buffer full probability in each of the degraded states weighted by the probability of the corresponding degraded state.

The required total blocking probability $\hat{W}_{ab}$ of abnormal traffic can be computed as the expected reward rate in the steady-state and is given by

$$\hat{W}_{ab} = \sum_{i=0}^{n-r} r_i \pi_i + \sum_{i=n-r+1}^{n} r_i \sum_{i=n-r+1}^{n} \pi_i$$
$$= \pi_0 + S_b(n-r)\pi_{n-r} + [\sum_{i=1}^{n-r-1} S_b(i)\pi_i + \sum_{i=n-r+1}^{n} S_b(i) \sum_{i=n-r+1}^{n} \pi_i]. \quad (12)$$

The total loss probability of abnormal traffic can be seen to consist of three summands: the first part is system unavailability U by failure-repair; the second part is the session blocking probability due to $(n-r)$ buffer full weighted by the probability that the system is up, where buffer is used by both normal traffic and abnormal traffic; and the bracketed part on the right-hand side of Equation (12) is the buffer full probability in the sum of the degraded states weighted by the probability of the corresponding degraded state under maximum $(n-r)$ buffer available for abnormal traffic.

4.3 Effective Traffic Throughput

In performance aspect, effective traffic throughput is important. In this section, we derive effective traffic throughput between controlling and blocking abnormal traffic. The effective new session attempts λ_{ec} in the ATC with control policy is given by

$$\lambda_{ec} = \lambda + \zeta Q_{ea} \quad (13)$$

Then, effective new session attempts λ_{eb} in the ATC with blocking policy is given by

$$\lambda_{eb} = \lambda \quad (14)$$

In the same arrival rate, effective traffic of the ATC with blocking abnormal traffic is lower than that of the ATC with controlling abnormal traffic, even though the required blocking probability is reversed. The ratio E_{bc} of effective throughput between con-trolling and blocking abnormal traffic is given by

$$E_{bc} = \frac{\lambda_{eb}}{\lambda_{ec}} = \frac{\lambda}{\lambda + \zeta Q_{ea}} \quad (15)$$

5 Numerical Results

In this section, numerical examples for the ATC are shown. As performance measures, the required blocking probabilities are derived for each of conventional network node with abnormal traffic and the ATC with controlling abnormal traffic.

Assumptions are given as follows: The mean new session attempts rate $\lambda = 0.1 \sim 1.0$ sessions/sec; the mean session duration time $1/\mu = 100$ sec/session; no. of channels $n = 20$; no. of reserved channels $r = 2$; session duration time

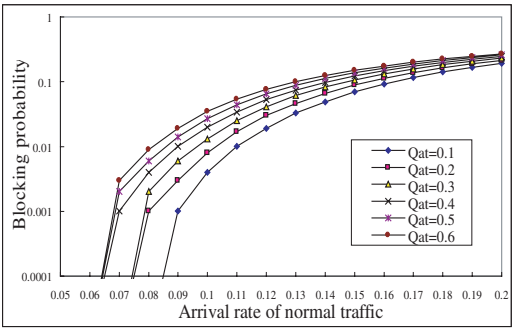


Fig. 5. The required blocking probabilities of the ATC as arrival rate increases

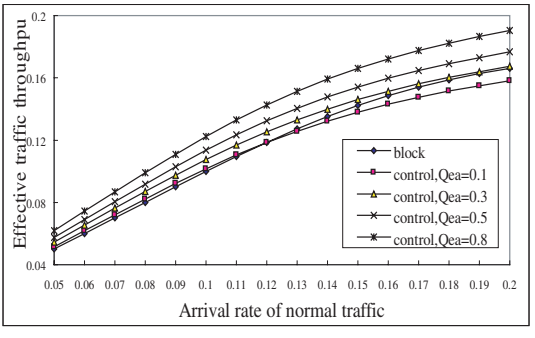


Fig. 6. The effective traffic throughput of the ATC with blocking policy and control policy in varying Q_{ea} of 0.1, 0.3, 0.5 and 0.8

follows exponential distribution; the ratio Q_{at} of abnormal traffic to total arrival traffic = 0.3; and the ratio Q_{ea} of effective traffic to abnormal traffic = 0.8; the times to channel failure and repair $1/\gamma = 10000$ (53 minuts/year) and $1/\tau = 1/\mu$, respectively; the inter-arrival time and service time of abnormal traffic session $1/\zeta$ and $1/\nu$ are the same as those of normal traffic session, respectively. Mathematica V4.2 package was used for numerical calculation [8].

Figure 5 shows required blocking probabilities of the ATC with abnormal traffic as arrival rate of normal traffic increases. Here, the ratio Q_{at} of abnormal traffic to total arrival traffic is considered in the interval of $0.1 \sim 0.6$. As Q_{at} increases, blocking probability increases in the same arrival rate of normal traffic. But, the difference between blocking probabilities in different Q_{at} is reduced quite in comparison with conventional network node. For an example, when arrival rate of normal traffic is 0.15, blocking probabilities in Q_{at} of 0.1, 0.3 and 0.6 are 0.07, 0.107 and 0.150, respectively.

Figure 6 shows the comparison of the effective traffic throughput of the ATC with blocking policy and control policy on abnormal traffic. Here the assumption

is the same as above, except for the ratio Q_{ea} of 0.1 $\sim$ 0.8. As Q_{ea} increases, effective traffic throughput increases in the same arrival rate of normal traffic. For an example, when arrival rate of normal traffic is 0.15, effective traffic throughput in blocking policy and control policy with Q_{ea} of 0.1, 0.3, 0.5 and 0.8 are 0.142, 0.138, 0.146, 0.154 and 0.166, respectively. If Q_{ea} is higher than 0.3, the ATC with control policy is better than the ATC with blocking policy. Else the ATC with blocking policy is better.

6 Conclusions

The ATC is presented as next generation security technology to securely support reliable Internet service in traffic-intended unknown attack. When fault factors exist continuously and/or are repeated, abnormal traffic control guarantees service completeness as much as possible. In blocking probability and effective traffic throughput aspects, as shown in numerical results, the ATC with control policy on abnormal traffic is superior to the ATC with blocking policy as well as conventional network node.

In the future, the network attack response technology such as the ATC will gradually apply to the edge point and access point of networks. That is because users cannot have all security functions in their systems and also ISPs have to guarantee users for quality such as SLA (Service level agreement) on Internet infrastructure providing e-services.

References

[1] J.Pescatrore, M. Easley, R.Stiennon, "Network security platform will transform security markets," Gartner, Nov. 2002.
[2] "State of the NGN : Carriers and vendors must take security seriously," Gartner, March 2003.
[3] DARPA FTN, http://www.iaands.org/iaands2002/ftn/index.html.
[4] Arbor Inc., peakflow, $http://www.arbornetworks.com/products_platform.php$.
[5] Vern Paxson, Sally Floyd, "Wide-area traffic: The failure of poisson modeling," IEEE/ACM Transaction on networking, 3(3), pp.226-244, June 1995.
[6] K. S. Kim, M. H. Cho and T. Y. Nam, "Analysis of Session Admission Control based on Area (SACA) for Software Download in Cellular CDMA Systems," ICOIN'2003 Feb. 2003.
[7] Kishor S. Trivedi, Xiaomin Ma and S. Dharmaraja, "Performability modeling of wireless communication systems," Int. Journal of communication systems, pp.561-577 May 2003.
[8] Wolfram Inc., Mathematica V4.2, http://www.wolfram.com.

Design of Traceback System Using Selected Router

Jeong Min Lee, In Gu Han, and Kyoon Ha Lee

Department of Computer Science and Engineering, Inha University at 253 Yonghyun-dong
Nam-gu, Incheon 402-751, Korea
verion@nate.com
inguhan@aiblue.inha.ac.kr
khlee@inha.ac.kr

Abstract. As increasing of Internet user and fast development of communication, many security problems occur. Because of Internet is design and development for speed not security, it is weak to attack from malicious user. Furthermore attack is more developed to have high efficiency and intelligent. We proposed effective traceback system in network and consider that ability of constitution. Traceback by Selected Router system is consists of managed router and manager system. Selected router marks router ID to packet which passing selected router, and use this router ID for traceback and filtering. Consequently this system reduces damage of attack.

1 Introduction

This paper describes a traceback system for tracing anonymous attacks in the Internet back to their source effectively. This work is motivated by the increased frequency and sophistication of Denial-of-Service attacks and by the difficulty in tracing packets with incorrect or "spoofed" source addresses by IP Spoofing attack [2].

In this paper we describe a traceback system bases on managed router and has two phases for traceback. Our approach allows identify the attack paths traversed by an attacker without requiring interactive operational support from Internet Service Providers (ISPs). Moreover, this traceback can reduce damage of attack by filtering method in managed router.

In the following Chapter we describe previous works in the area of attack traceback. Chapter 3 describes Traceback system by Selected Router (TBSR) and comparative results appear in Chapter 4. Finally, we conclude in Chapter 5.

2 Related Works

As increasing of Internet user and fast development of communication, many security problems are occurred. Denial of Service attack (DoS) is one of representative problems. DoS makes whole system resources are exhausted or monopolized. Consequently, it prevents victim system from offering pertinent service

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 877–885, 2004.

to other user [1, 3]. A serious problem to fight these DoS attacks is that attack systems use incorrect or spoofed IP address in the attack packets and hence disguise the real origin of the attacks. Due to the stateless nature of the Internet, it is difficult problem to determine the source of these spoofed IP packets, which called the IP traceback problem.

For solve this problem, victim system finds indication of attack as fast as he can and processes countermeasure for offering pertinent service to users. But DoS paralyzes system by large number of packet, passive methods like Intrusion Detection System (IDS) or Intrusion Prevention System can not defend victim system from attack.

2.1 Node Append, Node Sampling, Edge Sampling, Fragment Marking Scheme Algorithm

These methods proposed for tracking attack system which uses Spoofed IP address. As attack systems use incorrect or Spoofed IP address, it is hard to find source. One promising solution, proposed by Savage et al., is to let routers probabilistically mark packets with partial path information during packet forwarding. This solution proposed four different algorithms [7].

Node Append Algorithm is simplest marking algorithm. It appends related router's IP address to end of packet. However, Node Append Algorithm has critical limitation that needs additional space and has overhead to router [7, 8].

Node Sampling Algorithm uses probabilistically methods for solving overhead of router. It sets up probability p, and generates random number x. And if x is smaller than p, router marks his IP address to packet. However, Node Sampling Algorithm needs many amount of packet for reconstructing attack path, because of Sampled IP address arrived to victim system. And multiple attack systems which have same distance from victim system can exist. In this case this method can not be distinguished [7, 8].

Edge sampling algorithm is to write edge information into the packets. This scheme reserves two static fields of the size of IP address, start and end, and a static distance field in each packet. Each router updates these fields as probabilistically method. This algorithm is strong to multiple attack system, but needs more packets to make attack path [7, 8].

Fragment Marking Scheme (FMS) encoding scheme splits each router's IP address and redundancy information into eight fragmentation. This encoding scheme works well with just a single attack system. But in case of a distributed Denial-Of-Service attack, FMS suffers from high computation overhead, because it needs to check a large number of combinations of the fragment [7, 8].

2.2 Advanced Marking Scheme (AMS) Algorithm

e efficient than FMS. If victim system knows the map of its upstream routers, it does not need the full IP address in the packet marking to reconstruct the attack path, and hence the marking scheme can be more communication and

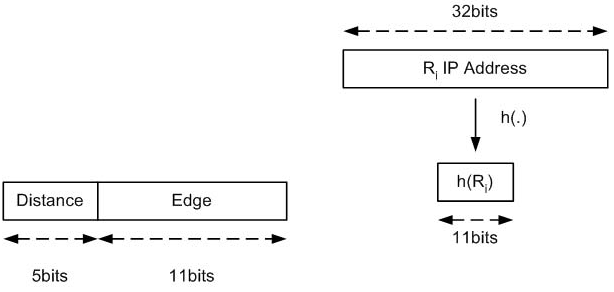


Fig. 1. Encoding in Advanced Marking Scheme I

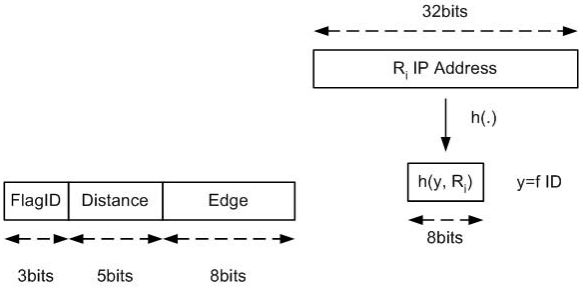


Fig. 2. Encoding in Advanced Marking Scheme II

computation efficient. Edge Sampling or FMS algorithms use the IP address for marking edge information, but AMS algorithm uses hash function value.

It is distinguish AMS-I from AMS-II by difference of hash function. AMS-I algorithm divides 16bits ID field to 11bits hash value for router identification and 5bits distance value. AMS-II algorithm is suggested for solving collision of hash function problem. It divides 16bits ID field to 8bits hash value for router identification, 5bits distance value and 3bits hash function classification [8]. In figure 2, 'FlagId' classify hash function collection which apply to packet.

However these algorithms should know the map of its upstream routers. If some routers are changed, it must reflect this effect to hash function.

3 Traceback by Selected Router (TBSR)

Marking based traceback methods are proposed for solving Spoofed IP address problem. But these methods have several limitations. First they need time to gather router information for reconstructing attack path. Second if attack systems try to fake trackback system, they can not distinguish. Third, they know the map of its upstream routers or control routers which exist between victim system and attack system. At last, they generally try to reconstruct attack path in victim system, so it gives another overhead to victim system.

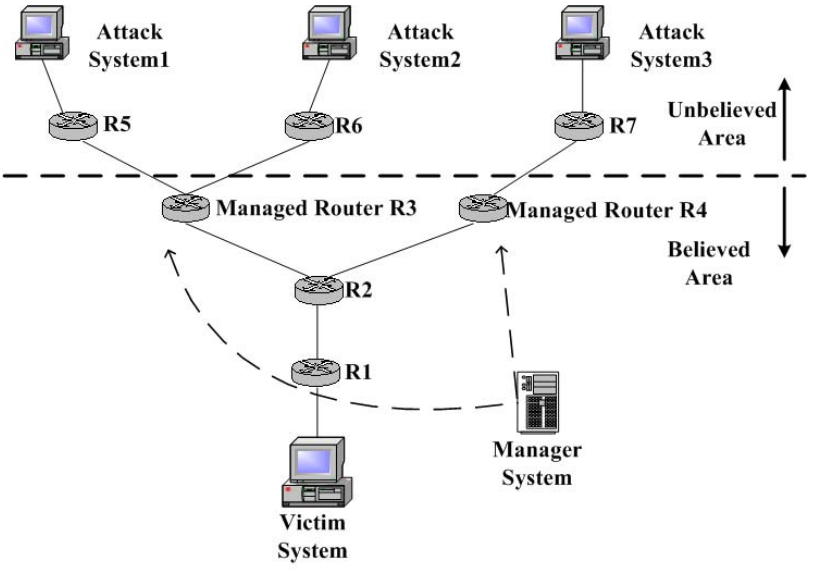


Fig. 3. Organization of TBSR

3.1 Overview

In this section, we describe our Traceback system by Selected Router. This traceback system divide network into two parts and execute traceback operation by two phases. We observe that network can be divided two areas, one is able to manage routers and the other is not able to manage routers.

We divide traceback operation to two phases, victim system – managed router and managed router – attack system. As attack systems send to attack packet to victim system, existing marking schemes execute traceback operation in victim system or IDS. In Figure 3, victim system tries to reconstructs attack path, R1 → R2 → R3 → R5 → AS1. The other side, proposed TBSR can execute traceback operation in R3 and R3 tries to reconstructs attack path R5→AS1. It is possible by managing router R3, R4. It can reduce requiring time and effort to reconstruct attack path. And it can disperse attack packet to managed routers, it can reduce overhead of DoS attack and reconstructing computation. At last if managed routers have filtering ability, it does not deliver attack packet to victim system. It can reduce damage of DoS attack.

3.2 Basic Idea

Proposed TBSR consists of managed routers and manager system. Managed routers do traceback operations and marking router ID to packet which are passing managed routers. These routers make man- aged area like Figure 3 by encircling area. Every packets from outside managed area, are must pass the

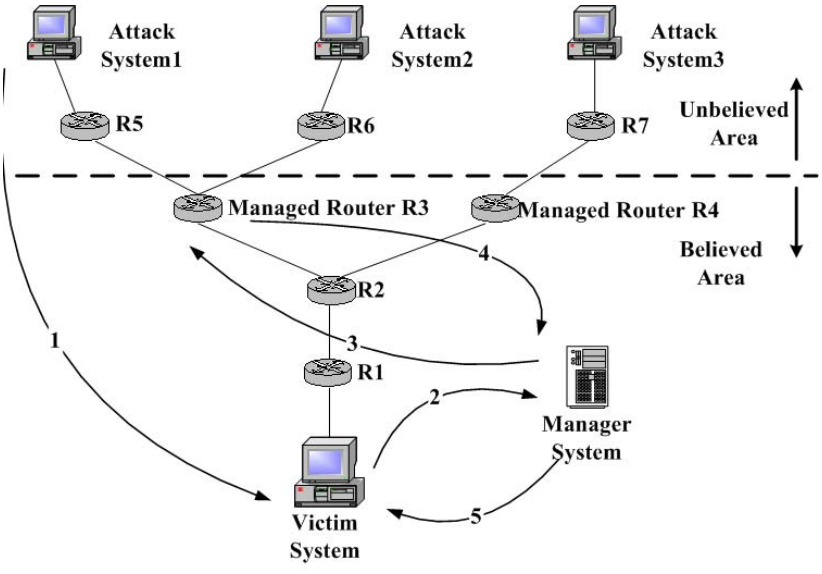


Fig. 4. Process of traceback in TBSR

managed router and managed router marks its router ID to packet's Identification field of IP header. Consequently victim system knows one router which marks router ID to packet. Managed router marks every passing packet, so that its overhead is smaller than probability methods.

Manager system administers router ID, managed router and traceback operations. As attack system A1 attacks victim system like Figure 3, victim system perceives this attack and sends traceback request to manager system with attack information.

- IP address of victim system
- Router Identification value inside attack packet's IP header
- Source address inside attack packet's IP header

After manager system receives these information, it orders managed router which have same router ID, filtering and doing traceback operation with attack information. It prevents additional attack packets arriving to victim system. After that this managed router do traceback operation by traditional algorithm, it can reduce distance between traceback system and attack system. Consequently it can reduce time and effort to find attack system. In proposed system, we suppose using Edge Sampling Algorithm for reconstructing attack path.

4 Performance Evaluation

The directed acyclic graph (DAG) rooted at V in figure 4 represents the network as seen from a victim system V and a distributed DoS attack from A1. V could

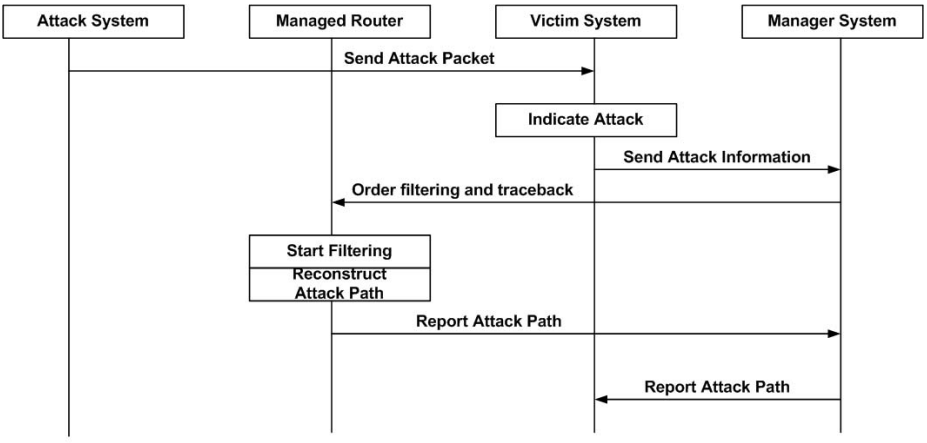


Fig. 5. Time Process of traceback in TBSR

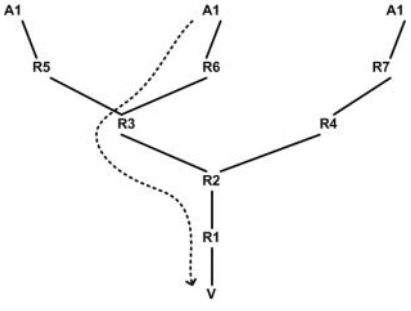


Fig. 6. Directed Acyclic Graph (DAG)

be either a single host under attack or a network border device such as a firewalls representing many such hosts. Node Ri represents the routers, which we refer to as upstream routers from V [9].

For each attack path with distance d and each router marks the packet with a probability p, the expected number of packets needed to reconstruct the path is,

$$E(X) = \frac{1}{p(1-p)^{d-1}}. \tag{1}$$

Because the probability of receiving a sample is geometrically smaller the further away it is from the traceback system, the time for marking algorithm to converge is dominated by the time to receive a sample from the furthest router in expectation.

However, there is a small probability that traceback system receive a sample from the furthest router, but not from some nearer router. We can bind this effect to a factor of ln(d). It can conservatively assume that samples from all

of the d routers appear with the same likelihood as the furthest router. Since these probabilities are disjoint, the probability that a given packet will deliver a sample from some router is at least,

$$p(1-p)^{d-1}. \tag{2}$$

Finally, as per the well known coupon collector problem, the number of trials required to select one of each of d equi-probable items is,

$$d(\ln d + O(1)). \tag{3}$$

Therefore the number of packets X, required for the traceback system to reconstruct attack path has the following bounded expectation [9].

$$E(X) < \frac{\ln d}{p(1-p)^{d-1}}. \tag{4}$$

In Figure 7, increasing required packets for reconstruct attack path, as distance between traceback system and attack system faraway.

Proposed TBSR, managed router does traceback operation, but not victim system. Consequently it can reduce distance between traceback system and attack system. The distance between victim system and managed router α, the expected number of packets needed to reconstruct the path is,

$$E(X) < \frac{\ln (d-\alpha)}{p(1-p)^{d-1-\alpha}}. \tag{5}$$

In Figure 8, probability p is 0.2 and distance between victim system and managed router α is 3. Distance between traceback system and attack system is 10 to 30, since few path exceed 25[10, 11, 12]. As Figure 8 shows, proposed TBSR needs smaller number of packets for reconstructing attack path, because it reduce distance between traceback system and attack system.

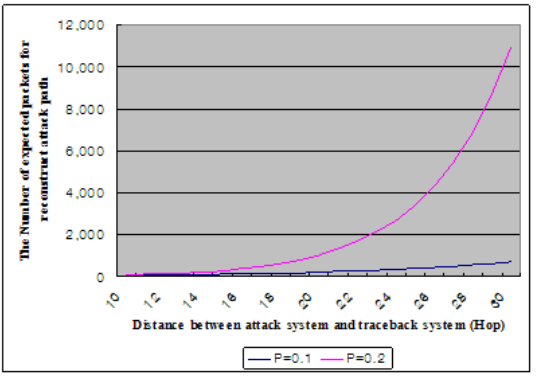


Fig. 7. The number of expected packets for reconstruct attack path in Edge Sampling

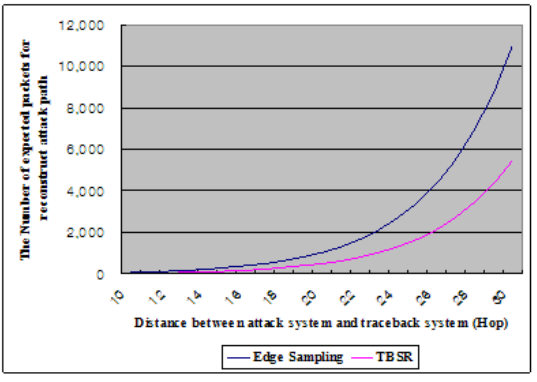


Fig. 8. The number of expected packets for reconstruct attack path (P=0.2, α=3)

5 Conclusion

DoS attack is simplest and most harm attack in internet environment. Internet speed and Computer performance highly increase, but it is hard to prevent DoS attack. The fundamental measurement of DoS attack is not defense attack, but it is finding source of attack and prevent re-attack from this attack system. But present IP technology has not authentication and information about sender, so that it is hard to find source of attack.

In this paper, we proposed new Traceback system by Selected Router for finding source of attack efficiently and decreasing damage of DoS attack. It consists of two phases, victim system – managed router and managed router – attack system. Managed routers encircle the safety area and mark its router ID to every packet through inside area. Manager system identify managed router by this router ID and select this router. Selected router does traceback and filtering operations by manger system order. It can reduce overhead of DoS attack and required packet number for reconstructing attack path. We will study more about extension this system, optimal number of managed router and preparation to IP v.6 environment. References

References

[1] Computer Emergency Response Team (CERT), "CERT Advisory CA-2000-01 Denial-of-Service developments," http://www.cert.org/advisories/CA-2000-01.html, Jan. 2000

[2] Computer Emergency Response Team (CERT), "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks," http://www.cert.org/advisories/CA-1996-21.html, Nov. 29, 2000

[3] Computer Emergency Response Team (CERT), "CERT Advisory CA-2003-04 MS-SQL Server Worm," http://www.cert.org/advisories/CA-2003-04.html, January 27, 2003

[4] David A.Curry, "Unix System Security," Addison Wesley, pp36-80, 1992 5. R. Stone "CenterTrack: An IP Overlay Network for Tracking DoS Floods," In to appear in Proceedings of the 2000 USENIX Security Symposium, Denver, CO, July, 2000
[5] R. Stone "CenterTrack: An IP Overlay Network for Tracking DoS Floods," In to appear in Proceedings of the 2000 USENIX Security Symposium, Denver, CO, July, 2000
[6] S. M. Dellovin, "The ICMP Traceback Messages," Internet Draft: draft-bellovin-itrace-00.txt, http://www.research.att.com/~smb, Mar. 2000
[7] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, "Practical network support for IP traceback," in Proc. of ACM SIGCOMM, pp295 306, Aug.2000
[8] Dawn Xiaodong Song and Adrian Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," in Proc. IEEE INFOCOM, Apr. 2001
[9] W.Feller, "An Introduction to Probability Theory and Its Applications (2nd edition)," volume 1. Wiley and Sons, 1966
[10] R.L Carter and M. E. Crovellam, "Dynamic Server Selection Using Dynamic Path Characterization in Wide-Area Networks," In Proc. of the 1997 IEEE INFOCOM, Kobe, Japan, Apr. 1997
[11] W. Theilmann and K. Rothermel, "Dynamic Distance Maps of the Internet," In Proc. 2000 IEEE INFOCOM, Tel Aviv, Israel, Mar 2000
[12] "Cooperative Association for Internet Data Analysis. Skitter Analysis," http://www.caida.org, 2000

Construct Efficient Hyper-alert Correlation for Defense-in-Depth Network Security System*

Nen-Fu Huang[1,2,3], Hsien-Wei Hung[2], Chia-Nan Kao[2], Gin-Yuan Jai[1], and Yi-Ju Sung[2]

[1] Department of Computer Science, National Tsing Hua University Taiwan, ROC
[2] Institute of Communication Engineering, National Tsing Hua University Taiwan, ROC
[3] Broadweb Corp., Hsin-Chu Industrial Science Park Hsin-Chu, Taiwan, ROC nfhuang@cs.nthu.edu.tw

Abstract. The current intrusion detection systems faced the problem of generating too many false alerts. The raising alerts are too elementary and do not accurate enough to be managed by a security administrator. Several alert correlation techniques have been proposed to solve this problem, such as hyper-alert correlation. The hyper-alert correlation takes advantage of the prerequisites and consequences of the attack to correlate the related alerts together. But the performance of this approach highly depends on the quality of the modeling of attacks. On the other hand, with growing of the network attacks, specifying the relationship for alert correlation would be quite complex and tedious task to perform mutually. This paper presents a practical technique to address this issue for hyper-alert correlation. On the basis of the attack signatures and the hyper-alert types defined in hyper-alert correlation, the proposed approach constructs alert relationship automatically. Furthermore, to take the various kinds of attacks into consideration, some of the relationships between attacks may be neglected. At this time, fine tuning the relationship by human user can efficiently deal with the above problem.

1 Introduction

As the Internet become more widespread and advanced, there is a higher risk of accidents, attacks, and failure. In the recent years, intrusion detection products have become widely available and used by many enterprises from the viewpoint of security. An intrusion detection system does exactly as the name which means it detects possible intrusions. More specifically, IDS aims to detect computer attacks and/or computer misuse, and to report the proper individuals upon detection. Intrusion detection is the process of monitoring computers or networks for unauthorized entrance, activity, or file modification. IDS can also be used to

* This work was partially supported by the MOE Program for Promoting Academic Excellence of Universities under Grant 89-E-FA04-1-4.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 886–894, 2004.

monitor network traffic, thereby detecting if a system is being targeted by a network attack such as a denial of service (DoS) attack.

Traditional intrusion detection systems mostly focus on low level attacks and anomalies. It will raise alerts independently even if the alerts have the relative connection. Inevitably, they often generate many false positives and false negatives. Therefore, it is necessary to have the techniques to help to improve the accuracy and quality of alerts. The ideal alert should report the real malicious behaviors of the intruder and inform the security administrator to take proper countermeasures to deal with the problem.

As time goes on, the network intrusions are getting more and more sophisticated and organized. Most intrusions are not isolated, but related as different stage of attacks. In other words, the intrusions with early stages will prepare for the later attacks. The hyper-alert correlation [1,2,3] approach constructs attack scenario on the basis of *prerequisites* and *consequences*of attacks. The prerequisites of an intrusion are the necessary condition for an intrusion to be successful. The consequences of an intrusion are the possible result of the intrusion. It presents a practical technique to correlate alerts and can reveal the attack strategies of intrusion. However, the quality of modeling the attacks dominates the performance of alert correlation and this approach still need the people to describe the relationship of all attacks. It is not friendly for the human to specify the connections between the attacks and inevitably people will miss some characteristics of the attacks. In this paper, we address the limitations of hyper-alert correlation and present an automatic technique to construct alert relationship for hyper-alert correlation.

As a result, it is difficult for human users to realize all of the attacks and even to model the attacks. Therefore, we present a practical method to reduce the burden of modeling. We do not change the alert correlation approach but take advantage of the prerequisites and consequences of attacks to construct attack scenario. As the CERT overview incident and vulnerability trends [11] described, the types of network intrusions are limited. We can classify the network intrusions into different types according to the impact of intrusions through this discovery. As previously described, most IDS are signature based and include several signatures for collection of known attacks. These signatures of an attack contain the necessary conditions for this attack to be successful. We can define the template relationship of the attack according to the signatures and the end result type of the attack. Then the prerequisites and consequences of this attack can be generated automatically. The template relationship still not perfect, because of the diversity of attacks. Therefore, we make use of the implicit correlation and explicit correlation to help human users to modify the relationship more completely.

2 Related Work

There are several alert correlation approaches have been proposed. These methods can be roughly separated into three classes. The first class correlates alerts

based on the similarities between alert attributes. For instances, the probabilistic alert correlation [6] considers the similarity of alerts and correlate them together if they have common feature overlap. Such features include the source of the attack, the target of the attack, the port of attack (source and target), the class of the attack, and time information. Though it is very effective for some alerts with same features, it can not fully discover the casual relationship between related alerts.

The second class correlates alerts based on the attack scenarios specified by the people or learned through training datasets. Several languages have been proposed to represent attacks, including LAMBDA [7], STAT [8], MuSig [9], and etc.

The third class correlates alerts based on the pre-conditions and post-conditions of individual attacks. It correlates alerts if the post-conditions of earlier alert are satisfied with the pre-conditions of the later alert. It can reveal the casual relationship between alerts and does not depend on the pre-defined attack scenarios. The Hyper-alert Correlation [1,2,3] and the MIRADOR project [5,6] both belong to this class.

In particular, processing of MIRADOR alert correlation is regard as different independent functions while the hyper-alert correlation allows the alert aggregation during the alert correlation. This convenience makes Hyper-alert Correlation approach easily and practically to achieve.

3 A Framework for Alert Correlation

Figure 1 shows the alert correlation architecture that we suggest to be developed for intrusion detection. The main objective of IDS cooperation is to correlate the related alerts in order to expose a more condensed view of current observations and then use the plan recognition to predict the next possible action of attackers. If the plan recognition can figure out the candidate plan of the attackers, the attack prevention will try to direct the underlying IDS to take proper countermeasures to prevent the occurrence of more advanced malicious actions. And the alert aggregation is designed to assist the IDS to detect the scanning attempts of attackers.

This paper presents the works for the purpose of preparing alert correlation. As mentioned before, it would be tedious for the human users to specify the appropriate prerequisites and consequences of attacks. We propose the template correlation, implicit correlation, and explicit correlation to model the prerequisites and consequences of attacks. The template relationship can be automatic generated from the rules of IDS and the end result types of attacks. We define the end result types of attacks in order to determine the consequences of attacks. Then the IDS's rules contain the required situations for attacks to be successful and these characteristics just correspond to the prerequisites of attacks. To combine above information that makes the automation possible. On the other side, by reason of a variety of attacks, the implicit correlation and the explicit correlation can be used to fine tuning the relationship of attacks and it will be

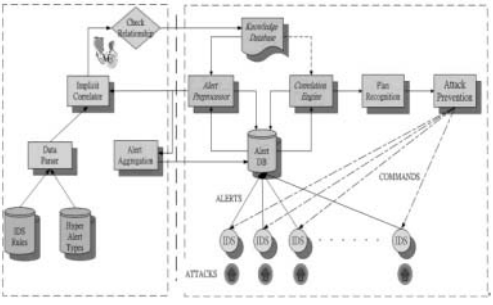


Fig. 1. Alert Correlation Architecture

a great help to specify the alert relationship. This paper focuses on modeling of the attacks. The alert aggregation, plan recognition, and attack prevention are out of the scope of this paper.

3.1 Modeling the Alert Relationship

Figure 4 shows the flow chart of modeling attack relationship and how it to assist hyper-alert correlation. There are two main blocks in current architecture. Initially, the data parser in the left block acquires the required information about the prerequisites and the consequences of attacks from IDS rules and hyper-alert type database. And then the parser generates the fundamental relationship of the attacks to the knowledge base in the right block based on the template correlation. In this moment, the alert preprocessor and correlation engine have enough information to process the alerts from the underlying IDS. The alert preprocessor in current architecture transfer the hyper-alerts not only to correlation engine but also to the implicit correlator. The implicit correlator will attempt to discover the complementary relationships while the template correlation does not provide. Eventually, it is up to human users to choose the proper relationships from this complementary relationships. Moreover, we provide the explicit correlation tools to fine tuning the inadequate alert relationships.

Template Correlation The typical attack scenario that an attacker launches a system attack from the target system consists of five phrases:

1. The attacker probes the network and finds the live IP.
2. The attacker attempts to probe the available service on target system.
3. The attacker attempts to gain admission to the target system via the service vulnerability.
4. The attacker installs the backdoor to the target system in order to take better control.
5. The attacker launches DDOS attack from the daemon or attacks other hosts from this compromised system.

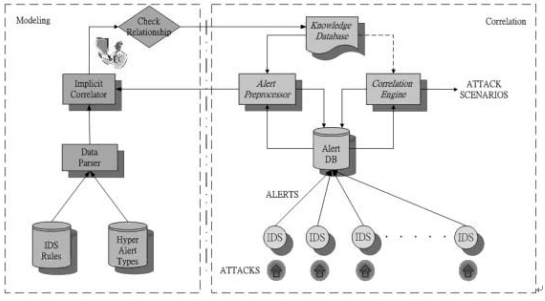


Fig. 2. Modeling vs. Correlation

In the other typical attack scenario, an attacker may launch a system attack directly after step 2. Therefore, our taxonomy is classified according to the end result types of the attack directly observable at above two typical attack scenarios. Our taxonomy follows:

1. IP_PROBE: e.g., IP Scan attack
2. GAIN_INFORMATION: e.g., Port Scan attack, Gain Info attack
3. GAIN_ACCESS: e.g., User Access attack, Root Access attack
4. DAEMON: e.g., trojan horse
5. SYSTEM_ATTACK : e.g., dos attacks, system attacks

Furthermore, we define five fundamental types of template relationship to describe the prerequisites and consequences of attacks according to the end result types of attacks. Table 1 to Table 5 show these fundamental template relationships.

This paper considers automated translation of the characteristics of Snort rules to the prerequisites and consequences of attacks. Finally, we can specify the prerequisite and consequence of attacks according to the end result types of attacks and the rules of IDS. There are two examples of the DDOS attack scenario as follow and each of them represents the final stage of the attack scenario.

Example 1: Sadmind Ping
P: ExistHost (VictimIP)
S: Sadmind_Ping(VictimIP,VictimPort)&
VulnerableRPCService(VictimIP, VictimPort)

Example 2: Sadmind Buffer Overflow
P: VulnerableRPCService(VictimIP, VictimPort)&ExistHost (VictimIP)
S: Sadmind_Overflow(VictimIP)&GainRootAccess(VictimIP)

Implicit Correlation Implicit correlation correlates the alerts based on the end result type of attacks and this analysis will bring out some complement

Table 1. Template Relationship – Vulnerable Host

	Vulnerable Host (type = IP_PROBE)
Fact	VictimIP
Prerequisite	null
Consequence	HyperAlert(VictimIP) & ExistHost(VictimIP)

Table 2. Template Relationship – Vulnerable Service

	Vulnerable Service (type = GAIN_INFORMATION)
Fact	VictimIP, VictimPort
Prerequisite	ExistHost(VictimIP)
Consequence	HyperAlert(VictimIP, VictimPort) & ExistService(VictimIP, VictimPort)

Table 3. Template Relationship – Gain Access on Target System

	Gain Access on Target System (type = GAIN_ACCESS)
Fact	VictimIP, VictimPort
Prerequisite	ExistService(VictimIP, VictimPort)
Consequence	HyperAlert(VictimIP) & GainAccess(VictimIP)

Table 4. Template Relationship – Install Daemon on Target System

	Install Daemon on Target System (type = DAEMON)
Fact	VictimIP, VictimPort
Prerequisite	SystemCompromised(VictimIP) & ExistService(VictimIP, VictimPort)
Consequence	HyperAlert & ReadyToLaunchAttack

Table 5. Template Relationship – System Attack

	System Attack (type = SYSTEM_ATTACK)
Fact	VictimIP, VictimPort
Prerequisite	ReadyToLaunchAttack & ExistService(VictimIP, VictimPort) & ExistHost(VictimIP)
Consequence	HyperAlert(VictimIP) & SystemAttack(VictimIP)

mappings. Then the security administrator chooses the appropriate relationship from implicit correlation. Table 6 shows the alert relationship before implicit correlation while Table 7 shows the alert relationship after implicit correlation. In this situation the attacker may launch Finger_Null_Request attack first to gain user access via finger vulnerability. Then the attacker may take advantage of user's right to execute system commands with exploit code via SMTP vulnerability in order to gain privileged right. As a result of the connections between the same end result type of attacks do not provided by template correlation, implicit correlation will be a necessary complement to template correlation.

Explicit Correlation With explicit correlation, security administrator can modify the appropriate relationship that he knows. Table 8 shows the alert relationship before explicit correlation while Table 9 shows the alert relationship after explicit correlation. Consider this example, Mstream_Zomebie is dedicated to Mstream_Attack. Other DDOS attacks can not make use of Mstream_Zomebie to launch DDOS attack. At this time, we can add the consequence of DDOS_Mstream_Zomebie (DDOS_Mstream_Zomebie) to the prerequisite of DDOS_Mstream_Attack and delete the ReadyToLaunchDDOSAttack from the consequence of DDOS_Mstream_Zomebie in order to specify the appropriate relationship between these alerts.

Table 6. Before Implicit Correlation

	FINGER_Null_Request	SMTP_Majordomo_ifs
Prerequisite	1.VulnerableFINGERservice 2.ExistHost	1.VulnerableSMTPservice 2.ExistHost
Consequence	*1.FINGER_Null_Request* 2.GainUserAccess	1.SMTP_Majordomo_ifs 2.GainRootAccess

Table 7. After Implicit Correlation

	FINGER_Null_Request	SMTP_Majordomo_ifs
Prerequisite	1.VulnerableFINGERservice, 2.ExistHost	1.*FINGER_Null_Request* 2.VulnerableSMTPservice 3.ExistHost
Consequence	1.*FINGER_Null_Request* 2.GainUserAccess	1.SMTP_Majordomo_ifs 2.GainRootAccess

Table 8. Before Explicit Correlation

	DDOS_Mstream_Zombie	DDOS_Mstream_Attack
Prerequisite	SystemCompromised	1.ReadyToLaunchDDOSAttack 2.ReadyToLaunchAttack 3.ExistService
Consequence	1.*DDOS_Mstream_Zombie* 2.ReadyToLaunchDDOSAttack	1.DDOS_Mstream_Attack 2.DOSAttack

Table 9. After Explicit Correlation

	DDOS_Mstream_Zombie	DDOS_Mstream_Attack
Prerequisite	SystemCompromised	1.*DDOS_Mstream_Zombie* 2.ReadyToLaunchDDOSAttack 3.ReadyToLaunchAttack 4.ExistService
Consequence	*1.DDOS_Mstream_Zombie* 2.ReadyToLaunchDDOSAttack	1.DDOS_Mstream_Attack 2.DOSAttack

4 Experimental Result

To evaluate the effectiveness of our method in modeling the alert relationship, the following test plan is proposed.

4.1 Multiple Attackers Vs. Single Victim

Consider the situation in Figure 7, where three attackers participating in this attack scenario. The six phrases of this attack scenario are:

1. The first attacker A2 launches IP_Scan to probe the live IP and find victim V2 is alive.
2. The attacker A2 passes this information (V2) to the next attacker A3 in other location.
3. The attacker A3 knows the victim V2 is alive and launches SMTP_Expn_Root to probe SMTP service.
4. The attacker A3 passes this information (V2,SMTP) to the next attacker A4 in other location.
5. The attacker A4 launches SMTP_ETRN_Overflow to victim V2 in order to gain root access.
6. If the attacker A4 fails to gain the root access, he can probe another machine or probe another service on target system.

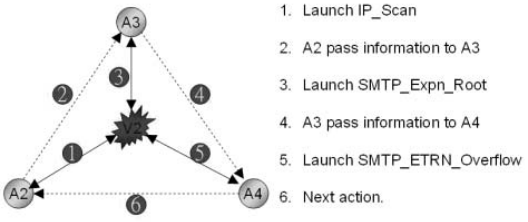


Fig. 3. Attack Scenario - via SMTP Vulnerability

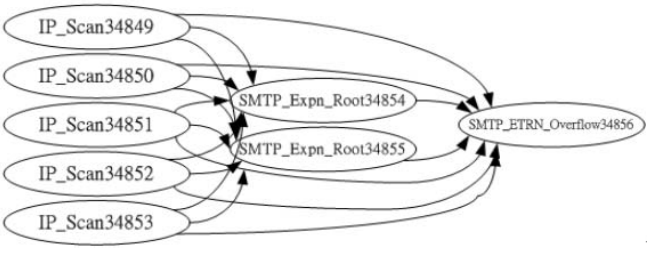


Fig. 4. Correlation Graph of Attack Scenario

And the Figure 8 shows the hyper-alert correlation graph discovered from this attack scenario.

5 Conclusions

This paper addressed a framework of constructing hyper-alert correlation for defense-in-depth network security system. A practical method to construct alert relationship automatically for hyper-alert correlation has been proposed. Without depending on predefined attack scenarios to discover sequences of related attacks, the human users can use the implicit correlation and explicit correlation to modify the difference of attacks. This reduces the considerable burden for specifying correlation relationship and the detection rules specified by the venders of IDS can be reused. The experiments show that our methods perform as well as the hyper-alert correlation with well specified relationship.

As a result of the alert correlation, the correlation system depends on the underlying IDS to provide alerts. Generally specking, most of the alert correlation systems are focus on reducing the false alerts in order to improve the accuracy of alerts but it is hardly to improve the detection rate of attacks. We suggest that applying the algorithm of finding frequent sequential patterns during alert aggregation stage in order to improve the alert detection rate, especially for the scanning attempt of attackers.

In our experiments, we only correlated the alerts generated by the same type of IDS. As mentioned previously, different types of IDS have their own strength

and weakness. It will be a critical issue to cooperate the different types of IDS. Although the Intrusion Detection Message Exchange Format [10] (IDMEF) is intended to be a standard data format and it tries to define common data formats which may be used in the different IDS, it still has no ability to demonstrate the endemic characteristics of different types of IDS.

References

[1] Peng Ning, Yun Cui, Douglas S. Reeves, "Analyzing Intensive Intrusion Alerts Via Correlation". In *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, Zurich, Switzerland, October 2002.

[2] Peng Ning, Yun Cui, Douglas S. Reeves, "Constructing Attack Scenarios through Correlation of Intrusion Alerts". In Proceedings of the 9th ACM *Conference on Computer & Communications Security*, Washington D. C., November 2002.

[3] P. Ning, D. Reeves, and Yun Cui, "Correlating alerts using prerequisites of intrusions". *Technical Report TR-2001-13*, North Carolina State University, Department of Computer Science, Dec. 2001.

[4] F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework". In Proceedings of the *2002 IEEE Symposium on Security and Privacy*, May 2002.

[5] F. Cuppens, "Managing alerts in a multi-intrusion detection environment". *17th Annual Computer Security Applications Conference(ACSAC)*. New-Orleans, December 2001.

[6] A. Valdes and K. Skinner, "Probabilistic alert correlation". In Proceedings of the *4th Int'l Symposium on Recent Advances in Intrusion Detection (RAID 2001)*, pages 54-68, 2001.

[7] F. Cuppens and R. Ortalo. "LAMBDA: A language to model a database for detection of attacks". In *Proceedings of Recent Advances in Intrusion Detection (RAID 2000)*, pages 197-216, September 2000.

[8] Vigna, G. and Kemmerer, R. A. "NetSTAT: A network-based intrusion detection system". In /it Journal of Computer Security. 7, pages 37-71, 1999.

[9] Sheyner, O., Haines, J., Jha, S., Lippmann, R. and Wing, J. "Automated generation and analysis of attack graphs". In *Proceedings of IEEE Symposium on Security and Privacy*, May 2002.

[10] John McHugh, Alan Christie, and Julia Allen. "Intrusion detection implementation and operational issues". *CERT*, January 2001.

[11] D. Curry and H. Debar "Intrusion detection message exchange format data model and extensible markup language (xml) document type definition". *draft-ietf-idwg-idmef-xml-10.txt*, January 2003.

Rethinking of Iolus: Constructing the Secure Multicast Infrastructure*

Wen Tao Zhu, Jin Sheng Li, and Pei Lin Hong

Department of Electronic Engineering and Information Science
University of Science and Technology of China
P.O. Box 4, Hefei, Anhui 230027, P.R. China
wtzhu@ustc.edu
{jinsheng, plhong}@ustc.edu.cn

Abstract. To provide multicast confidentiality, the traffic data is encrypted with a session key known only to certificated group members, which should be updated dynamically whenever there is a change in the group membership. We present the Secure Multicast Infrastructure as a general solution for securing many-to-many group communications that can be relied on by concurrent multicast applications. Rekey algorithms and protocols for data transmission are specified. The main features of our solution include strong scalability and observable reliability.

1 Introduction

On the Internet, multicast has been used successfully to provide an efficient, best effort delivery service to large groups that are dynamic in nature. As a result, multicast security has become a critical networking issue since the original Internet protocols paid little attention to security concerns [1][2]. Specifically, the Internet Group Management Protocol was designed to provide an open group model. It does not provide an access control mechanism; anyone can join the group and obtain a copy of every multicast message from the sender by simply sending membership reports to its neighboring router. It would be very easy to launch a theft of service when the multicast data is transmitted unencrypted.

While protocols have been proposed for Internet security (e.g., IPSec, TLS) that allow unicast messages to travel encrypted through the network, it proved to be much more difficult to secure group communications [1]-[3]. The design issue of secure multicast is to maintain group communication secrecy over an untrustworthy network medium. Cryptography is a practical approach for secure multicast, and we concentrate on simple and efficient symmetric cryptosystem (e.g., DES). A **session encryption key** (SEK) is used by the sender for traffic encryption, and every member in the group should share the identical SEK so as to decrypt the traffic. A trusted key server called the **group controller**

* This work was supported by the National Natural Science Foundation of China under Grant No. 60272043 and the National High-Tech Research and Development Plan of China under Grant No. 2002AA121067.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 895–904, 2004.

(GC) is introduced to perform the key management. Only authenticated users should be able to decrypt the multicast messages even if the data is leaked to the entire network. To ensure that only valid members of the group have access to the multicast communication, the GC should change the SEK on each member join/leave. Such a **rekey** process insures that a joining entity is not able to access previously multicast data and a leaving entity is not able to continue to access data multicast after its departure.

Management of the SEK of a dynamic group is a complex business. There are generally two types of scalability failures which are specific to multicast [3]:

1. a 1 affects n type failure, which occurs when one member affects the entire group. For example, joins exhibit a 1 affects n scalability failure because joins require all members to rekey. In this case, the GC may multicast the new session key, SEK′, encrypted with SEK, to the current members. We simply denote this rekey message as SEK(SEK′). The GC may then separately apprise the joining member of SEK′ encrypted with a pre-established pair-wise key. The pair-wise key is used between the GC and the individual for unicast delivery of the SEK, thus being a **key encryption key** (KEK). When member M_i joins, the unicast message is denoted as $KEK_i(SEK')$, where KEK_i stands for M_i's pair-wise key.
2. a 1 does not equal n failure, which occurs when the protocol cannot deal with the group as a whole and instead, must consider the members on an individual or a subset basis. For example, leaves suffer from a 1 does not equal n scalability failure. When a member leaves, it would be more difficult to distribute SEK′ to the residual members since there is no efficient means to communicate. For instance, when there are N residual members, the GC has to be involved in N unicast deliveries to send SEK′ to each of them.[1] This is inefficient when N is very large or when the group is highly dynamic.

The rest of this paper is organized as follows. In section 2, we investigate and classify secure multicast key management protocols proposed in the literature. In section 3 we review Iolus [3] in detail, on which our solution is based. Section 4 presents the framework of our Secure Multicast Infrastructure (SMI), which is an attempt as a general solution that can be relied on by concurrent applications. SMI not only solves the key management problem in a scalable and reliable manner, but is also adaptive, and this is further discussed in Section 5. Finally we conclude our work in section 6.

2 Related Works

To blind multicast data to unauthorized users, one inefficient but secure way of rekeying is to have the GC share a pair-wise KEK with every member. To add or delete a user, the GC uses the pair-wise KEK of every valid member to securely

[1] Alternatively, the N separate $KEK_i(SEK')$ can be sent as a combined message to all group members via multicast. However, this does not yield a substantial difference.

communicate the new group key (SEK′). This solution is called Simple Key Distribution Center (SKDC), with a rekey communication cost growing linearly with the group size N. The Internet standard GKMP [4] is similar to SKDC.

Other than GKMP, SKDC has led to quite a few mathematical approaches including those based on polynomial interpolation or number theories (e.g., Chinese Remainder Theorem, Euler's Theorem). However, all involving a rekey cost proportional to N, they are not scalable to large or dynamic groups. SKDC and its various transformations are generally referred as **flat** schemes. To be applicable to large-scale multicast, **hierarchical** structure of either a logical one or a physical one is introduced towards solving the scalability problem.

Wong et al. [5] and Wallner et al. [6] independently proposed a scalable scheme by constructing a logical key tree. Their approach is generally known as Logical Key Hierarchy (LKH) and has led to a family of key management schemes for secure multicast [1][7][8]. In LKH each multicast group member is represented in a virtual key tree by a unique leaf node and is pre-assigned the individual's pair-wise key. The inner nodes are associated with extra intermediate KEKs and the root node is associated with the SEK. The set of keys associated with the nodes along the path from a leaf node to the root are assigned to the member represented by that leaf node, which include its pair-wise key with the GC, the intermediate KEKs, and the SEK used for traffic encryption. The main idea of LKH is to have the GC distribute intermediate KEKs in addition to the SEK. Each of the intermediate KEKs can be used to securely multicast rekey messages to users that are leaves of the corresponding inner node's subtree, thus materializing a "restricted multicast" to a specific set of users. Deletion of a member is accomplished by rekeying all the keys on the path from that particular leaf node to the root except its pair-wise key. By taking advantage of these auxiliary intermediate KEKs, a logarithmic communication cost is obtained.

Another approach is to employ a physical hierarchical structure, typically the Iolus framework proposed by S. Mittra [3]. Iolus decomposes a large group into many geographical subgroups that are relatively autonomous. In Iolus there is no globally shared SEK. A new member joins in the global communication by joining to a certain subgroup and only the local subgroup key is distributed to it. This work is performed by special trusted entities called **Group Security Agents** (GSAs). Deleting a member from the whole group is simply done by removing it from its subgroup; other subgroups are not affected and they need not rekey. Our solution is based on a rethinking of the Iolus model.

3 Iolus: The Base Model

Iolus organizes the multicast group into independent subgroups. In Fig. 1 there is a hierarchy of subgroups of two levels. On the user level there are m subgroups in the charge of m GSAs respectively. The m GSAs then form another subgroup_0 which is on the agent level. This is our case implementation of Iolus framework and we will use such a hierarchy for illustration in the following discussion.

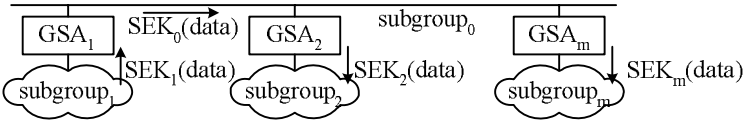


Fig. 1. A two-level case implementation of Iolus

There is no global SEK for the whole group; GSA_i is responsible for a local SEK_i within its user-level $subgroup_i$. A new member joins the multicast by joining to a geographically nearby $subgroup_i$ and only SEK_i is distributed to it (i = 1, 2, ... m). When a member is removed from the multicast, it is removed by the GSA from the subgroup and a local rekey process is triggered without affecting other subgroups. Since each GSA_i maintains SEK_i independently (e.g., applying flat key management schemes), the scalability problem is greatly mitigated.

All the GSAs share a SEK_0 for the agent-level $subgroup_0$. The GSAs, serving as interpreters, translate and forward the traffic throughout subgroups. Iolus facilitates both one-to-many and many-to-many secure multicast. For example, user M_{1a} within $subgroup_1$ may send a message to the entire group by multicasting the message encrypted with SEK_1, which is only understandable by members of $subgroup_1$. With each GSA_i being aware of both SEK_i and SEK_0, the ciphertext from M_{1a} is then decrypted by GSA_1 and again encrypted with SEK_0, and then forwarded to $subgroup_0$. As illustrated in Fig. 1, every GSA_i (except GSA_1) may decrypt the traffic and again encrypt it with SEK_i, and then multicast to $subgroup_i$ respectively. In this way all the members that do not belong to $subgroup_1$ can then read M_{1a}'s message after three encryptions (on the sender side and the outgoing interfaces of the GSAs) and three decryptions (on the incoming interfaces of the GSAs and the receiver side).

Iolus is easy to understand and it offers strong scalability. Unlike the flat or the logical hierarchical schemes, there is no centralized entity such as the GC, thus a setup explosion as well as a single point of failure is avoided. Flat key management schemes such as SKDC may be applied within user-level subgroups whose maximum sizes can be pre-scaled[2], and thus each user only needs to remember its pair-wise KEK plus the subgroup SEK. This cuts down the key storage overhead of the users as compared with LKH, in which auxiliary intermediate KEKs are kept and updated.

Similar to LKH, Iolus has also led to other solutions using a physical hierarchical structure. A detailed comparison of Iolus alike solutions and LKH based schemes can be found in [9]. It is observed that physical hierarchy based approaches fare better than logical hierarchy based ones: the former protocols incur less encryption cost compared to the latter ones, and Iolus scales much better than LKH as the number of simultaneous members in a multicast session

[2] This may be practical in the typical case of a metropolitan network. For example, each subgroup more or less corresponds to a residential area and thereby the potential subgroup size can be well investigated.

increases. Another comparison can be found in [10], where Iolus is supposed to be not only more scalable than LKH but also reusable. That is, Iolus can be relied on by many different multicast groups (i.e., concurrent multicast applications). Therefore, it would be practical and promising to build a secure multicast infrastructure on the basis of Iolus.

Chen et al. [11] also reviewed the key management schemes and classified them as stateful or stateless according to the interdependency of rekey messages. LKH based approaches [1][5]-[8] are stateful in that members should have correctly received past rekey messages to decrypt current rekey messages. What's more, LKH uses restricted multicast to distribute rekey messages, hence a reliable multicast delivery is mandatory. On the other hand, stateless rekey protocols such as SKDC and Iolus only use the individuals' pair-wise keys to decrypt unicast rekey messages. Those messages are independent of each other and consequently members going offline can decrypt the SEK without involving multiple re-transmissions of a chain of intermediate KEKs. In our case implementation of the Iolus framework, a new member joins the multicast group by contacting a geographically nearby GSA, which makes the unicast delivery of a local SEK more reliable. It is also observed that stateless key management schemes perform better if batch rekeying is adopted [8][11]. We would thereby include batch rekeying as one of the basic policies in our Secure Multicast Infrastructure.

We conclude this section with a summary of Iolus. The main idea is to have each GSA distribute a local SEK. When a member joins or leaves, the corresponding GSA rekeys its subgroup without affecting others, thus reducing the scalability problems. However, this improvement is not for free: as subgroups have different SEKs, multicast packets should be decrypted and re-encrypted by GSAs whenever they pass from one subgroup to another. We would try to answer similar open questions with our Secure Multicast Infrastructure (SMI).

4 The SMI Framework

The SMI framework seeks to be a general solution for securing many-to-many group communications that can be relied on by concurrent multicast applications. We would first concentrate on protocols for subgroup SEK generation and distribution. A version-based key management is applied.

4.1 Key Generation and Distribution

We begin with the agent-level subgroup_0, which performs a periodic rekeying. Each GSA is required to maintain a timer, according to which SEK_0 is synchronously and independently calculated. Inspired by the one-way functions used in [1][7], we have the GSAs generate the group key as $\text{SEK}_0(\text{n, t}) = \text{f}_0(\text{s, n, v}_0)$. Note that no rekey exchanges are involved within subgroup_0 at all.

– s is a secret seed of the one-way function f_0 known only to the GSAs. It may be securely negotiated by the GSAs during session initialization.

- n is the session number. It is an application identifier, which is also assigned during the initialization of the secure multicast session.
- v_0 (actually v_{0n}) is a discrete time index, which starts from zero at a negotiated time point[3] and increases by one after every rekey period denoted as T_{max}. It is the version of SEK_0 for session n during the period from v_0T_{max} to $(v_0+1)T_{max}$. Since $subgroup_0$ is static and the GSAs are all trusted entities, it is desirable to choose a relatively long T_{max}.[4]

On the other hand, batch rekeying [8][11] is applied in the user-level subgroups. In batch rekeying, the GC (here the GSA) collects join/leave requests during an interval and rekeys after a batch has been collected. Batch rekeying saves server cost substantially, and flat schemes outperform LKH in the case of large dynamic groups. Similar to the generation of SEK_0, we have GSA_i generate the $subgroup_i$ key for multicast session n as $SEK_i(n, t) = f_i(s, n, v_i)$:

- s is the same secret seed in the one-way function f_0 known only to the GSAs, hence it is impossible for the group members to calculate SEK_i themselves.
- n is the session number identifying a specific multicast scenario. SMI supports concurrent multicast applications that overlap in the geographical subgroups. Suppose there are two users in $subgroup_1$: M_{1a} joining session a and M_{1b} joining session b. The geographical $subgroup_1$ then serves both scenarios by having GSA_1 distribute SEK_1(a,t) to M_{1a} and SEK_1(b,t) to M_{1b}.
- v_i is $subgroup_i$'s key version for session n. It increases by one after the GSA has collected every B join/leave requests and triggers a local rekey process. Therefore, instead of beginning with zero, v_i starts from one. B is the batch rekeying threshold for membership changes. It is observed that flat schemes perform better when the batch size increases [11], so it is desirable to choose a relatively larger B. However, the larger B is, the longer the rekey interval tends to be, which is undesirable for the integrity of the subgroup. Take this into consideration, we set T_{max}, the rekey period of $subgroup_0$, as the upper limit of $subgroup_i$'s rekey interval. A lower limit, T_{min}, is also introduced to prevent v_i from increasing too quickly (i.e., to prevent a highly dynamic group from batch rekeying too frequently).[5] In a nutshell, the rekey interval is always not less than T_{min} and not more than T_{max}, and it is only less than T_{max} when the GSA has collected at least B membership changes.

On unicast deliveries of the rekey messages, along with $SEK_i(n, t)$ for session n, GSA_i also delivers both n and v_i to the users within $subgroup_i$, which may consist of several user sets that belong to different multicast sessions. On receiving the rekey message, a member (which may join multiple sessions) updates its SEK as well as the key version according to the session number n.

[3] The negotiated time point is denoted as $t = 0$. Protocols for network clock synchronization (e.g., NTP) may be needed.

[4] v_0 may be recycled after some time. For instance, v_0 is 16-bit wide and T_{max} is 8 seconds, v_0 will then overflow and reset to zero after $2^{16}T_{max} = 524288$ seconds (approximately 6 days). Fortunately, it is less likely for a secure multicast scenario to last more than one day. What's more, the session number n can be re-assigned.

[5] For instance, T_{max} is 8 seconds and T_{min} is 1∼2 seconds.

4.2 Data Transmission

For the rekeying of the group key, care must be taken that all members should receive the key updates and they are synchronized in rekeying. Members that do not rekey duly may cause transient security breaches [3]: receivers failing to receive a key update will not be able to continue decrypting the traffic and may also accept communications from evicted members; senders failing to receive a key update will continue to encrypt their transmissions with an outdated SEK, causing receivers unable to decrypt the data while departed group members can continue decrypting the transmissions.

The above is known as the out-of-sync problem between keys and data, which may require a user to hold many historical SEKs and buffer a large amount of traffic that it temporarily can not decrypt.[6] Problems may be even exacerbated when the group has a highly dynamicity, which results in flurries of rekey messages and increases the probability of transient security breaches and confusion.

SMI alleviate this problem by taking advantage of version-based key management. It requires the sender to attach its SEK version to the ciphertext before multicasting to the subgroup. For example, user M_{1a} within $subgroup_1$ participating in session a may multicast a message to $subgroup_1$ (ultimately the entire group of session a) by sending a combined message of the key version v_1 and the data encrypted with $SEK_1(a, t)$.[7] Note that the session number a is not included in the combined message, since the destination address of the multicast packet itself identifies the group that the packet is sent to. In addition, a local timer may be maintained by each member, and a sender may be required to examine its key version before any data transmission; if it has been using a SEK_i for longer than T_{max}, other than simply waiting for a rekey message, it may pull the latest local SEK from the GSA on its own initiative.

On receiving M_{1a}'s message, members within $subgroup_1$ participating in the same session a should first compare M_{1a}'s SEK version v_i with its own v_i' before further processing ($i = 1$ in $subgroup_1$). Since the subgroup is geographical formed and the members are regionally nearby, with a moderate lower limit for the batch rekey interval (i.e., T_{min}) there may only be three possibilities:

1. $v_i = v_i'$: the sender and the receiver are synchronized in rekeying, and the data can be correctly decrypted with $SEK_1(a, t)$.
2. $v_i = v_i'$-1: the sender may not have received the latest SEK, but the data can still be decrypted by the receiver with the previous key, i.e., each member needs to keep one (and only one) historical SEK.
3. $v_i = v_i'$+1: the receiver may not have received the latest SEK and the data should be buffered and hopefully it would be "decodable" within time T_{max}.

If v_i does not match any of the three cases, the receiver may simply discard the traffic. In this way, security breaches and confusion can be avoided effectively.

[6] It may be even troublesome in the case of LKH based schemes: a user may have to hold many historical intermediate KEKs and buffer a large amount of rekey messages.

[7] Digital signature may be needed, which is not discussed here.

On receiving M_{1a}'s message, GSA_1 verifies M_{1a}'s SEK version in the same way a common $subgroup_1$ member does. Since M_{1a}'s key version can never be newer than GSA_1's, GSA_1 needs no buffering. If the key version is valid (i.e., the same as GSA_1's, or less than it by only one), the traffic can be decrypted and then encrypted again with SEK_0(a, t) and multicast to other GSAs. GSA_1 may need to attach both the session number and its SEK_0 version to the ciphertext.

On receiving GSA_1's traffic, each GSA_i (i = 2, 3, ... m) first checks the attached session number a to avoid delivering multicast packets to its subgroup where there are no interested receivers. If there is at least one member within $subgroup_i$ participating in session a, GSA_i should then translate and forward the traffic to $subgroup_i$. Again only the SEK version is attached to the ciphertext, while the destination address of the multicast packet identifies session a's group.

On receiving GSA_i's message, members within $subgroup_i$ participating in session a should first check its own SEK version before trying to decrypt. Buffering should be performed if the member holds an outdated key that was for the previous interval. In this way, session a's members in $subgroup_1$ as well as in other subgroups can all be informed M_{1a}'s message in a secure and reliable way.

5 Enhancements and Adjustments

SMI inherits Iolus' scalability owing to the distributed agents (i.e., GSAs). SMI also provides reliability to secure multicast communications owing to the version-based key management. In this section, we further extend SMI by specifying other behaviors of the GSAs.

5.1 SEK Forenotice

Multicast tends to become an important and well-used Internet paradigm especially in the area of effective multimedia content distributing. Therefore, while securing the communications, it is preferable to guarantee the quality of service (QoS) of those applications which ask for a real-time data delivery. In the SMI framework, a recipient who has not obtained the latest SEK needs to buffer the received data, which leads to undesirable delays and possible jitters. However, this may be avoidable when SEK forenotice is introduced. The main idea is simple: by the end of each rekey interval, instead of releasing SEK_i of version v_i, we have GSA_i pre-distribute SEK_i of version v_i+1 to the residual members within $subgroup_i$. An exception is that at the end of the beginning interval (referred as interval 0), GSA_i should release SEK_i of both version 1 and version 2. In general, the GSA collects join/leave requests during interval v_i-1, instead of rekeying for the next interval, it pre-rekeys for interval v_i+1.[8] Note that SEK forenotice also mitigates the GSA's rekey pressure: other than having to be involved in flurries of deliveries of the latest SEK at the very beginning of every rekey interval, it now only needs to dispatch the pre-rekey messages without haste to the residual members right before the end of every rekey interval.

[8] As described in Section 4.1, v_i starts from one.

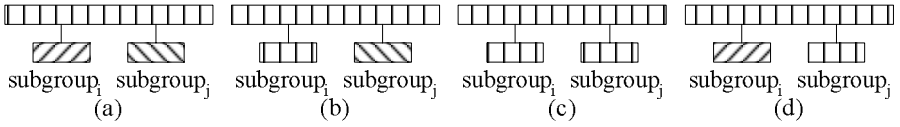


Fig. 2. Group merging and detaching

5.2 Group Merging

Thus far we have specified many aspects and behaviors of the GSAs. However, one major disadvantage of such a physical hierarchy has been receiving scant attention. The merits of the SMI framework is not for free: as subgroups have different SEKs, traffic have to be decrypted and re-encrypted by the GSAs, incuring an observable computational overhead. One straightforward idea is to have the subgroups hold the same SEK. However, since the use-level subgroups are nonadjacent, we would choose an indirect way as a measure to deal with the problem.

The main idea of our solution is to have a user-level subgroup merge into the agent-level subgroup$_0$ when it tends to be less dynamic. Suppose in a certain multicast session n, GSA$_i$ collects join/leave requests during a rekey interval as usual, see Fig. 2(a). When the frequency of membership changes is less than a threshold[9], GSA$_i$ would then merge subgroup$_i$ into subgroup$_0$ by distributing SEK$_0$(n, t) to subgroup$_i$ instead of SEK$_i$(n,t), i.e., in subgroup$_i$, batch rekeying is replaced by agent-level periodic rekeying. This is depicted in Fig. 2(b). Note that GSA$_i$ may be responsible for converting v$_0$ to corresponding value of v$_i$. If later anther subgroup$_j$ also becomes less dynamic, in the same way it switches to SEK$_0$, and thus subgroup$_j$ is also merged into subgroup$_0$. Since subgroup$_i$ and subgroup$_j$ have the same SEK with subgroup$_0$, the two GSAs now simply forward the traffic between its own subgroup and subgroup$_0$ without involving decryption and re-encryption, as depicted in Fig. 2(c). Suppose that after some time GSA$_i$ detects a local dynamicity, to reduce the 1 affects n problem, it then switches back to local batch rekeying, and subgroup$_i$ is thus detached from subgroup$_0$, as depicted in Fig. 2(d).

A multicast application would probably experience a high number of requests to join the group at the beginning (e.g., TV broadcast of a football game) and a high number of requests to leave near the end. In the middle of the session the group usually tends to be less dynamic. By adaptively merging separate user-level subgroups into a large agent-level subgroup, multiple data decryptions and re-encryptions can be avoided and thus the computational overhead of the GSAs is substantially reduced. Note that to alleviate the scalability problem, a GSA should switch back to normal operations when requests are frequent, like during the startup or teardown of a secure multicast session.

[9] A subgroup may be judged as "less dynamic" when the GSA collects less than B membership changes when the rekey timer expires at T$_{max}$.

6 Concluding Remarks

Securing multicast communications has become a critical Internet design issue and hierarchical structures have been presented to address the key management scalability problem. Based on a rethinking of the Iolus model, we propose our Secure Multicast Infrastructure as a general solution for securing many-to-many group communications that can be relied on by concurrent multicast applications. The primary motivation behind our design was enhanced scalability as well as reliability. Periodic rekeying is applied in the agent-level subgroup without involving either multicast or unicast rekey exchanges. Batch rekeying with a constrained rekey interval is separately applied within each user-level subgroup without affecting others. Both rekey algorithms are based on the one-way functions. A novel version management is introduced both in key management and in data transmission. Other contributions of this paper include proactive rekeying and reduced computational overhead owing to adaptive group merging.

References

[1] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas: Multicast security: a taxonomy and some efficient constructions. INFOCOM'99, Mar 1999, 708–716 895, 897, 899

[2] P. Judge, M. Ammar: Security issues and solutions in multicast content distribution: a survey. IEEE Network, 17(1), Jan/Feb 2003, 30–36 895

[3] S. Mittra: Iolus: a framework for scalable secure multicasting. ACM SIGCOMM Computer Communication Review, 27(4), Oct 1997, 277–288 895, 896, 897, 901

[4] H. Harney, C. Muckenhirn: Group Key Management Protocol (GKMP) Specification. RFC2093, Jul 1997 897

[5] C. K. Wong, M. Gouda, S. S. Lam: Secure group communications using key graphs. IEEE/ACM transactions on networking, 8(1), Feb 2000, 16–30 897, 899

[6] D. Wallner, E. Harder, R. Agee: Key management for multicast: issues and architectures. RFC2627, Jun 1999 897

[7] A. T. Sherman, D. A. McGrew: Key establishment in large dynamic groups using one-way function trees. IEEE Transactions on Software Engineering, 29(5), May 2003, 444–458 897, 899

[8] X. S. Li, Y. R. Yang, M. G. Gouda, S. S. Lam: Batch rekeying for secure group communications. 10th international conference on World Wide Web, May 2001, 525–534 897, 899, 900

[9] L. R. Dondeti, S. Mukherjee, A. Samal: Comparison of scalable key distribution schemes for secure group communication. GLOBECOM'99, Dec 1999, 1774–1778 898

[10] Fan Du, L. M. Ni, A.-H. Esfahanian: Towards solving multicast key management problem. 8th International Conference on Computer Communications and Networks, Oct 1999, 232–236 899

[11] Weifeng Chen, L. R. Dondeti. Recommendations in Using Group Key Management Algorithms. DARPA Information Survivability Conference and Exposition 2003, Apr 2003, 222–227 899, 900

A DRM Framework for Secure Distribution of Mobile Contents

Kwon Il Lee[1], Kouichi Sakurai[2], Jun Seok Lee[3], and Jae Cheol Ryou[4]

[1] Daeduk College, Division of Computer Internet Information
48, Jang-dong, Yuseong-gu, Daejon, 305-715, Korea
kilee@mail.ddc.ac.kr
[2] Kyushu University, Faculty of Computer Science and Communication Engineering
6-10-1, Hakozaki, Higashi-ku, Fukuoka, 812-8581, Japan
sakurai@csce.kyushu-u.ac.jp
[3] ETRI, Computer and Software Research Laboratory
161, Gajeong-dong, Yuseong-gu, Daejon, 305-350, Korea
jslee@etri.re.kr
[4] Chungnam National University, Department of Computer Science
220, Gung-dong, Yuseong-gu, Daejon, 305-764, Korea
jcryou@home.cnu.ac.kr

Abstract. DRM technology allows content to be distributed in a controlled manner. Therefore, appropriate security mechanism is required. The Mobile DRM System is same as the general DRM system. Encryption technology is in use digital contents packaging. In case of Mobile DRM system, secure distribution and store of packaging encryption key is important. In this paper, we propose a DRM framework, SDRM (Secure mobile Digital Rights Management)*, to ensure secure distribution of mobile contents and rights. We considered being a secure DRM system to contain appropriate security solution.

1 Introduction

DRM technology allows content to be distributed in a controlled manner. There are many DRM solutions but these solutions do not cooperate with each other. The standards for the mobile DRM is in need. There are two standardization organizations in mobile DRM. One of them is 3GPP [1], another is OMA (Open Mobile Alliance) [2, 3, 4, 5, 6]. The 3GPP's DRM model has been converged into OMA DRM Model. OMA DRM Version 1.0 [2] is defined by OMA and this is the first DRM standard.

OMA DRM model has no definition about explicit secure delivery mechanism for rights. In addition, the solution of authentication problem for device is not suggested. In this paper, we propose DRM framework, SDRM (Secure mobile Digital Rights Management), to ensure secure distribution of rights and contents. SDRM follows the tradition of OMA DRM, at the same time it provides secure

* This research was supported by University IT Research Center Project.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 905–914, 2004.

contents distribution, authentication of device and it uses public key mechanism. The threat model is different between WIM and DRM therefore it is of no use if we apply existing WIM[7]/WPKI[8] method on DRM. Generally, the end user is the target of attacks in mobile environment whereas the end user itself is the attacker in DRM. In this paper, we describe about SDRM focus on protection of contents and rights from end user.

MPEG-21 distribution model has been studied and applied at wired DRM environment already [9]. In wireless DRM, the study of MPEG-21 distribution model is just started. We did not consider MPEG-21 distribution model in this paper, but we will start to study this field soon.

The organization of this paper is as follows. The related research that contains standardization is in section 2, the proposal of mobile contents are in section 3, the description and evaluation of SDRM in section 4 and conclusion in section 5.

2 Related Works 2.1

2.1 OMA DRM

The OMA DRM version 1.0 will govern the use of mobile-centric content types, whether it is received by WAP download or MMS. The OMA DRM provides three DRM methods: Forward-lock, Combined delivery, and Separate delivery.

Forward-lock intended for the delivery of mobile contents such music, images and information that should not be sent on to others. The device cannot forward digital contents. Combined delivery enables usage rules to be set for the rights. Separate delivery provides delivering of the digital contents and rights via separated channels. In the separate delivery method the digital contents always encrypted using symmetric encryption and converted into the DCF format[4]. Superdistribution is facilitated by allowing DCF [4] formatted digital contents to be forwarded from device to device and by enabling device to obtain rights for superdistribution contents.

2.2 Analysis of OMA DRM

The Forward-lock method and combined delivery method it will face threats on such exposure of the contents for they are not encrypted. In SDRM that we propose in this paper, exclude such criteria in the fear of disclosure of digital contents. OMA DRM model still has problems in the view of the security. Those are as follows.

- OMA DRM model did not explain explicit secure delivery mechanism of rights.
- OMA DRM model did not propose mechanism of device authentication.
- OMA DRM model did not address problem that unauthorized entities such as text editor to access to DRM entities such as rights or contents in the client device.

We suggest one framework, SDRM, to solve above security problem in this paper.

3 Distribution Model

3.1 Distribution Scenario

SDRM is a framework to support secure distribution of mobile contents. SDRM provides following distribution scenarios.

- **Separate distribution of contents and rights.** The contents provider makes protected contents using symmetric encryption and registers CEK (Contents Encryption Key) to the rights issuer (RI). Encrypted contents and rights including CEK are distributed to the client device separately.
- **Superdistribution.** The contents are allowed to pass from one mobile device to another mobile device through any channel, with the rights being obtainable from the rights issuer via WAP PUSH [11].

3.2 The Distribution Model

The distribution model of SDRM is made based on the following concepts.

- WTLS [10] assures secure communication between client and the contents distributor.
- The client's public key was certificated by Certification Authority (CA)
- The contents distributor and the rights issuer are administrated by same subject. So communication channel with the contents distributor and the rights issuer is secure.
- Technically it is impossible to explain every each distribution model therefore, we need to select and support some of them. In this paper we consider separate distribution and superdistribution.

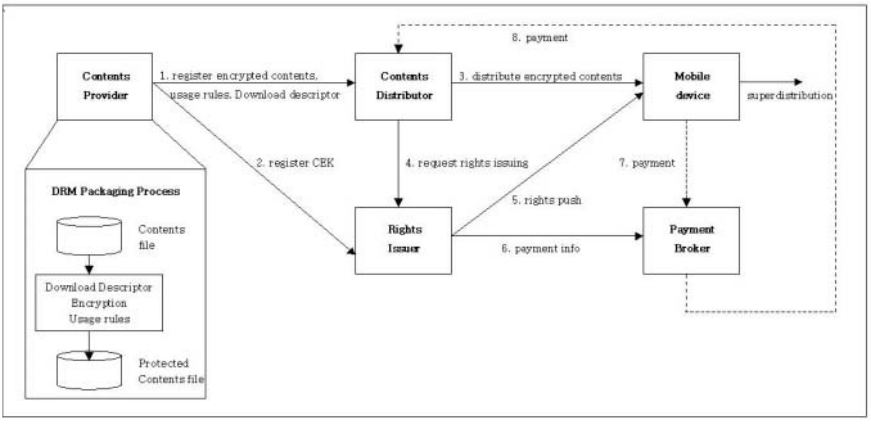


Fig. 1. The distribution model of SDRM

Fig. 1 shows distribution model of SDRM. The contents provider encrypts contents and pack for DRM contents. Download descriptor and contents encryption key are produced by the contents provider. Download descriptor is metadata that contains contents packaging information.

The contents distributor as portal server presents contents information to the client devices, maintains packaged contents and distributes packaged contents to the clients. The rights issuer issues rights for contents. If the client wants to use downloaded packaged contents from the contents distributor, the client has to get rights of contents issued by the rights issuer. The client device means mobile device such as PDA, smart phone or cellular phone. The client device could use DRM contents those with rights.

The flow of our distribution model is briefly explained in the followings.

1. The contents provider registers encrypted contents, download descriptor and usage rules to the contents distributor as on/offline
2. The contents provider registers CEK to the rights issuer as on/off-line.
3. The contents distributor distributes encrypted contents to the client device.
4. The contents distributor requests right issuing for the client device to right issuer after distribution of encrypted contents.
5. The rights issuer distributes rights to the client device via Wap push [11]
6. The rights issuer sends payment information to payment broker

4 DRM Framework

4.1 Metadata

In this section, we describe metadata that was defined in SDRM.

Content Packaging Format (CPF). We define contents packaging format for the encrypted contents based on OMA DRM content format [4]. In addition to encrypting contents the content packaging format supports metadata such as

- Original contents type of the mobile contents
- Unique identifier for this DRM protected digital contents to associate it with rights
- Information about the encryption detail
- Information about the rights issuing service for this DRM protected mobile content
- The URI of the contents distributor to support superdistribution (We insert this field into the OMA DCF [4])

Download Descriptor(DD). The download descriptor is a metadata for packaged contents, and a collection of attributes, used to describe a mobile content at a URI or URL. The defined attributes are specified to allow the download agent of the client device to identify, retrieve, and install contents. The descriptor allows the device to verify that the desired mobile contents are suitable for the device before being loaded. The syntax of download descriptor is expressed as XML [12]. This is a by-product of contents package.

Usages. Usages are metadata to specify candidates of usage for contents. The user who orders contents packaging offers some usage rules for contents usage. The usages are defined using XML. The user can select one item among the usages that was presented by contents provider according to the development of user's situation.

Rights. Rights define rules for how the digital contents should be used. Rights can be limited using both time and count constraints. Rights are used to specify the access a client device is granted to DRM contents. We defined rights using OMA REL (Rights Expression Language) [5] was specified based on ODRL (Open Digital Rights language) [13]. The REL [5] is defined as a mobile profile of ODRL. The structure of the rights expression language enables the following functionality:

1. Metadata such as version and content ID
2. The actual rights specification consisting of
   A. Linking to and providing protection information for the content, and
   B. Specification of usage rights and constraints

4.2 Contents Encryption

Fig. 2 shows CEK generation and contents encryption process. The contents provider generates nonce using random number generation mechanism, and generates CEK using hash function f(). Packaging process gets CEK, Kri (public key of the rights issuer) and raw contents as input and generates encrypted results as follows:

- E_{CEK}[contents]: encrypted contents using key CEK
- E_{Kri}[CEK|CID]: encrypted CEK and CID with key Kri, this is delivered to the rights issuer.

The encrypted contents E_{CEK}[contents] is delivered to the contents distributor via on/offline, and E_{Kri}[CEK|CID] is delivered to the rights issuer. The CEK and CID is encrypted using rights issuer's public key Kri to protect information about rights from abnormal access on network.

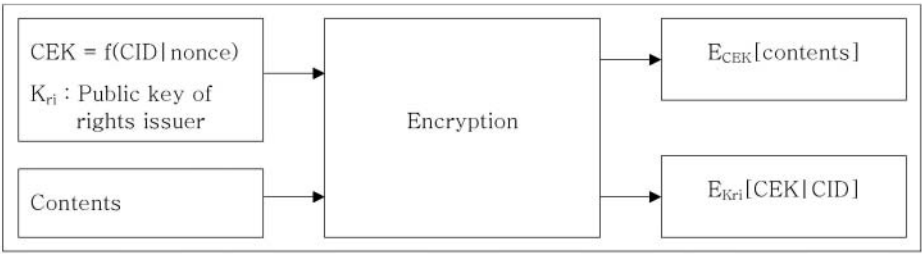


Fig. 2. Key Generation and Contents Encryption Process

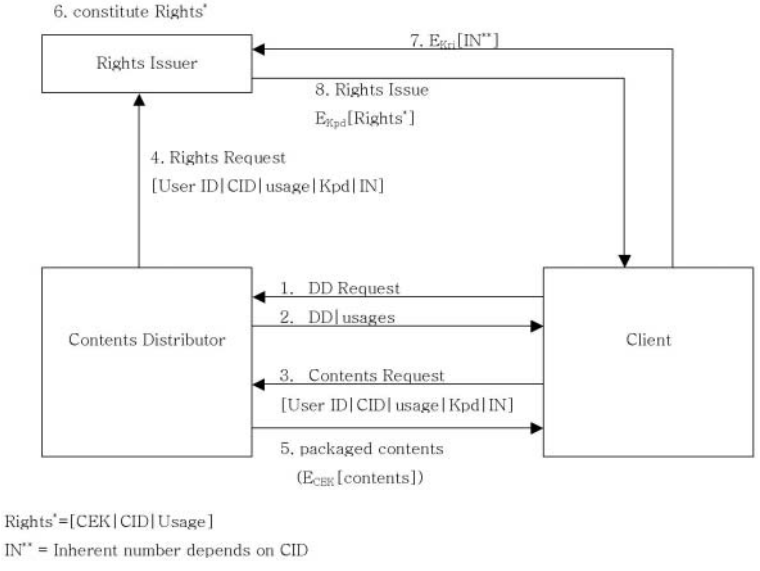


Fig. 3. Contents Distribution and Right Issuing Process

4.3 Contents Distribution and Rights Issuing Process

In SDRM, contents distribution is processed by the contents distributor. Fig. 3 shows contents distribution and rights issuing process.

We prove integrity and confidentiality of contents on distribution path using encryption mechanism. CEK have to be saved and distributed securely to protect contents from illegal access. We used RSA [14] public key mechanism to protect CEK on the distribution path.

Contents distribution and rights issuing process as follows:

1. The client device initiates distribution process of contents and sends DD request message.
2. The contents distributor sends DD and usages to give information about contents. The end user chooses one usage among usages that is presented on the window of client device.
3. Contents request message that is composed with User Id, CID, chosen usage, and client public key Kpd is sent to the contents distributor.
4. The contents distributor requests rights issuing to rights issuer. Rights request message to the rights issuer is made with user ID, content Id (CID), usage is selected by end user, client public key Kpd, and IN (inherent number).
5. The contents distributor delivers encrypted contents to the client device. The client generates IN depends on CID. The rights issuer use IN to authenticate client device.

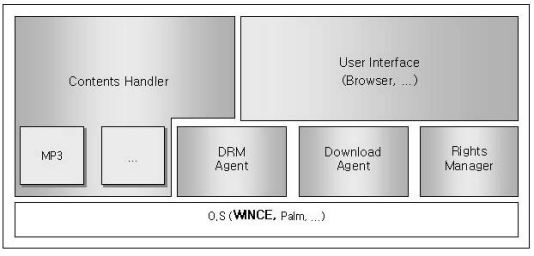


Fig. 4. Client Structure

6. The rights issuer constitutes rights with CEK, CID, and usage rule, and encrypts rights using key Kpd to ensure integrity of rights. The rights issuer pushes a URI of encrypted rights E_{Kpd}[rights] to the client.
7. When the client receives URI of E_{Kpd}[rights], client device sends E_{Kri}[IN] to rights issuers using http post.
8. The rights issuer confirms client's request using IN. And the rights issuer issues an encrypted right E_{Kpd}[rights] to the client. When the client receives E_{Kpd}[rights], the download agent of client can decrypt E_{Kpd}[rights] and saves secure storage of the client.

4.4 DRM Client

Figure 4 shows our DRM client's structure. User Interface is a Web Browser. Contents Handler is an application program to play DRM contents. Contents handler plays DRM contents using DRM Agent. DRM Agent is a user agent that enforces the rights and controls consumption of DRM contents on the client device. Download agent is a user agent responsible for downloading a DRM contents described by a download descriptor and downloadable transaction from the client perspective. It is triggered by the reception or activation of a download descriptor. Rights Manager generates and manages public key and private key of client device. Also manages rights pool securely using our 256-DES [15] algorithm. Encryption/decryption of rights with K_{RM} was executed by this program.

The DRM client must protect contents and rights from illegal access. We use security mechanism to save contents and rights securely in the client device. The malicious user wants to crack contents and rights in him/herself's client device and tries to use or distribute illegally. In SDRM, downloaded rights were encrypted by DRM client's key K_{RM}. Key K_{RM} was made up as follows

- $K_{RM} = K^1{}_{RM} + K^2{}_{RM}$ = f(nonce|device ID|User Id);
- $K^1{}_{RM}$: the client device keeps $K^1{}_{RM}$
- $K^2{}_{RM}$: the rights issuer keeps $K^2{}_{RM}$

DRM client's key K_{RM} is separated to two parts, $K^1{}_{RM}$ and $K^2{}_{RM}$. DRM client only keeps $K^1{}_{RM}$ part, $K^2{}_{RM}$ part was sent to the rights issuer. DRM client

tries to get K^2_{RM} to construct K_{RM} when DRM client program was launched (send E_{Kpd}[User ID+ get (K^2_{RM})]. The rights issuer sends $E_{Kpd}[K^2_{RM}]$ to the client device. Only DRM client program knows about key K_{RM} because key K_{RM} was initiated and maintained in memory of client device when DRM client program started.

4.5 Evaluation

In this section, we describe evaluation of SDRM. In section 2.2, we point some weakness of OMA DRM as viewpoint of security. SDRM proposes solution of security problem of OMA DRM.

Case 1: SDRM ensures secure distribution of rights. Proof) Assume that abnormal client (AC) intercepts E_{Kpd}[rights]that is pushed to normal client (NC). The abnormal client tries to decrypt E_{Kpd}[rights] to get rights to use DRM contents illegally. The AC may be presumed normal client's public key. Let Kpd' is NC's public key was presumed by AC. AC get rights' that was guessed by AC through decryption using guessed client's public key Kpd' ($D_{Kpd'}[E_{Kpd}[rights]]$). As a result of attacking AC gets rights'. Rights' is not equal to rights, so AC cannot get rights abnormally and rights is distributed to the client securely.

SDRM suggests method of device authentication. Proof) NC sends Kpd and IN to rights issuer via the contents distributor. The rights issuer generates rights and encrypts rights using client's public key Kpd (E_{Kpd}[rights]) and push E_{Kpd}[rights]'s URI to NC. Assume that AC find out E_{Kpd}[rights]'s URI, guess IN of NC (IN'), encrypts IN' using key Kri (E_{Kri}[IN']) and sends http post request with E_{Kri}[IN'] to E_{Kpd}[rights]'s URI. Then the rights issuer get IN' using decryption ($D_{Kpri}[E_{Kri}[IN']]$), and compare IN' with IN. The rights issuer knows that IN' is not equal to IN, so the rights issuer did not send rights to AC. In conclusion, the rights issuer can authenticate NC.

SDRM prevents that unauthorized entities(UE) access to DRM entities in the client device. UE cannot guess rights. Proof) UE have knowledge about user ID, device ID but UE cannot calculate K_{RM} because K^2_{RM} which only known to DRM client in memory. So, UE cannot calculate K_{RM}. In conclusion UE cannot guess rights, so UE cannot access to DRM entities.

Table 1 shows a comparison of OMA DRM and proposed DRM in this paper. SDRM more considers secure distribution between DRM components than OMA DRM. We select contents distributor as subject of rights issuing because the portal vendor govern current business environments of digital contents.

5 Conclusion

In this paper, we propose secure DRM framework following on standard model of OMA. We consider secure DRM frame with appropriate terminal key manage-

Table 1. Comparison of DRM system

Item		SDRM	OMA DRM
Security technology	Authentication of Client Device (or User)	Public Key mechanism (RSA [14]) and IN	Not proposed
	Access to DRM contents and rights from unauthorized entities	Save rights and contents securely using encryption mechanism with DRM client's key	Not proposed
	Encryption Mechanism	Using our encryption library (implements DES [15] RSA [14])	AES [16]
Subject of right issuing		Contents distributor	Contents provider
Interoperability with wire DRM		Consider	Not proposed
DRM contents format		CPF (OMA DCF+) (specify contents distributor's URL)	DCF [4]
Download descriptor		OMA	Based on XML [11]
Rights		OMA	REL [5] based on ODRL [13]
Usages Format		XML [12]	Not proposed
Private player		Not necessary	Not proposed

ment mechanism using public key. Also SDRM adapts separate rights delivery mechanism, also supports superdistribution. The security issues in mobile DRM system are user authentication problem and differentiation problem of unauthorized entities (e.g. text editor, calculator, ...) and authorized entities (e.g. DRM agent, download agent, rights manager) to DRM objects such as DRM contents or rights within the same device. To solve of the authentication problem, we adapt separate delivery of digital content and rights and supports superdistribution of digital contents. The differentiation problem of unauthorized entities (e.g. text editor, calculator, ...) and authorized entities was solved using encryption mechanism of rights on the client device. Only authorized entities share rights encryption key.

References

[1] 3GPP: Digital Rights Management Technical Specification, http://www.3gpp.org/ (2002) 905
[2] OMA: Digital Rights Management V1.0, http://www.openmobilealliance.org/ (2002) 905
[3] OMA: Download Architecture V1.0, http://www.openmobilealliance.org/ (2002) 905
[4] OMA: DRM Contents Format V1.0, http://www.openmobilealliance.org/ (2003) 905, 906, 908, 913

[5] OMA: DRM Rights Expression Language, http://www.openmobilealliance.org / (2003) 905, 909, 913
[6] OMA: Generic Content Download Over The Air Specification V1.0, http://www.openmobilealliance.org/ (2003) 905
[7] WAP Forum: Wireless Application Protocol Identity Module, http://www.wapforum.org/ (2001) 906
[8] WAP Forum: Wireless Application Protocol Public Key Infrastructure Definition Specification, http://www.wapforum.org/ (1999) 906
[9] Junseok Lee, Seong Oun Hwang, Jae Cheol Ryou, " A DRM Framework for Distributing Digital Contents through the Internet", ETRI, J. vol.25, no. 6, Dec. 2003, pp. 423-436 906
[10] WAP Forum: Wireless Session Protocol Version 1.0, http://www.wapforum.org/ (2002) 907
[11] WAP Forum: WAP Push Architectural Overview version 0.8, http://www.wapforum.org/ (1999) 907, 908
[12] W3C: Extensible Markup Language (XML), http://www.w3.org/XML/ 908, 913
[13] W3C: Open Digital Rights Language (ODRL) Version 1.1, http://www.w3.org/TR/2002 /NOTE-odrl-20020919/ 909, 913
[14] Rivest, R. L., Shamir, A.Adleman, L.:A method for obtaining digital signatures and public key cryptosystems. Communications of ACM, 21 (1978), 120-126 910, 913
[15] NIST: Data Encryption Standard, FIPS PUB 46-3, http://csrc.nist.gov/publications/fips /fips46-3/fips46-3.pdf 911, 913
[16] NIST: Advanced Encryption Standard, http://csrc.nist.gov/CryptoToolkit/aes/ 913

Analysis and Countermeasure on Vulnerability of WPA Key Exchange Mechanism

You Sung Kang[1], KyungHee Oh[1], ByungHo Chung[1], Kyoil Chung[1], and DaeHun Nyang[2]

[1] Information Security Research Division
Electronics and Telecommunications Research Institute
Daejeon, 305-350, Korea
{youskang,khoh,cbh,kyoil}@etri.re.kr
[2] The Graduate School of Information Technology and Telecommunications
InHa University
Incheon, 402-753, Korea
nyang@inha.ac.kr

Abstract. In this paper, we analyze some weaknesses in WPA authenticator key management state machine and propose the countermeasures to overcome these problems. Our researches on IEEE 802.11i authenticator state machine that is WPA authenticator key management state machine reveal that the state machine cannot support the stable group key setting and is vulnerable to the replay attack and DoS attack. We describe 3 problems related to these vulnerabilities, propose the respective solutions and reconstruct WPA authenticator key management state machine to which the alternative solutions are applied.

Keywords: WLAN security, WPA, 802.11i, 802.1X

1 Introduction

Most of the WLAN (Wireless Local Area Network) products are based on the IEEE 802.11 standard and certified by Wi-Fi Alliance [1]. However, it is common knowledge that these products provide only limited support for confidentiality through the WEP (Wired Equivalent Privacy) protocol that contains significant flaws in the design [2]. The IEEE 802.11 TGi (Task Group i) has proposed the security architecture for IEEE 802.11 standard in order to enhance the security function [3]. In addition, Wi-Fi Alliance has released WPA (Wi-Fi Protected Access) specification as Wi-Fi security standard that is a subset of IEEE 802.11i Draft 3.0 [4]. The Wi-Fi Alliance is a nonprofit international association formed in 1999 to certify interoperability of WLAN products based on IEEE 802.11 specification. The Wi-Fi Alliance in conjunction with the IEEE 802.11 WG (Working Group), has driven an effort to bring strongly enhanced, interoperable Wi-Fi security. The result of this effort is WPA [5].

The IEEE 802.11i standard is intended to provide strong authentication, access control, key management, key establishment, and cipher algorithm. Unfortunately, our researches on IEEE 802.11i authenticator state machine that is WPA

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 915–924, 2004.

authenticator key management state machine reveal that the state machine cannot support the stable group key setting and is vulnerable to the replay attack and DoS (Denial of Service) attack. In this paper, we analyze the vulnerabilities on WPA authenticator key management state machine and propose the countermeasures to overcome these problems. We describe 3 problems and propose the respective solutions. These problems are as follows. The first is vulnerability due to the number of stations to exchange GTK, the second is vulnerability due to the reuse of ANonce when PTK update request is received, and the last is vulnerability due to the incorrect transition after timeout event. In addition, we reconstruct WPA authenticator key management state machine to clarify group key distribution procedure.

The paper is organized as follows. In section 2, the overview of WPA standard is explained briefly. Section 3 describes the technologies related to the general WLAN key establishment. In section 4, we describe the AP action in the WLAN system and analyze weaknesses of WPA key management mechanism. In addition, we explain the respective countermeasures in detail and reconstruct a new key management state machine to which the solutions are applied. Finally, section 5 concludes the paper.

2 WPA Overview

To strengthen user authentication, WPA implements IEEE 802.1X standard as a basis for access control and EAP (Extensible Authentication Protocol) as a framework for authentication message [6, 7]. This framework utilizes a central authentication server, such as RADIUS (Remote Authentication Dial In User Service) [8], to authenticate each user. After acquisition of PMK (Pairwise Master Key) from a central authentication server or PSK (Pre-Shared Key), WPA performs 4-way handshake and group key handshake to distribute PTK (Pairwise Transient Key) and GTK (Group Transient Key), respectively. The default cipher algorithm in WPA standard is TKIP (Temporal Key Integrity Protocol) with the Michael integrity check.

In IEEE 802.11, all the stations must get each PTK because an associated station is treated as a logical port. But, they must have the same GTK that is generated in the AP at the time of the first station connection. If the AP is required to update GTK, the AP must maintain both of the old GTK and a new GTK. In addition, the AP must manage the total number of stations to be sent the group key and the number of stations left to have group key updated. Therefore, the AP must have a state machine performing GTK generation, GTK setting, and GTK update. WPA Authenticator key management state machine (refer [3] section 8.5.6) performs these functions. In section 4 in this paper, we give an explanation about problems included in this state machine.

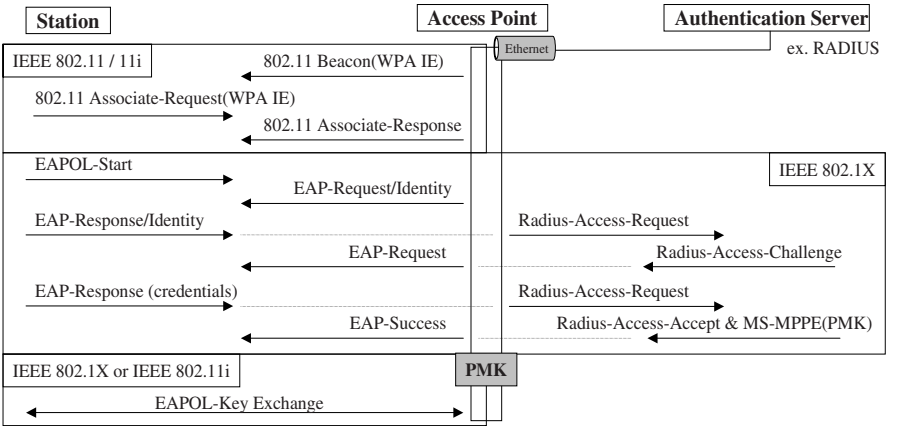


Fig. 1. PMK acquisition using IEEE 802.1X with EAP authentication

3 Key Establishment Mechanism

There are two methods for the dynamic key exchange. One is IEEE 802.1X key transmission using IEEE 802.1X authenticator key transmit state machine (refer [6] section 8.5.5). The other is WPA key exchange using WPA authenticator key management state machine. Both of two must be the next step of PMK acquisition.

3.1 PMK Acquisition

To get PMK, the AP supports two authenticated key management protocols in infrastructure mode using IEEE 802.1X with pre-shared key and with EAP authentication. Fig. 1 shows a sequence of PMK Acquisition using IEEE 802.1X with EAP authentication. EAP-TLS protocol is a typical EAP-method for getting PMK [9].

After association between a station and an AP, the exchange of EAP authentication frame happens. The second box in Fig. 1 illustrates a supplicant-initiated authentication conversation. The AP can get PMK for the associated station via MS-MPPE (Microsoft Point-to-Point Encryption) attribute as a result of IEEE 802.1X with EAP Authentication [10]. If the station is the non-WPA station the AP can use IEEE 802.1X authenticator key transmit state machine to send WEP key to the station. On the other hand, if the station is the WPA station supporting TKIP encryption the AP can use WPA authenticator key management state machine to establishment TKIP key.

3.2 IEEE 802.1X Key Transmission

According to IEEE 802.1X standard, the IEEE 802.1X authenticator key transmission state machine is an option of implementation [6]. In addition, the standard does not define an ACK of a received EAPOL-Key (EAP Over LAN) frame,

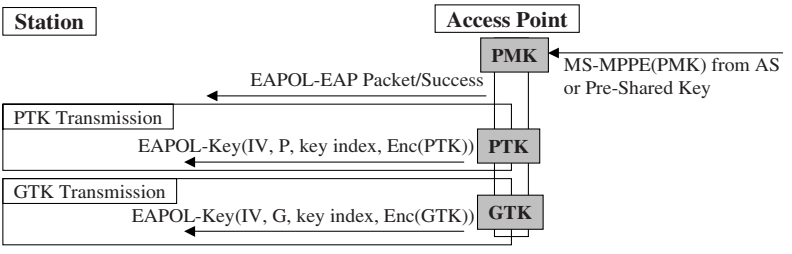


Fig. 2. IEEE 802.1X key transmission procedure

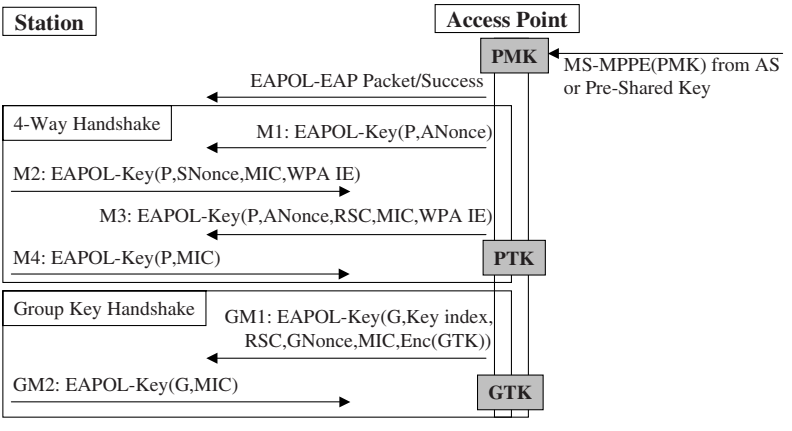


Fig. 3. WPA key exchange procedure

therefore the AP is not responsible for a complete key establishment with a non-WPA supplicant. The IEEE 802.1X key transmission procedure for non-WPA supplicant appears as illustrated in Fig. 2. After PMK acquisition, the AP sends EAP-Success frame to the station. The station receiving EAP-Success frame waits for EAPOL-Key frame. And then, the AP sends pairwise EAPOL-Key frame containing encrypted PTK and group EAPOL-Key frame containing encrypted GTK to the station sequentially. The encryption key used to encrypt the PTK or the GTK is a concatenate string of IV and PMK.

3.3 WPA Key Exchange

WPA defines the 4-way handshake for pairwise key exchange and the group key handshake for group key exchange. Fig. 3 illustrates the WPA key exchange procedure including 4-way handshake and group key handshake.

The 4-way handshake confirms the liveness of the peers communicating directly with each other over the IEEE 802.11 link, guarantees the freshness of the their shared PTK, and synchronizes the usage of the PTK to secure the IEEE 802.11 link [3]. The AP uses the group key handshake to send a new GTK to the

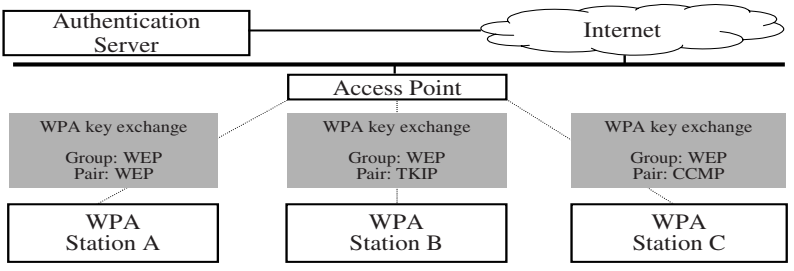


Fig. 4. WLAN system with many WPA clients

station. The PTKs are used between a single AP and a single station but the GTK is used between a single AP and all the stations authenticated to that AP. After PMK acquisition, the AP sends EAP-Success frame to the station. And then, the AP sends the first pairwise EAPOL-Key frame containing a random number called ANonce (Authenticator Nonce). The second frame comes from the station and includes SNonce (Supplicant Nonce). Next, the AP calculates PTK by means of PRF (Pseudo Random Function) using the PMK, the ANonce, and the SNonce as function parameters. After successful 4-way handshake, the AP sends group EAPOL-Key frame containing encrypted GTK to the station. The encryption key used to encrypt the GTK is a part of the preceding PTK.

4 Problems and Proposed Solutions

In a WLAN system with many WPA clients, a likely scenario is that AP supports the respective unicast keys and the common broadcast key. Fig. 4 shows the WLAN system including the WPA AP and many WPA stations.

WPA AP operating in the WPA WLAN system must support WPA key exchange procedure and may use the various cipher algorithms for unicast data. But the AP must use only one cipher algorithm and key for broadcast data. For example, if the stations supplicate the AP to exchange pairwise key and group key by using WPA authenticator key management state machine and each station uses each cipher algorithm as shown in Fig. 4, the AP must use the WEP as group cipher algorithm because the WEP is a common algorithm that all the stations can support. However, the AP uses the respective algorithm for unicast data according to the negotiation result between each station and the AP.

The WPA station employs WPA key exchange procedure shown in Fig. 3 as the key exchange procedure. The state machine related to 4-way handshake and group key handshake is WPA authenticator key management state machine. Fig. 5 shows the WPA authenticator key management state machine. According to the IEEE 802.11i specification, this state machine is responsible for group key exchange with the associated station.

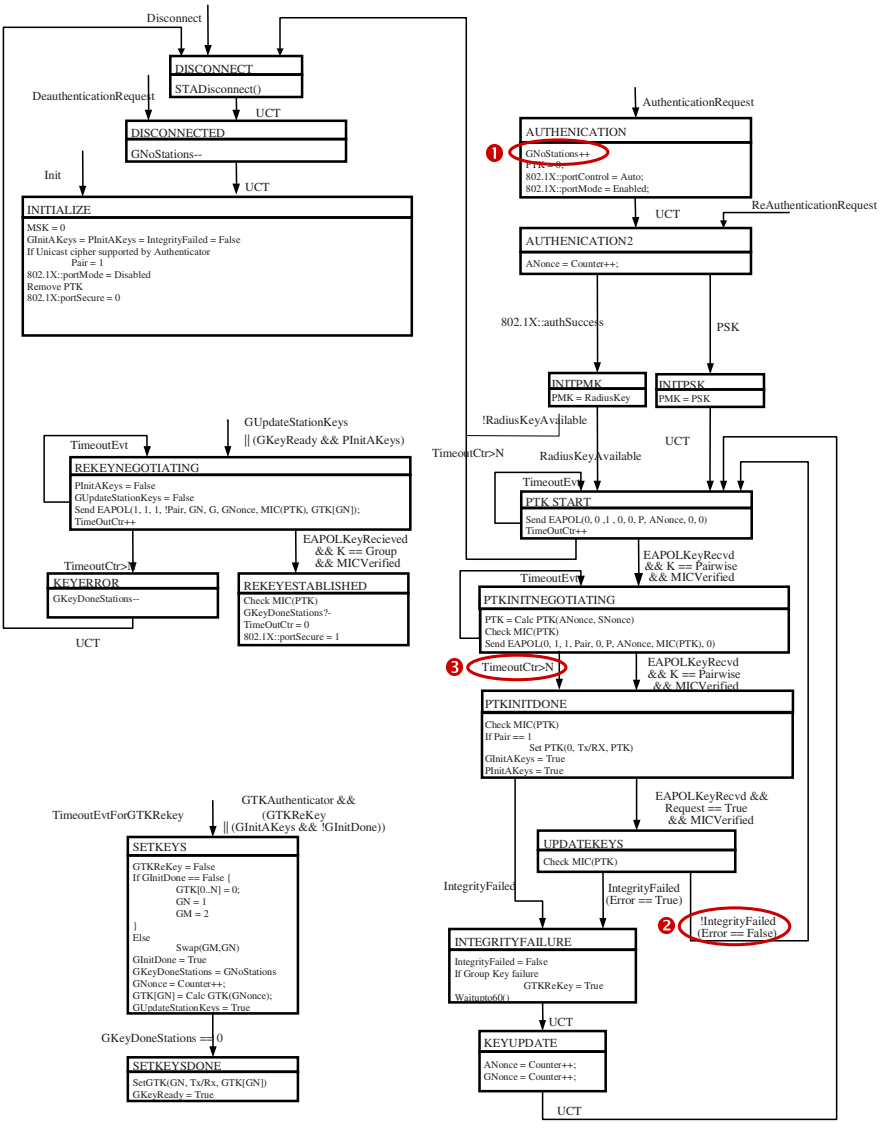


Fig. 5. Weak points in WPA authenticator key management state machine

Unfortunately, this state machine cannot support the stable group key setting and is vulnerable to the replay attack and DoS attack. In this section, 3 problems indicated by circles in Fig. 5 are described in detail and the alternative countermeasures are suggested, respectively. Fig. 6 shows the enhanced WPA authenticator key management state machine to which the alternative solutions are applied.

4.1 Vulnerability Due to the Number of Stations to Receive GTK

This weakness has a close relation to system stability. After the association of the first WPA station, this state machine will successfully support pairwise key exchange, PTK setting, group key exchange, and GTK setting. And then, the sequential WPA stations establish each PTK and the same GTK through the normal operation of this state machine. The normal operation means that the AP and the associated station acquire PMK after IEEE 802.1X authentication with EAP or PSK, the AP counts the authenticated station in the total number of stations to receive GTK, and the AP excludes the station succeeding in group key establishment from the total number of stations to receive GTK. Therefore, the AP fails to set new GTK when the AP counts the associated station in the total number of stations to receive GTK but it does not exclude the station from the number.

In Fig. 5, the variable of 'GNoStations' is the number of stations to receive GTK. This variable increases when the associated station wants the AP to be authenticated. The variable of 'GKeyDoneStations' is the number of stations left to have group key updated. In SETKEYS state, 'GKeyDoneStations' is substituted for 'GNoStations' and GTK is set in SETKEYDONE state when 'GKeyDoneStations' becomes zero.

However, a procedure with a station stays at AUTHENTICATION2 state if the AP fails to perform IEEE 802.1X authentication or the station does not request IEEE 802.1X authentication. In this case, though the variable of 'GNoStations' increases in AUTHENTICATION state the variable of 'GKeyDoneStations' for this station does not decrease because the procedure cannot reach REKEYESTABLISHED state including the operation of 'GKeyDoneStation--'. As a result of stay at AUTHENTICATION2 state, the AP fails to set new GTK.

A simple solution to resolve this problem could be to add a state transition from AUTHENTICATION2 to DISCONNECT when the authentication fails. This transition can decrease 'GNoStations' after the authentication failure. But this solution has a weak point of the processing overhead due to the disconnection and re-initialization of the station, whenever the authentication is retried.

In order to stably set the group key, we let the operation of 'GNoStaions++' be located in INITPMK state or INITPSK state as shown in Fig. 6. This modification makes the AP consider only the stations completing IEEE 802.1X authentication as an object of the operation of 'GNoStations++'. This solution can correctly control 'GNoStations' under consideration of the authentication result and does not require the additional processing overhead. In actual implementation, an indicator informing the AP that the operation of 'GNoStations++' was performed may be used. It helps the AP manage the operation of 'GNoStations++' and 'GNoStations--'.

4.2 Vulnerability Due to the Reuse of ANonce after PTK Update Request

This weakness has a close relation to the replay attack. PTK is calculated using the parameters such as PMK, ANonce, SNonce, AP hardware address, and

station hardware address. In case of the same PMK, new ANonce and new SNonce must be required to calculate new PTK. The transition line ❷ shown in Fig. 5 results from the station's PTK update request. The line directly reaches PTKSTART state from UPDATEKEYS state when the AP receives EAPOL-Key frame demanding PTK update. In other words, the existing state machine reuses ANonce when PTK update is required. The reuse of ANonce may be a potential defect such as a replay attack. For example, when an attacker who eavesdrops the previous 4-way handshake messages requests EAPOL-Key update and receives the first EAPOL-Key frame of 4-way handshake from the AP, he can successfully forge the second EAPOL-Key frame by using the previous messages because the first EAPOL-Key frame is the same frame as the previous frame.

In order to overcome this vulnerability, we let the line ❷ go through KEYUPDATE state to PTKSTART as shown in Fig. 6. If the AP receives EAPOL-Key frame demanding PTK update, ANonce increases in KEYUPDATE state. Therefore, the AP can send the first EAPOL-Key frame of 4-way handshake, containing new ANonce.

4.3 Vulnerability Due to the Incorrect Transition after Timeout Event

This weakness has a close relation to the DoS attack. The variable of 'TimeoutCtr' maintains the count of EAPOL-Key frame receive timeouts. If this value exceeds the defined number the AP must disconnect the station that is exchanging EAPOL-Key frames. However, PTKINITNEGOTIATING state is changed to PTKINITDONE in spite of the fact that the variable of 'TimeoutCtr' exceeds the limited number as shown in Fig. 5. The current incorrect transition cannot appropriately cope with the wrong EAPOL-Key frame. This problem makes a cause of the DoS attack because the AP continuously wastes its resource to process the bad EAPOL-Key frame.

In order to overcome this vulnerability, we made the AP's state be changed from PTKINITNEGOTIATING state to DISCONNECT state when the variable of 'TimeoutCtr' exceeds the limited number as shown in Fig. 6. That is, the AP disconnects the station that sends incorrect EAPOL-Key frames.

5 Conclusions and Future Works

Our researches on WPA authenticator key management state machine demonstrate that it is inappropriate to be implemented without modification. Fig. 6 shows the reconstructed WPA authenticator key management state machine to which our solutions are applied. The proposed countermeasures help the AP solve 3 problems investigated in this paper, such as the incorrect position of 'GNoStations++', the reuse of ANonce, and the incorrect treatment of timeout event.

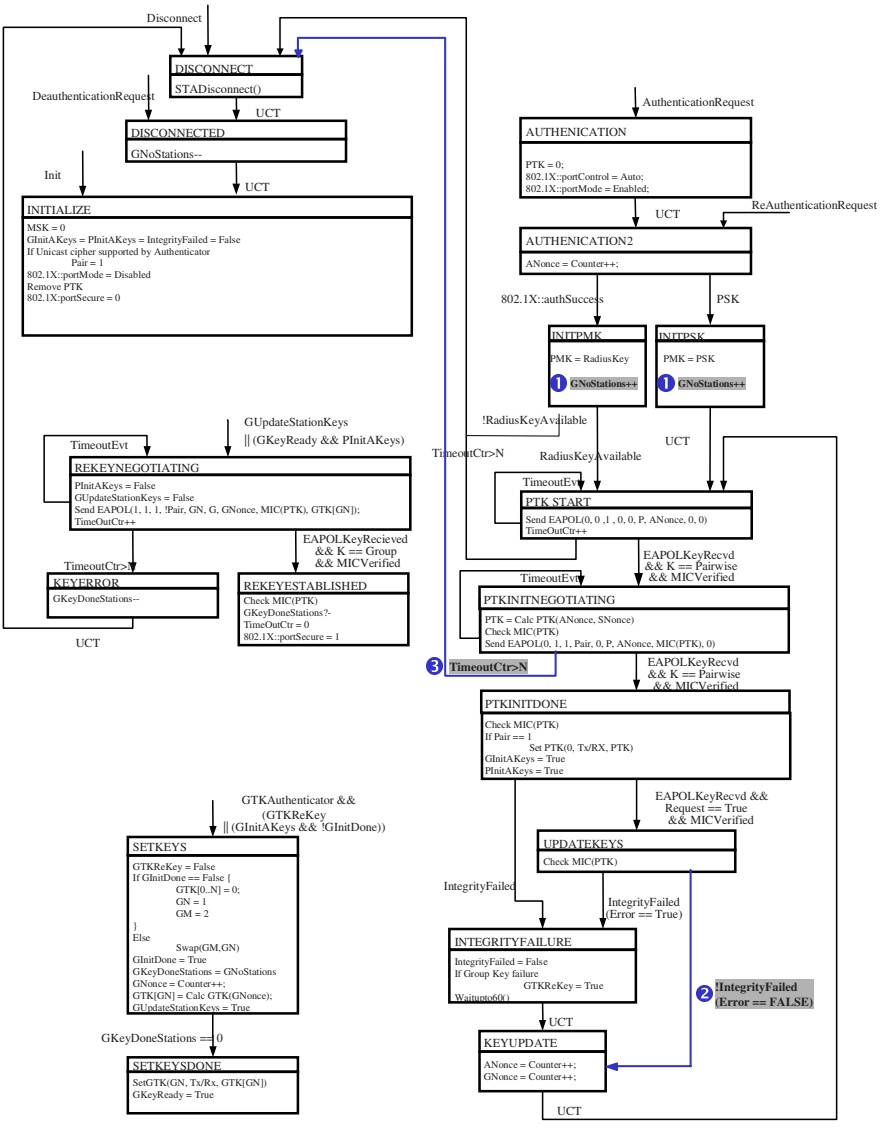


Fig. 6. Reconstruction of WPA authenticator key management state machine

The proposed state machine did not consider the functions supporting the technology related to secure hand-off or global roaming, though it can support user authentication, secure key exchange, and data confidentiality in single BSS (Basic Service Set). The IEEE 802.11 standards body is now working on significant improvements such as IAPP (Inter-Access Point Protocol) and pre-authentication [12, 13]. Therefore, the further study is expected to advance in

order to support distributed authentication for global roaming, context transfer for pre-authentication, improved cipher algorithm.

References

[1] ISO/IEC 8802-11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ISO/IEC 8802-11 (1999) 915
[2] J. R. Walker: Unsafe at any key size; An analysis of the WEP encapsulation. IEEE 802.11-00/362 (2000) 915
[3] IEEE 802.11: LAN/MAN Specific Requirements- Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specification: Specification for Enhanced Security. IEEE Std 802.11i/D3.0 (2002) 915, 916, 918
[4] Wi-Fi Alliance: Wi-Fi Protected Access (WPA). WPA Version 2.0 (2003) 915
[5] Wi-Fi Alliance: Overview Wi-Fi Protected Access. http://www.wi-fi.org/OpenSection/pdf/Wi-Fi_Protected_Access_Overview.pdf 915
[6] IEEE 802.1: Standard for Local and metropolitan area networks- Port-Based Network Access Control. IEEE Std 802.1X (2001) 916, 917
[7] L. Blunk and J. Vollbrecht: PPP Extensible Authentication Protocol (EAP). IETF (1998) 916
[8] C. Rigney: Remote Authentication Dial In User Service (RADIUS). IETF (2000) 916
[9] B. Aboba, D. Simon: PPP EAP TLS Authentication Protocol. IETF (1999) 917
[10] G. Pall, G. Zorn: Microsoft Point-To-Point Encryption (MPPE) Protocol. IETF (2001) 917
[11] IEEE 802.1: Standard for Local and metropolitan area networks- Port-Based Network Access Control- Amendment 1: Technical and Editorial Corrections. IEEE P802.1aa/D6.1 (2003)
[12] IEEE 802.11: Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. IEEE Std 802.11f/D5 (2003) 923
[13] B. Aboba: IEEE 802.1X Pre-Authentication. IEEE 802.11-02/389r1 (2002) 923

Optimizing Authentication Mechanisms Using ID-Based Cryptography in Ad Hoc Wireless Mobile Networks

WonJun Lee[1⋆] and Wiroon Sriborrirux[2]

[1] Korea University, Seoul, South Korea
wlee@korea.ac.kr
[2] Burapha University,Chonburi, Thailand
wiroon@buu.ac.th

Abstract. To achieve reliable and secure authenticated broadcast routing, we propose a novel certificate-based scheme in AODV (in short, CBS-AODV) and an identity-based signature scheme in AODV (IBS-AODV)[1], which apply the use of ID-based cryptography to abate overhead effect by exchanging the certificate public key. Using our proposed schemes, we could reduce the routing load up to 24.8as well as dramatically save the storage space consumption of mobile node. Our protocols can reduce the size of key while providing the same security level as that of RSA and DSA.

Keywords: Ad Hoc networks, secure routing, CBS-AODV, IBS-AODV, elliptic curve.

1 Introduction

IP-based computer communication over wireless network, which started in the 1970's is a research subject that has become ever more interesting over the last few years because of the fast growing Internet. The many possible applications for networks of this type, for example mobile conferencing scenarios, search-and-rescue operations, disaster scenarios and police matters, have created need for efficient routing protocols. The concept of Ad hoc network [1] is attractive due to the following reasons: ease of deployment, speed of deployment, and decreased dependence or infrastructure. The range of applications varies from military to commercial purposes. Also since the mobile devices are extremely limited in computational resource, and the available location-limited channels do not permit trusted exchange of secret data, the impromptu nature of the Ad hoc network formation thus makes it hard to distinguish between trusted and non-trusted nodes. In general, nodes may leave and join the network at will. Due to dynamic nature of Ad hoc networks, the trust relationship between

⋆ Corresponding Author

[1] This work was supported by grant No. R01-2002-000-00141-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 925–934, 2004.

nodes also changes, so any security solution with static configuration would not suffice. It is desirable that the security mechanisms adapt on the fly to those changes. They should also be scalable to handle large networks. Those current proposals for authenticated broadcast to the entire in network can be seen that they are impractical for Ad hoc network. Therefore, security is a very critical issue in any wired/wireless networks. The dynamic nature of Ad hoc networks makes it very challenging to ensure secure communication in these networks. There are several security aspects to secure including the end-to-end data authentication, routing security and link level security. Especially in routing protocols, a node and its neighbors trust to each other. It will trust that its neighbors will forward packets for it and also assumes that the received packets from its neighbors are all authentic. Unfortunately, the native trust model allows an adversary to inject erroneous routing requests into a network, which can paralyze the entire network. Thus recently several research groups [3], [4], [5], [6] have proposed security extensions that include mechanisms for authenticating the routing control packet in the network to deal with such attacks. One intuitive solution is to authenticate all packets so that a node only forwards packets from authorized nodes. So an attacker cannot insert spurious information into the network, because its neighbors will drop packets immediately if verification fails. This paper places important in securing the routing protocol, using AODV (Ad hoc On Demand Distance Vector) [2] before starting sending user data between nodes, that helps in achieving authentication with minimal overheads and security requirements' satisfaction. The remainder of this paper is organized as follows: Related work is discussed in Section 2. In Section 3, we propose our two authentication mechanism schemes. In Section 4, the algorithm of each proposed scheme will be presented and the simulation result for each situation will be discussed. We give conclusions with lessons learned in Section 5.

2 Related Work

One of traditional solutions is to use a network-wide key shared by all nodes [7][11] based on symmetric cryptographic mechanisms. Idea behind these mechanisms is that each node uses this shared key to compute message authentication codes (MACs) appended along with packets. MACs based on cryptography could identify and authenticate nodes that participate in the routing, thereby detecting the fabricated and distorted information and preventing nodes from impersonation [3]. Moreover, Encryption could protect routing messages from disclosure. Auditing combined with authentication could detect non-cooperative behaviors from nodes, such as dropping packets [6]. After that when received by receiver, packets will be verified using the same shared key whether it is authenticated. These schemes however have several disadvantages. First, an outsider breaking the global key could break down the protocol. Second, it is difficult to identify the compromised node when the global key is divulged. Third, it is expensive to recover from a compromise because it will invoke a global re-key, as described in [12]. Another traditional solution is to use a authentication techniques based on

Table 1. The notations of two proposed schemes

Notation	Definition
A, B, C, D	Specific mobilenodes
$Digest_A$, X	Digestmessage
$A \rightarrowtail B$	Node A sends data to node B
$Sign_{PrivKey_B}$,Sign	Signature generated by using A's private key
Hash(ID_{node})	Hash message of node's identity
S_{ID}, Q_{ID}	Identifier based key pair
$Cert_A$	Public Certificate
$GroupKey_{ID}$	Group Key based on identity
P, Q	Points on Elliptic Curve [21]
$\hat{e}$(P,Q)	Bilinear mapping based on the Tate Pairing
G_1, G_2	Groups of prime order q (Additive notation, multiplication notation)
s	System master secret

public key or asymmetric key cryptography. As Hu, Perrig and Johnson [7] proposed to use an online trusted KDC to help establish trust relationship between pairs of nodes. Unfortunately, such techniques are impractical and do not adept well to Ad hoc networks having limited characteristics, as we are going to show it in simulation environment. Another scheme proposes that each node select a number of certificates to store. The public key s obtained when a chain of certificates is discovered because every pair of nodes will merge the two certificate repositories [5]. Also from [14] in 1984 and [15], an Identity-based signature was proposed.

3 Proposed Authentication Mechanism Schemes

The notations shown in Table 1 will be used throughout this paper.

In this paper, we evaluate two main applied security mechanisms in Ad hoc network based on AODV routing protocol. First, we apply the use of certificate-based model with digital signature verification when two nodes (source and destination) desire to communicate to each other in Ad hoc network. Before sending messages, AODV routing protocol will be invoked to discover the proper route between them by flooding the Route Request message (RREQ) to source's neighbors until reaching the destination. Destination then replies the source by sending (unicasting) Route Reply message (RREP) to the route that it receives the fastest RREQ message. During these processes, certificate-based scheme is used to provide the verification of digital signature for routing authentication in Ad hoc network. Second, to optimize the performance particularly routing overhead, we proposed to use of Identity-based signature scheme, which will be explained after evaluating this first proposed security scheme. Since a certificate system providing an asymmetric digital signature before using the public key of

a user, the participants so must first verify the certificate of the user. As consequence, this system requires a large amount of computing time, exchanging the public certificate (CA) for verification, and storage. Also it leads to high communication overhead as well as routing overhead. These schemes are based on the following assumptions. The network consists of a group of mutually trusting nodes. For our first proposed certificate-based scheme; CBS-AODV, before each node participate in the network, it has to be given a public/private key pair by the authority. Also for two system keys created by the authority, first the system private key is used to sign the public keys of all nodes and the second system public key, which is used when two nodes desire to communicate at first time by exchanging the certificate public key, is stored in all nodes. Earlier mentioned, they are taken place off-line before the node can join the network. For our second proposed identity-based scheme; IBS-AODV, all nodes keep the system parameters G_1, G_2, $\hat{e}$, P, $P_{pub} = s.P$ where s is the system master secret. Each node registers with the authority and obtains a private key s.Qnode = s.Hash(IDnode). Also, they are taken place off-line before the node can join the network. Whenever, two nodes desire to communicate to each other, they do not need the certificate exchange but they will use the public key generated from sender's ID to verify the received signature as explained in next section. All inks between the nodes are bi-directional. The nodes have enough power energy and computationally powerful enough to execute our security algorithms.

3.1 Proposed Certificate-Based Scheme in AODV Routing Protocol (CBS-AODV)

To provide secure routing authentication without the need of an encryption layer, in this thesis we apply to use an asymmetric digital signature and public certificate key method exchange for verification of signature. The procedures of route discovery in AODV are shown below.

Step 1:
$A \rightarrowtail B$: $[Digest_A]Sign_{PrivKey_A}, Cert_A$ where,
$Digest_A$=HMAC($GroupKey_{ID}$,[$RREQ|Nonce_A|$timestamp]

When B node receives it, B first verifies the A's certificate using the CA's public key and validate the signature signed with A's Private key, and then B will record A's Nonce and take A's public key from extracting A's certificate by using CA's public key as followed traditional certificate-based scheme.

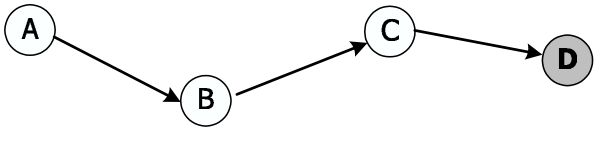


Fig. 1. Simple 4-nodes network

Step 2:
$A \leftarrowtail B$: [X]$Sign_{PrivKey_B}$,$Cert_B$ where,
X=HMAC($GroupKey_{ID}$,[$Nonce_B \oplus$timestamp]

For generating one-way hash code using $GroupKey_{ID}$, which is secret key shared in Group ID before attendants enter to the conference scenario (e.g. Conference Room no. 1; Key group ID no. 1). Typically the authentication tag will be transmitted along with a ACK message and B's Nonce (Pseudorandom number) signed with B's Private key to allow the receiving node (means A node) to verify the message's authenticity by extracting B's Public key from $Cert_B$ using the CA's public key. After that, B node will make reverse path toward A node.

Step 3:

A verifies B's signature and checks X value received from B node whether is equal. If yes, A and B node will authenticate to each other. We proposed to use sending ACK message back to previous node in order to make the connection between every pair of nodes in the network has more private connection and also we believe that it could protect some internal attacker who know group key for example. The method is between A and B node, they should update Key group ID by using X value, which is generated from A's Nonce and B's nonce, to XOR Key group ID to be new key of both. Finally, A node can send data packet and message digest of data packet, which is used with new updated group Key ID in order B node to check message digest of data packet sent by A node whether is equal or not. However, this work is not our concern in this thesis.

Step 4:
$B \rightarrowtail C$: [$Digest_B$]$Sign_{PrivKey_B}$,$Cert_B$ where,
$Digest_B$=HMAC($GroupKey_{ID}$,[$RREQ|Nonce_B|$timestamp]

...

Step n:
$A \leftarrowtail ... \leftarrowtail D \leftarrowtail X$: [$Digest_X$]$Sign_{PrivKey_X}$,$Cert_X$ where,
$Digest_X$=HMAC($GroupKey_{ID}$,[$RREQ|Nonce_X|$timestamp]

Until reaching the destination (X), the destination then generates the signature along with the RREP message the same as RREQ broadcasting. However, they need no to use ACK message during unicasting this RREP message to source (A node) because now the route to source has more enough secure due to using ACK message between each pair.

3.2 Proposed Identity-Based Signature Scheme in AODV Routing Protocol (IBS-AODV)

In this thesis, we built the code in *C* programming style with *GNU MP Library* for arbitrary precision arithmetic and also *IBE Library* for Tate pairings

computation. From [10], we then implemented the method of creating system parameters, signing a message and verifying the signature as full codes implemented by us. To provide secure routing authentication without the need of the certificate exchange, we implement by using the techniques as explained above. The procedures of route discovery in AODV are similar as our first proposed certificate-based scheme.

3.2.1 IBS-AODV from Pairings on Elliptic Curve

The Tate paring is operations on pairs of points of the same order on an elliptic curve. See [27] for some early applications, especially in breaking certain elliptic curve based cryptosystems. Let G1 and G2 denote two groups of prime order q in which the discrete logarithm problem is believed to be hard and for which there exists a computable bilinear map

$$\hat{e} : G_1 \times G_1 \mapsto G_2.$$

A good source of groups and pairings is elliptic curves, in particular, *supersingular* elliptic curves. Elliptic curves have long been used in cryptography, to produce public-key cryptosystems and protocols that are more bandwidth efficient or less processor intensive than primitives like Diffie-Hellman or RSA. Moreover, we used elliptic curves in this thesis since the Elliptic curve discrete logarithm problem (ECDLP) appears to be significantly harder than the DLP, the strength-per-key-bit is substantially greater in elliptic curve systems than in conventional discrete logarithm systems.

Moreover, an elliptic curve $E(Z_P)$ with a point P $E(Z_P)$ whose order is a 160-bit prime offers approximately the same level of security as DSA with 1024-bit modulus p and RSA with a 1024-bit modulus n. Thus, smaller parameters can be used in elliptic curve cryptosystems than with old discrete logarithm systems but with equivalent levels of security. The advantages that can be gained from smaller parameters include speed (faster computations) and smaller keys. These advantages [28] are especially important in environments where processing power, storage space, bandwidth, small hardware processors, and power consumption are constrained like the Ad hoc network.

– *Types of Public/Private Key Pairs*

We require the following two types of keys: A standard public/private key pair is a pair (R, r) where R G_1 and r F_q with $R = rP$ For some given fixed point P G_1.

An identifier based key pair is a pair (Q_{ID}, S_{ID}) where Q_{ID}, S_{ID} G_1 and there is some trust authority (TA) with a standard public/private key pair given by (R_{TA}, s), such that the key pair of the trust authority and the key pair of the identifier are linked via $S_{ID} = \mathrm{s}.Q_{ID}$ and $Q_{ID} = H_1$ (ID), where ID is the identifier string.

3.2.2 IBS-AODV Authentication Algorithm

At first, the TA publishes system parameters G_1, G_2, $\hat{e}$, P, $P_{pub} = s.P$ where s is the system master secret. We also need two more cryptographic hash functions, H_2 which maps arbitrary strings onto integers. As usual A registers with the TA and obtains a private key $s.Q_A = s.H(ID_A)$.

Where we have used the bi-linearity properties of $\hat{e}$. Thus a valid signature will always satisfy the check. Notice that signing does not involve any pairing calculations, and so can be very quickly even on low-end processors. Signatures are also rather short - the size of only two group elements from G_1. Verification computations can also be reduced, down to a single pairing plus an exponentiation in G_2 and a few hash function calls, if one is verifying many signatures from a fixed entity A.

4 Simulations and Results

In this paper, we implemented AODV routing protocol to support two kinds of proposed security schemes incorporated in the version 2 .1b9a based on RedHat version 7.3 Operating System with Pentium III 700 MHz. Also the Ns-2 distribution used is the ns-allinone distribution, version 2.1b9a. In addition, we took the incorporation from OpenSSL in providing the traditional signing/verification signature as well as IBE Open source with GNU MP Library for alternative public key cryptography. The radio model uses characteristics similar to a commercial radio interface, the 914MHz Lucent's WaveLANTM DSSS radio interface. WaveLAN is modeled as a shared-media radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 meters. In our experiments for CBS-AODV, 30 and 27 nodes move around in a rectangular area of 900m X 800m according to a mobility models i.e., Random waypoint, Fixed waypoint, and Brownian [20]. For the work related to energy-aware routing we assume long-lived sessions. The session sources are CBR (constant bit rate) and generate UDP packets at 4 packets/sec with each packet being 512 bytes long in 900-second simulated time. To make our simulation more practical, we then created the scenario similar to some convention events. The people's behavior could move toward some place along with another like moving group and also some people could move toward some place independently as seen in Fig.2 below. Group of nodes, which is moving, use fixed waypoint mobility model and another use Brownian mobility model in this scenario. As shown in fig. 3, the conference scenario also was created in our simulation. We created this event let have 2 speakers (node 0 and node 26) who move in horizontal and vertical direction using fixed waypoint respectively. There are 25 audiences participating in this conference. The audiences' movement behavior could move toward some place independently using Brownian mobility model.

After each simulation was processed, trace files recording the traffic and node movement are generated (almost 3 GB). We then parsed those trace files in order to extract the information and statistics needed to measure each performance

Table 2. Simulation variables of each scenario

Scenario Type	Convention Scenario	Conference Scenario
Nodes	30	27
Mobility model	Pursue and Brownian	Fixed Waypoint and Brownian
Pause time(s)	2s	2s
Speed (m/s)	2-8 m/s (1 group) and 2-5 m/s (24 individual guests)	2-10 m/s (2 speakers) and 2-5 m/s (25 audiences)
Connections	5,10,15,20,25	5,10,15,20,25

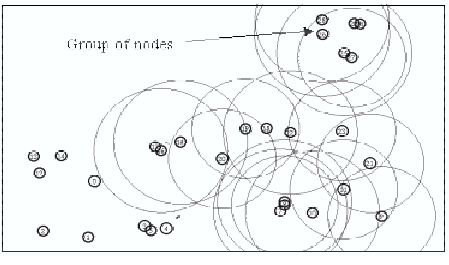


Fig. 2. Packet Sending in Convention Scenario

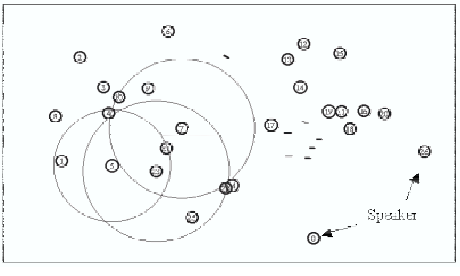


Fig. 3. Packet Sending in Conference Scenario

metric and then fed the Gnu and graph plotter with those results. We started simulating next with a convention scenario and a conference scenario respectively.

The simulation results bring out some important characteristic differences when we apply the security extension to Ad hoc network. Let us assume in our first proposed CBS-AODV, we used 16-byte group key, 32-byte MAC, 64-byte (512 bit RSA) for digital signature, and 256-byte public certificate. For the second proposed IBS-AODV, we used 16-byte group key, 20-byte MAC, and 168-byte digital signature. We believe this amount of overhead extended in AODV routing protocol is reasonable for a security service. Performance metrics include:

- **Packet delivery fraction Effect:** Our certificate-based scheme could work well in two scenarios because the effect of throughput of the network is small around 7-11of connections between nodes were increased from 5 connections to 25 connections.
- **End-to-end delay Effect:** This is the average End-to-End delay of each scenario. The results are fairly low between with authentication and without authentication extension.
- **Normalized routing load Effect:** The number of routing packets increases when our scheme is incorporated. We computed the results by using mathematical equation called polynomial (power 2).

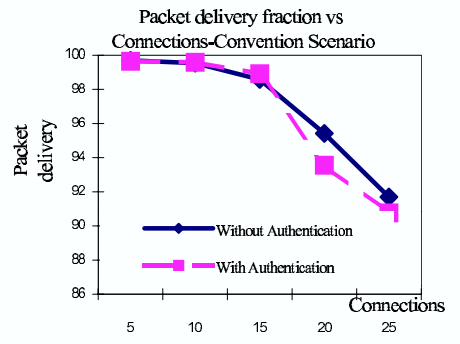


Fig. 4. PDF (percentage) vs Connections in Convention Scenario

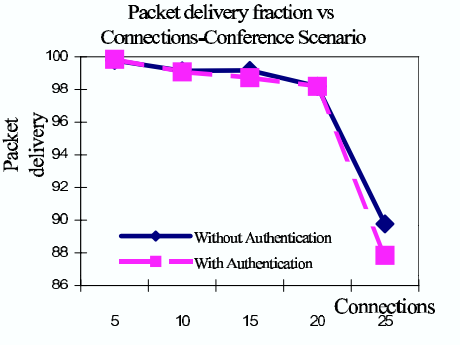


Fig. 5. PDF (percentage) vs Connections in Conference Scenario

5 Conclusions

In the first proposed CBS-AODV, every node has to maintain a list of certificate public key of many other nodes. It thus could cause the limited storage of mobile node. In addition, every pair of nodes must exchange its certificate public key to each other in order to let them verify the signature. This also could cause the very high overhead to the network when the size of network and number of nodes become large. However, we then proposed the use of uniquely identity as the public key based on pairings on elliptic curve digital signature having very short ciphertexts/signatures and efficient computation times. Our second proposed IBS-AODV can eliminate the storage consumption and the certificate public keys exchange (an RSA public key plus the CA's signature thereon requires 2048 bits; an identity can be 168 bits) dramatically when the network scalability is increased. The advantages that can be gained from smaller parameters include speed (faster computations) and smaller keys.

Acknowledgements

We would like to thank Dr. Jeff Boleng for his valuable suggestions, Dr. Kenneth G. Peterson for accepting codes designed/implemented by us, and Dr. Benjamin Lynn for providing IBE libraries for developing the Tate pairings.

References

[1] Charles E. Perkins, Ed., "Ad hoc Networking", Addison Wesley, 2001.
[2] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das., "Ad hoc Demand Distance Vector (AODV) Routing", http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt, Internet draft, Internet Engineering Task force, February 2003. Work in Progress.

[3] P. Papadimitratos and Z. J. Hass, "Secure Routing for Mobile Ad Hoc Networks", SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 2002.
[4] D. Balfanz, D. K. Smetters, P. Stuart, H. C. Wang, "Talking to Strangers: Authentication in Ad Hoc Networks", Network and Distributed System Security Symposium, San Diego, CA, February 2002.
[5] J. P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks", 2nd MobiHoc, Long Beach, CA, October 2001.
[6] S. Bhargava and D. P. Agrawal, "Security Enhancement in AODV Protocol for Wireless Ad Hoc Netoworks", Vehiculat Technology Conference, vol. 4, 2001.
[7] Y. Hu, A. Perrig, D. Johnson., "Ariadne: A Secure On-demand Routing Protocol for ad-hoc networks", MobiHoc 2002.
[8] B.Dahill, B. Levine, E. Royer, C. Shields, "A secure Routing Protocol for ad-hoc networks", Technical report, UMass-CS-2001-037.
[9] Y. Hu, A. Perrig, D. Johnson., "SEAD: Secure Efficient Distance Vector Routing for mobile wireless Ad hoc netowkrs", June 2002.
[10] S. Yi, P. Nadurg, R. Kravets., "Security-Aware-Ad-Hoc routing for wireless networks", Technical Report, UIUCDCS-R-2001-2241, UILU-ENG-2001-1748.
[11] S. Basagni, K. Herrin, E. Rosti, D. Bruschi,, "Secure Pebblenets", MobiHoc 2001.
[12] Tony Larsson, Niklas Hedman, "Scenario-based performance analysis of routing protocols for mobile Ad hoc networks", August 1999.
[13] Levente Buttyan, Jean-Pierre Hubaux, "Report on a Working Session on Security in Wireless Ad hoc Networks", Mobile Computing and Communication Review Volumn 6 Number 4, 2002.
[14] Shamir, "Identity-based cryptosystems and signature schemes", Advances in Cryptology (Proceedings of Crypto '84), Lecture Notes in Computer Science, vol 196, Springer-Verllag, 1985.
[15] D. Boneh, M. Franklin, "Identity based encryption from the Weil pairing", extended abstract in Advances in Cryptology - Crypto 2001, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, pp. 231-229, August 2001.
[16] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks" Proc. 7th Ann. Intl. Conf. Mobile Computing and Networks (MobiCom 2001), Rome, Italy, 2001.
[17] K. G. Paterson., "ID-based signatures from pairings on elliptic curves". Electronics Letters, 38(18):1025-1026, 2002.
[18] K. G. Paterson., "Cryptography from Pairings: A Snapshot of Current Research", Nuffield Foundation, November 2001.
[19] A. J.Menezes, "Elliptic Curve Public Key Cryptosystems", Kluwer International series in Engineering and Computer Science, 1993.
[20] Ratish J. Punnose, Pavel V. Nikitin, Josh Broch, Daniel D. Stancil, "Optimizing Wireless Network Protocols Using Real-Time Predictive Propagation Modeling", Carnegie Mellon University Technical Report, 1999.
[21] Don Johnson, Alfred Menezes and Scott Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom Research, Canada, 2001.

Robust Remote User Authentication Scheme

Eun-Jun Yoon, Eun-Kyung Ryu, and Kee-Young Yoo

Department of Computer Engineering, Kyungpook National University,
Daegu 702-701, Republic of Korea
{ejyoon,ekryu}@infosec.knu.ac.kr
yook@knu.ac.kr

Abstract. Recently, Wu and Chieu proposed an improvement to Sun's scheme, whereby users could choose and change their passwords freely through a secure channel when using a remote system. However, this improved scheme is still susceptible to impersonation attacks and does not provide mutual authentication. Accordingly, the current paper demonstrates the vulnerability of Wu and Chieu's scheme to impersonation attacks and presents an enhancement to resolve such problems. As a result, the proposed scheme enables users to update their passwords freely without the help of a remote system, while also providing mutual authentication.

1 Introduction

User authentication is an important part of security, along with confidentiality and integrity, for systems that allow remote access over untrustworthy networks, like the Internet. As such, a remote password authentication scheme authenticates the legitimacy of users over an insecure channel [1-9], where the password is often regarded as a secret shared between the remote system and the user. Based on knowledge of the password, the user can use it to create and send a valid login message to a remote system to gain the right to access. Meanwhile, the remote system also uses the shared password to check the validity of the login message and authenticate the user. In 1981, Lamport [1] proposed a remote password authentication scheme using a password table to achieve user authentication. In 2000, Hwang and Li [7] pointed out that Lamport's scheme [1] suffers from the risk of a modified password table and the cost of protecting and maintaining the password table. Therefore, they proposed a new user authentication scheme using smart cards to eliminate the risk and cost. Hwang and Li's scheme can withstand replaying attacks and also authenticate users without maintaining a password table. Later, Sun [8] proposed an efficient smart card-based user authentication scheme to improve the efficiency of Hwang and Li's scheme [7], and more recently, Wu and Chieu [10] proposed an improvement on Sun's scheme [8] to make the protocol a user-friendly remote authentication scheme through which the user can choose and change their password based on a secure channel. They claimed that their scheme provided effective authentication and also eliminated the drawback of Sun's scheme [8] that requires the

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 935–942, 2004.

assignment of unfriendly lengthy passwords. However, Wu-Chieu's scheme is vulnerable to an impersonation attack and does not provide mutual authentication, plus a user can only update their password through a secure channel when using a remote system. Accordingly, the current study demonstrates that Wu-Chieu's scheme is susceptible to impersonation attacks, where an attacker can easily impersonate other legal users to access the resources at a remote system, then presents an enhancement to the scheme to isolate such problems. The proposed scheme enables users to freely update their passwords without the help of a remote system, and also provides mutual authentication.

The remainder of this paper is organized as follows: Section 2 briefly reviews Wu-Chieu's scheme, then Section 3 demonstrates an impersonation attack on Wu-Chieu's scheme. The proposed scheme is presented in Section 4, while Section 5 discusses the security and efficiency of the proposed scheme. Some final conclusions are given in Section 6.

2 Wu-Chieu's Scheme

This section briefly reviews Wu-Chieu's scheme, which has a registration, login, and authentication phase, as explained in the following:

Registration Phase: The user U_i submits their identifier ID_i and chosen password PW_i to the remote system. These private data must be sent in person or over a secure channel. Upon receiving the registration request, the system performs the following steps:

1. Compute $A_i = h(ID_i,\ x)$, where x is a secret key maintained by the system and $h(\cdot)$ is a collision resistant one-way hash function.
2. Compute $B_i = g^{A_i \cdot h(PW_i)} (\mathrm{mod} p)$, where p is a large prime number, and g is a public, primitive element in $GF(p)$.
3. The system then personalizes the smart card with the secure information: $\{ID_i, A_i,\ B_i,\ h(\cdot),\ p,\ g\}$.

Login Phase: If the user U_i wants to login, they attach their smart card to the card reader and key in their identifier ID_i and password PW_i^*, then the smart card performs the following operations:

1. Compute the following two integers:
   $B_i^* = g^{A_i \cdot h(PW_i^*)} (\mathrm{mod} p)$ and $C_1 = h(T \oplus B_i)$, where T is the current date and time of the input device.
2. Send a message $m = \{ID_i,\ B_i^*,\ C_1,\ T\}$ to the remote system.

Authentication Phase: Upon receiving message m at time T', the remote system authenticates the user based on the following steps:

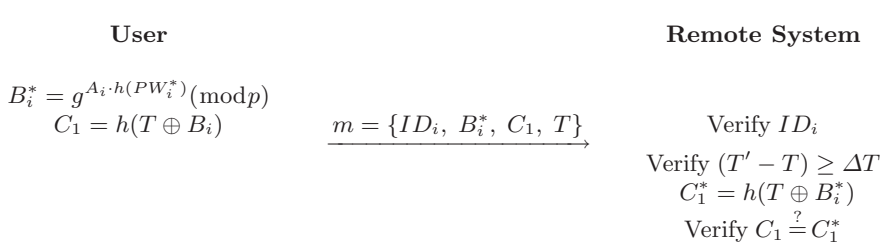


Fig. 1. Login and Authentication Phases in Wu-Chieu's scheme

1. Verify the format of ID_i. If the format is incorrect, the system rejects the login request.
2. Verify the validity of the time interval between T and T'. If $(T' - T) \geq \Delta T$, where ΔT denotes the expected valid time interval for a transmission delay, the remote system rejects the login request.
3. Compute $C_1^* = h(T \oplus B_i^*)$, and compare C_1 and C_1^*. If they are equal, this indicates that the password PW_i^* is equal to PW_i, then the system will accept the login request; otherwise the login request is rejected.

According to Wu-Chieu's scheme, when an authorized user U_i wants to change their password, they have to perform the following procedures:

1. The user just submits their smart card to the system and chooses a new password PW_i' via a secure channel.
2. The remote system then performs the new B_i' as $B_i' = g^{A_i \cdot h(PW_i')} (\mathrm{mod} p)$ and writes the new B_i' into the user U_i's smart card to replace the original B_i. After the replacement of B_i, the user U_i can use the new password PW_i' to login.

3 Impersonation Attack on Wu-Chieu's Scheme

This section demonstrates that Wu-Chieu's scheme is vulnerable to an impersonation attack, where an attacker can easily impersonate other legal users to access the resources at a remote system.

Suppose that an attacker has eavesdropped a valid message $m = \{ID_i,\ B_i^*,\ C_1,\ T\}$ from an open network. In the Login Phase, the attacker can choose a random number X and let $B_A^* = X$. Then, they let $C_A = h(T_A \oplus B_A^*)$ and send a message $m = \{ID_i,\ B_A^*,\ C_A,\ T_A\}$ to the remote system, where T_A is the attacker's the current date and time for succeeding with Step 2 of the Authentication Phase. It is easy to check whether the system will accept this message, as $C_A = C_A^* = h(T_A \oplus B_A^*)$. Finally, the system accepts the attacker's login request, making Wu-Chieu's scheme insecure.

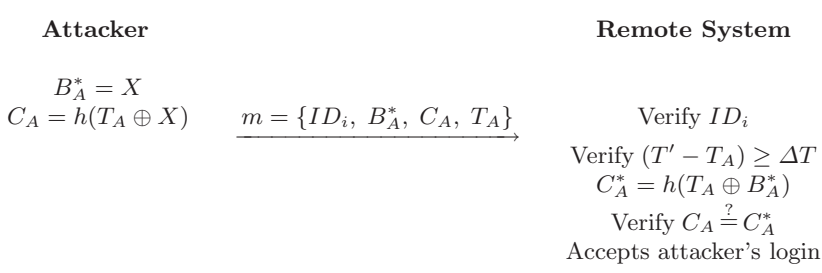


Fig. 2. Impersonation attack on Wu-Chieu's scheme

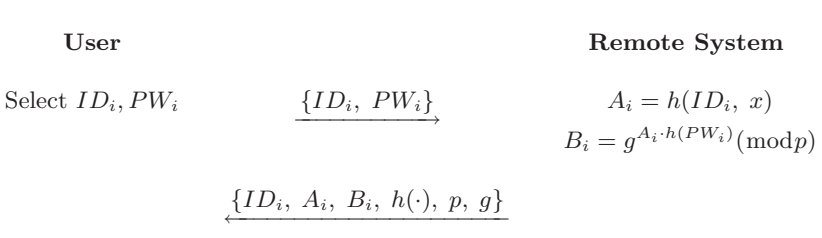


Fig. 3. Registration Phase in proposed scheme

4 Proposed Scheme

This section proposes an enhancement to Wu-Chieu's scheme that can withstand an impersonation attack. In addition, the proposed scheme also allows users to update their passwords freely without the help of a remote system and provides mutual authentication between the user and a remote system. The security of the proposed scheme is based on a one-way hash function and discrete logarithm problem, and consists of a registration, login, and verification phase.

Registration Phase: Like Wu-Chieu's scheme, let x be a secret key maintained by the system. The user U_i submits their identifier ID_i and chosen password PW_i to the system. These private data must be sent in person or over a secure channel. Upon receiving the registration request, the system performs the following steps:

1. Compute $A_i = h(ID_i,\ x)$, where x is a secret key maintained by the system and $h(\cdot)$ is a collision resistant one-way hash function.
2. Compute $B_i = g^{A_i \cdot h(PW_i)} (\mathrm{mod} p)$, where p is a large prime number, and g is a public, primitive element in $GF(p)$.
3. The system personalizes the smart card with the secure information: $\{ID_i,\ A_i,\ B_i,\ h(\cdot),\ p,\ g\}$.

Login Phase: If the user U_i wants to login, they attach their smart card to the card reader and key in their identifier ID_i and password PW_i^*, then the smart card performs the following operations:

1. Compute the following two integers:
   $B_i^* = g^{A_i \cdot h(PW_i^*)} (\mathrm{mod} p)$
   and
   $C_1 = h(T, A_i, B_i)$,
   where T is the current date and time of the input device.
2. Send a message $m = \{ID_i,\ B_i^*,\ C_1,\ T\}$ to the remote system.

Verification Phase: Upon receiving the authentication request message $m = \{ID_i,\ B_i^*,\ C_1,\ T\}$, the remote system and smart card execute the following steps for mutual authentication between the user U_i and the remote system.

1. The system verifies the format of ID_i. If the format is incorrect, the system rejects the login request.
2. The system verifies the validity of the time interval between T and T'. If $(T' - T) \geq \Delta T$, where ΔT denotes the expected valid time interval for a transmission delay, the remote system rejects the login request.
3. The system computes the following two integers:
   $A_i^* = h(ID_i,\ x)$
   and
   $C_1^* = h(T, A_i^*, B_i^*)$,
   and compares C_1 and C_1^*.
   If they are equal, this indicates that the password PW_i^* is equal to PW_i. The system then accepts the login request and proceeds to Step 4, otherwise it rejects the login request.
4. The system acquires the current time stamp T'' and computes
   $C_2 = h(T'', A_i^*, C_1^*)$. The system sends back the message $\{C_2,\ T''\}$.
5. Upon receiving the message $\{C_2,\ T''\}$, the user U_i verifies the validity of the time interval between T'' and T''', then computes $C_2^* = h(T'', A_i, C_1)$ and compares C_2 and C_2^*. If they are equal, the user U_i believes that the responding part is the real system and the mutual authentication is complete, otherwise the user U_i interrupts the connection.

If the user U_i wants to change their password, they only need to perform the procedures below, without any help from the remote system:

1. Compute $V = (B_i)^{h(PW_i^*)^{-1}} = g^{A_i \cdot h(PW_i) \cdot h(PW_i^*)^{-1}} = g^{A_i} (\mathrm{mod} p)$.
2. Compute g^{A_i} using stored A_i and verify $V = g^{A_i}$.
3. Select new password PW_i' and compute $h(PW_i')$.
4. Compute $B_i' (= g^{A_i \cdot h(PW_i')} (\mathrm{mod} p))$.
5. Store B_i' in smart card in place of B_i.

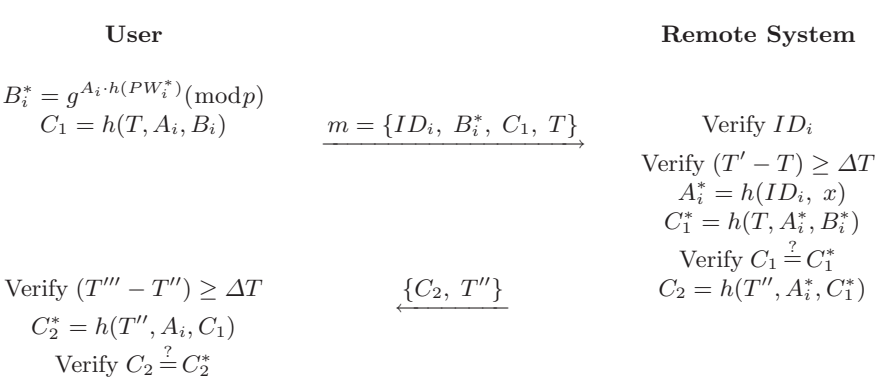


Fig. 4. Login and Verification Phases in proposed scheme

5 Security Analysis and Comparison

The following analyzes the security of the proposed scheme:

1. Due to the fact that a one-way hash function is computationally difficult to invert, it is extremely hard for any attacker to derive the system secret key x from $A_i = h(ID_i,\ x)$. Even if the smart card of the user U_i is picked up by an attacker, it is still difficult for the attacker to derive x.
2. If an attacker tries to forge a valid parameter C_1, they must have the system secret information x, because C_1 must be derived from PW_i and A_i. However, this is infeasible, as A_i has to be obtained from the system secret information x.
3. For replay attacks, neither the replay of an old login message$\{ID_i, B_i^*, C_1, T\}$ in the login phase nor the replay of the remote system's response message in Step 4 of the verification phase will work, as it will fail in Steps 2 and 5 of the verification phase due to the time interval $(T' - T) \geq \Delta T$ and $(T''' - T'') \geq \Delta T$, respectively.
4. Given a valid request message $m = \{ID_i,\ B_i^*,\ C_1,\ T\}$, it is infeasible that an attacker can compute PW_i using equation $B_i = g^{A_i \cdot h(PW_i)} (\mathrm{mod} p)$, because it is a one-way property of a secure one-way hash function and a discrete logarithm problem.
5. The proposed scheme can resist an impersonation attack. An attacker can attempt to modify a message $\{ID_i,\ B_i^*,\ C_1,\ T\}$ into $\{ID_i,\ B_A^*,\ C_A,\ T_A\}$, where T_A is the attacker's current date and time, so as to succeed in Step 2 of the verification phase. However, such a modification will fail in Step 3 of the verification phase, because an attacker has no way of obtaining the value $A_i (= h(ID_i,\ x))$ to compute the valid parameter C_1.
6. If a masqueraded server tries to cheat the requesting user U_i, it has to prepare a valid message $\{C_2,\ T''\}$. However, this is infeasible, as there is no way to derive the value $h(ID_i,\ x)$ to compute the value C_2, due to the one-way

property of the secure one-way hash function. Plus, a replay message can be exposed because of the time stamp.

7. Because of smart card verify $V = g^{A_i}$ using stored A_i in Step 2 of the password change, when the smart card was stolen, unauthorized users cannot change new password of the card.
8. Several server spoofing attacks have been recently proposed [6]. The attacker can manipulate the sensitive data of legitimate users via setting up fake servers. Therefore, a secure password-based authentication scheme with smart card must have the ability to work against such attacks. Proposed scheme provide mutual authentication to withstand the server spoofing attack.

The security properties of Wu-Chieu's scheme and the proposed scheme are summarized in Table 1.

Table 1. Comparison of Wu-Chieu's scheme with proposed scheme

	Wu-Chieu	Proposed
Impersonation attack	Yes	No
Verification table	No	No
Change password	Yes (complex)	Yes (simple)
Mutual authentication	No	Yes

6 Conclusion

The current paper demonstrated that an attacker can easily impersonate other legal users to access the resources at a remote system in Wu and Chieu's scheme. Thus, an enhancement to Wu-Chieu's scheme was proposed that can withstand an impersonation attack and allow for users to easily update their passwords without the help of a remote system. The proposed also provides mutual authentication between the user and a remote system and achieves the same advantages as Wu and Chieu's scheme.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This work was supported by the Brain Korea 21 Project in 2004.

References

[1] L. Lamport: Password authentication with insecure communication, Communications of the ACM, 24 (11) (1981) 770-772. 935
[2] C. C. Chang, and T. C. Wu: Remote password authentication with smart cards, IEE Proceedings-E, 138(3) (1991) 165-168.

[3] T. C. Wu, and H. S. Sung: Authentication passwords over an insecure channel, Computers & Security, Vol. 15 No. 5 (1996) 431-439.
[4] K. Tan, and H. Zhu: Remote password authentication scheme based on cross-product, Computer Communications, 18 (1999) 390-393.
[5] W. H. Yang, and S. P. Shieh: Password authentication schemes with smart card, Computers & Security, Vol. 18 No. 8 (1999) 727-733.
[6] N. Aoskan, H. Debar, M. Steiner, and M. Waidner: Authentication public terminals, Computers Networks, Vol. 31 (1999) 861-970. 941
[7] M. S. Hwang, and L. H. Li: A new remote user authentication scheme using smart cards, IEEE Trans. On Consumer Electronics, Vol. 46 No. 1 February (2000). 935
[8] H. M. Sun: An efficient remote user authentication scheme using smart cards, IEEE Trans. on Consumer Electronics, Vol. 46 No. 4 November (2000). 935
[9] H. Y. Chien, J. K. Jan, and Y.-M. Tseng: An efficient and practical solution to remote authentication: smart card, Computers & Security, Vol. 21 No. 4 (2002) 372-375.
[10] S. T. Wu, and B. C. Chieu: A user friendly remote authentication scheme with smart cards, Computers & Security, Vol. 22 No. 6 (2003), 547-550. 935

A Combined Data Mining Approach for DDoS Attack Detection

Mihui Kim[1], Hyunjung Na[1], Kijoon Chae[1], Hyochan Bang[2], and Jungchan Na[2]

[1] Dept. of Computer Science and Engineering, Ewha Womans University, Korea
{mihui,hjna,kjchae}@ewha.ac.kr
[2] Electronics and Telecommunications Research Institute, Korea
{bangs,njc}@etri.re.kr

Abstract. Recently, as the serious damage caused by DDoS attacks increases, the rapid detection and the proper response mechanisms are urgent. However, existing security mechanisms do not provide effective defense against these attacks, or the defense capability of some mechanisms is only limited to specific DDoS attacks. It is necessary to analyze the fundamental features of DDoS attacks because these attacks can easily vary the used port/protocol, or operation method. In this paper, we propose a combined data mining approach for modeling the traffic pattern of normal and diverse attacks. This approach uses the automatic feature selection mechanism for selecting the important attributes. And the classifier is built with the theoretically selected attribute through the neural network. And then, our experimental results show that our approach can provide the best performance on the real network, in comparison with that by heuristic feature selection and any other single data mining approaches.

1 Introduction

Distributed Denial of Service (DDoS), is a relatively simple, yet very powerful technique to attack Internet resources as well as system resources. Distributed multiple agents consume some critical resources at the target within the short time and deny the service to legitimate clients. As a side effect, they frequently create network congestion on the way from source to target, thus disrupting normal Internet operation and making the connections of many users be lost. Recently, the side effect seriously threatens our real networks together with worm viruses. As we consider the serious damage caused by DDoS attacks, rapid detection and proper response are urgent.

As the damage by DDoS attack increase, many research for detection mechanism have performed, but the existing security mechanisms do not provide effective defense against these attacks or the defense capability is only limited to specific DDoS attacks as we explained comparatively them [1]. The large number of attacking machines and the use of source IP address spoofing make the traceback impossible. Although the router performs the ingress filtering, a lot of spoofing packets can pass it because some DDoS tools provide the several

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 943–950, 2004.

spoofing levels in order to pass the ingress filtering router. The use of legitimate packets for the attack and the variation of packet fields disable characterization and filtering of the attack streams. The distributed nature of the attacks calls for a distributed response, but cooperation between administrative domains is hard to achieve, and security and authentication of participants incur high cost. There are the automated DDoS tools of various types, so beginner can easily operate them also.

In order to detect these attacks, we can monitor some features of each server or router, merge those results, and output the synthetic judgment. So many existing proposes used the network monitoring data such as tcpdump and SNMP MIB [2]. Tcpdump needs to go through multiple iterations of data pre-processing to extract meaningful features and measures, since tcpdump is not intended specifically for security purposes. Also this process requires basically a lot of domain knowledge, and may not be easily automated. SNMP MIB is meaningless if few systems adopt the SNMP, and it supplies the features for system itself such as in/out packet count, octets, error packet count, and so on. Then it is necessary to integrate the MIP entries of each system and analyzing the combined outputs. And we can use the RMON MIB that provides the features of network traffic generated in a segment. It is more useful for detecting these attacks, but the operation of RMON considerably decreases the system performance, so many operators usually turn off the RMON feature.

So we used the NetFlow that is developed by Cisco systems. It is originally developed as the network accounting technology, and it answers questions regarding IP traffic like who, what, where, when, and how. This framework defines a flow with seven unique keys that are source IP address, destination IP address, source port, destination port, layer 3 protocol type, TOS byte (DSCP) and input logical interface (ifIndex). Although the NetFlow is only provided at the cisco systems, you can use the sFlow instead of Netflow. The sFlow(RFC 3176) was standardized at the IETF and provide the similar features with NetFlow. The flow-based traffic analysis is valuable for detecting the DDoS attacks, because most of DDoS attacks suddenly increase the number of flow, exhaust the maximum flows on the ingress switch, and make the Internet connections of the network close up.

And we used the data mining techniques for modeling the traffic pattern based on the NetFlow data. This model can be used for both the misuse detection and the anomaly detection according to modeling data and method. At first, we manually selected the meaningful features(attributes) for DDoS attack, and only we used the data mining technique creating the classifier for attack detection[7]. However, because the deciding upon the right set of features is difficult and time consuming, an automated tool is necessary. For example, many trials were attempted before we came up with the current set of features, and we couldn't corroborate that our manually selected set is the really right set. So we also used another data mining technique, decision tree algorithm, for selecting the important features for each DDoS attack. This technique eliminated the helpless

Table 1. Features of DDoS Tools

	Trinoo	Synk4	TFN2k	Stacheldraht
Attack Type	UDP flood	SYN flood	UDP/SYN/ICMP flood, Smurf	UDP/SYN/ICMP flood, Smurf
Source IP	Not Spoofing	Spoofing	Capable of control the spoofing level	Automated spoofing
Source Port	Not allow to specify	Automatic selection	Automatic selection (at random or sequentially)	Automatic selection (at random or sequentially)
Target Port	Not allow to specify	Specify the range	Allow to specify	Specify the range
Etc.			-Uni-directional control -Encrypted communication	- Automated agent update - Encrypted communication

features, and informed the priority of the features. Therefore we could build the efficient model for detecting the various DDoS attacks.

This paper is divided into five sections. In Section 2, we explain briefly DDoS attacks and attack tools, and then we introduce the proposed combined data mining approach. Next we explain our experimental environment and results. Finally, a brief conclusion and future work are presented.

2 DDoS Attack & Attack Tool

An intrusion can be defined as "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource"[2]. Among these compromises, the DoS/DDoS attacks has compromised the availability of the network resource as well as the system resource itself, and the damage by the DDoS attacks is increasing as the time goes on. And because the DDoS attacks usually use the normal protocol packet like legitimate users, these attacks aren't enough with the intrusion prevention techniques only, such as user authentication and encryption. Therefore detection mechanisms are indispensable, as a first line of DDoS attack defense.

The one of reasons why the DDoS attacks are very threatening is the automated tool. Because of using the automated attack process, if once the attacker finds the systems with weak security, it dose not take above 5 seconds to install the tool and attack the victim. And it takes thousands of hosts only one minute to be invaded. As the representative DDoS attack tools, there are Trinoo, Synk4, TFN, TFN2k, and Stacheldraht, and we usually used TFN2k and Stacheldraht, the most powerful tools, for the experiments. We comparatively sum up the features of these tools at table 1. These tools use the specific port number and protocol, but it can be easily changed. So, it is difficult to detect proactively the DDoS attack by means of monitoring the control command between the attacker and master, and between master and agent.

3 Proposed Combined Data Mining Approach

As the first experimental step, we proposed the classifier built by the neural network technology for DDoS attack detection [3]. At that experiment, we also gathered the NetFlow data not only in the normal case, but also in the attack case. However, because the NetFlow on the access router only turned on the gathering feature of input traffic, that are on their way from Internet to our network, the NetFlow mainly gathered the TCP SYN flood attack traffic although we mounted several attacks. And, we heuristically selected the input attributes of the neural network, depicting the comparative graphs for the normal and attack case. The selected attributes were the number of flow, the number of octets per flow, and the number of packets per flow, because most of DDoS attack tools generate many flows using few octets and few packets. And the built classifier was designed to output the normal or the abnormal. Conclusively, we could get the 90.9% detection rate at that experiment.

As the enhanced approach, we propose the combined data mining approach. It uses the automatic feature selection mechanism and builds the classifier by the neural network technology with the automatic selected attributes. For the selection of the important attributes, heuristic method can't prove that the choice is the best, and the many trials and the many processing time are required [2]. So, we propose the decision tree algorithm, one of the data mining technologies, as the automatic feature selection mechanism. It can output the best attributes set for the candidate attributes and their priority, using the entropy or the chi-square theory. This algorithm theoretically provides insight into the patterns that may be exhibited in the data. And, the output of this decision tree is used as the input in order to build the neural network classifier like figure 1. Such mapping approach between decision tree and neural network was proposed for the goal to accurately specify the number of units, layers, connection and initial setting of parameters of neural network [4]. This combined approach can be more overhead, but classifier generation can be performed as off-line process and generated classifier can use for real-time detection as other data mining approaches.

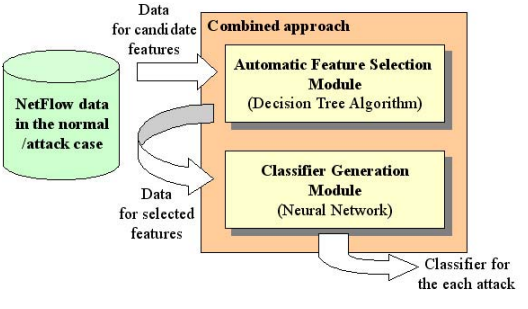


Fig. 1. Proposed approach structure

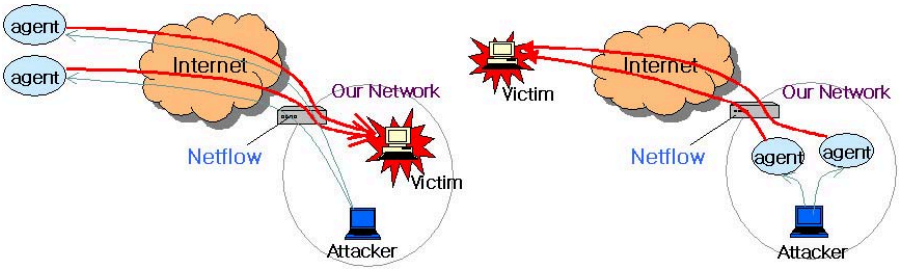


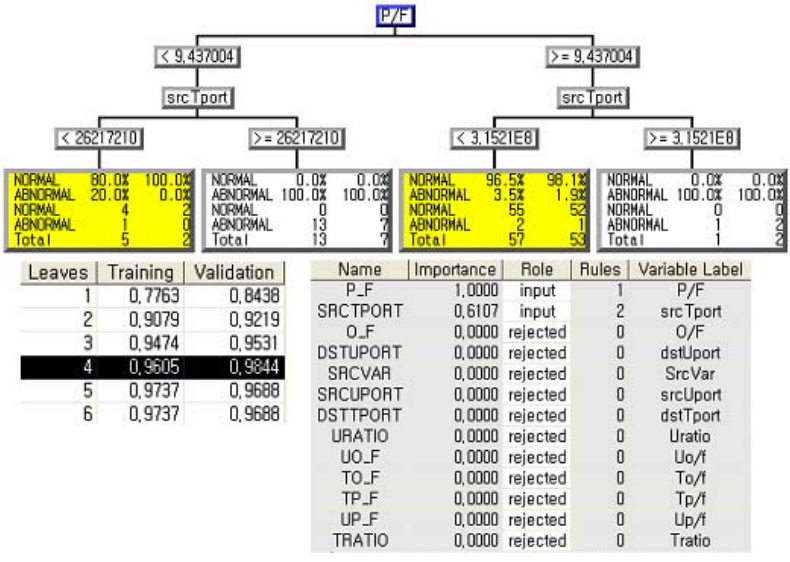
Fig. 2. Attack Scenario

4 Experimental Results

In order to especially prove the performance for the DDoS attack detection, the experiment on the real network is important and essential. The pattern of normal traffic may affect the performance of DDoS attack detection, because most of DDoS attacks use the general protocol packet, such as TCP, UDP and ICMP. So we gathered the real network traffic using the NetFlow, that was composed of the normal data and the attack data. The Router performing the NetFlow is the access router that connects the Internet and our network like figure 2.

For the various experiments, we mounted the DDoS attack in the two cases. First, the agents were located at the external network and the victim is located at our internal network, and second is vice versa. We chiefly mounted the Stacheldraht and TFN2k, because these are the strongest DDoS attack tool and they provide the various DDoS attack types, like the TCP flood, UDP flood, ICMP flood, smurf attack, and mix attack. We performed all of the attack types, but we could get only TCP flood attack traffic, UDP flood attack traffic and mix traffic of TCP/UDP, because the firewall on our network filtered ICMP traffic. So we could get the 176 normal runs, 15 TCP attack runs, 15 UDP attack runs, and 4 mix attack runs. Each run is 5-minute statistical data. And we used the 40% of each run for training, the 30% of each run for model validation, and the 30% of each run for test. We turned on the gathering feature for the egress traffic as well as the ingress traffic, different from the previous experiment [3].

At fist, we made the simple decision tree with the candidate attributes that we considered as the important input for the classifier. The candidate attributes are octet count per flow(O/F), packet count per flow(P/F), TCP octet count per flow(TO/F), TCP packet count per flow(TP/F), UDP octet count per flow(UO/F), UDP packet count per flow(UP/F), source port variance for TCP traffic(srcTport), source port variance for UDP traffic(srcUport), destination port variance for TCP traffic(dstTport), destination port variance for UDP traffic(dstUport), source IP address variance(srcVar), TCP traffic ratio(Tratio), and UDP traffic ratio(Uratio). Fist we designed decision tree with the simple output that is normal or abnormal in order to compare the previous experiment [3]. The result of decision tree by the chi-square is like figure 3.



Leaves	Training	Validation
1	0,7763	0,8438
2	0,9079	0,9219
3	0,9474	0,9531
4	0,9605	0,9844
5	0,9737	0,9688
6	0,9737	0,9688

Name	Importance	Role	Rules	Variable Label
P_F	1,0000	input	1	P/F
SRCTPORT	0,6107	input	2	srcTport
O_F	0,0000	rejected	0	O/F
DSTUPORT	0,0000	rejected	0	dstUport
SRCVAR	0,0000	rejected	0	SrcVar
SRCUPORT	0,0000	rejected	0	srcUport
DSTTPORT	0,0000	rejected	0	dstTport
URATIO	0,0000	rejected	0	Uratio
UO_F	0,0000	rejected	0	Uo/f
TO_F	0,0000	rejected	0	To/f
TP_F	0,0000	rejected	0	Tp/f
UP_F	0,0000	rejected	0	Up/f
TRATIO	0,0000	rejected	0	Tratio

Fig. 3. Decision Tree by chi-square

This case selected P/F and srcTport attribute through minimizing the number of leaves and maximizing the model validation rate. Also, the result of decision tree by the entropy selected same attributes although the threshold of each rule and the number of rules are somewhat different from each other. The leaf nodes present the each class: white is abnormal, and the other is normal. Second column of each leaf node is the classification result of training data, and third column is the classification result of validation data. In the first leaf node case of figure 3, it is normal class, and 4 normal and 1 abnormal training data, and 2 normal validation data are classified in this class.

To compare the model by theoretic selection with the model by heuristic selection, we built two classifiers by neural network with the selected attributes that were P/F and srcTport, and with all candidate attributes that were same with the input of decision tree. The misclassification rate was outputted as the test result like table 2. We could increase the detection performance in the case of neural network model by the theoretic selection, as compared with the case of neural network model by the heuristic selection and with the case of decision tree model.

To provide more information by the output of classifier, we designed the output of decision tree became normal case and each attack type such as the TCP attack, UDP attack or MIX attack. This added output information can easily add the filter rule on the firewall or IDS. The result of decision tree by entropy is like figure 1. At this case, decision tree outputted the P/F, Tratio and srcTport as the important attributes. Also the decision tree by chi-square outputted the same attributes although the number of rules and the threshold

Table 2. Misclassification Rate by heristic and theoretic selection(target status : normal/abnormal)

Used Technologies	Misclassification Rate	Comments
Heuristic Selection + Neural Network	0.0428571429	Heuristically selected attributes : Flow, O/F, P/F
Decision Tree	0.0428571429	All candidate attribute as input
Theoretic Selection (Decision Tree)+ Neural Network	0.0285714286	Theoretically selected attributes : P/F, srcTport

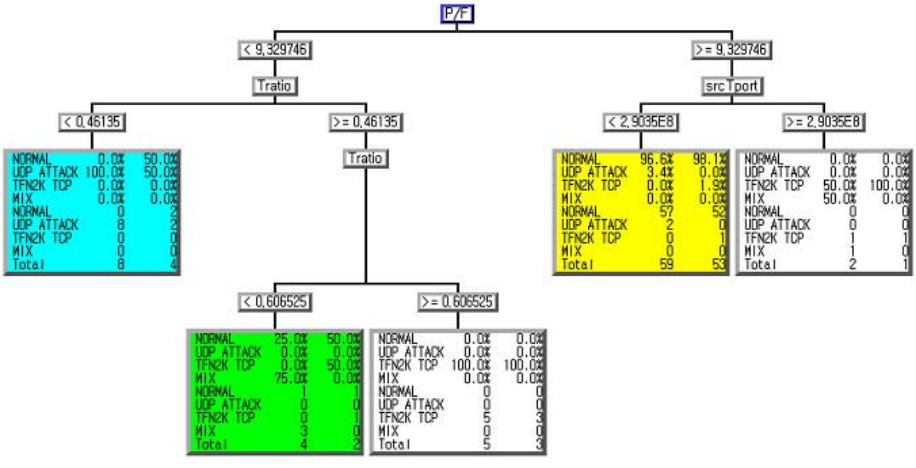


Fig. 4. Decision Tree by entropy

Table 3. Misclassification Rate by single data mining and combined data mining approach (target status : normal/each attack type)

Used Technologies	Misclassification Rate	Comments
Neural Network	0.0434782609	All candidate attribute as input
Decision Tree (entropy)	0.0724637681	All candidate attribute as input
Decision Tree (chi-square)	0.0869565217	All candidate attribute as input
Theoretic Selection (Decision Tree) + Neural Network	0.0289855072	Theoretically selected attributes : P/F, srcTport, Tratio

of each rule were different. We compared the misclassification rate of one data mining technology like decision tree or neural network with that of proposed combined data mining approach at table 3. The combined approach provided

the best performance for the DDoS attack detection, and the neural network is the next.

5 Conclusions

In this paper, we have proposed a combined data mining approach for the DDoS attack detection of the various types, that is composed of the automatic feature selection module by decision tree algorithm and the classifier generation module by neural network. For proving the practical detection performance of our approach, we gathered the real network traffic in the normal case and the attack case. We mounted the most powerful DDoS attack changing attack types, so we could get the attack traffic of various types. And we used the NetFlow data as the gathering data, because the analysis per flow is useful in the DDoS attack detection. Because the NetFlow provides the abstract information per flow, we don't need the extensive pre-processing, different with the tcpdump.

At first, we designed the target status became normal or abnormal, in order to compare new approach with our previous approach [3]. As the result of experiment, we compared the misclassification rate by the automatic selection with that by heuristic selection, and our approach resulted in the twice performance, in comparison with that by heuristic selection. Next, we devised the target status became normal or each attack type, in order to provide the extra information for attack type. Also, we compared the misclassification rate of classifier by single data mining approach and by our combined approach, and our approach provided the best performance.

The future works include the comparative experiments using various data mining technologies, and comparative experiments between the data mining approach and the pure statistic approach. And we couldn't gather the many attack runs because the DDoS attack could severely affect our network, we have a plan to gather sufficient attack runs and normal runs with enough time.

References

[1] Mihui Kim, et al.: A Combined Data Mining Approach for DDoS Attack Detection. Proc. of ICOIN (2004) 1365-1374 943

[2] Wenke Lee, Salvatore J. Stolfo: Data Mining Approaches for Intrusion Detection. Proc. of the 7th USENIX Security Symposium (1998) 79-94 944, 945, 946

[3] Hyunjung Na, et al.: Distributed Denial of Service Attack Detection using Netflow Traffic. Proc. of the Korea Information Processing Society (2003) 946, 947, 950

[4] LI Aijun, LIU Yunhui and LUO Siwei: Mapping a Decision Tree for Classification into a Neural Network. Proc. of the 6th International Conference on Computational Intelligence & Natural Computing (2003) 946

Detecting Traffic Anomalies Using Discrete Wavelet Transform*

Seong Soo Kim[1], A. L. Narasimha Reddy[1], and Marina Vannucci[2]

[1] Department of Electrical Engineering, Texas A&M University
College Station, TX 77843-3128, USA
{skim,reddy}@ee.tamu.edu,
[2] Department of Statistics, Texas A&M University
College Station, TX 77843-3143, USA
mvannucci@stat.tamu.edu

Abstract. We propose a traffic anomaly detector operated in postmortem and real-time by passively monitoring packet headers of traffic. We analyze the correlation of destination IP addresses of outgoing traffic at an egress router. Based on statistical bounds on normal traffic patterns of the correlation signal of destination addresses, sudden changes can be used to detect anomalies in traffic behavior. For more computational efficiency, we suggest a correlation calculation using a simple data structure. These correlation data are processed through coefficient-selective discrete wavelet transform for effective and high-confidence detection. We present two kinds of mechanisms for postmortem and real-time detection modes. We evaluate the effectiveness of those two mechanisms by employing network traffic traces.

1 Introduction

The frequent attacks on network infrastructure, using various forms of bandwidth attacks, have led to an increased interest for developing techniques for analyzing and monitoring network traffic. If efficient analysis tools were available, it could become possible to detect the attacks, anomalies and to appropriately take action to suppress the attacks before they have had much time to propagate across the network. In this paper, we study the possibilities of traffic-analysis based mechanisms for attack and anomaly detection.

Our approach monitors a packet header of network traffic at regular intervals and analyzes it to find if any abnormalities are observed in the traffic. By observing the traffic and correlating it to previous states of traffic, it may be possible to see whether the current traffic is behaving in an ordinary manner. In the case of bandwidth anomalies such as flash crowds and denial of service (DoS) attacks, the usage of network may be increased and abnormalities may show up

* This work is supported by an NSF grant ANI-0087372, Texas Higher Education Board, Texas Information Technology and Telecommunications Taskforce and Intel Corp.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 951–961, 2004.

in traffic pattern. Abrupt increase or decrease of traffic access pattern could signify the onset of an anomaly such as worm propagation. Our methodology relies on analyzing packet header data in order to provide indications of possible abnormalities in the traffic. Our approach to detecting anomalies envisions two kinds of detection mechanisms: off-line and on-line modes.

2 Related Work

In terms of bandwidth consumption, long term high-rate flow can be identified using partial state information [8]. Using a sample and hold approach, once a suspicious entry is detected; every subsequent packet belonging to the flow is kept monitoring and will be updated [11].

Many rule-based approaches, such as intrusion detection systems, try to match the established rules to the potential DoS attack from external incoming traffic near the victims. Moreover, the pushback is a cooperative defending mechanism through which the information exchanges amongst core routers [9, 10]. In contrast, some approaches proactively seek a method that suppresses the overflowing of traffic in the source [5]. It usually uses a rate-limited control for reducing the monopolistic consumption of available bandwidth to diminish the effect of attack [5, 7, 10].

Traditionally, various forms of signature have been utilized for representing the whole contents or certain identities. By expanding utilization of signature, network securities use the signature-based algorithm prevalently. The disproportion of bidirectional flows can be used as the signature of anomalistic traffic [4]. The changing ratios (i.e., the rate of decrease) between the flow numbers of neighboring specific bit-prefix aggregate flows can be calculated and used for peculiarities [6]. Besides, there are some distinguished transform approaches to emphasize the anomaly of the traffic [1, 3, 12]. Using traffic volume such as byte counts as signal, a wavelet system shows performance to expose the variance of the anomalies.

3 Our Approach

3.1 Traffic Analysis at the Source

The (spoofed) source address of outbound traffic from an AD (administrative domain) could be filtered because router can know internal address range unlike destination addresses. On the other hand, destination address is likely to have a strong correlation with itself over time since the individual accesses have strong correlation over time. Recent studies have shown that the traffic can have strong patterns of behavior over several timescales [3]. It is possible to infer that some correlation exists on their weekly or daily consumption patterns. We hypothesize that the destination addresses will have a high degree of correlation over longer timescales. If this is the case, sudden changes in correlation of outgoing addresses can be used to detect anomalies in traffic behavior.

3.2 General Mechanism of the Detector

Our detection mechanisms can be organized in three steps. The first step is traffic parser, in which the correlation coefficient signal is generated from packer header traces or NetFlow records as input, in section 4. The second step processes wavelet transforms to study the address correlation over several timescales. We selectively reconstruct decomposed signal across specific timescales based on the nature of attacks and network administrator's focus, in section 5. The final stage is detection, in which attacks and anomalies are detected using historical thresholds of traffic to see whether the traffic's characteristics are out of regular norms. Sudden changes in the analyzed signal will lead to some indication that could be used to alert the network administrator of the potential anomalies in the network traffic. In this paper, we consider some statistical summary measures as explained in section 6.

3.3 Traces

To verify the validity of our approach, we run our algorithm on two kinds of traffic traces. First, we examine the detector on traces from the University of Southern California which contains real worm attacks. Additionally to inspect the sensitivity of our detector over attack of various configurations, we employ simulated virtual attacks on the University of Auckland traces of addresses collected over a campus access link. These traces range in length from 3 days to several weeks.

3.4 Attacks

We consider nine kinds of virtual attacks as shown in Table 1. These simulated attacks cover many kinds of behaviors and allow us to deterministically test diverse attacks.

- **Persistency.** The first 3 attacks send malicious packets in on-off type. More sophisticated attackers attempt to conceal their intentions through repeating attack and pause periods. So, it is intended to model intelligent and crafty attackers that attempt to dilute their trails. The other remnant attacks continue to assault throughout the attack.
- **IP address.** The 1^{st} attack among every 3 attacks targets for a (semi) single destination IP address. This target may be really one host in case of 32-bit

Table 1. The Nine kinds of simulated Attacks

	1 (2,I,SD)	2 (2,I,SR)	3 (2,I,R)	4 (2,P,SD)	5 (2,P,SR)	6 (2,P,R)	7 (1,P,SD)	8 (1,P,SR)	9 (1,P,R)
Duration	2 hours	2 h.	2 h.	2 h.	2 h.	2 h.	1 hour	1 h.	1 h.
Persistent	intermittent	int.	int.	persistent	per.	per.	per.	per.	per.
IP	single destination	semi-random	random	single dest.	semi-rand.	rand.	single dest.	semi-rand.	rand.

prefix, occasionally aggregated neighboring hosts in case of x-bit prefix. The 2^{nd} attack style composes the IP address in which specific portion of the address structure preserves the identical value and the rest of the address is generated randomly for the infiltration efficiency. The 3^{rd} type is randomly generated.

Our attacks can be described by a 3-tuple (duration, persistency and IP address). We superimpose these attacks on ambient traces, which are the University of Auckland traces [2]. The mixture ratios of normal traffic and attack traffic range 2:1 to 10:1 in packet count. Replacement of normal traffic with attack traffic is easier to detect and hence not considered here.

4 Signal Generation

4.1 The Correlation Coefficient

Our approach collects packet header data at an AD's edge over a sampling period. Individual fields in the packet header are then analyzed to observe anomalies in the traffic. To study the correlation embedded in the IP address, we use its correlation coefficient which is a normalized measure of the linear relationship in random variable.

For each address, a_m, in the traffic, we count the number of packets, p_{mn}, sent in the sampling instant, s_n. In order to compute address correlation coefficient signal, we consider two adjacent sampling instants *n-1* and n . We can define IP address correlation coefficient signal in sampling point n as

$$\rho(n) = \frac{\sum_m (p_{mn-1} - \overline{p_{n-1}}) * (p_{mn} - \overline{p_n})}{\sqrt{\sum_m (p_{mn-1} - \overline{p_{n-1}})^2} \sqrt{\sum_m (p_{mn} - \overline{p_n})^2}} \quad (1)$$

When correlation coefficient signal varies between -1 and 1, if an address a_m spans the two sampling points, we will obtain a positive contribution to $\rho(n)$. A popular destination address a_m contributes more to $\rho(n)$ than an infrequently accessed destination.

4.2 Data Structure for Computing Correlation Coefficient

In order to minimize storage and compute efficiently, we employ a simple but powerful data structure. This data structure consists of 4 arrays *"p[4]"* . Each array expresses one of the 4 bytes in an IP address. Within each array, we have byte-sized 256 locations, for a total of 4*256 bytes = 1024 bytes. A location *p[i][j]* is used to record the packet count for the address j in i^{th} field of the IP address through scaling. This provides a concise description of the address instead of unique 2^{32} locations. We generate this approximate signal by computing a correlation coefficient over the address in two success samples, i.e., by computing

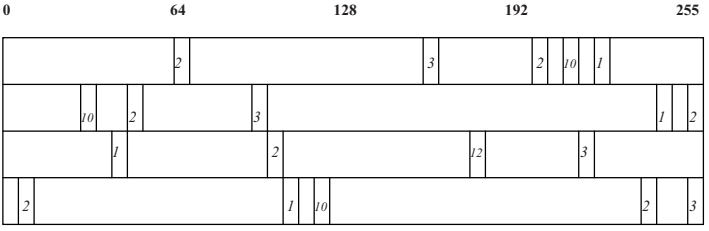


Fig. 1. Data structure for computing correlation coefficient. Suppose that only five flows exist, their destination IP addresses and packet counts are as follows. IP of F1 = 165. 91.212.255, P1 = 3; IP of F2 = 64. 58.179.230, P2 = 2; IP of F3 = 216.239. 51.100, P3 = 1; IP of F4 = 211. 40.179.102, P4 = 10; IP of F5 = 203.255. 98. 2, P5 = 9

$$\rho_{in} = \frac{\sum_{j=0}^{255}(p[i][j][n-1] - \overline{p[i][n-1]}) * (p[i][j][n] - \overline{p[i][n]})}{\sqrt{\sum_{j=0}^{255}(p[i][j][n-1] - \overline{p[i][n-1]})^2}\sqrt{\sum_{j=0}^{255}(p[i][j][n] - \overline{p[i][n]})^2}},$$

where i=1,2,3,4

We present simple example to illustrate the information can be stored this data structure as shown in Fig. 1. The packet counts of each flow are recorded to the corresponding position of each IP address segment.

Our approach could introduce errors when the addresses segments match even if addresses themselves don't match. From comparison between correlation coefficient signal of the full-32 bit address and our data structure, we see that the difference are negligible i.e., our approach does not add significant noise. Moreover, we examine the similarity of above signals with cross-correlation coefficient, which has actually $\rho_{XY} \approx 0.74$. We think that these signals have a close positive correlation interchangeably.

5 Discrete Wavelet Transform

5.1 DWT (Discrete Wavelet Transform)

Wavelet technique is one of the most up-to-date modeling tools to exploit both non-stationary and long-range dependence. In real situations, we encounter signals which are characterized by abrupt changes and it becomes essential to relate to the occurrence of an event in time. Wavelet analysis can reveal scaling properties of the temporal and frequency dynamics simultaneously unlike Fourier Transform used in [12]. Through signal can be detected in certain timescales and in certain position of the timescales, we can induce the frequency and temporal components respectively. We compute a wavelet transform of this correlation signal over a given time. A multilevel one-dimensional DWT consists of decomposition (or analysis) and reconstruction (or synthesis) [13].

For decomposition, starting from a signal s , the first step of the transform decomposes s into two sets of coefficients, namely approximation coefficients cA_1, and detail coefficients cD_1. The input s is convolved with the low-pass filter Lo_D to yield the approximation coefficients. The detail coefficients are obtained by convolving s with the high-pass filter Hi_D. This procedure is followed by down sampling by 2. The second step decomposes the approximation coefficient cA_1 into two sets of coefficients using the same method, substituting s by cA_1, and producing cA_2 and cD_2, and so forth. At level j , the wavelet analysis of the signal s has the following coefficients, [cA_j, cD_j, cD_{j-1}, cD_{j-2}, ... , cD_2, cD_1].

For reconstruction, starting from two sets of coefficients at level j , that is cA_j and cD_j, the inverse DWT synthesizes cA_{j-1}, up-samples by inserting zeros and convolves the up-sampled result with the reconstruction filters Lo_R and Hi_R. For a discrete signal of length n , DWT can consist of $\log_2 n$ levels at most.

5.2 Our Specification: Coefficient-Selective Reconstruction

Our specification is daubechies-6 two-band filter. The filtered signal is down-sampled by 2 at each level of the analysis procedure; the signal of each level has an effect that sampling period extends 2 times. Consequently it means that the wavelet transform can identify the changes in the signal over several timescales. When we use t seconds as sampling period, the time range at level j extend $t * 2^j$ seconds. And a 1-minute sampling interval and 30-second sampling duration are used to reduce the amount of data and computational complexity.

The network operators can select reconstructed levels that they wish to be captured. We assume that the network administrators are interested in detecting shorter anomalies of sufficient intensity and anomalies of more than 30-minute duration. In order to detect these attacks, we extract only the 1^{st}, 5^{th}, 6^{th} and 7^{th} levels in decomposition and reconstruct the signal based only on coefficients at these levels.

6 Detection

6.1 Thresholds Setting through Statistical Analysis

We develop a theoretical basis for deriving thresholds for anomaly detection. We first investigate the distribution of the wavelet reconstructed signal in the University of Auckland free of attacks. To examine random variable density, we select only some levels of the DWT decomposition of the ambient trace without attacks and reconstruct the signal based on those levels. We then look at some statistical properties. The Fig. 2(b) and 2(e) show the distribution and histogram of the reconstructed signal of the ambient traces in postmortem mode. We verify normality through the Lilliefors test for goodness of fit to normal distribution with unspecified mean and variance. The postmortem transformed data have a normal distribution at 5% significance level, namely $X \tilde{} N(0, 0.026^2)$. By selecting some of the levels through selective reconstruction, we have removed

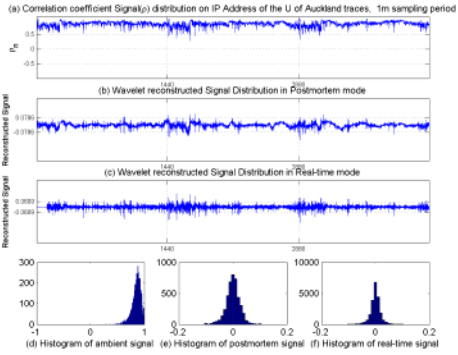


Fig. 2. Various distribution of the ambient traces in the Univ. of Auckland

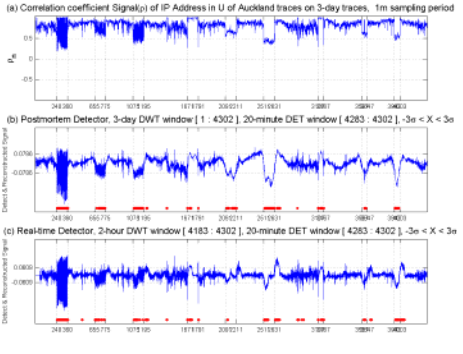


Fig. 3. The Univ. of Auckland detection results in postmortem and real-time mode

some of the features from the signal that were responsible for the non-normality in the original signal.

We gather the 4-week traces and analyze their statistical summary measures. The statistical parameters of network traffic, such as mean, variance and autocorrelation function, are stationary distributed given different days. From the viewpoint of communications, the ambient trace could be considered as wide-sense stationary Gaussian white noise, on the other hands, the attack and anomaly could be considered as random signal.

When we set the thresholds to -0.078 and 0.078 respectively, these figures are equivalent to $\pm 3.0\sigma$ confidence interval for random variable X.

$$P(\mu - 3.0\sigma < X \leq \mu + 3.0\sigma) \approx 99.7\% \tag{2}$$

This interval corresponds to 99.7% confidence level by (3). With such thresholds, we can detect attacks with error rate of 0.3%.

We also analyze the reconstructed signal at our selected levels of ambient trace in real-time mode, which shows approximately normal distribution. The Fig. 2(c) and 2(f) show the distribution and histogram of the reconstructed signal of the ambient traces in real-time mode.

6.2 Detection Anomalies Using the Real Attack Trace

Our postmortem and real-time approaches are applied to the USC traces which contain real network attacks. Detection results are shown in Fig. 4. The Fig. 4(a) illustrates a correlation coefficient signal of IP addresses that is used for wavelet transform. The Fig. 4(b) is the wavelet-transformed and reconstructed signal in postmortem and its detection results. The detection signal is shown with dots at the bottom of the each sub-picture. The Fig. 4(c) shows the wavelet-transformed and reconstructed signal in real-time and its accumulated results. Through traffic engineering, we can identify the attack in which a specific internal compromised machine continued to attack a few external destinations.

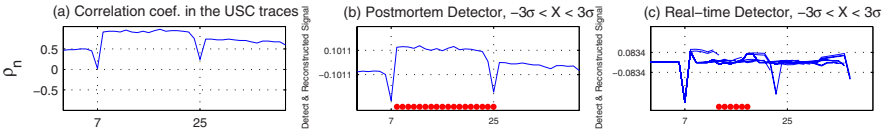


Fig. 4. The USC trace detection results in postmortem and real-time mode

6.3 Detection Anomalies Using the Simulated Attack Trace

Detection results on the University of Auckland traces in 3 days are shown in Fig. 3. The Fig. 3(a) illustrates a correlation coefficient signal of IP addresses and the Fig. 3(b) is the wavelet-transformed and reconstructed signal in postmortem and its detection results and the Fig. 3(c) shows the accumulated results in real-time.

The simulated nine attacks are staged between the vertical lines, shown in the figure. Overall, our results show that our approach may provide a good detector of attacks in both modes. Moreover, the detections in the early points of every day, namely sampling points are near at the 1450 and 2900, are turned out the regular flash crowds included in the original traces such as file backup.

Discussion of Postmortem Analysis. The postmortem analysis and detectors can rely on datasets over long periods of time and we use whole 3-day correlation data all at once. The reconstructed signals of first 3 attacks (*,I,*) show an oscillatory fashion because of their intermittent attack patterns, while the remaining six attacks, namely (*,P,*), give a shape of hill and dale at attack times due to persistence.

The attacks on a single machine, the 1st attack among every 3 attacks described in (*,*,SD), reveal the high valued correlation which means the current traffic is concentrated on (aggregated) a single destination. Detection signals in the form of dots show that these typed attacks can be detected effectively. On the other hand, the semi-random typed attacks, that is (*,*,SR), and random styled attacks, namely (*,*,R), illustrate low correlations which means traffic is behaving in irregular pattern. Consecutive detection signals indicate the length of attacks and also imply the strength of anomalies.

In order to evaluate the effectiveness of employing DWT, we compare the detection results of our scheme employing DWT with a scheme that directly employs statistical analysis of the IP address weighted correlation signal. When confidence levels of most interest (90% ~ 99.7%) are considered, DWT provides significantly better detection results than the simpler statistical analysis without applying of DWT. This shows that DWT offers significant improvement in the detection of anomalies.

Discussion of Real-Time Analysis. The real-time detection requires that the analysis and the detection mechanism rely on small datasets in order to keep such

Table 2. The Relation between Latencies and Confidence levels in Nine attacks in Real-time mode

	c	1	2	3	4	5	6	7	8	9	f	f
	l^a	(2,I,SD)	(2,I,SR)	(2,I,R)	(2,P,SD)	(2,P,SR)	(2,P,R)	(1,P,SD)	(1,P,SR)	(1,P,R)	p^b	n^c
1.0σ	68	0^d	1	2	0	0	0	0	0	0	11	0
1.5σ	86	0	1	2	0	0	0	0	2	0	7	0
2.0σ	95.5	1	2	5	0	0	0	0	5	2	5	0
2.5σ	98.5	1	2	5	0	3	0	0	7	2	3	0
3.0σ	99.7	1	3	5	0	10	2	0	8	6	2	0
3.5σ	99.95	1	5	10	0	36	2	0	11	6	2	0
4.0σ	99.99	2	15	13	0	X^e	6	0	X	8	1	2

[a] confidence level in percentage
[b] false positive is counted a series of relevant signal as 1
[c] false negative
[d] latency is measured by minute unit
[e] X means non-detection

on-line analysis feasible. If we want to investigate a specific level j , it requires 2^j samples for reconstruction at least. So, the most recent 2-hour correlation data make use of detecting an attack of less than 2-hour duration. We take the moving window and majority to accommodate faster detection while reducing the false alarms. As the Fig. 3(c) shows, our detector achieves acceptable attack detection performance in on-line analysis as well as in off-line analysis.

Table 2 shows the overall timing relationship between detection latency and the setting of the confidence level of our simulated attacks in real-time mode. As we expect, the higher the confidence level, the higher the detection latency. The detection against the (*,*,SD) typed attacks, (aggregated) single destination, can achieve a prompt response. And the (*,*,R) typed attacks, randomly generated destination, can generally be detected more quickly than the semi-random type attacks described in (*,*,SR) because of the resulting lower correlation with random attacks.

6.4 Adaptive Filtering in Real-Time Analysis

In order to detect anomalies of different unknown durations, we considered adaptive filtering of the traffic signal. Adaptive filtering continues to search for the proper levels of timescales suitable to the nature of the attack. At normal times, the detector monitors only the reconstructed signal based on lower level coefficients (say, aggregated 1, 2 and 3 levels), which can identify the most detailed and instantaneous change in the traffic signal. Once a possible detection of anomaly is identified at these levels, the detector expands its investigative scope into higher levels gradually, for example at levels 2, 3 and 4, for improving the identification accuracy or robustness. It may help to reveal the substance of attack and to diminish the false alarm under unknown conditions. False alarms can be reduced by not declaring the detection of an anomaly until consecutive alarms are raised at multiple levels. The traffic signal at higher levels is considered as the identification progresses. On the other hand, when any anomaly is not detected, the reconstructed signals return to lower levels gradually. The Table 3 shows the results of such an approach. It may induce more false positives but achieve faster detection campared to non adaptive method. The results indicate

Table 3. The Latencies in Nine kinds of attacks in Adaptive Filtering

	c	1	2	3	4	5	6	7	8	9	f_p	f_n
	l	(2,I,SD)	(2,I,SR)	(2,I,R)	(2,P,SD)	(2,P,SR)	(2,P,R)	(1,P,SD)	(1,P,SR)	(1,P,R)		
3.0σ	99.7%	1	2	2	0	1	1	4	1	1	5	0

that it may be feasible to detect traffic anomalies with low latency even when we consider attacks of unknown length.

7 Future Work and Conclusion

The duration of the samples and the number of samples have a strong impact on the accuracy of the results and the latency for detecting an attack. The samples with a smaller sampling period cause to more false positives but lead to faster identification of the attack. It is a pivotal factor for the real-time approach we discussed. Thus, as a further research, the relation between sampling rate and latency should be investigated from statistical point of view.

We plan to study containment approaches along the multiple dimensions of addresses, port numbers, protocols and other such header data based on a detection tool. We also plan to study the effectiveness of the analysis of traffic at various points in the network, at the destination network and within the network core.

We studied the feasibility of analyzing packet header data through wavelet analysis for detecting traffic anomalies. Specifically, we proposed the use of correlation coefficient of destination IP addresses in the outgoing traffic at an egress router. Our results show that statistical analysis of aggregate traffic header data may provide an effective mechanism for the detection of anomalies within a campus or edge network. We studied the effectiveness of our approach in postmortem and real-time analysis of network traffic. The results of our analysis are encouraging and point to a number of interesting directions for future research.

Acknowledgement

We are very grateful to Deukwoo Kwon for his comments and reviews on statistical analysis, to Alefiya Hussain at USC for her help in accessing traces.

References

[1] Ramanathan, A.: "WADeS: A Tool for Distributed Denial of Service Attack Detection", *TAMU-ECE-2002-02, Master of Science Thesis*, August 2002

[2] National Laboratory for Applied Network Research (NLANR), measurement and operations analysis team: "NLANR network traffic packet header traces", accessed in August 2002

[3] Barford, P., Kline, J., Plonka, D., Ron A.: "A Signal Analysis of Network Traffic Anomalies", in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002
[4] Gil, T., Poletto, M.: "MULTOPS: A Data-Structure for Bandwidth Attack Detection", in *Proceedings of the 10th USENIX Security Symposium*, Washington, D. C., USA, August 2001
[5] Mirkovic, J., Prier, G., Reiher P.: "Attacking DDoS at the Source", in *10th IEEE International Conference on Network Protocols*, Paris, France, November 2002
[6] Kohler, E., Li, J., Paxson, V., Shenker, S.: "Observed Structure of Addresses in IP Traffic", in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002
[7] Garg, A., Reddy, A.: "Mitigation of DoS attacks through QoS regulation", in *Proc. of IWQOS workshop*, May 2002
[8] Smitha, Kim, I., Reddy, A.: "Identifying long term high rate flows at a router", in *Proc. of High Performance Computing*, December 2001
[9] Mahajan, R., Bellovin, S., Floyd, S., Ioannidis, J., Paxson, V., Shenker, S.: "Controlling High Bandwidth Aggregates in the Network (Extended Version)", in *ACM SIGCOMM Computer Communication Revies, Volume 32, Issue 3*, July 2002
[10] Ioannidis, J., Bellovin, S.: "Implementing Pushback: Router-Based Defense Against DDoS Attacks", in *Proceedings of Network and Distributed System Security Symposium*, San Diego, California, February 2002
[11] Estan, C., Varghese, G.: "New Directions in Traffic Measurement and Accounting", in *ACM SIGCOMM 2002*, Pittsburgh, PA, USA, August 2002
[12] Cheng, C., Kung, H., Tan, K.: "Use of spectral analysis in defense against DoS attacks", in *Proc. of IEEE Globecom*, 2002
[13] The MathWorks. Inc.: MatLab software, ver 6.1.0.450 Release 12.1, May 2001

The Causality Analysis of Protocol Measures for Detection of Attacks Based on Network*

Il-Ahn Cheong[1], Yong-Min Kim[2], Min-Soo Kim[2], and Bong-Nam Noh[3]

[1] Interdisciplinary Program of Information Security
Chonnam National University, 500-757, Gwangju, Korea
mir@lsrc.jnu.ac.kr
[2] Linux Security Research Center, Chonnam Nat'l Univ.
ymkim@chonnam.ac.kr, phoenix@athena.jnu.ac.kr
[3] Div. of Electr-Comput. & Inform-Engine., Chonnam Nat'l Univ.
bbong@chonnam.ac.kr

Abstract. New intrusions have being tried continuously due to vulnerability of TCP/IP on the computer networks. Many studies have been progressed about the method that is based on the signature and anomaly behavior in order to detect the attacks using vulnerability of networks. However the detection of intrusion from an enormous network data is very difficult and required much load of work. In this paper, for the effective detection, we studied the combination of network measures from the data packets which is generated by various DoS attacks using the vulnerability of TCP/IP. As the result, we were able to find the causality of network measures for the DoS attacks based on networks and detect similar attacks as well as existing attacks using it. Furthermore, the detection by possible combination of selected measures has a high accurate rate, and also the causality of network measures can be used to generate real-time detection patterns.

1 Introduction

TCP/IP is designed without consideration of vulnerability for many of threats. However, it was found defects on design including recently various attack behaviors, and by exploiting the vulnerabilities, the network environments were damaged by many attacks such as Smurf, ICMP disconnection, IP spoofing, Denial of Service, UDP/SYN flooding, etc.

In general, intrusion detection techniques can be classified into two categories according to detection method: signature-based and anomaly-based. The signature-based method makes a pattern by analyzing known attacks and detects the attack matched with the pattern. However, this method can not detect various or unknown attacks. The anomaly-based method is introduced to overcome this weakness. This method detects the intrusion through modeling network data

* This research was supported by University IT Research Center Project.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 962–972, 2004.

using various methods such as statistics, data mining and other artificial intelligence algorithms [1, 2, 3, 4, 5]. Nevertheless in this case it has some problems such as the detection rate is low or the false rate is high in according to method of measure selection. Consequently, we need an adequate method which detects various or unknown attacks efficiently.

In order to extract their own properties of network attacks, it needs to find out the causality between attacks and fields in network packet. Analyzing the causality enables not only to explain the attacks, but also to select the measures for effective detection from attacks. Selected measures and theirs combinations can be used to detect variant or new attacks and to generate detection patterns for these kinds of intrusions. Furthermore, a load of detection works could be decreased by monitoring only related measures. In this paper, we analyzed the properties of DoS attacks based on networks and selected the combined measures to detect effectively by approving the causality of network measures by using the Causality Analysis System(CAS).

2 Related Works

The research to detect anomaly behavior using packet analysis based on network protocol is being recently taken at Ohio [6], Florida University [7], U.C. Davis University [8] and Boeing Company [9]. Ohio University evaluated meaning of doubtful packets and effects of packets by analyzing IP and TCP packets. Mainly the error or occupied packet rate about fields such as Length of packet, Checksum, IP address and Flag was analyzed statistically. However, in the research, analysis about packets occurred by actual attacks was not carried out and analyzed properties for anomalous packets in its network.

Florida University analyzed anomalous distribution of packet headers such as Ethernet, IP, TCP, UDP and ICMP. This research used clustering to get together only each range of packet fields to profile packet header and method to calculate anomaly score, as generating probability of fields in packets is smaller, as anomaly score is calculated higher. The result represents such a low detection rate about total attacks and better detection rate only about Probes and DoS attacks. On the other hand, selected attack feature such as TTL field has nothing to do with attack types and it is not sufficient to explain attacks as detected result. Furthermore, it had not disadvantage to detect attacks in application layer included in payload.

U.C. Davis has studied host and routing-based method to detect spoofed packets. These methods have been help to decide whether received packets have spoofed source address or not. This study analyzed attack types of spoofed packets such as SYN-flood, Smurf, TCP hijacking connection, bounce scan and Zombie control. These kinds of attacks were detected by routing and non-routing method. This study analyzed mostly attack types of spoofed packet and evaluated mostly TTL value based on IP source address about each protocol's TTL, value of IP ID, OS fingerprinting, ACK packet, etc. Boeing company has researched into identifying DDoS(Distributed DoS) as calculating entropy and

Table 1. Causality for DoS attack(e.g., in case of *neptune* attack in Table 2)

Attack Name	Cause	Effect
SYN Flood (Neptune)	• every TCP/IP implementation is vulnerable	Protocol=6(TCP)
	• IP-spoofed packets requesting new connections faster than the victim system can expire the pending connections • the neptune program compiled from C code allow the user to use int the invented packets	srcIP==fabricated IP
	• send twenty SYN packets to all port(1~1024) on a Solaris 2.6 system	dstPort=1~1024
	• look for a number of simultaneous SYN packets destined for a particular machine that are coming from an unreachable host	Flag=2(SYN), 20(ACK&RST)
	• only detected attack source with ICMP_ECHO message(*potential attack possibility*)	ICMP_ECHO

frequency-sorted distribution of properties for selected packets. Furthermore, they have studied detection-response prototype to effectively cope with DDoS attacks. Entropy calculation, one of detection algorithms used in this study, measured entropy amounts of change based on a fixed size of windows for continuous packets and Pierson's chi-square distribution for dispersed properties such as SYN flag of TCP. These studies have limits to detect or analyze using by data of its network in case [6, 8], and [9]. Also it was not sufficient to explain attacks as detected result which used by each single measure in case all.

3 The Analysis of Causality

3.1 The CAS for Measures of Network Protocols

The Causality Analysis System(CAS) [10][1] represents the structure of system that analyzing causality among measures of network. The CAS consists of three parts such as part of the DARPA IDS Evaluation Datasets in MIT Lincoln Lab., part of causality analysis between network protocols and attacks, part of detection, verification and saving in database of generated combination patterns with combination of analyzed measures. Attack types of DoS including the 1999 DARPA IDS Evaluation Datasets (DARPA99) analyzed based on classification of Kendall and Das [11, 12, 13], as a result we analyze causality and mainly common features by using the CAS. Table 1 represents a part of analyzed contents and Table 2 represents causality between DoS attack types and measures of network protocols. It is proved that a common field about entire attacks is Protocol field of IP. Because the most of attacks target in each network protocol, it is classified by Protocol field of IP again. Two signs to mark items of fields about each attack represent causality degree of attacks. If causality is high, then marked '◯', if it is low, then marked '△', and other cases are not marked. For example, in the case of *neptune* attack(bold-box in Table 2) that is one of attacks against TCP, the targeting protocol is TCP so that it is marked '◯' in *PT(Protocol)* filed. It is possible for source address to be fabricated and this importance is low so that it was marked '△'. Because the destination *Port* targets ports of specific

[1] For more details, we present in *http://lsrc.jnu.ac.kr/~mir/icoin2004/causality.html*

Table 2. The causality analysis between DoS attacks and network protocols

Type of Attacks \ Protocol Measures		IP								TCP												UDP			ICMP			Data
		VE	HL	TO	TL	ID	FR	OF	TT	PT	HC	SI	DI	SP	DP	SQ	AK	HD	RV	UG	AC	PS	RS	SY	FN	WS	CK	UP	US	UD	UL	UC	TY	CO	IC	IN	SN	SZ
DoS (tcp)	apache2									○					○						○			○														
	arppoison																																					
	back						○		△	○				△	○					△				△		△		△										○
	crashiis									○					○																							○
	dosnuke								△	○					○																							
	land						△	△		○		○	○	○	○									○														△
	Mailbomb									○					○																							△
	neptune									○		△		△	○						○		○	○										△	△			△
	Proc. Tbl						△	△		○					○																							
	sshproc.tb						△	△		○					○																							
	Tcpreset									○													○															
DoS (udp)	Syslogd									○																			○									△
	Teardrop						○	○		○																			○	○	○							
	Udpstorm									○																			○									
DoS (icmp)	POD						○	○	△	○																							○	○	○			△
	selfping									○					○																			○					
	smurf						○	○		○																							○	○	△	△	△	△

* IP: VE(VErsion), HL(Header Length), TO(TOS), TL(Total Length), ID, FR(FRagmentation), OF(OFfset), TT(TTL), PT(ProTocol), HC(Header Checksum), SI(Source IP address), DI(Destination IP address)
* TCP: SP(Src Port), DP(Dst Port), SQ(SeQuence number), AK(AKnowledgement number), HD(HeaDer length), RV(ReserVed), UR(URG), AC(ACK), PS(PSH), RS(RST), SY(SYN), FN(FIN), WS(Window Size), CK(CheckSum), UP(Urgent Point), FL(UR, AC, PS, RS, SY, FN)
* UDP: US(UDP Src port), UD(UDP Dst port), UL(UDP Length), UC(UDP Checksum)
* ICMP: TY(TYpe), CO(COde), IC(ICMP Checksum), IN(IdentificatioN), SN(Sequence Number). * Data: SZ(SiZe of Data)
* Degrees of Association: ○(high), △(low), Blank(none)

range from 1 to 1024, it is marked '◯' to represent high causality. The case of *Flag(UG, AC, PS, RS, SY, FN)* fields of TCP marked '◯' to the main item occurred in these attacks. Even though it was not represented in the Evaluation Datasets, as the result of analysis for attacks it has potentiality to attack protocol as a target so that it was marked '△' in item *TY(Type)* and *CO(Code)* field of ICMP protocol to have high causality. Finally, in this paper only protocol header is analyzed to reduce load of work and effectively detect attacks in an enormous data of network, so that *Data* field didn't verify the contents and it is represented whether it has causality among attacks to target only size of data. It is analyzed and summarized about other attacks in the same method above.

3.2 The Analysis of Measures and Method for Detection of Attacks

In this paper, we experiment it using *tcpdump* log of network data that records are from simulating the 1999 DARPA IDS Evaluation Datasets by each week. We use a sampling of DoS attack from week 2 of training data with attack behavior and use the data of week 1 and 3 to analyze optimal measure that could identify from abnormal behavior. We use the data of week 2 to verify combination pattern of sampling measure through above procedure and experiment on detection efficiency as apply verified patterns to the data of week 4.

If we choose all possible combination of measures on network protocol, the number of cases would be numerous. Also if we select wrong combination, we couldn't identify normal behavior. Therefore in this experiment, we choose the combination of protocol measures which showed high causality in Table 2 and calculate the *FPP*(Field Per Packets) of interesting target.

$$FPP = \frac{the\ amount\ of\ interested\ field\ value}{the\ total\ amount\ of\ interested\ packets\ in\ datasets} \times 100 \qquad (1)$$

This represents how to calculate DR, FR and DS to compare detection efficiency from each combination of measures.

$$Detection\ Rate(DR) = \frac{DA+NA}{TA}\ where, \quad (2)$$
$$DA: the\ number\ of\ detected\ attacks$$
$$NA: the\ number\ of\ new\ detected\ attacks$$
$$TA: the\ number\ of\ attacks\ in\ test\ datasets$$

$$False\ Rate(FR) = \frac{the\ number\ of\ false\ alarm(FA)}{the\ number\ of\ alarm\ in\ detection(FS)} \quad (3)$$

$$Detection\ Score(DS) = DR + \left(\frac{NA}{NA+DA} \cdot \frac{DR}{DR+FR}\right) - \left(\frac{1}{FS} \cdot \frac{FR}{DR+FR}\right) \quad (4)$$

In equation (2), DR is derived from the number of detection from attacks classified training data in test data and other number of detection. In equation (3), FR indicates the rate of false alarms toward all alarms. And in equation (4), DS is total evaluation of detection efficiency that calculated by add weight of detection rate to DR and subtract weight of fault rate from DR. To select the most optimal measure combination, we should find out DS from training data and from test data. DS_{train} shows the detection efficiency of previous attacks and the reduction of false rate. Therefore, the most optimal measures should indicate high DS_{train} and DS_{test}. Method of detection decided by whether is the attack or not as finding out the distribution of these measures which selected by rule of pattern on the unit of fixed time unit ("per second" in this paper). That is, the attack point of time is solved by calculating the occupying ratio of these measures by means of above method.

4 The Experiments

In this chapter, we find most optimal combination to increase the detection rates through the possible combination of measures. Also, through these experiments we verify the propriety of the combinations and evaluate the efficiency of detection.

4.1 Experimental Data

DoS attack types could be classified by each interesting protocol(TCP, UDP, ICMP), as well as by each service(based on target port) in Table 2. We analyzed the distribution of each service port at first, and find out optimal combination of measures to detect attacks by analyzing flag field and the distribution of data size.

Table 3 shows an arrangement for the distribution of service port in each week and DoS attacks against target port. We experiment with the attacks that TCP attack against SMTP service(port 25), UDP attack against HTTP service(port 80) and ICMP attack not against specific port. Table 4 shows an arrangement for the number of DoS attacks targeting TCP, UDP and ICMP protocol in each evaluation datasets, and it used by an evaluation of efficiency in experiment. Also, we add type of attacks such as Probes, R2L and Data to find out whether new attacks except DoS attacks can be detected or not.

Table 3. DoS attacks according to type of service

Type of Protocols / Type of Attacks	TCP	UDP	ICMP
DoS Attacks *port(services)*	20(ftp-data), 21(ftp): warez master/client 22(ssh): processtable, 23(telnet): land 25(smtp) : land, mailbomb, processtable 80(http): back, crashiis, apache2 139(netbios-ssn): dosnuke 1~1024: neptune	80(http): teardrop 514(syslog): syslogd	pod smurf

Table 4. Construction of attacks for evaluation datasets

Attacks / Evaluation datasets	Attack Type	TCP			UDP		ICMP	
		Attack Name(counts)	Attack Counts	Total Attacks (TA)	Attack Name (counts)	Total Attacks (TA)	Attack Name (counts)	Total Attacks (TA)
Week 2	DoS	land(1), neptune(2), maibomb(2)	5	10	-	0	pod (2)	2
	Probes	portsweep(3),satan(2)	5					
Week 4	DoS	crashiis(1),mailbomb(4), processtable(1)	6	20	teardrop (1)	1	smurf (3)	3
	Probes	portsweep(5), satan(1), sshtrojaninstall(1)	7					
	R2L	sendmail(1),netcat(1), netbus(2), ppmacro(2)	6					
	Data	secret(1)	1					

4.2 The Experiments for Detection of DoS Attacks

4.2.1 Selection of Measures

In case of TCP protocol in Table 2, as mostly DoS attacks interrupt normal services according to create many SYN requests for connection, RST of disconnecting and RST-ACK of disconnection directly, it analyzed affecting by destination *Port* and *Flag* value.

In this section, we analyzed the combination of target port by each week and flag measures to detect DoS attacks against SMTP service port*(Port 25)* based on TCP. For example, in this paper we show FPP for destination *Port 25* as Table 5. *Flag* has 1(FIN), 4(RST), 18(SYN-ACK) and 20(RST-ACK), the *Flag* contains more these attacks in week 2 than week 1 and 3. Comparing the results about sample dataset of DoS attacks(dos_tcp) and distribution of each week, the distribution of DoS attacks packet shows high values at which *Flag* is 2(SYN), 4(RST), 16(ACK), 17(FIN-ACK), 18(SYN-ACK), 20(RST-ACK), 24(PSH-ACK). Here if *Flag* is 2, 16, 17 and 24, the priority for selection of measure should be lowered because distribution by each week is similar. Also in case of *Flag* has 1, the priority for measure of selection should be lowered because not including DoS attack packets. Therefore, the *Flag* 4, 18 and 20 values having priority selected by excluded those values: the results that selecting values having high rate of DoS attacks in distribution of each week. Therefore, we experiment with selected values that 2(SYN) value occurred connection established and (4, 18, 20) values in previous analyzed results.

Table 5. FPP for combination of destination Port and Flag in each week

dstPort.Flag	dos_tcp	Week 1	Week 2	Week 3	Selection
25.01	0.000	0.000	0.000	0.000	N
25.02	3.174	0.330	0.443	0.336	Y
25.04	0.034	0.000	0.000	0.000	Y
25.16	9.336	1.308	1.599	1.184	N
25.17	3.108	0.329	0.432	0.332	N
25.18	0.003	0.000	0.000	0.000	Y
25.20	0.003	0.000	0.000	0.000	Y
25.24	15.547	2.626	3.383	2.651	N

4.2.2 The Experiment for Causality Analysis among Measures

Table 6 shows the experimental results whether it is detected by combination of *Flag* values analyzed prior to SMTP service port of TCP, UDP and ICMP protocols. In this experiment we use Week 2 dataset for verifying dataset, and Week 4 dataset for normal behavior dataset(test dataset). As you see Table 3, total number of DoS attacks is 10(TA=10) in verification dataset(Week 2) and 20(TA=20) in test dataset(Week 4). Especially in case of *land* attack, packet is forced as source address is equal to destination address, even though this kind of attack doesn't show the distribution property of the packet, we include this attack to find out whether is detected or not.

In case *Flag* has 2 at the experimental results from verification dataset in Table 6, as the result of experiment for DoS attack in Table 3, it could not detect entirely existing attacks(SA=10), but detect *satan*(Probes) which one of new attacks is detected in two points(NA=2).

We also calculate the rest combination using above mentioned method, also do for test dataset. In Table 6, values in parenthesis below DR, FR section means the total number of detection and warning message when attacks detected and values at "AT(Attack)" section means "(# of detection for new attacks / # of warning messages when detects)". Among detected attacks, the characterized '*' attack means that new DoS attacks are detected not in sample dataset. As the result of experiment to detect DoS attacks against TCP protocol only with *Flag*, in the case of 2 and 18 DoS attack was not detected in verification dataset (Week 2) and in the case of 4 and 20 it was also not detected in test dataset exactly. And totally alarm messages took place very much, in the case of 2 it shows property to detect new attack much was showed in test dataset. In the case of 18 specially, detection rate was low in verification data and also exact detection not achieved in test data.

As the result of experiment with combination of destination *Port* and *Flag*, totally it was showed that detection and accuracy was increased a little bit. However both of low detection and low accuracy were still resulted in at the combination with 18. As the experimented result with combination of three *Flag* values, the combination of 4 and 20 is superior to other results. Specially, detection rate is increase and accurate detection was carried out in verification and test datasets, and *processtable* attack of new DoS was also detected. Then in

Table 6. The result of detection for TCP protocol

Subject Datasets / Combination		Verification Datasets(Week 2)					Anomaly Behavior Datasets(Week 4)				
		Detected Attacks	DR	FR	AT	DS_{train}	Detected Attacks	DR	FR	AT	DS_{test}
TCP (Flag)	2	satan(probes)	0.20 (2)	0.00 (2)	2/0	1.20	mailbomb(dos) processtable(dos)* portsweep(probes) sshtrojaninstall(r2l) secret(r2l) ppmarcro(r2l)	0.35 (7)	0.96 (272)	6/1	0.58
	4	neptune(dos)	0.20 (2)	0.00 (7)	0/2	0.20	netbus(r2l)	0.00 (0)	1.00 (27)	0/0	0.00
	18	-	0.00 (0)	1.00 (106)	0/0	0.00	portsweep(probes) secret(data)	0.10 (2)	0.98 (267)	2/0	1.02
	20	neptune(dos) portsweep(probes) satan(probes)	0.30 (3)	0.00 (186)	2/1	0.97	satan(probes)	0.05 (1)	0.98 (66)	1/0	0.08
TCP (dstPort, Flags)	25,2	neptune(dos) portsweep(probes)	0.20 (2)	0.67 (9)	1/1	0.23	mailbomb(dos) ppmarcro(r2l)	0.15 (3)	0.75 (16)	1/2	0.15
	25,4	neptune(dos)	0.30 (3)	0.25 (4)	2/1	0.55	mailbomb(dos) portsweep(probes)	0.20 (4)	0.86 (29)	3/1	0.31
	25,18	neptune(dos)	0.10 (1)	0.00 (1)	0/1	0.10	-	0.00 (0)	0.00 (0)	0/0	0.00
	25,20	neptune(dos)	0.10 (1)	0.00 (1)	0/1	0.10	-	0.00 (0)	0.00 (2)	0/0	0.00
	25, 2|20	neptune(dos)	0.20 (2)	0.33 (3)	0/2	0.00	mailbomb(dos) sendmail(r2l) ppmarcro(r2l)	0.20 (4)	0.69 (16)	2/2	0.26
	25, 4|20	neptune(dos) portsweep(probes)	0.40 (4)	0.00 (4)	2/2	1.40	mailbomb(dos) processtable(dos) portsweep(probes)	0.20 (4)	0.42 (31)	3/1	0.42
	25, 2|18|20	neptune(dos)	0.20 (2)	0.33 (3)	0/2	0.20	mailbomb(dos) sendmail(r2l) ppmarcro(r2l)	0.20 (4)	0.69 (16)	2/2	0.26
	25, 4|18|20	neptune(dos) portsweep(probes)	0.40 (4)	0.00 (6)	2/2	1.40	mailbomb(dos) processtable(dos) portsweep(probes)	0.20 (4)	0.42 (31)	3/1	0.42
	25, 2|4|20	neptune(dos)	0.20 (2)	0.00 (2)	0/2	0.20	processtable(dos)	0.05 (1)	0.00 (1)	1/0	1.05

the case of 18, there is no big change in the result when this combination pattern is combined to 18. As the fact, in case that *Flag* is 18, it didn't have severe effect on different combination and the detection rate was low as observed above. Therefore it is not possible that this value becomes the exact combination. And the detection rate and accuracy were low in case of combination for destination *Port* and *Flag* 20, but each the detection rate and accuracy became high as add combination of 2 or 4 to that combination.

Finally, detection of new attack was carried out in experiment of combination with destination port and three values(2, 4, 20), but the accuracy became low a little bit for the existing attack, and totally we found that types of Probes attacks showed the similar property of DoS attacks were detected in large numbers. As mentioned above, although the *land* attack is not grasped by property of distribution, it could be detected by verifying whether source address (or port) is same as destination address (or port). As the result, it was found that optimal combination of measures is (*destination Port*, *Flags*) = (25, 4|20) to detect DoS attacks for SMTP service port. And it showed the optimal combination of measures to detect various attacks in Table 7.

Table 7. The optimal combination of measures to detect attacks against each protocol

Attack Types	Combination of Measures / Attack Names	IP	IP/Data	TCP	UDP	ICMP
		FR:OF	*FR:DZ*	*DP:FL*	*UD:FR*	*TY:CO*
DoS	neptune, mailbomb	-	-	(25,4|20)	-	-
	processtable	(2,0), (0,0)	(0,0), (2,44)	(25,4|20)	-	-
	sshprocesstable	(2,0), (0,0)	(2,0)	-	-	-
	land, smurf	(0,0)	-	-	-	-
	teardrop	(0,3), (1,0)	-	-	(80,0), (80,1)	-
	pod	-	-	-	-	(0,0)
Probes	portsweep	(2,0), (0,0)	(2,44), (2,80)	(25,4|20)	-	(8,0)
	ipsweep	(2,0)	-	-	-	-
R2L	xlock	(2,0)	-	-	-	-
	guest	(0,0)	-	-	-	-
	named	-	(2,80)	-	-	-
Data	secret	-	(0,1460)	-	-	-

Table 8. The Comparison of detected attacks and contributed measures

Detected attacks / Researcher		Florida Univ.	This Paper
DoS	crashus	TTL=253	-
	mailbomb	TTL=253	dstPort=25 & Flag=2|20 or 4|20
	processtable	IP src addr	dstPort=25 & Flag=4|20, Frag=2 & DataSize=44
Probes	portsweep	FIN-ACK, IP src/dst addr, packet size, TTL=36~52	dstPort =25 & Flag=4|20, Frag=2 & DataSize=44 or 80
	satan	packet size, TTL	Flag=20
R2L	netbus	TTL=126	Flag=4
	ppmacro	-	dstPort =25 & Flag=2|20
	secret	-	Flag=2, Frag=0 & DataSize=1460
	sendmail	Outgoing/Incoming IP src/dst addr	dstPort =25 & Flag=2|20
	sshtrojaninstall	-	Flag=2

4.3 The Comparison and Analysis

In this section, we compare with DoS attacks included in Week 4 of test datasets. Table 8 shows the comparison for detection of attack and contribution of detection with research of Florida University and result of experiment in this research. A Research of Florida University decided in many cases that detected with was irrelevant to the characteristic of attacks such as *TTL* field, also the detected results did not give us entire satisfaction of explanation for attack. In Table 8, in case of *TTL* value is 253 on DoS attack, *crashiis* and *mailbomb* is detected but it could not explain the characteristic of attack of two kinds.

Because *TTL* value is largely depended on operating system, in case of value is 253 it is caused by the phenomenon that was appeared in operating system of mainly BSD series, Solaris, HP-UX, etc. Also, in the case of *processtable* it found out that the result was detected by the source IP address, but the source IP address is possible to be fabricated by attacker, accordingly detecting by IP address did not give us enough explanation for characteristic of attack. However, observing the detected result of this paper, it was detected as measures correspond to the characteristic of attacks. That is, in case of DoS attack it was detected by combination because rates of SYN, RST, RST-ACK had the charac-

teristic that is occurring by far rate than at normal condition. This combination also detected other similar attack such as Probes or R2L(Remote to Local).

In the result of our experiment that the optimal combination of analyzed measures were defined as pattern, we found that it can be detect not only new attacks as processtable attack, but also existing attacks as *neptune* or *mailbomb* attack. Further, we showed to be detected *portsweep* and *satan* for Probes and several R2L similar DoS attack.

5 Conclusion and Future Works

Abnormal behavior method has a problem that detection rate is low or false alarm rate is high to detect modified types of existing attacks or new attacks according to the selection of measures and difficulty of real-time detection because much works are required from an enormous network data.

To solve these problems, we analyzed attack types by each network protocol and experiment with DoS attack methods which attack each service of network protocols to find out the optimal composition of measures in possible compositions of measures. As the result of experiments, not only existing attacks but similar attacks were detected and the optimized combinations of measures were found. This can be used to generate detection pattern for purpose of effectively and rapidly detecting an attack from an enormous network data.

In Future, there is a necessity for analyzing more attacks and finding out combination of patterns for effectively detecting DoS attacks of TELNET, FTP and HTTP services. Furthermore, the combination of network measure for detecting Probes and R2L attacks should be more analyzed and optimized.

References

[1] D. Anderson, T. Lunt, H. Javitz, A. Tamaru, A. Valdes, "*Detecting Unusual Program Behavior Using the Statistical Component of the Next-generation Intrusion Detection Expert System(NIDES)*," TR SRI-CSL-95-06, SRI C&S Lab., 1995. 963

[2] J. Cannady, "Artificial Neural Networks for Misuse Detection," NISSC, 1998, 443-456. 963

[3] V. Paxson, "Bro: A system for detection network intruders in real-time," Computer Networks, 31(23-24), Dec. 1999, 2435-2463. 963

[4] R. Sekar, Y. Guang, S. Verma, T. Shanbhag, "A High-Performance Network Intrusion Detection System," ACM Conference on Computer and Communications Security, 1999, 8-17. 963

[5] S. Mukkamala, A. Sung, "Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques," International Journal of Digital Evidence, Vol. 1, 2003. 963

[6] M. Bykova, S. Ostermann, "Statistical Analysis of Malformed Packets and Their Origins in the Modern Internet," 2nd IMW, 2002. 963, 964

[7] M. Mahoney, P. Chan, "*PHAD: Packet Header Anomaly Detection for Indentifying Hostile Network Traffic*," Florida Tech., TR CS-2001-4, Apr. 2001. 963

[8] S. Templeton, K. Levitt, "Detecting Spoofed Packets," Proc. of the DARPA Information Survivability Conferences and Exposition(DISCEX'03), 2003. 963, 964

[9] L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred, "Statistical Approaches to DDoS Attack Detection and Response," Proceedings of the DARPA Information Survivability Conferences and Exposition(DISCEX'03), 2003. 963, 964
[10] http://lsrc.jnu.ac.kr/~mir/icoin2004/causality.html 964
[11] K. Kendall, "*A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*," MIT Master's Thesis, Jun 1999. 964
[12] K. Das, "*Attack Development for Intrusion Detection Evaluation*," MIT Master's Thesis, Jun 2000. 964
[13] T. Ptacek, T. Newsham, "*Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*," TR, 1998. 964

Network Processor Based Network Intrusion Detection System

Hyeyoung Cho[1], Daeyoung Kim[1], Juhong Kim[1], Yoonmee Doh[1], and Jongsoo Jang[2]

[1] Information and Communications University
103-6, Munji-dong, Yuseong-gu, Daejeon, 305-714, Korea
{hycho,kimd,scarlet,ydoh}@icu.ac.kr
[2] Electronics and Telecommunications Research Institute
161, Gajung-dong, Yuseong-gu, Daejeon, 305-350, Korea
jsjang@etri.re.kr

Abstract. To achieve fast packet processing and dynamic adaptation of intrusion patterns that are continuously added, a new high performance network intrusion detection system using Intel's network processor, IXP1200, is proposed. Unlike traditional intrusion detection engines, which has been implemented by either software or hardware so far, we propose an optimized architecture and algorithms, exploiting the features of network processor. Through implementation and performance evaluation, we show the proprieties of the proposed approach.

1 Introduction

Recently with the explosive growth of Internet applications, the attacks of hackers on network are increasing rapidly and becoming more seriously. Thus information security is emerging as a critical factor in designing a network system and much attention is paid to *Network Intrusion Detection System* (NIDS), which detects hackers' attacks on network and handles them properly. But, the performance of current intrusion detection system cannot catch the increasing rate of the Internet speed because most of the NIDS's are implemented by software. To implement fast enough NIDS for high speed network backbones and access networks, it is necessary to find a new mechanism for fast intrusion detection.

An *Intrusion Detection Engine* (IDE) is a core component to detect any network intrusion. IDE's can be categorized into hardware-based IDE and software-based IDE. Software-based IDE, such as *Snort*, Hogwash, etc., is implemented by pure software on general microprocessors. Therefore, software-based IDE has an advantage of dynamic change of rule signature database, while having a poor performance. On the other hand, to reduce a software overhead, hardware-based IDE uses special hard-wired processors for high speed packet processing or new techniques utilizing specific hardware for efficient pattern matching. For example, implementing *non-deterministic finite automata* (NFA) for fast pattern matching in FPGA is introduced [1] and automatic conversion from the rule signature

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 973–982, 2004.

used in open-source NIDS into JHDL (Java based Hardware Description Language) is studied [2,3]. However, the hardware-based approaches are not flexible for dynamic updates of rule signature database. From the viewpoints of performance and flexibility, neither software-based nor hardware-based IDE fully satisfy the requirements of building real-time/high-speed NIDS.

In this paper, we propose a *Network Processor based Network Intrusion Detection System* (NP-NIDS) whose network processor is being popularly used in building a giga-/tera-bit high-speed network router or firewall. Exploiting dedicated network packet processing engines and software programmability of network processor, we could take both advantages of software-based and hardware-based IDS's. We use an open source software-based NIDS, *Snort*, as a reference model in building NP-NIDS with the support of it's rule signature database. The *Intel IXP1200* network processor is chosen to implement NP-NIDS due to its higher programmability.

The rest of the paper is organized as follows. We introduce intrusion detection system and network processor in Section 2, and describe the architecture of proposed NP-NIDS in Section 3. In Section 4, we discuss the algorithms of network processor based intrusion detection engine. The performance evaluation is given in Section 5. In Section 6, a short conclusion and future works are given.

2 Intrusion Detection System and Network Processor

An *intrusion detection system* (IDS) is a kind of security system that detects unauthorized use, misuse, and abuse of networks and computer systems, and takes appropriate actions to handle the intrusion [4]. According to the origin of data, intrusion detection systems (IDS's) can be classified into host-based IDS and network-based IDS. IDS's can also be divided into misuse detection model and anomaly detection model based on analysis method of intrusion detection. Misuse detection model detects intrusion based on signature analysis, which finds anomalous packet by searching signature database. The signature database contains previously known intrusion information related with network packets. Anomaly detection model detects intrusion by looking for activities that are different from normal behaviors, mainly using statistical analysis, data mining, and expert system. In this paper, we are targeting misuse detection model and network-based IDS.

Snort is one of the most popular software-based NIDS's and used on various hardware platforms with the moderate flexibility. The detection engine in *Snort* examines every packet comparing it with rule signatures that keep previously known intrusion patterns. When an intruding packet is detected, *Snort* takes an appropriate action following its response policy, which is described by the rule with many formats such as *syslog*, *tcpdump* format log, alarm, windows popup, and so on [5,6,7]. In our implementation, *Snort* version 2.0.0 that supports 1,267 rules is referenced.

```
alert tcp 1.1.1.1 any -> 2.2.2.2 any (flags:F; msg:"FIN Scan";)
```

The above example shows a rule for TCP packets. If a TCP packet, whose source and destination IP address is '1.1.1.1' and '2.2.2.2' with any IP port number and *F* flag bit is set, is detected, *Snort* sends alert signal that includes *"FIN Scan"* message. In this paper, NP-NIDS supports the same rule as *Snort* provides.

Intel's IXP1200 network processor consists of StrongArm core, 6 microengines, SDRAM/SRAM/PCI interfaces, and IX Bus unit [8]. It supports various features for fast packet processing, such as multi-processing, distributed data storage architecture, and hardware multi-threading. StrongArm core is a 32bit RISC microprocessor operated on 232MHz clock frequency and used for management and control. Six 32bit multi-threading RISC microengines are equipped to support up to 24 hardware threads and make zero-overhead context-switching possible. We also use IXP2400 network processor, which supports XScale technology, operates on 600MHz clock frequency, and has 8 microengines [8]. Each microengine can support 4K instructions and 8 hardware threads.

3 NP Based High Performance NIDS

In NP-NIDS, Intrusion detection that have been functioned in software on general microprocessors are performed on network processor. Network processor allows multiple packets to be processed concurrently so that it responds to any intrusion in real-time, leading to higher performance IDS.

While monitoring network packets in real-time, NIDS should detect intruding packets by comparing them with stored rule signatures. And it takes an appropriate response based on the directive action found in matched rule signature. Of the procedures, intrusion detection engine spends most of the processing time on comparing packets with rule signatures and finding anomalous packets. With 6 microengines and 4 hardware threads per each engine, the IXP1200 can handle the received packets at high speed without any help of StrongArm core. The specially designed instruction sets for network packet processing such as bit, byte, word, and long word operations enable microengine to process packets in high performance. In addition, NP-NIDS can also achieve flexibility by implementing intrusion detection algorithms on network processor. High programmability of network processors makes rule signatures easily updated along with the changes in intrusion detection policy.

As shown in Fig. 1, NP-NIDS is structured in a three-level hierarchy; host processor and StrongArm and microengines on IXP1200. Host processor manages rule signatures and provides an interface to higher-level network management systems (NMS). StrongArm has a role of arbiter between the host processor and microengines. It downloads rule signatures from the host and transforms them into proper format for detection engine algorithms. It also manages all the modules on IXP1200. A microengine carries out the functions of Ethernet MAC and device driver such as receiving and sending Ethernet frames and checking errors. The main role of microengine is to run an intrusion detection algorithm, which makes a decision whether intrusion occurs or not by searching rule signa-

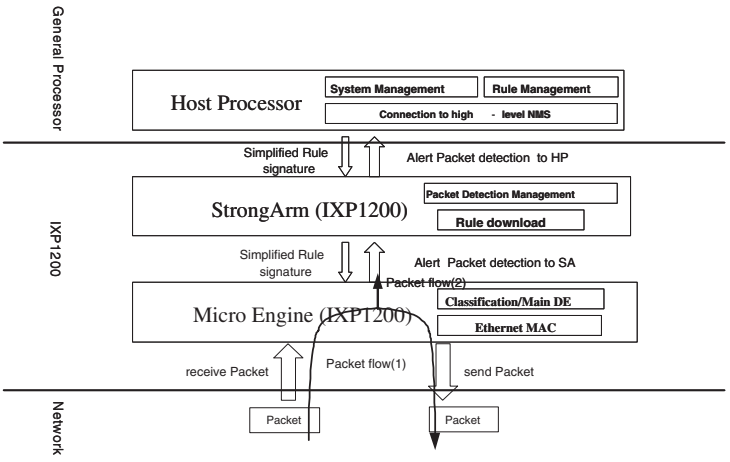


Fig. 1. The Architecture of Network Processor IXP1200 based NIDS

tures for corresponding intrusion pattern. If no matching is happened, NP-NIDS just forwards the packet into the network like packet flow (1) as shown in Fig. 1. If an intrusion is detected, NP-NIDS sends appropriate warning message to StrongArm like packet flow (2). We use Linux kernel 2.4.17 for the host and embedded Linux 2.3.99 for the StrongArm of IXP1200. To keep the compatibility with *Snort*, we use the same rule signatures as *Snort* has. Only a few functions and rules are excluded. In system initialization, the rule signatures stored in host processor are downloaded to StrongArm. The rule signatures on StrongArm are compressed and optimally restructured for detection engine algorithm, and finally stored in shared SRAM. For prototyping NP-NIDS, we use Radisys's ENP2506-P board with 256MB SDRAM and 8MB SRAM. To provide simultaneous access to both network packet and rule signatures, we store network packets and rule signatures in SDRAM and SRAM, respectively.

4 Design of High Performance IDE

An *Intrusion Detection Engine* (IDE) performs time-consuming algorithms, which compare every received packet with each rule signature, so that it is a major factor to decide the performance of an IDS. In the proposed NP-NIDS, IDE is implemented as a hardware thread in each microengine of IXP1200. An IXP1200 has 6 microengines and each microengine supports 4 hardware threads such that 24 hardware threads are available [8]. In case of IXP2400, 64 hardware threads are available because it provides 8 microengines and each microengine has 8 hardware threads [8]. To make the best of multi-processing capability for building a high performance intrusion system, it is essential to take account of optimizing the usage of available threads.

For different kinds of network protocols such as TCP, IP, UDP, and ICMP, there exist various patterns of intrusion for each of them and each intrusion pattern has a corresponding rule signature. In this paper, we propose a *Distributed Intrusion Detection Engine* (DIDE) to deal with different kinds of rule signatures effectively, exploiting multi-processing capability of IXP network processor. In DIDE approach, every IDE is assigned to a specific protocol and performs searching rule signature. Currently, we support the packet types of TCP, UDP, IP, and ICMP with four different types of detection engine algorithms like the reference model of *Snort* 2.0.0.

Since a DIDE uses the detection algorithm optimized to the assigned protocol, it is possible to get small code size, fast detection, and optimized rule signature database for a specific type. However, exclusive allocation of detection engines to specific protocols may not keep DIDE work conserving. It means that a packet should wait for the dedicated IDE of the same type to be free, even though other types of IDE are idle. We also have the other type of detection engine called *Combined Intrusion Detection Engine* (CIDE), which processes all types of packets. But due to the limit of the paper length, we skip its explanation. We choose DIDE over CIDE in this paper, because IXP1200 does not have enough instruction memory to accommodate CIDE that requires much longer program size and more complicated rule signature database. The packet processing procedure of DIDE is shown in Fig. 2 and the explanations are as follows.

1. A microengine receives a packet from MAC device.
2. The received packet is classified into one of protocol types, TCP, IP, UDP, and ICMP. If the protocol field in IP header is not any one of TCP (0x06), UDP (0x11), and ICMP (0x01), the packet is determined as the type of IP.

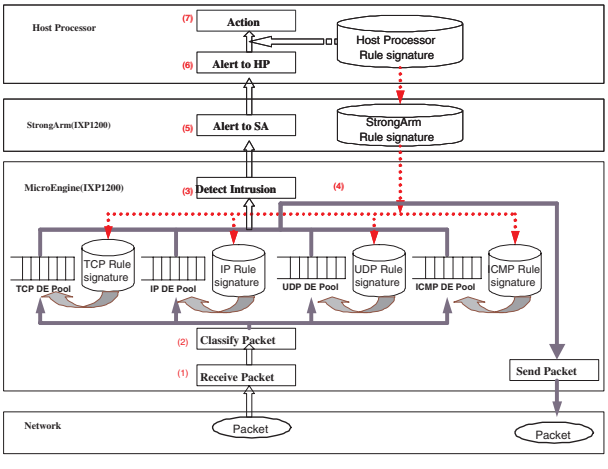


Fig. 2. Packet Processing Procedure in Distributed Intrusion Detection Engine

3. The classified packet is processed by detection engine dedicated to the protocol type. The engine searches corresponding rule signature database for any possible intrusion pattern.
4. If the packet does not have any match pattern with rule signatures, it is forwarded to network as a healthy packet.
5. Otherwise, the engine notifies the StrongArm of an intrusion with the matched rule identifier and packet content.
6. StrongArm then informs the host processor of the matched rule identifier and the packet content.
7. Host processor logs the received rule identifier and the packet content, and then takes proper actions like massage printing or alerting to the NMS.

Detection Engines are heterogeneous in DIDE because engines execute different codes for each protocol. Thus, we keep four types of pools for available detection engines. Detection Engines are also divided into *DIDE-Serial* (DIDE-S) and *DIDE-Parallel* (DIDE-P) with respect to execution style.

Fig. 3(a) shows the procedure of processing packets in *DIDE-S*. Idle DE's are kept in the corresponding pool of the same type protocol. We assume that the packets are received in order of P1 (TCP), P2 (IP), P3 (TCP), and P4 (ICMP). Since *P1* is a TCP packet, it is assigned to the engine *E1* from the TCP DE pool. The engine *E1* examines n TCP rule signatures one by one in serial. In the same way, the packet *P2* is assigned to the engine *E2* such that it examines m IP rule signatures in serial.

Fig. 3(b) shows the behavior of *DIDE-P*. As shown in the figure, more than one engines can be assigned to process a packet in *DIDE-P* method. The more engines are assigned, the less time is consumed for intrusion detection. But, a proper number of detection engines must be determined for the system-level performance. In Fig. 3(b), two TCP engines are assigned to the packet *P1*. The TCP rule signatures are evenly divided into two groups such that *E1* processes rules, R1, R3, R5, etc. and *E2* does R2, R4, R6, and so on.

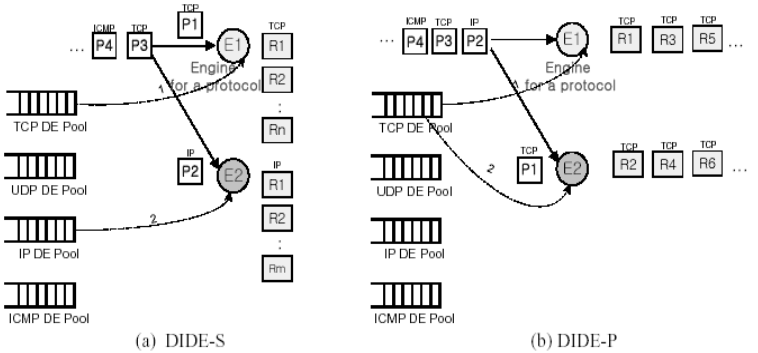


Fig. 3. Packet Processing Flow in *DIDE-S* and *DIDE-P*

5 Performance Evaluation

5.1 Time Measure for Number Field and String Pattern Matching

Rule signatures are largely divided into content and non-content rules according to whether it needs string comparison or not. Content rule is activated by including *'content'* field to search for a specific string pattern in the payload of a received packet. On the other hand, non-content rule examines only number field without performing string pattern matching. In case of IP protocol, which has TTL and Flag number fields in packet's header, non-content rule corresponds to the comparison of header field.

As a performance metric for detection engine, times taken to check number and string field are used. To measure the elapsed time of IXP1200 instruction execution, we use 64-bit Cycle Count register of FBI CSR, which is operated by the core clock frequency (232MHz, 4.3ns/cycle) [8]. The time is obtained by taking the difference between the counter values right before and after checking number or string field.

Table 1 shows an example of microcode to count clock cycles for checking a number field. Its short description of the microcode is given in the right column. In our experiments, it takes an average of 108 clock cycles to verify one number field of a rule signature. For the string field checking, the same method is used. Table 2 also shows the microcodes for string field checking, which performs basic string matching algorithm. The total length of used packet is 74 bytes including 32-byte length payload. In searching a packet for 14-byte length of pattern, an average of 5,416 clock cycles is taken.

Table 1. An Example Code to Measure the Time Taken for Number Field Checking

IXP1200 Code	Description

`.local startTime endTime runtime`
`xbuf_alloc($cycle_xfer, 2)`
`csr[read, $cycle_xfer[0], CYCLE_CNT], ctx_swap`
`alu[startTime, --, B, $cycle_xfer[0]]`
`xbuf_free[$cycle_xfer]`
	`allocate SRAM Transfer Register`
	`read from Cycle Count Registe to $cycle_xfer`
	`copy $cycle_xfer to startTime`

`.if( checkField & CHECK_ICMP_TYPE)`
`sdram[read, $$packet_data[0], data_ptr, 4, 2], sig_done`
`ctx_arb[sdram]`
`xbuf_extract(packet_byte, $$packet_data, 0, 2, 1)`
	`check checkField`
	`read packet from SDRAM`
	`extract icmpType(1byte) field from packet_data`

`sram[read, $rule_data[0], rules_addr, 4, 2], sig_done`
`ctx_arb[sram]`
`xbuf_extract(rule_byte, $rule_data, 0, 4, 1)`
	`read rule from SRAM`
	`extract icmpType field from rule_data`

`.if( packet_byte != rule_byte)`
`br[get_next_rule#]`
`.endif`
	`compare icmpType field on packet with`
	`icmpType value on rules`

`xbuf_alloc($cycle_xfer1, 2)`
`csr[read, $cycle_xfer1[0], CYCLE_CNT], ctx_swap`
`alu[endTime, --, B, $cycle_xfer1[0]]`
`alu[runTime, endTime, -, startTime]`
`xbuf_free[$cycle_xfer]`
`.endlocal`
	`read from CSR to $cycle_xfer1`
	`copy $cycle_xfer1 to endTime`
	`subtract startTime from endTime`
	`(runTime=endTime-startTime)`

Table 2. An Example Code to Measure the Time Taken for String Field Checking

briefly described IXP1200 Microcode (1)

```
/*Start Time Read*/
allocate SRAM Transfer Registerread
from CSR(Cycle Count Register) to $cycle_xfer
copy $cycle_xfer to startTime

/*String Check Code */
.if( checkField & CHECK_ICMP_TYPE)

get_one_byte#:
  extract each 1byte(rule_byte,packet_byte) from
  packet_data and rule_data
  .if(rule_byte == packet_byte)
  br[byte_match#]
  .else
  br[byte_not_match#]
  .endif
byte_match#:
  .if(curr_rule_len>= max_rule_len)
  set match_result to 1
  br[cmp_end#]
  .endif
  .if(packet_len >= max_packet_len)
  set match_result to 0
  br[cmp_end#]
  .endif
  set rule_read_addr and packet_read_addr
  br[get_one_byte#]
```

briefly described IXP1200 Microcode (2)

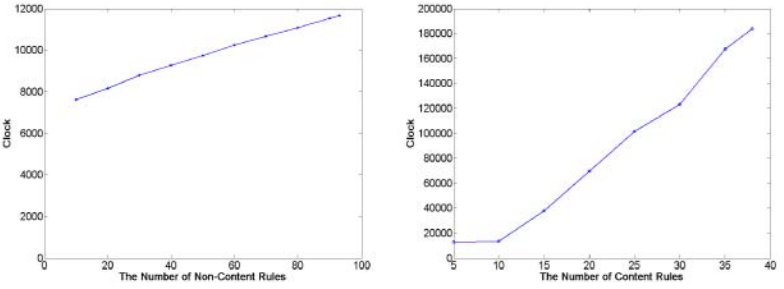
```
byte_not_match#:
  increse packet_NB_offset and curr_packet_base_len
  .if( curr_packet_len >= max_packet_len)
  br[get_next_rule#]
  .endif
  br[NB_read_sdram#]
cmp_end#:
  .if(match_result==0)
  br[mepass#]
  .else
  br[exception#]
  .endif
exception#:
 alert to StrongArm
 br[end#]
mepass#:
  :pass the packet to Engress Block
end#:
  go next to rule check
.endif

/* End Time Read*/
read from CSR to $cycle_xfer1
copy $cycle_xfer1 to endTime
subtract startTime from endTime(runTime=endTime-startTime)
```

5.2 Performance Evaluation of DIDE-S for ICMP

In this section, we describe the performance measuring of ICMP packet processing with ICMP *DIDE-S* engine and estimate the processing speed of the engine. For the experiment, we implement an ICMP *DIDE-S* that supports the same 133 ICMP rules, 95 non-content rules and 38 content rules, as *Snort* 2.0.0 has. Fig. 4 shows the result for the ICMP detection engine, where the time is measured by clocks consumed for processing 74-byte ICMP packets. When none of packets contains intrusion patterns, no rule signature is matched. At worst case, the detection engine must search the signature database thoroughly until the end.

Fig. 4(a) represents the result for ICMP non-content rules. The time till any matching occurred is measured by increasing the number of non-content rules from 10 to 93. The line shows a monotonic increment, having an offset(initialization) cycles that is required for the first comparison. The initialization time comes from the cycles needed to move received packet from SDRAM to SDRAM transfer register and rule signature from SRAM to SRAM transfer register. With respect to the increase by 10 in the rule numbers, the cycles are increased by 500. It takes 11,665 clocks to check all 93 non-content rules, so an average of 125 clocks is consumed for a single non-content rule. Furthermore, 11,665 clocks is required to check a 74-byte ICMP packet, which means that the throughput of each engine is roughly 12Mbps.

Fig. 4(b) represents the result of processing time for ICMP content rules, increasing the number of rules from 5 to 38 by 5. It takes 183,754 clocks to check all 38 content rules, showing an average clocks of 4,835 and resulting in less



(a) Non-content rule detection time (b) Content rule detection time

Fig. 4. Non-Content and Content Rules Detection Time

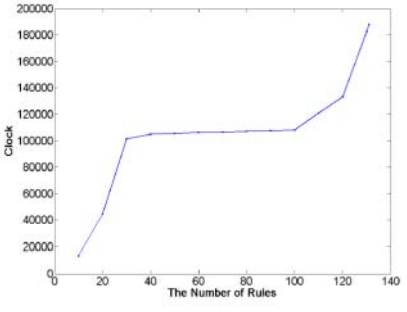


Fig. 5. Total ICMP Rule Detection Time

average clocks than the ones for string field checking of 14-byte patterns (5,416 clocks). The difference can be explained by the fact that the string fields can be checked only when all number fields have passed the non-content rules.

Fig. 5 illustrates the processing times for all 133 ICMP rules consisting of mixed content and non-content rules. The average clocks for processing an ICMP rule is 1,436. Over the regions of 0-30 and 100-133, the line shows steep increases. It is due to the fact that content rules spread over the ranges and the rules take much more time than non-content ones. Thus, the performance of detection engine is quite dependent on the arrangement of content rules. In the experiment, we use the same order of rules as *Snort*'s.

From the results shown in Fig. 4 and 5, the throughput is estimated when all 24 hardware threads of IXP1200 are fully functioned as detection engines and listed in Table 3. When all 24 detection engines and existing 133 rules are used, a packet can be handled with a throughput of 17.52 Mbps. Since the throughput does not take into account receiving and transferring packets but comparison of a packet with the rule signatures, the actual performance of an IXP1200 is expected to be slightly lower than 17.52 Mbps. In addition, the microcode for IXP1200 can be run easily on IXP2400, which operates on 600 MHz clock frequency, has 64 hardware threads, faster memory and high-

Table 3. Packet processing time for ICMP detection engine on IXP1200

	Clocks for a rule (232MHz)	Clocks for all rules (232MHz)	Mbps for all rules (with a DE)	Mbps for all rules (with 24 DE's)
Total Rules	1,436	188,141	0.73	17.52

speed packet interfaces. Therefore much greater performance can be achieved for IXP2400.

6 Conclusion

In the paper, we propose high performance network intrusion detection system, utilizing the features of RISC based network processors, fast-packet processing and flexibility of software blocks. The proposed architecture of NP-NIDS offers not only high performance, but also flexibility in refreshing rule signatures. From the result of performance analysis, much better performance can be expected if we use either IXP2400 or IXP2800, while IXP1200 still has many constraints in speed and control storage. To minimize the number of rule comparison, the rule data structures will be further improved, taking account of rule optimization. And an efficient string matching mechanism on microengine will be further studied.

References

[1] R. Sidhu and V. K. Prasanna, "Fast Regular Expression Matching using FPGAs," IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM01), (2001)
[2] B. L. Hutchings, R. Franklin, and D. Carver, "Assisting Network Intrusion Detection with Reconfigurable Hardware," IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'02), (2002)
[3] Y. H. Cho, S. Navab, and W. H. Mangione-Smith, "Specialized Hardware for Deep Network Packet Filtering," FPL 2002, LNCS 2438, pp. 452-461, (2002)
[4] B.Mukherjee, L. T.Heberlein, and K. N.Levitt, "Network Intrusion Detection," IEEE Network, Volume 8, Issue 3, pp. 26-41, (1994)
[5] *Snort* homepage, available at http://www.snort.org
[6] M. Roesch and C. Green, Snort User Manual, Verson 1.8.6/2.0.0, (2002/2003)
[7] N. Desai, "Increasing Performance in High Speed NIDS," A look at Snort's Internals, (2002)
[8] Intel corporation, "Intel IXP1200/2400 Network Processor Family Hardware Reference Manual," (2001/2003)

Part VI

Internet Application

A SIP-Based Voice-Mail System with Voice Recognition

Yasutaka Otake[1], Yasuhiro Tajima[2], and Matsuaki Terada[2]

[1] Graduate School of Technology, Tokyo University of Aguriculture and Technology
2-24-16, Naka-machi, Koganei-shi, Tokyo 184-8588, Japan
yasutaka,@cc.tuat.ac.jp
[2] Department of Computer, Information and Communication Sciences
Tokyo University of Aguriculture and Technology
2-24-16, Naka-machi, Koganei-shi, Tokyo 184-8588, Japan
{ytajima,m-tera}@cc.tuat.ac.jp

Abstract. In this paper we propose a new voice mail service incorporating voice recognition, a function not realized in existing unified messaging systems. The proposed service, which we have also implemented and evaluated, is characterized by a voice-to-text conversion function with delivery of text and associated voice message by email. A Web-based GUI (Graphical User Interface) provides easy user access to voice messages. Telephone calls to the service are VoIP (Voice over IP) using SIP (Session Initiation Protocol) call signaling.

1 Introduction

Expectations are high for VoIP (Voice over IP) technology to drive demand for a new generation of communications, not only in the traditional use of telephony for telephone calling but in the convergence of data and voice as well.

With VoIP, integrated text (e-mail), voice (phone calls), and fax can all be managed at the desktop. New unified messaging services allow users to retrieve information from anywhere in almost any desired form. For example, services are available to read out text as audible speech, allowing users to have their e-mail read to them over a cellular phone.

In this paper we propose a new integrated voice and data service not found in current unified messaging systems. The proposed service converts voice into text using an existing voice recognition engine. We do not examine in detail voice recognition systems themselves.

2 Voice Mail Overview

Voice mail systems record telephone calls and store them as messages in the appropriate user mailbox of a voice mail center. Users can access their mailboxes and listen to the recorded messages. Voice mail, as with e-mail, is considered to be a highly efficient means of contact and communication and is widely used

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 985–994, 2004.

in many large companies. However, voice mail systems using the traditional telephone network have some disadvantages such as complicated operation and inaccurately recorded messages.

With VoIP-based voice mail systems, voice is handled the same as any other IP data, providing seamless integration with the Web. Management becomes easier with browser-based GUI interfaces, and other applications can take advantage of the compatibility of digitized voice data, such as the sending of voice messages as e-mail file attachments. These are just a few examples of the advanced services possible with an IP-based voice mail system.

We are proposing a new additional service that has not been offered in existing VoIP voice mail systems.

3 Voice Mail with Voice Recognition

Our proposed voice mail system is shown in Fig.1. It has the following functions.

1. A SIP server, using SIP (Session Initiation Protocol) [1, 2] as the call signaling protocol, accepts calls from cellular or traditional telephones through a VoIP gateway or directly from VoIP terminals.
2. The voice mail system records voice stream from callers and saves them as voice message files.
3. The recorded voice message files are converted to text using an existing speech recognition engine and saved as text files.
4. The converted text and voice message files are attached to e-mails that are then sent to the appropriate end user e-mail address.
5. The web-based system allows users to retrieve and manage their recorded voice messages.

To realize our proposed voice mail system, we had to overcome three difficulties. The first concerned the SIP connection for establishing a session with the voice mail system. Ideally, any incoming call that does not connect successfully to the intended recipient should be processed by the voice mail system. Such calls can be handled as direct SIP client connections through a SIP server, or they can be handled with an extended SIP protocol. The latter method was not realistic for us, so we needed to develop a flexible solution.

The second difficulty concerned the voice recognition function. There are two methods for voice recognition: speaker-dependent and speaker-independent. The speaker-dependent method compares waveform patterns from a voice stream to patterns previously registered by a speaker, dynamically contracting and expanding the patterns as needed, in order to recognize the spoken words. The speaker-independent method compares components of the voice stream input with a model of speech components constructed from statistical information and, using linguistic rules, outputs the best voice recognition candidates. Examples of the use of these methods in current voice recognition technology are

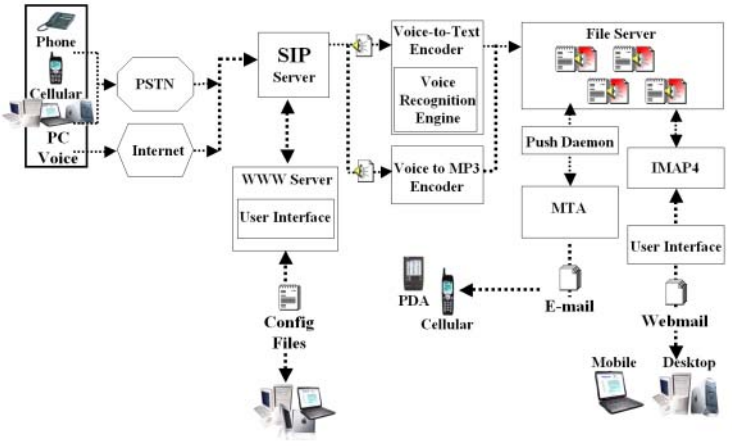


Fig. 1. Proposed Voice Mail System

the speaker-dependent "ViaVoice" engine of IBM Corporation and the speaker-independent "Julius" [3], a large-vocabulary continuous speech recognition engine. Voice recognition in conventional VoIP technology is primarily speaker-dependent and limited to individual names and words. Few systems can handle continuous recognition of spoken sentences. As most voice mail systems are required to accept calls from any caller, not just those who have registered their speech patterns, the speaker-independent method is required when voice recognition will be used with voice mail.

The third difficulty concerned the support of other applied technologies. While our objective was simply to voice-recognize and convert a received voice message to text, and then attach the resulting text and voice message files to an e-mail, we needed to consider the possible use of other technologies such as those supporting the synchronization of text and voice. For example, if the appropriate application is applied to the files produced by our proposed voice mail system, a user could select a specific part of the text in the voice-recognized text file in order to play back only that portion of the message in the voice file.

4 Design

This section describes the proposed voice mail system configuration environment, functions and items to be satisfied by the system design.

4.1 Call Signaling

Call signaling in the voice mail system is performed by SIP (Session Initiation Protocol). A SIP address, similar to that used with e-mail, uniquely identifies each user. This addressing scheme allows a user's location to be determined dynamically and user information to be managed easily.

By interacting with SIP forking proxy servers and redirect servers, users can arrange for incoming SIP calls that cannot be answered for some reason to be forwarded to the voice mail server. In this case the SIP server, upon failing to connect to one or more of a user's registered locations, would then issue an invite request, and subsequently establish a session with, the caller and the voice mail system.

After the SIP server and voice mail server establish a session, the two servers then use SDP (Session Description Protocol) [4] to exchange connection parameters such as the media type and format acceptable to the caller and the destination for media data. SDP messages consist of SIP invite requests, responses and other messages. In this case, the voice mail server must select the appropriate voice codecs for the voice recognition engine, and supply the recorded voice data stream for the conversion process.

We have designed the voice mail server so that it need not issue SIP requests to other servers; it only receives requests and responses and only issues responses. The single SIP function required of the voice mail server is that of an SIP user agent. This is because the voice mail server should only operate on behalf of callers whose incoming connection requests to users fail for some reason. In this system, the voice mail server itself never receives direct connection requests from SIP clients Generally, incoming connection requests to users pass through initial SIP servers, such as forking proxy servers or redirect servers. The SIP server queries for the user's location information and attempts to complete the connection based on that information. If the user has multiple locations registered with one or more SIP forking proxy or redirect server, one location of which may be the voice mail system, the SIP server will attempt to connect to each location in turn until it succeeds. By adopting this configuration, we were able to realize a flexible system for forwarding failed calls to our voice mail system without having to extend SIP server functions to the voice mail server itself.

4.2 Voice-to-Text Conversion by Voice Recognition

Generally, voice mail systems will take and record any call. To convert a caller's speech to text, the voice recognition engine must therefore be speaker independent. A speaker-dependent system would require prior registration of every potential caller's speech patterns in the voice recognition system, which is impractical. In our proposed voice mail system, we selected the speaker-independent Julius [3] voice recognition system for Japanese language. Julius is free and distributed under open license together with the source code. It can process continuous complete sentences, has a vocabulary of tens of thousands of words and has a greater than 90% success rate for a 20,000 word vocabularies dictation task [3]. Rather than develop a voice recognition technology component ourselves, we took advantage of open source Julius and incorporated it as a module in our proposed system.

4.3 VoIP Client Emulation

A general VoIP client is required in our system to acquire the voice data for recognition and conversion to text. Basically, a VoIP client must support jitter control, packet loss detection and other functions in order to provide acceptable real-time communication. IP is a "best effort" packet-switched architecture, and packet delay, fluctuation in delay and packet loss tends to occur. Delay is due to traffic processing in the various routers in the network as well as coding and packetization processing at terminals. Jitter is caused by fluctuations in network load, among other reasons, and packet loss is caused by network congestion. As a result, the VoIP client must perform jitter suppression and packet loss compensation. But in our case, the voice mail system, as a nonhuman entity, does not require real-time communication. As long as the packets arrive, real-time processing for delay and jitter can be disregarded. However, in the case of the voice recognition engine, the success rate may be affected by packet loss due to degradation in the voice stream. Therefore, in this system, packet loss must be compensated

4.4 Message Delivery Function

Converted text from voice-recognized recordings are sent to SMTP (Simple Mail Transfer Protocol) servers as specified by user information registered into the voice mail system. Both the converted text and the original voice message file are attached to the sent e-mails, thereby integrating text-based email and voice mail.

4.5 User Interface Function

The user can access his or her voice mail through a web browser-based GUI display, offering an overall view of the user's mailbox. A message list with reception information, such as the voice recognition result, as well as message contents can be displayed and downloaded.

4.6 Other Functions

There are additional functions for management of user configuration information, adding and deleting users, file management and others.

5 Implementation

5.1 Overview

The proposed system comprises SIP clients, SIP servers, a location server, a voice mail server and e-mail servers. The voice mail system comes into play only when the SIP server, having failed to complete a connection request from an originating SIP terminal to a destination SIP terminal (a voice mail system

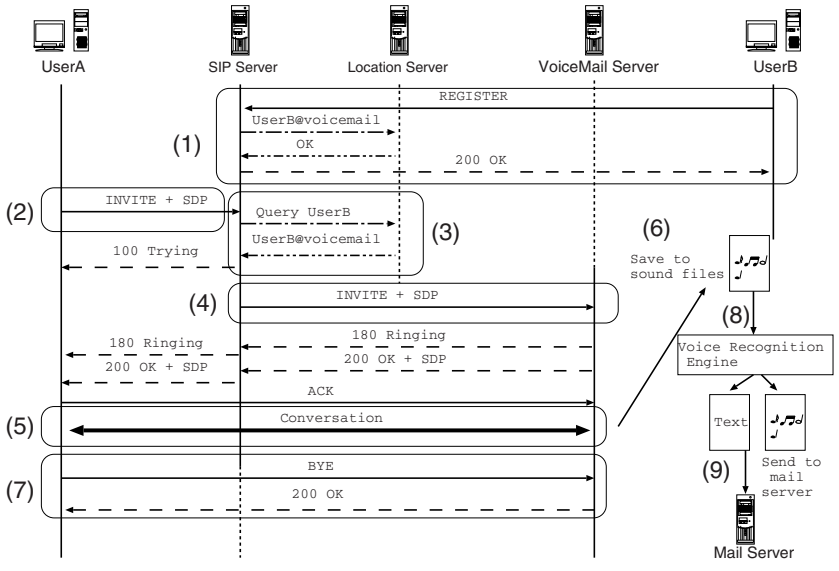


Fig. 2. Location Registration and Voice Mail Call Process Flow

user), then establishes a connection between the originating SIP terminal and the voice mail server. The voice mail system then records the voice stream into a voice message file, voice-recognizes the recorded voice message and converts it to a text file, and then sends both the text file and voice file to a mail server for transmission as an email with text and voice file attachments.

5.2 Basic Operation

The SIP server is not aware of the voice mail system location. It retrieves this information from the location server using normal SIP procedures. Users must individually inform the location server, via the SIP server, as to the SIP address of the voice mail system using the `REGISTER` request, a standard SIP function. The process for registering the location of the voice mail server, and the subsequent flow for an incoming SIP call, are shown in Fig.2.

1. UserB sends a `REGISTER` request to the SIP server indicating the location of the voice mail server.
2. UserA sends an `INVITE` request to the SIP server inviting UserB to connect.
3. The SIP server queries the location server database for location information on UserB.
4. The SIP server sends `INVITE` request/s to one or more UserB location based on UserB's returned location information. If there is no response or an error is returned from each location, the SIP server then transfers the `INVITE` request to the final UserB location, the voice mail server registered in step 1.
5. A session is established between UserA and the voice mail server.

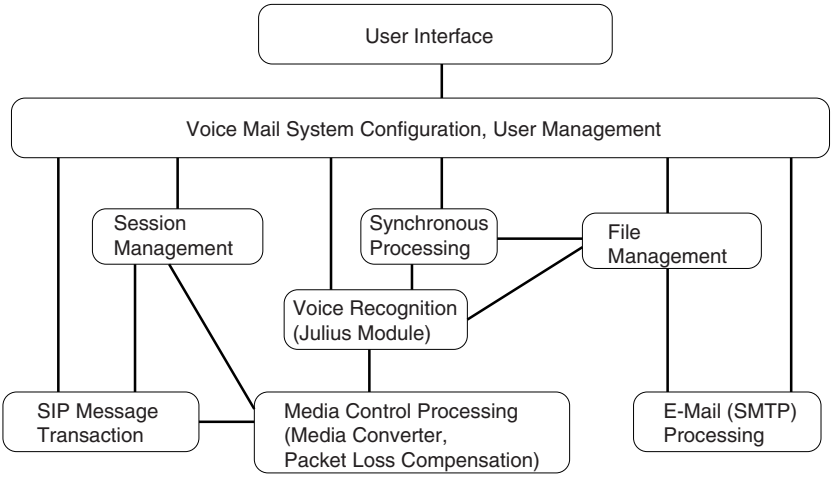


Fig. 3. Software Architecture

6. The voice mail server receives UserA's voice stream and records it to a file.
7. When UserA sends a `BYE` request or the session exceeds a specified time period, the voice mail server closes the session.
8. The voice mail server hands the voice data over to the speech recognition engine module which converts it to text.
9. The voice file is compressed and sent with the converted text to an SMTP server as specified in the user's configuration file.

5.3 Software Architecture

The software architecture for the proposed voice mail system is shown in Fig. 3. There are nine functional components: SIP message processing, synchronous processing, voice recognition processing, file management, media control processing and e-mail (SMTP) processing (Fig. 4). Setup and user management functions are provided through a standard web browser (Fig. 5).

Our goal was to offer a new unified messaging service, that is, a voice mail system with voice recognition. We implemented the prototype software on a FreeBSD 4.8-STABLE operating system with Dual Pentium III 800 MHz processors and 896 MB of RAM. The SIP server was implemented with the "sipdaemon" daemon, and we used the Microsoft Windows RTC Client on Windows XP for the SIP terminal.

6 Evalution

At present, we have not obtained satisfactory results from voice recognition under VoIP. We can suggest the following problems and possible solutions. The

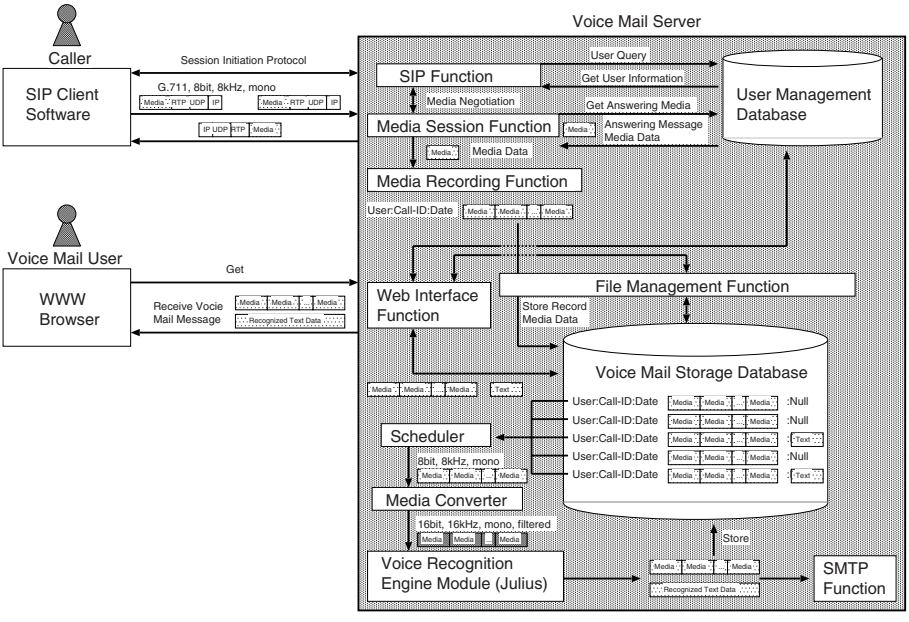


Fig. 4. Module Architecture

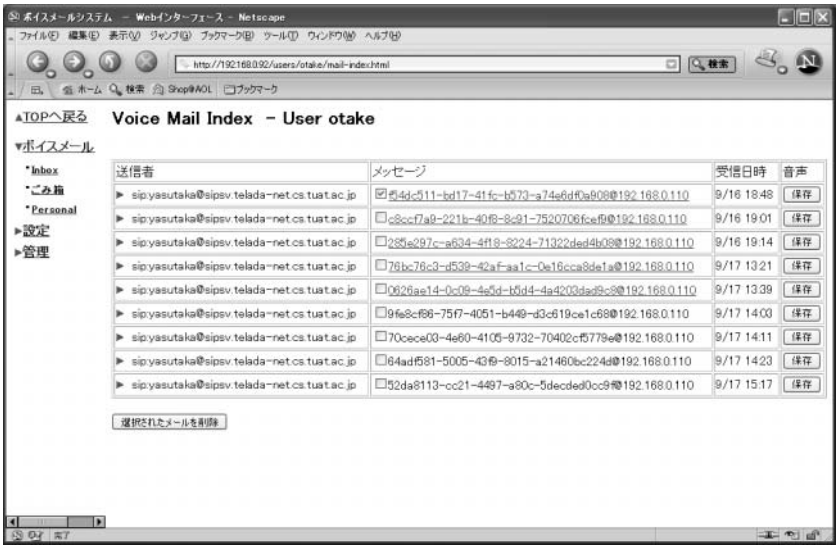


Fig. 5. Web Interface

voice codec of the VoIP terminals are not well suited to the voice recognition engine. The terminals support 8 kHz 8-bit PCM, basic G.711, which offers no compression. The voice recognition engine, on the other hand, requires a 16k Hz 16-bit PCM format [5], which is relatively rare but is the format used in the commercial SIREN codec of the Windows RTC client [6]. We were able to create VoIP client software to support that format, however, the output exceeds available bandwidth available and is therefore not a realistic method. Then, when we transformed voice data to speech recognition engine, we are changing into the supported format for Julius and coped with the problem. In addition, the conversion between voice encoding formats results in a degradation of voice quality and volume, affecting the performance of the voice recognition function. We found it necessary to adjust the strength and clarity of the voice source. Specifically, we had to enhance the frequency band between 300Hz and 3,500Hz. And, as a solution, we check the strength of the voice source. Then, we calculate the peak of the voice level and the average of the voice level in all the sections, and their values are stored. The voice strength is classified into six levels and a volume level is adjusted to suitable level for the voice recognition engine according to them. Moreover, if it is -48dB or less, volume adjustment will not be performed because of no speaking or the silence. Further investigation in this area is required, although we have achieved some improvement in of the voice recognition success rate.

We also suspect load presented by simultaneous VoIP voice stream reception processing and voice mail system processing is resulting in insufficient packet loss compensation and other processing. The CPU load for voice recognition is very high. One solution could be to distribute the voice recognition, voice stream reception and other functions to separate servers, as needed. Another possibility is to serially process voice recognition at times of low system load, at night for example. In this case, users who need their voice mails immediately could be handled with a "priority" field in the SIP headers. One advantage of using G.711 in this system is the reduced CPU load required for voice decoding as compared with other voice codec formats, leaving more CPU processing power for the voice recognition function.

7 Conclusion

We proposed, implemented and evaluated a new voice mail service not realized in current unified messaging systems characterized by a voice-to-text conversion function using an existing speech recognition engine. The proposed service features (1) SIP for call signaling; (2) easy user access to voice mail through seamless integration with the Web; (3) integration with e-mail; (4) compatibility with other applications that handle digitized data. These features are not commonly realized in conventional voice systems.

Acknowledgement

We thank Mr. Toshinobu Himori (Graduate School of Technology, Tokyo University of Aguriculture and Technology) for kindly providing us useful comments for the implementation of our proposed voice mail system.

References

[1] M.Handley, H.Schulzrinne, E.Schooler, J.Rosenberg.: SIP: Session Initiation Protocol. IETF RFC2543 (1999) 986
[2] J.Rosenberg, H.Schulzrinne, G.Camarillo, A.Johnston, J.Peterson, R.Sparks, M.Handley, E.Schooler.: SIP: Session Initiation Protocol. IETF RFC3261 (2002) 986
[3] Kyoto University: Julius — Open source real-time large vocabulary speech recognition engine. online documentation http://julius.sourceforge.jp 987, 988
[4] M.Handley, and V.Jacobson: SDP: Session Description Protocol. IETF RFC2327 (1998) 988
[5] Kyoto University: Julius-3.3 document (in Japanese). online documentation http://julius.sourceforge.jp/3.3/Julius-book-3.3-ja.pdf 993
[6] Microsoft Corp: Media Support in the Microsoft Windows Real-Time Communications Client. online documentation http://msdn.microsoft.com/library/en-us/dnwxp/html/mediainrtcclient.asp 993
[7] H.Schulzrinne, S.Casner, R.Frederick, and V.Jacobson: RTP: A transport protocol for real-time applications. IETF RFC1889 (1996)
[8] G.Camarillo: *SIP Demystified*. McGraw-Hill TELECOM. (2001) 94–190
[9] A.Johnston: *SIP – Understanding the Session Initiation Protocol*. Artech House Publishers. (2001)
[10] A.Johnston, H.Sinnreich: *Internet Communications Using SIP*. Wiley Computer Publishing. (2001)

Network and Application Security in Mobile e-Health Applications

Ramon Martí, Jaime Delgado, and Xavier Perramon

Universitat Pompeu Fabra (UPF)
Pg. Circumval·lació 8, E-08003 Barcelona, Spain
{ramon.marti,jaime.delgado,xavier.perramon}@upf.edu

Abstract. Different IT applications require different network and application security services. We have been working in the area of e-health applications in mobile environments, and we have needed to integrate security services therein. This paper presents a specification of such network and application security services for mobile e-health applications and how we have implemented them. First, various security threats specific of e-health applications are described, like patients' data eavesdropping and manipulation. The different security mechanisms to address these specific security threats are then described, e.g., data confidentiality and integrity. Following, the specification of the network and application security requirements and the implementation possibilities to address them in the mobile e-health applications are described. As an example of network and application security services integrated into an e-health system, the paper includes the description of the mobile e-health application MobiHealth, an application developed within the European Commission co-funded MobiHealth project (IST–2001–36006), focusing on the security services added to it.

1 Introduction

In a digital society, one of the services that will contribute to improve the citizens' quality of life is electronic healthcare, or e-health. A further step is the use of mobile communication technologies to provide the so-called m-health service. Depending on the severity of their diseases, patients will not need to stay at hospitals, but they will be able to lead a normal life while their medical data are being monitored by healthcare professionals.

In this context, data protection and security is a key aspect in order to increase users' acceptance of these new technologies, given the highly sensitive nature of personal health data to be transmitted to and from mobile terminals, and because of the perceived risk for the user's health.

In this paper we present an overview of security threats in mobile e-health applications, and how to introduce safeguards against these threats in an m-health system. Our work is centred on an m-health system architecture based on the concept of a BAN (Body Area Network) linked via mobile communications with a hospital or a healthcare centre. Then we focus on the security services that must be provided by this m-health system and how to implement them.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 995–1004, 2004.

2 Security Threats in Mobile e-Health Applications

2.1 Types of Security Threats

A mobile e-health application, like all information technology systems, is subject to different kinds of security threats. We will not consider here threats of environmental origin (fire, etc.) or accidental ones (user error, software malfunction, etc.). The deliberate threats that we will consider can be categorized into 4 groups:

- Threats to confidentiality.
- Threats to integrity.
- Threats to authenticity (including non-repudiation).
- Threats to system performance (availability, reliability and accountability).

2.2 Protection of Communications against Security Threats

Different security mechanisms are used to provide the security services that are required for data: confidentiality, integrity, authentication, non-repudiation, and access control.

These mechanisms can be incorporated into the different communication layers by using existing security standards, thus providing different security features depending on the layer where security is included. In a mobile e-health application, communications security includes protection of communications between the system components at the different layers:

- Data link layer, where protection can be provided by the communication technologies used in each link, e.g., Bluetooth, Zigbee, or GPRS/UMTS.
- Network layer, where protection can be provided by IPsec [1].
- Transport layer, where protection can be provided by protocols based on SSL/TLS [2, 3], including HTTPS [4].
- Application layer, where protection (data encryption, signature, etc.) can be provided directly by the applications.

One important issue in the final implementation of the m-health application security, apart from the security requirements, is the use of dynamic IP addresses for two of the system components:

- The mobile terminal at the patient's side will normally have a dynamic IP address provided by the network operator.
- The healthcare professional monitoring the patient's data can access the system from the hospital, or from outside the hospital (e.g., from a mobile computer). In the former case, network access will probably be done from a static IP address. In the latter case, however, the computer may have a dynamic IP address provided by the ISP.

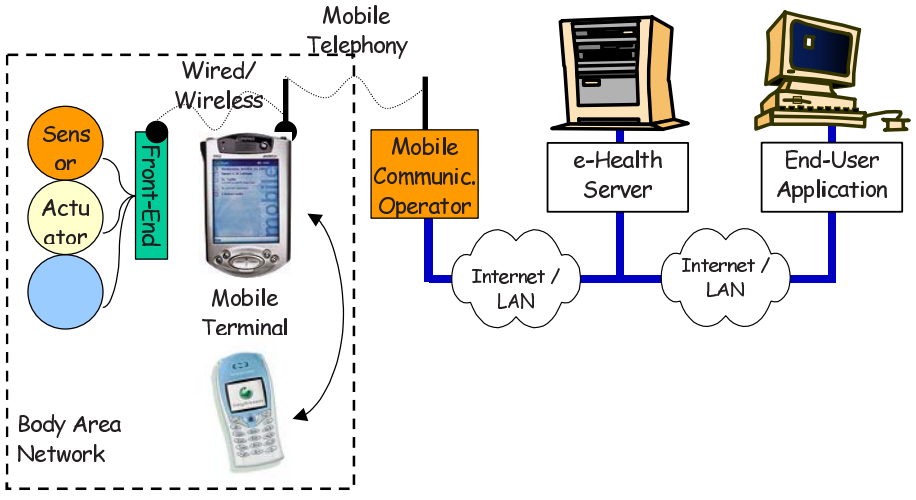


Fig. 1. Overview of a mobile e-health system

3 A Mobile e-Health System Architecture

Mobile e-health systems provide medical staff (doctors and nurses at a hospital or healthcare centre) with real-time remote access to patients' health data. This section gives an overview of a possible architecture for mobile e-health systems, including the description of all components and communication interactions between them. Figure 1 presents the components and communications of this mobile e-health architecture, which are described in the following subsections.

3.1 Mobile e-Health System Components

According to this architecture, an m-health system consists of a BAN (Body Area Network) linked to a hospital or healthcare centre via mobile communications. The concept of Body Area Network is a specialization of Personal Area Network that has recently been introduced in the literature [5, 6, 7]. For our purposes, the BAN is a network of sensors (e.g., a pulse metre or a glucose metre) and/or actuators (e.g., an insuline pump) attached to the patient's body, and interconnected in a star topology to a hub or concentrator, where all data are collected. This interconnection can be through wires, but also through short-range wireless techniques, such as IEEE 802.15.1/Bluetooth or the more recently developed IEEE 802.15.4/Zigbee [8]. The hub is a device directly connected to a mobile communications terminal, typically a cellular phone, through which data can be transmitted virtually anywhere using Internet protocols, and in particular to the hospital or healthcare centre where the patient is being monitored.

In this architecture, a mobile e-health system consists of the following components:

- Sensors, i.e., devices such as a photoelectric cells, that receive and respond to signals or stimuli. Sensors in the BAN can measure pulse, blood pressure, oxygen level, glucose level, etc.
- Actuators, which allow to actuate mechanical devices, such as those connected to a computer by a sensor link. In an e-health system, an actuator can be an insuline pump for patients with diabetes.
- Front-end: hub for all the sensors and actuators in the BAN. It records all the data from all the sensors and actuators, and can send them to the mobile terminal.
- Mobile terminal: a mobile phone or some device with wireless communication capabilities, e.g., a PDA.
- Body Area Network (BAN): patients network, composed of sensors, actuators, a front-end and a mobile terminal.
- Mobile communications operator: network operator providing mobile communications access to Internet.
- e-Health Server (eHS): main server where medical data is received and distributed. It can be installed in the mobile communications service provider or in the hospital.
- End-User Application (EUA): application installed in the hospital (or healthcare centre) for accessing the information from the sensors and actuators, and for sending new configuration parameters to the BAN, through the eHS. It can be run either on a main server in the hospital, which accesses the eHS data and stores them in the existing patients database, or on computers used by authorized employees, which access the eHS data from inside or outside the hospital.

3.2 Mobile e-Health System Communications

Different communication interactions exist in a mobile e-health system. These communications can be done in two ways: from the BAN to the EUA, and from the EUA to the BAN. The communication path between the mobile terminal and the e-Health Server is used to transport application data through application layer protocols.

4 Specification of Network and Application Security Requirements for Mobile e-Health Systems

A preliminary step of our work has consisted in the specification of different security requirements related to the architecture, the system components and the communications of mobile e-health applications [9].

The following general security services are defined to avoid security threats: confidentiality, integrity, authentication, non-repudiation, access control, secure data storage, and secure time stamping. The security service requirements to be addressed by a mobile e-health application, and how they can be implemented, are presented in the following subsections.

4.1 Confidentiality

Confidentiality protects data making it (practically) impossible to interpret for a non-authorized user during communication or storage. These are the confidentiality requirements to be addressed by a mobile e-health application:

- Data transmitted between sensors/actuators and the mobile terminal must not be read by unauthorized persons.
- Data transmitted externally to or from the BAN must not be read by unauthorized persons.
- Data transmitted externally to or from the eHS must not be read by unauthorized persons.
- Traffic characteristics of the transmissions to or from the BAN (how many data are sent, how often, from where to where, etc.) must be concealed so that non-authorized observers cannot infer information about the patient.

4.2 Integrity

Integrity protects data against non-authorized modification, insertion, reordering or destruction during communication or storage. The following are the confidentiality requirements to be addressed:

- Data transmitted between sensors/actuators and the mobile terminal must not be modified by unauthorized persons.
- Data transmitted externally to or from the BAN must not be modified by unauthorized persons.
- Data transmitted externally to or from the eHS must not be modified by unauthorized persons.

4.3 Authentication

Authentication provides the way to corroborate identity of the entities, i.e., sender and receiver, implied in the data creation or communication (entity authentication). It can also provide authentication of the data (data authentication). The following are the authentication requirements to be addressed:

- It must be possible to verify that data collected from the sensors were genuinely produced by the sensors and not forged nor tampered with.
- It must be possible to verify that data transmitted from the BAN were sent by the authentic BAN worn by the authentic patient.
- It must be possible to verify that data received by the BAN from the eHS were sent by the authentic originator.
- It must be possible to verify that data transmitted from the EUA were sent by an allowed user.
- It must be possible to verify that data received by the EUA from the eHS were sent by the authentic originator.

4.4 Non-Repudiation

Non-repudiation protects against unilateral or mutual data repudiation. The following is the non-repudiation requirement to be addressed:

- It must not be possible for a data sender to repudiate the transmission of these data.

4.5 Access Control

Access control protects the system and resources against unauthorized use. The following are the access control requirements to be addressed:

- It must not be possible for unauthorized users to access the BAN.
- It must not be possible for unauthorized users to access the eHS.

4.6 Secure Data Storage

Data storage security protects the stored data against unauthorized use. Depending on the security level desired for every type of application, it may be necessary to fulfill some of the following requirements:

- Data collected from the sensors must not be stored locally in the BAN, except for temporary storage for later transmission while network connection is off.
- A log of data collected from the sensors must be stored in the BAN.
- A log of data transmitted externally to or from the BAN must be kept locally.

4.7 Secure Time Stamping

The following is the time stamping requirement to be addressed:

- It must be possible to unforgeably determine at what time data were originated.

5 Implementation of Network and Application Security in Mobile e-Health Applications

The security requirements, together with use of dynamic IP addresses, have different implications in the security of communication protocols, which must be taken into account in the implementation of security in a mobile e-health application.

- IPsec provides communication security with data encryption and node authentication, based on node addresses. IPsec is not suitable for providing communications security from components with dynamic IP address. Therefore, IPsec is not suitable to provide security to the mobile terminal $\leftrightarrow$ eHS communication, or to the eHS $\leftrightarrow$ EUA communication.

– SSL/TLS provides transport level security to communications requiring data encryption and user authentication, based on server public key certificates. SSL/TLS, and hence HTTPS, is suitable for providing communications security from components with dynamic IP address. Therefore, it is suitable for mobile terminal ↔ eHS or eHS ↔ EUA security.
– Current mobile communication technologies and Bluetooth are suitable for communications requiring data encryption and terminal authentication.

6 Example of Network and Application Security in a Mobile e-Health Application: MobiHealth

This section describes the MobiHealth [10] system and the security services it provides, as an example of the security services of a mobile e-health application. It includes the description of its architecture, its components and the communications between them. MobiHealth is a mobile e-health application developed under the MobiHealth Project, co-funded by the European Commission (IST–2001–36006). The main goal of this project is the development of an m-health service based on new generation mobile communications: the so-called 2.5G (GPRS: General Packet Radio Service) and the new 3G (UMTS: Universal Mobile Telecommunications System).

6.1 MobiHealth Architecture

Figure 2 presents the main components of the MobiHealth system and the communication interactions between them, which are described in the following subsections.

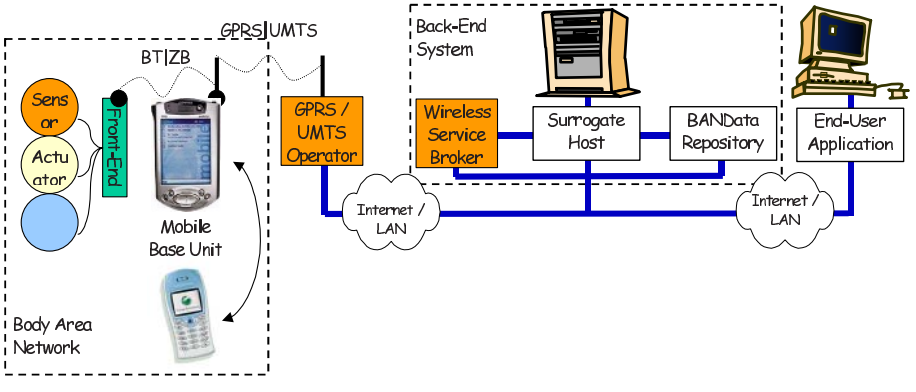


Fig. 2. Overview of the MobiHealth System

6.2 MobiHealth Components

The MobiHealth architecture is largely based on the m-health architecture presented in the previous sections, with some adaptations: the mobile terminal is called the Mobile Base Unit (MBU) in MobiHealth, and a new component is introduced, the Wireless Service Broker (WSB), as detailed below. These are the specific components of the MobiHealth system:

- Mobile Base Unit (MBU): it corresponds to the mobile terminal.
- Back-End System (BEsys): it corresponds to the e-Health Server. It is a system composed of a Wireless Service Broker, a Surrogate Host, and a BANData Repository. A BEsys can be installed in the GPRS/UMTS service provider, or in the hospital.
- Wireless Service Broker (WSB): authenticates and authorizes mobile terminals.
- Surrogate Host (SH): main server, where wireless sensor and actuator objects are "surrogated" inside the wired Internet, and where medical data are received.
- BANData Repository (BDR): a process that acts as client to the Surrogate Host (implemented as a Jini [11] service user of the MBU service provider). In addition, the BDR writes the medical data (i.e., measurements) to persistent storage.
- End-User Application (EUA): in MobiHealth, it accesses data from the BANData Repository (part of the eHS).

6.3 MobiHealth Communications

Different communication interactions exist in MobiHealth. These communications are carried out in both directions: from the BAN to the EUA, and from the EUA to the BAN. The communication path between the MBU and the BEsys is used to transport application data through application layer protocols.

6.4 MobiHealth Security Implementation Summary

Taking into account the different issues related to the security and to the MobiHealth requirements and architecture, several security options have been integrated, based on existing standards. This subsection enumerates the security functionalities integrated into the MobiHealth system.

- Bluetooth / Zigbee security for encrypted and authenticated data transmission between the front-end and the MBU.
- HTTPS with user and server X.509 certificates for encrypted and authenticated data transmission between the MBU and the WSB.
- HTTPS with user and server X.509 certificates for encrypted and authenticated data transmission between the WSB and the SH, if they are on different systems.

- RMI (Remote Method Invocation) security (based on SSL) for the BDR access to the SH data, when both are in different systems.
- HTTPS security for the EUA access to the BDR data.
- No data storage in the "disk", but provision for some data storage for buffering, for the front-end, MBU, GPRS/UMTS operator, WSB and EUA (for employees' computers).
- Secure data storage, with confidentiality and user access authentication, for the BDR and EUA (for hospital workstations with the patients database).

This implementation of the security in the MobiHealth system presents the following advantages:

- Use of standard user-oriented security mechanisms.
- No use of IPsec host-oriented security.
- All communications and data from the mobile terminal to the Surrogate Host are secured through authentication and encryption, independently of the underlying network.

The main disadvantage of this implementation is the following:

- Since security services are added, there is an increase in traffic, which may represent a penalty in system performance. However, according to the measurements we have carried out, the decrease in throughput is small (around 6%).

7 Conclusions

This paper presents the security services required for mobile e-health applications. The introduction of security in these applications will enhance the quality of the service in the form of increased trust and acceptance from the users of m-health services, which will add to the social and economical advantages of mobile healthcare for an important number of citizens. All potential users of the healthcare system, i.e., every individual, can benefit from the improvement in quality of life that m-health represents.

The MobiHealth project has also been presented as an example. The main technical objective of this project has been to prove the feasibility and the advantages of using 2.5G–3G mobile communications in a specific area of application, namely m-health. But it has also other side benefits, one of which has been the implementation, based on existing standards, of the security services that we have presented in this paper, to enhance the quality of the service.

References

[1] Kent, S., Atkinson, R.: Security Architecture for the Internet Protocol, RFC 2401 (1998) 996

[2] Frier, A., Karlton, P., Kocher, P.: The SSL 3.0 Protocol, Netscape Communications Corp. (1996) 996

[3] Dierks, T., Allen, C.: The TLS Protocol Version 1.0, RFC 2246 (1999) 996

[4] Rescorla, E.: HTTP Over TLS, RFC 2818 (2000) 996

[5] Zimmerman, T.: Personal area networks (PAN): Near-field intra-body communication. IBM Systems Journal **35** (1996) 609–618 997

[6] Jones, V., Bults, R., Konstantas, D., Vierhout, P. AM: Healthcare PANs: Personal Area Networks for trauma care and home care. Fourth International Symposium on Wireless Personal Multimedia Communications (WPMC), Aalborg, Denmark (2001) 997

[7] Van Dam, K., Pitchers, S., Barnard, M.: From PAN to BAN: Why Body Area Networks? Proceedings of the Wireless World Research Forum (WWRF) Second Meeting, Helsinki, Finland (2001) 997

[8] IEEE wireless standards web page: `http://standards.ieee.org/wireless/overview.html#802.15` 997

[9] Martí, R., Delgado, J.: Security in a Wireless Mobile Health Care System. MobEA (Emerging Applications for Wireless and Mobile Access) Workshop collocated with WWW2003 conference, Budapest, Hungary (2003) (`http://www.research.att.com/~rjana/Marti-Delgado.pdf`) 998

[10] MobiHealth website: `http://www.mobihealth.org/` 1001

[11] Jini Network Technology: `http://www.sun.com/jini/` 1002

Internet-Based Device Communication Protocol with the Client/Server Role Exchange

Inwhee Joe

College of Information and Communications, Hanyang University
Area: Internet Application
Tel: 02-2290-1088, FAX: 02-2290-1886
iwjoe@hanyang.ac.kr

Abstract. Recently, the Internet-based communication method has been adopted as an open networking solution in the field of remote control and data acquisition. In the current Internet, most networking applications are developed according to the client-server approach. In this paper, we propose an innovative device communication protocol that exchanges the traditional role between client and server to provide a uniform device interface over the Internet. The proposed protocol is implemented as a networking application running on top of the standard TCP/IP protocols. Since the complicated server functionality is excluded from the field devices as a result, their functionality can be minimized as the thin client, thereby making them as small and light as possible in terms of H/W and protocol stack. Moreover, a common Web server is introduced to provide a uniform device interface over the Internet to various client devices. To validate our proposed protocol, we have built a test bed consisting of two types of field devices for security system over the Internet. From these experiments, we ensure that our proposed protocol has been validated, because it works correctly in function and satisfactorily in performance.

1 Introduction

In the field of remote control and data acquisition, traditional communication methods are based on a master/slave approach where a central master controller is connected to various field devices (e.g., sensors and actuators) by a proprietary network using various protocols and interfaces. The master controller is in charge of its devices scattered in the field, and it collects data from the field devices to produce a report to the users or control them in real time. If the field devices are located remotely, this approach causes a high cost in communication and, moreover, in maintenance due to the high complexity from the different communication methods for different devices in a proprietary network.

Recently, the Internet-based communication method has been adopted as an open networking solution to enable the users to access the field devices cost-effectively from anyplace with a simple Web browser using the standard TCP/IP protocols [4],[5]. In the current Internet, most networking applications are developed according to the client-server approach, where one side is the client and

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 1005–1014, 2004.

the other the server [2]. The server is designed to provide some defined service for clients. If this approach applies to the industrial communication field for remote control and data acquisition, a master controller will be the client and field devices will be the servers in that the devices acquire data and offer a service to the controller when it is requested. Actually, this is an ordinary way to develop commercial products nowadays in this field when with the use of the Internet.

In this paper, we propose an innovative device communication protocol that exchanges the role between client and server to provide a uniform device interface over the Internet. Since there are a lot of industrial devices like sensors or actuators spread in the field, it is not practical to implement them as servers. Moreover, the trend is that they are expected to become small, low-cost, and low-power devices (so called "thin client") especially for the ubiquitous computing environment [1]. Therefore, our approach is to minimize the functionality of the field devices by excluding the complicated server functionality from them. That is, their role is changed from the server to the client, thereby making them as small and light as possible in terms of H/W and protocol stack. More importantly, our approach provides a uniform device interface over the Internet to various client devices by introducing a common Web server before their respective service servers.

The remainder of this paper is organized as follows. Section 2 presents a field system for security that consists of two types of field devices, with particular emphasis on how each device is organized and connected to the Internet. Section 3 proposes a novel application-layer protocol running on top of the TCP/IP protocols to minimize the functionality on the device side by exchanging the traditional role between client and server over the Internet. Section 4 describes how the test bed is built for security system over the Internet and the experimental results. Section 5 concludes the paper.

2 Security System

This section describes two types of field devices for security system over the Internet: Access Control and DMR (Digital Multiplex Recorder). The Access Control device is used for access control with access cards, while the DMR device is used for remote monitoring with video cameras. These devices are developed to offer an integrated security solution for the wireline and wireless environments through the Internet using the standard TCP/IP protocols and Web technology.

In particular, the Access Control device provides user authentication by checking the identity. It consists of the traditional access module and the TCP/IP interface module. The access module contains a keypad and a sensor to acquire access information directly from the key input or by sensing the access cards. It is connected to the TCP/IP interface module through the traditional serial communication interface (e.g., RS-232 or RS-485). In the past, the access modules used to communicate directly with the master controller using dial-up modems or leased lines over the proprietary network, which is very expensive and requires high maintenance. On the other hand, the use of the TCP/IP module enables

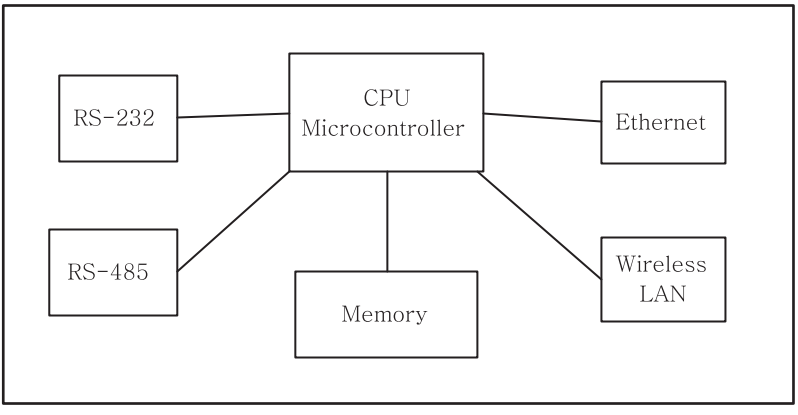


Fig. 1. TCP/IP Module Block Diagram

the access module to communicate over the Internet using the standard TCP/IP protocols as an open network approach.

As shown in Fig. 1, the TCP/IP interface module consists of CPU, memory, and I/O parts just like a normal computer. It functions as an interface between two different types of I/O communications, one for serial communication and the other for LAN-based communication. The serial communication I/O is used to connect to the access module through the RS-232 or RS-485 interfaces. On the other hand, the LAN-based communication I/O is used to communicate over the Internet using the TCP/IP protocols through Ethernet or Wireless LAN according to the connection environment. For the CPU, a micro-controller with the small flash memory is chosen to store and execute our application protocol as well as the TCP/IP protocols as one chip. To fit the small flash memory, the protocol functionality is minimized on the device side by exchanging the role between client and server, so that the Access Control device plays a simple role as the client rather than the traditional complicated server.

With regard to the DMR device, it is used for remote monitoring over the Internet with the video cameras installed in the field for security purposes. After the DMR device receives image sequences as input from the video cameras, it sends them to the TV screen as output, and at the same time, it saves them in the hard disk to be sent out to the Internet. Since the incoming image sequences are analog data, they should be converted into digital data first using the CODEC (Coder/Decoder) before the data are saved in the hard disk. Likewise, the DMR device also plays a simple role as the client in terms of the network application for consistency with other security devices like Access Control devices, thereby leading to a uniform device interface over the Internet. To communicate with the Web server across the Internet, our application protocol and the TCP/IP protocols both are stored in the DMR ROM (Read-Only Memory) and these protocols are running over Ethernet or Wireless LAN depending on the connection environment.

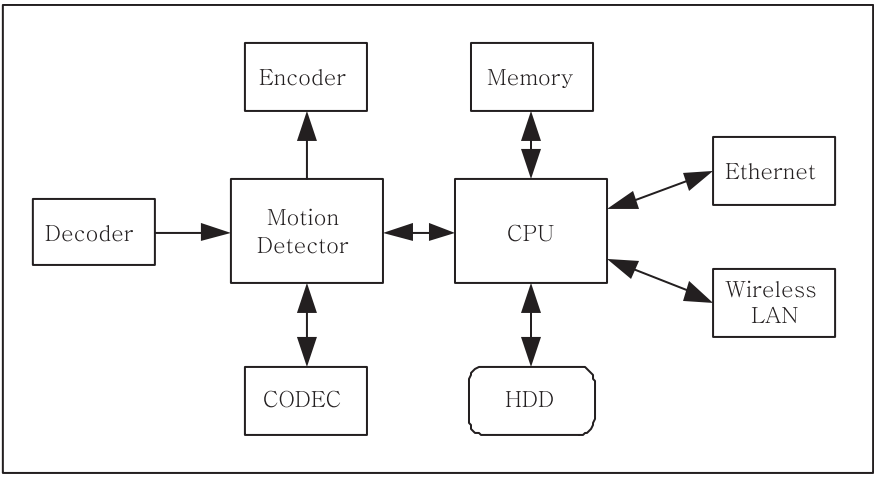


Fig. 2. DMR Block Diagram

As shown in Fig. 2, the DMR device consists of two modules inside, one for video processing and the other for normal computer. In the video processing module, the decoder receives image sequences from the video cameras as input, the encoder sends them to the TV screen as output, the CODEC converts analog data into digital using the Wavelet coding, and the motion detector detects motions according to the given detection conditions. On the other hand, the normal computer module consists of CPU, memory, and I/O parts. It saves the converted image sequences in the hard disk and sends them out to the Internet through Ethernet or Wireless LAN using the TCP/IP protocols and our application protocol residing in the ROM memory.

In addition to remote monitoring, several other services for the DMR device are also available over the Internet with a simple Web browser. For example, the current configuration of a particular DMR device is acquired or changed about the serial port configuration, compression rate, frame rate, sharpness, resolution, and so on. Also, the current configuration for motion detection is acquired or changed about the detection condition, threshold, velocity, detection area and so on. For remote control, two types of services are provided: DMR Reset and Server IP Modify. The DMR reset is used to reset a particular DMR device and start its reboot sequence, while the server IP modify is used to change the IP address of the DMR service server for some reason like service redirection. The DMR service server is a Web server to communicate over the Internet with the DMR client devices spread in the field. Finally, the image data files in the DMR hard disk can be accessed for search and playback.

3 Protocol Description

This section presents our proposed protocol at the application layer running on top of the standard TCP/IP protocols. Instead of UDP, the TCP protocol is chosen as the transport protocol to provide a reliable data transfer to the network applications for our security system. Since the security devices are connected to the Internet using the TCP/IP protocols, they can be accessed from anyplace over the Internet using a simple Web browser in order to control them or collect some data (e.g., access numbers or monitoring data) from them.

Like most Internet applications, the proposed protocol is based on the well-known client-server model. However, we propose an innovative approach to exchange the role between client and server over the Internet so that the functionality of the field devices can be minimized by excluding the complicated server functionality from them. In other words, the industrial devices in the field play a role as the client to send a request to the master controller and wait for a response. Minimizing the functionality allows for smaller size and lower cost as the thin client, which is important especially for the ubiquitous computing environment.

Our protocol is developed by using the FTP (File Transfer Protocol) as the reference, which is one of the most popular Internet applications [3]. Instead of the well-known ports (e.g., 21 for FTP), two ports 915 and 914 are statically assigned for the control port and the data port of the server, respectively. Like FTP, two TCP connections are used in the proposed protocol: Control and Data Connections. The control connection stays up for the entire connection to exchange commands and replies between client and server, while the data connection is created each time data is transferred between them. Like FTP, the control connection is established first in the normal client-server way, i.e., the client devices as the active opener send their connection requests to the server listening on the fixed control port as the passive opener.

On the contrary, the data connection is usually established in such that the server does an active open by sending its connection request to the client in case of FTP. However, our protocol takes a different approach due to the NAT (Network Address Translation) device. Since IP addresses are scarce in the current Internet based on the IP version 4, the NAT device is often used to get around this problem by assigning a small number of official IP addresses only for Internet traffic. Therefore, when a packet is sent out to the Internet, an address translation is needed at the NAT device to change from its internal IP address to an official IP address. However, the problem is that the NAT device recognizes only the well-known ports for incoming Internet traffic. Since both our control and data ports are not well-known ones obviously, the data connection is established in the same way as the control connection so that the client devices always work as the active-opener in the proposed protocol.

In the following, we address how the client device (e.g., Access Control or DMR device) finds its way to the corresponding server over the Internet using the proposed protocol. At power up, the device starts its booting sequence. In the last phase of the booting, it tries to contact the DHCP (Dynamic Host

Configuration Protocol) server to obtain its IP address. DHCP is widely used to alleviate administrative requirements for the installation and initial configuration of new IP machines on the Internet. In response with the request from the client device, the DHCP server returns necessary information like its IP address, subnet mask, default router, and DNS (Domain Name System) address.

Since the IP address of the DNS name server is obtained from the DHCP server, the client device is ready to resolve the domain name device.ipnpok.com of the DCP (Device Configuration Protocol) server, which is hard-coded in the protocol. The DNS is a distributed data base used to map between domain names and IP addresses. To obtain the IP address of the DCP server, the client device contacts its name server by sending a DNS query packet with the domain name in it, before the TCP connection can be established between the client device and the DCP server. As a result, the corresponding IP address is returned in a DNS reply packet to the client device.

Before the security system comes into service, all the security devices should be registered first so that their MAC (Medium Access Control) addresses are stored into the service table of the DCP server. The MAC address (e.g., Ethernet address) is chosen as the identifier of the device, because it is unique globally. Once the IP address of the DCP server is obtained, the client device attempts to contact the DCP server to find out its corresponding service server, such as the Access Control server for the Access Control devices and the DMR server for the DMR devices according to the device type. If the DCP server receives a DCP request from the client device, it searches its service table by the MAC address carried in the request and returns the IP address of the corresponding server provided that the device is registered as valid.

To communicate with the DCP server, the DCP packet format is used at the application layer over the TCP/IP protocols, as shown in Fig. 3. The first Address field indicates the 6-byte MAC address of the client device used as a unique identifier. Typical examples include IEEE 802.3 Ethernet or IEEE 802.11 WLAN MAC addresses. The second Class field represents the 2-byte device type, such as Access Control or DMR devices, which is used to determine the service server

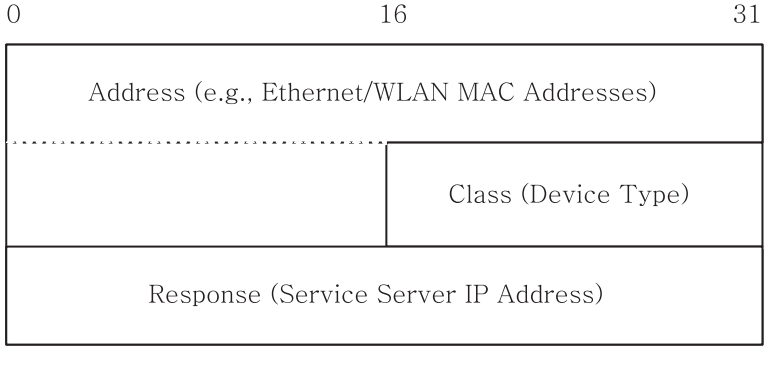


Fig. 3. DCP Packet Format

for each client device. The last Response field indicates the 4-byte IP address of the corresponding service server carried in the DCP reply packet from the DCP server. Once the IP address of the service server is obtained, the client device is now able to communicate with its service server directly. As mentioned earlier in this section, both control and data connections are established in the normal client-server way between client and server over the Internet.

The DCP server is implemented as a Web server over the Internet. It is used to verify that the client device is registered as valid and further, to find its way to the corresponding service server. That is, the DCP server is a common server for various field devices to provide a uniform device interface over the Internet. Like the DCP server, the service server is also implemented as a Web server. According to the device type, there are two types of service servers developed at this point: Access Control server and DMR server. The Access Control server is used for user authentication by checking the access information from its client devices, while the DMR server is used to provide various types of services including remote monitoring and remote control, as described in the previous section.

4 Experiments

As shown in Fig. 4, we built a test bed for security system over the Internet to validate our proposed protocol running on top of the TCP/IP protocols. There are two types of security devices employed in the test bed: Access Control device for user authentication and DMR device for remote monitoring. To check if our security system works as an integrated solution regardless of the connection environment, the test bed provides two different field environments through which these devices connect to the Internet, one for the wireline field with Ethernet and the other for the wireless field with Wireless LAN.

Before these security devices are connected to the Internet, the NAT device and the Firewall are located in between. The NAT device is used to do the

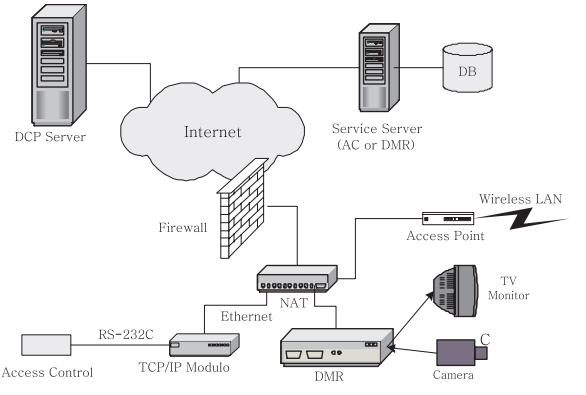


Fig. 4. Security System Test Bed

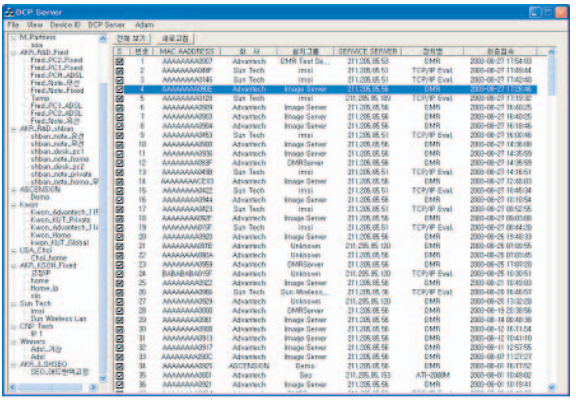


Fig. 5. DCP Server Log Data

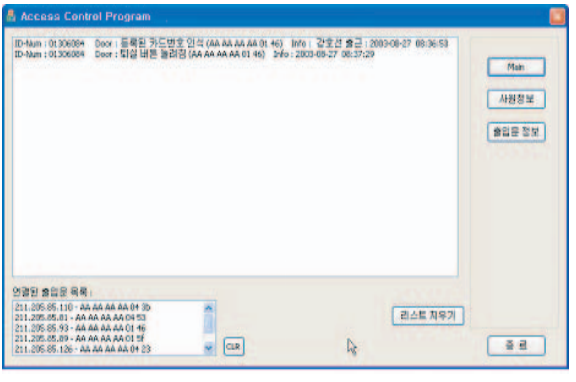


Fig. 6. Access Control Server Log Data

address translation, whereas the Firewall is installed to protect our own network from the outside. Since all traffic from the Web servers are filtered out initially, their IP addresses and port numbers should be registered so that the server traffic can pass the Firewall to reach the client devices in the field on our local network.

The proposed protocol is designed to work as an application protocol between security devices and Web servers over the Internet. If a security device is registered as valid, the DCP server looks it up with the MAC address and logs the current time as its last access time, as shown in Fig. 5. In this case, a DMR device is highlighted in the shaded region as an example. As mentioned earlier, the DCP server is introduced to provide a uniform device interface over the Internet. Once the device obtains the IP address of the corresponding service server from the DCP server, it attempts to reach its service server. Currently, there are two types of service servers. According to the device type, it could be the Access Control server or the DMR server.

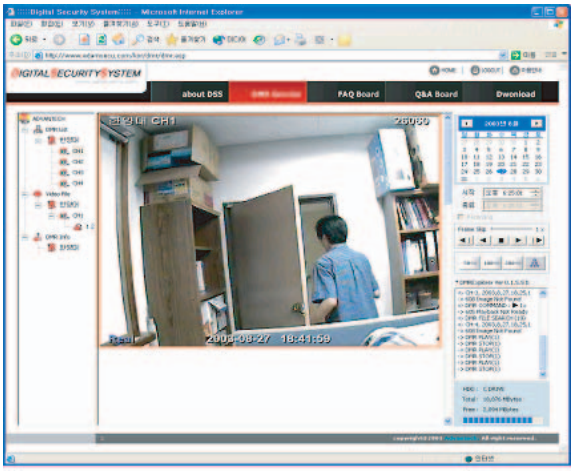


Fig. 7. DMR Server Output Display

Fig. 6 shows some log data produced by the Access Control server, when Access Control devices find their server and send some access information to the server. For example, the log data indicate that somebody checks in or out of the company at particular times. Finally, Fig. 7 shows some output display provided by the DMR server, when DMR devices find their server and send some remote monitoring data to the server. In this case, the output image indicates that somebody tries to break in. From these experiments with the test bed, we can make sure that our proposed protocol has been validated, because it works correctly in function and satisfactorily in performance.

5 Conclusions

In this paper, we have developed an innovative device communication protocol as a network application protocol running on top of the standard TCP/IP protocols in order to provide a uniform device interface over the Internet. The key idea in the proposed protocol is to exchange the role between client and server over the Internet so that the functionality on the device side can be minimized by excluding the complicated server functionality from them. Minimizing the functionality makes the field devices as small and light as possible in terms of H/W and protocol stack. Further, a common Web server is introduced to provide a uniform device interface over the Internet to various field devices. We have also presented experimental results from our test bed consisting of two types of field devices for security system over the Internet. From these experiments, the proposed protocol has been validated, because it works correctly in function and satisfactorily in performance.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci: A Survey on Sensor Networks, IEEE Communications Magazine, pp. 102-114, August (2002) 1006
[2] W. R. Stevens: TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley Publications, (1994) 1006
[3] J. Postel and J Reynolds: File Transfer Protocol (FTP), RFC 959, October (1985) 1009
[4] Pulse: Feature- A Revolution in Industrial Networking?, Industrial Networking and Open Control, June (2003) 1005
[5] B. Nakatani: The Rookery: Improving Network-based Industrial Automation Data Protocols, Linux Journal, July (2002) 1005
[6] R. Simon, C. Diedrich, M. Riedl, M. Thron: Field Device Integration, IEEE International Symposium on Industrial Electronics, June (2001)

FSL3/4 on NEDIA (Flow Separation by Layer 3/4 on Network Environment Using Dual IP Addresses)*

Kwang-Hee Lee and Hoon Choi

Department of Computer Engineering, Chungnam National University
220 Gung-dong, Daejeon 305-764, Korea
{khlee,hchoi}@ce.cnu.ac.kr

Abstract. Solutions for IP address depletion problem can be categorized by two methods. The first method, referred as IPv4-to-IPv6 transition, is the way of replacing IPv4 with IPv6, as a long-term solution. However, IPv4-to-IPv6 transition requires modification of both all IPv4 network equipments and all IPv4 hosts. It also needs incredible amount of times and expense. The latter is a usage of NAT (Network Address Translation), as a short-term solution. NAT translates the IP address and TCP or UDP port of the IPv4 packet header and performs demultiplexing of incoming data flows. The modification of packet is NAT's basic operation but leads to many problems such as IPSec supporting on local network, ALG dependency for supporting various Internet applications, degradation of router/gateway's packet forwarding performance, etc. In this paper, we propose NEDIA which is network environment for sharing public IP address assigned to a router/gateway at the network boundary. We also propose and implement FSL3/4 which is a data flow separation algorithm to overcome limitation of NAT.

1 Introduction

The early Internet, which was constructed to transmit only text document or mail, has grown explosively by advent of WWW (World Wide Web) and various Internet applications. Approximately 1.6 hundred million hosts have already connected to the Internet as of February, 2002[1]. It is anticipated that the IPv4 address will be depleted around the year 2010 because the address is allocated inefficiently, and by advent of various Internet services, home networking, Internet information electronics, and ubiquitous networking. The IP address depletion problem will emerge as a serious issue in developing new Internet services.

Solutions for the IP address depletion problem can be categorized into two methods. The first method, referred as IPv4-to-IPv6 transition[2], is a replacement of the IPv4 that has 32 bit address space with the IPv6 that has 128

* This research was supported by the program for Cultivating Graduate Students in Regional Strategic Industry of the Ministry of Commerce, Industry and Energy.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 1015–1024, 2004.

bit address space, as a long-term solution. However, IPv4-to-IPv6 transition requires modification of both all IPv4 network equipments and all IPv4 hosts. It therefore needs an incredible amount of time and expense.

The latter method is a usage of NAT (Network Address Translation) [3][4] technology, as a short-term solution. NAT is a general technology for IP depletion problem during IPv4-to-IPv6 transition progress. To support IP network communication between local host and global host, NAT translates the IP address and TCP or UDP port on the IPv4 packet header and performs demultiplexing of incoming data flows by referring to the translation table, created by outgoing data flows. The modification of the packet is NAT's basic operation but leads to many problems such as IPSec[5][6] supporting on local network, ALG (Application Level Gateway)[3] dependency for supporting various Internet applications, degradation of router/gateway's packet forwarding performance, etc.

In this paper, we propose NEDIA (Network Environment using Dual IP Addresses) which enables hosts with private IP address to share public IP address that is assigned to a router/gateway at network boundary. We also propose and implement FSL3/4 (Flow Separation by Layer 3/4) which is a data flow separation algorithm for supporting bidirectional communication on NEDIA. The FSL3/4 on NEDIA supports not only full Internet access by only referring to the L3/L4 information of packet but also the transport mode IPSec[6] session and does not degrade router/gateway's packet forwarding performance.

2 Related Works

2.1 NAT (Network Address Translation)

NAT is a short-term solution for the IP depletion problem and a general technology for interworking between local network using private IP address and global network such as the Internet. NAT technology can be divided into Basic NAT, which only translates the IP address and NAPT, translating both the IP address and TCP/UDP port.

Basic NAT runs in the gateway router between the local network and global network, and translates the private IP address in packet's source address field to the public IP address assigned to the NAT router. This method is simple and can be easily implemented and supports bidirectional communication using DNS_ALG[7]. However, the basic NAT has poor IP address reusability because one global IP address is dedicated to one private IP address for connecting between local host and global host.

NAPT (Network Address Port Translation) has overcome the basic NAT's limitation of poor IP address reusability. This technique supports sharing of one global IP address among many local hosts via translation of both the IP address and TCP/UDP port.

Although NAPT has higher IP address reusability than Basic NAT, it is more complicated and has lower translation speed because of using TCP/UDP port as the demultiplexing key. NAPT cannot support port-sensitive Internet application such as Talk, RealPlayer without a special ALG. NAPT must reassemble fragmented packets into one complete packet because only the first fragmented packet has TCP/UDP port information. Furthermore, it cannot support bidirectional communication between the local and global network.

2.2 NAT's Major Limitations

NAT has an inherent operation, i.e., the modification of packet to change the IP address. Therefore it is incompatible with the IPSec which does not permit any modification of packet except normal routing processing. It is also the major cause for degradation of packet forwarding performance of NAT router/gateway.

Many ALGs for supporting various Internet applications are necessary. This is a burden of the NAT router/gateway implementation because NAT router/gateway is often implemented as an embedded device with limited resources. Therefore, whenever a new Internet application appears, NAT software developer makes ALG for that Internet application and adds it on NAT router/gateway.

3 The Proposed Scheme for Sharing Public IP Address

In this section, we describe FSL3/4 on NEDIA as the new public IP address sharing technique. NEDIA is a small LAN segment such as HomeLAN or an office network, and consists of NEDIA hosts and an FSL3/4 router. A NEDIA host is a general personal computer running on Window or Linux operating system and has two IP addresses. One is the private IP address used for communication inside NEDIA. The other is the public IP address for communicating with global network such as the Internet. To support internetworking between NEDIA and global network, we define FSL3/4 as new data flow separation algorithm.

3.1 NEDIA (Network Environment Using Dual IP Addresses)

NEDIA is not a new conceptual network but is similar to NAT/NAPT network. It is a set of NEDIA hosts which have single NIC (Network Interface Card), a configured private IP address and public IP address. Like NAT/NAPT network, the NEDIA host uses the private IP address for communicating with NEDIA host. Unlike NAT/NAPT, the NEDIA host uses the public IP address assigned to the router/gateway's external interface for communicating with external hosts. NAT/NAPT network can be changed to NEDIA by configuring TCP/IP stack, simply.

The NEDIA host uses the private IP address ***"L"*** for communicating with the NEDIA host, and public IP address ***"G"*** for communicating with the external host. This selection of the source address is performed by normal host routing.

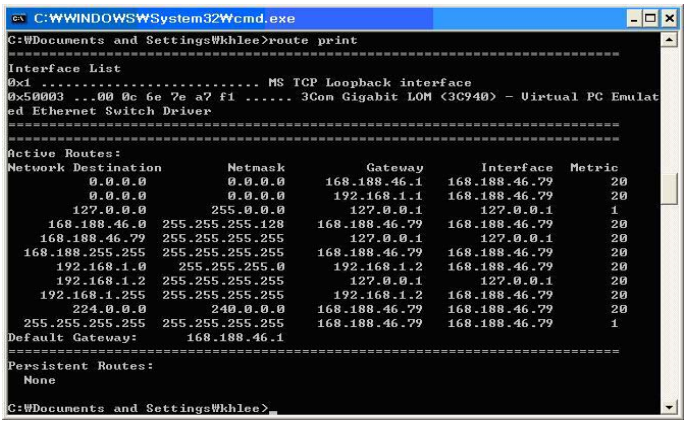


Fig. 1. Host Routing Table

Figure 1 represents the host routing table of a NEDIA host. This host has "192.168.1.2" as the private IP address and "168.188.46.79" as the public IP address. NEDIA host performs host routing for the selection of source address of the packet to be sent to other hosts. When the NEDIA host communicates with another NEDIA host, it selects "192.168.1.2" as the source address of packet by referring "192.168.1.0" routing entry in the host routing table; NEDIA hosts share network prefix "192.168.1.0" with each other. When the NEDIA host communicates with an external host, it selects "168.188.46.79" as the source address of packet by referring to default gateway address "168.188.46.1" in the host routing table because the destination IP address, resolved by DNS service, does not have the same network prefix. Therefore FSL3/4 router can distinguish data flow without any modifications of packet except normal routing processing. FSL3/4 router only refers packet's L3/L4 information for supporting connection between NEDIA and global network.

3.2 Data Flow Separation Algorithms for NEDIA

In this section, we describe FSL3/4 algorithm which runs on the router/gateway at the network boundary and distinguishes data flow by only referring Layer 3/4 information of packet.

The algorithm FSL3/4 consists of FSL3 and FSL4. FSL3 is a data flow separation algorithm which refers only Layer 3 information of packet. FSL4 is a data flow separation algorithm which refers to both Layer 3 and Layer 4 of packet. Although FSL3 is similar to Basic NAT using only the IP address as demultiplexing key, FSL3 distinguishes incoming data flow by the ***source address*** of incoming packet, unlike Basic NAT's ***destination address***. As this difference of demultiplexing key of income data flow between FSL3 and Basic NAT, FSL3 supports more IP address reusability than Basic NAT. In Basic NAT, if an IP address is already used for separation of data flow, it cannot be reused for

distinguishing other data flow. However, FSL3 can reuse one IP address for distinguishing multiple data flows except for the connection from multiple NEDIA host to the same external host.

To overcome the above limitation of FSL3, FSL4 uses both the IP address and TCP/UDP port as demultiplexing key to distinguish data flows.

FSL3/4 is a data flow separation algorithm and runs on the router/gateway at the network boundary. To support connection between NEDIA host and external host, FSL3/4 distinguishes data flow by only referring to packet's L3/L4 information such as protocol ID, source MAC address, destination address, source port, and destination port without any modifications of packet. FSL3/4 also performs L2 forwarding for transmitting incoming packet to NEDIA host. L2 forwarding is to transmit packet using the MAC address of the destination host without L3 routing process.

Figure 2 represents an example of applying FSL3/4 algorithm in NEDIA. NEDIA hosts share the public IP address ***"G"*** of FSL3/4 router's external interface. When a NEDIA host communicates with an external host on the Internet, FSL3/4 router creates an entry of FST (Flow Separation Table), consisted of {protocol ID, source MAC address, destination address as multiplexing IP, source port, destination port}. FSL3/4 router demultiplexes incoming data flow by referring to FST and performs L2 forwarding using FST's source MAC address. We can assume that NEDIA Host-1 knows the IP address of Telnet Server ***"W"*** through ordinary FQDN (Fully Qualified Domain Name) resolving process. When NEDIA Host-1 tries to connect Telnet Server "W", it performs host routing for selection of IP address "G" as the source address of packet. NEDIA Host-1 sends the packet to FSL3/4 router. FSL3/4 router looks up FST by packet's {protocol ID(6), source MAC address, destination address(W), source port(100), destination port(23)}. If a matching entry is not found, FSL3/4 router creates a tuple consisting of {6, NEDIA Host-1' MAC address, W, 100, 23} and inserts it in FST. FSL3/4 router sends the packet through normal routing process to Telnet Server "W". Packet sent by FSL3/4, arrives on Telnet Server "W" via normal routing in the Internet. Telnet Server "W" makes a response packet and sends it to the originator of packet. On receiving the response packet from Telnet Server "W" at FSL3/4 router, FSL3/4 router searches a tuple in FST and performs L2 forwarding using the source MAC address of the matching entry.

When NEDIA Host-2 tries to connect Telnet Server "W", NEDIA Host-2 also follows the same operation as NEDIA Host-1 and uses the same public IP address "G" as source address. FSL3/4 router supports N:1 connection between multiple NEDIA hosts and the same external host.

In Figure 2, NEDIA Host-n has an IPSec peer relationship with IPSec Gateway ***"Z"***. To support transport mode IPSec session between NEDIA Host and external IPSec gateway, FSL3/4 router performs FSL3 algorithm by packet's Protocol ID automatically. Because FSL3 algorithm refers only to IP address of packet to distinguish data flow, FSL3/4 router supports IPSec which does not permit any modifications of packet except normal routing processing.

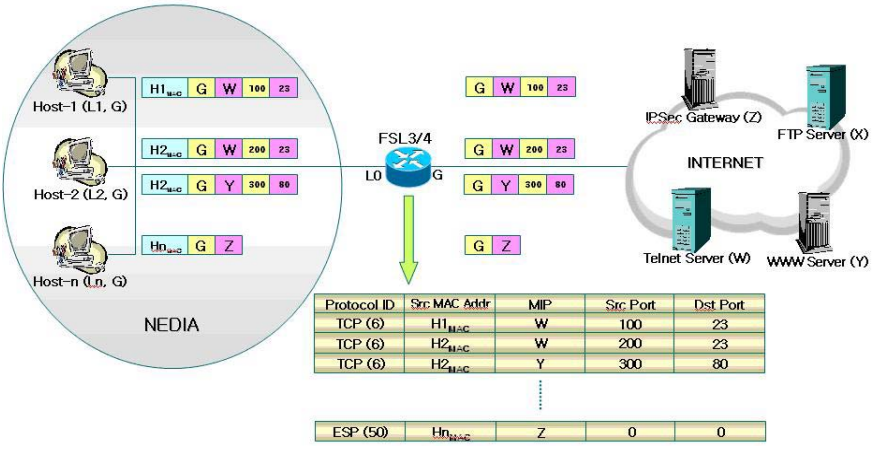


Fig. 2. FSL3/4 on NEDIA

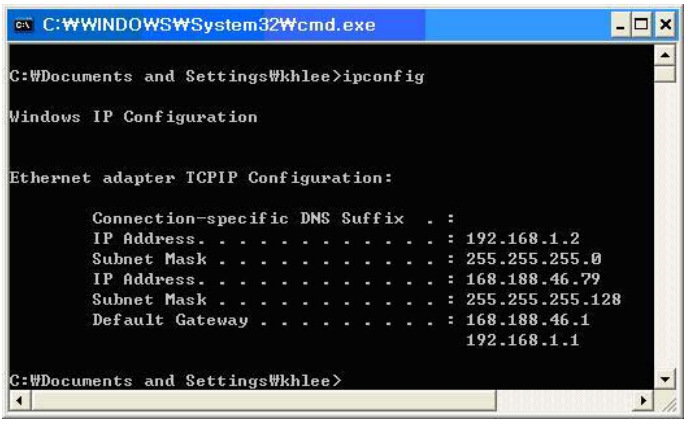


Fig. 3. TCP/IP Configuration on NEDIA Host

4 Implementations

4.1 NEDIA

To construct NEDIA, we used 10/100M Ethernet as transmission media and allocated the private IP address of 192.168.1.0∼24 on each NEDIA host for communication in NEDIA. We allocated the public IP address of 168.188.46.79 on each NEDIA host for communication with the external host.

Figure 3 shows TCP/IP configuration of a NEDIA host.

When a computer with Microsoft Windows operating system boots up or configures TCP/IP protocol stack, it broadcasts an ARP request message with its IP address on the local segment network. This operation, called Gratuitous

Fig. 4. FSL Table Entity Structure

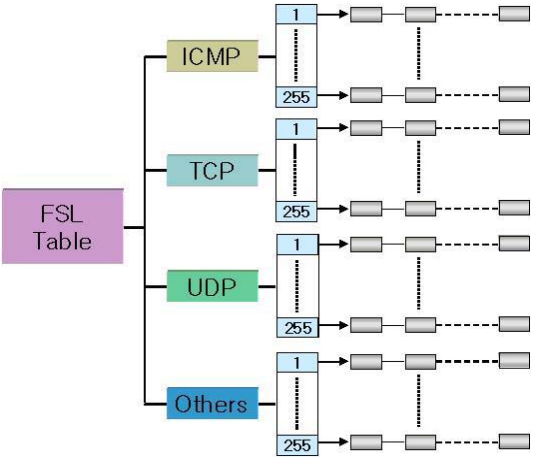


Fig. 5. FSL Table Structure

ARP, is to find a host using a duplicated IP address and to prevent using the same IP address in the local segment network. Because all the NEDIA hosts are configured with the same public IP address "168.188.46.79" with each other, Gratuitous ARP function must be disabled in NEDIA. This Gratuitous ARP is a default function for series of Windows operating system, which can be disabled by editing the window registry using "regedit" program[8]. Gratuitous ARP does not become a serious problem for construction of NEDIA.

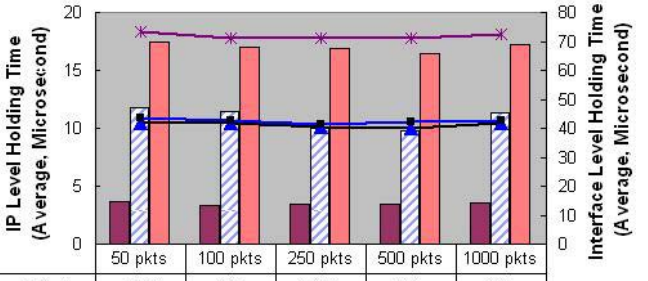
4.2 FSL3/4 Router

We implemented FSL3/4 router on a Linux machine with Pentium-Pro 233MHz, 128Mbyte Main Memory, two 3COM NICs and Linux Kernel 2.2.17. For implementing FSL3/4 algorithm, we modified Linux kernel source.

Figure 4 shows an entity of the table and Figure 5 shows FSL table structure.

5 Experimentation and Performance Evaluation

We constructed an experimental environment to prove practicality and to compare the performance of FSL3/4 on NEDIA with NAPT which is a popular technique to share the public IP address in private network.



	50 pkts	100 pkts	250 pkts	500 pkts	1000 pkts
Normal Routing	3.72	3.38	3.45	3.4	3.6
FSL3/4	11.8	11.48	10	9.79	11.33
NAPT	17.5	17.01	16.89	16.45	17.2
Normal Routing	41.94	41.64	40.52	39.95	41.64
FSL3/4	43.46	42.71	41.16	42.3	42.8
NAPT	73.22	71.21	71.29	71.28	72.36

Fig. 6. Packet Holding Time

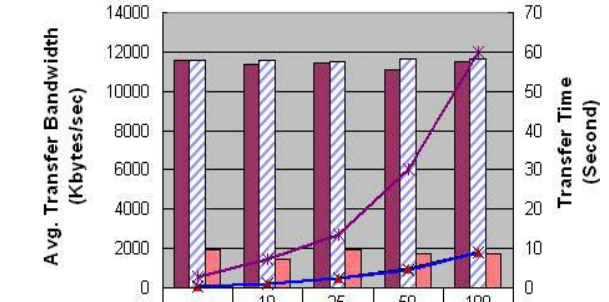
To investigate Internet application supportability of FSL3/4 on NEDIA, we executed Internet applications[9]. As a result, we found that NAPT cannot support many Internet applications without proper ALGs such as Talk, Realplayer, Starcraft, Peer-to-Peer File transfer, etc. These applications can be easily supported by FSL3/4 on NEDIA without additional efforts. Moreover, FSL3/4 on NEDIA supports full bidirectional connection using DNS_ALG.

We measured PHT (Packet Holding Time), average elapsed time and average transmit bandwidth for performance analysis of FSL3/4 on NEDIA.

5.1 PHT

PHT is the elapsed time of packet transmission in the Linux kernel. PHT consists of interface level holding time and IP level holding time. Interface level holding time is the average elapsed time of packet transmission from input interface to output interface. IP level holding time is the average elapsed time for processing routing and data flow separation algorithm in IP layer. We modified the Linux kernel source to measure PHT. The local host sends a ping packet to FTP server "168.188.44.2" by increasing from 50 packets to 1000 packets per second.

NAPT distinguishes data flow by modification of packet. Compared with normal routing or FSL3/4, NAPT has more overhead for packet processing such as additional IP Checksum calculation. Therefore, PHT of NAPT is the largest against FSL3/4 or normal routing. For forwarding of the packet, all three techniques manage a table. The portion of the average PHT time for creating and inserting entry is big when the number of packets to transmit is small. Once the entry is created, time to forward packets takes less. The more packets to transmit, the less average PHT we get. This is the why the PHT in case of 50 packets is larger than others.



	5 Mbyte	10 Mbyte	25 Mbyte	50 Mbyte	100 Mbyte
Normal Routing	11573.69	11385.19	11412.45	11074.95	11491.24
FSL3/4	11573.69	11573.69	11492.5	11650.84	11650.84
NAPT	1973.23	1446.31	1962.16	1743.09	1749
Normal Routing	0.45	0.92	2.3	4.73	9.13
FSL3/4	0.45	0.91	2.28	4.5	9
NAPT	2.66	7.25	13.36	30.08	59.95

Fig. 7. FTP Data Transfer

5.2 FTP Data Transfer

Figure 7 represents the effectiveness of FSL3/4 which distinguishes data flow by not modifying but only referring to L3/L4 of packet and performs L2 forwarding without routing decision.

NAPT modifies packet information of all packets and has additional overhead to calculate checksum. On the other hand, FSL3/4 does not modify incoming packet but only refers to L3/L4 of packet to demultiplex incoming packets. Therefore the performance of FSL3/4 is enhanced about 667 percent on average data bandwidth and reduces the time of transferring data about one seventh against NAPT. FSL3/4 dose not route and calculate the checksum of all incoming packets. This gives better efficiency to transfer data than ordinary normal routing mechanism. For instance, regarding the case of transferring data of 100M bytes, it is equivalent or better than ordinary normal routing.

6 Conclusions and Remarks

In this paper, we analyzed limitations of NAT/NAPT, short-term solutions of the IP depletion problem, and proposed FSL3/4 on NEDIA to overcome the limitation.

FSL3/4 on NEDIA consists of NEDIA as local network environment for sharing the public IP address and FSL3/4 as data flow separation algorithm in NEDIA. FSL3/4 on NEDIA distinguishes data flow by not modifying but only referring to L3/L4 of packet and performs L2 forwarding to transmit incoming packets. All NEDIA host share the public IP address assigned to the

router/gateway at the network boundary and can select the public IP address as the source address of packet to communicate with the external hosts by the host routing technique.

These characteristics of FSL3/4 on NEDIA solve many problems of NAT/NAPT. First of all, FSL3/4 on NEDIA supports bidirectional communication using DNS_ALG. Next, it supports transport mode IPSec session between NEDIA host and external IPSec gateway without additional processing. The third, the performance is improved. In transmitting FTP data, performance of FSL3/4 on NEDIA is enhanced about 667 percent on average transfer bandwidth and reduces the time of transferring data about one seventh against NAPT.

FSL3/4 on NEDIA is more efficient technology than NAT or NAPT.

References

[1] Internet Host Count, http://www.isc.org/ds/WWW-200207/hosts.gif
[2] R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers RFC2893," IETF, August 2000.
[3] P. Srisuresh and M. Holredge, "IP Network Translator (NAT) Terminology and Considerations," RFC2663, IETF, August 1999.
[4] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translation (Traditional NAT)," RFC3022, IETF, January 2001.
[5] M. Holdrege and P. Sriuresh, "Protocol Complications with the IP Network Address Translation," RFC3027, IETF, January 2001.
[6] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC2401, IETF, November 1998.
[7] P. Srisuresh, G. Tsirtsis, P. Akkiraju and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)," RFC2694, IETF, September 1999.
[8] "How to Disable the Gratuitous ARP Function," Microsoft Technical Document, http://support.microsoft.com/support/kb/articles/Q219/3/74.asp, May 2003.
[9] Internet Application List Supported by NAPT, http://www.tsmservices.com/masq/cfm/applist.cfm?group=A

An Analysis of the End System Heterogeneity in Many-to-Many Application Layer Multicast

Kyungran Kang, Sunghoon Kim, and Dongman Lee

School of Engineering, Information and Communications University
119 Munjiro, Yuseong, Daejeon, 305-714, Republic of Korea
{korykang,kimsh,dlee}@icu.ac.kr

Abstract. The lack of the IP multicast deployment and the increased interest of the multi-party applications promotes the use of application layer multicast. The existing many-to-many application layer multicast schemes, Narada, TBCP, and NICE do not concern the heterogeneity of the processing capability of the end systems. We propose the modification of the existing schemes that considers the processing delay in the delivery tree construction algorithm. The performance of the modified versions are evaluated and compared with the original version using the network simulator. The analysis results and the modification proposal can be used as the basis to design a scalable and adaptable application layer multicast schemes.

1 Introduction

The proliferation of the Internet has driven the development of various multi-party applications such as network games, networked virtual environments, collaboration and distributed interactive simulation [9]. The lack of the IP multicast [5] deployment and the increased interest of the multi-party applications promotes the use of application layer multicast, where the underlying network is abstracted by constructing a virtual network using unicast channel among the end systems. The multicast routing is implemented in end systems such that the data delivery tree is constructed over the virtual network and data is forwarded from a node to the child nodes along the delivery tree. The throughput of a multicast session depends on the quality of a delivery tree in application layer multicast.

The representative many-to-many application layer multicast schemes, Narada [3], TBCP [8], and NICE [1] build a delivery tree approximating a unicast routing topology to avoid the redundant use of network links. However, these schemes do not consider another important factor in determining the quality of a delivery tree. In practice, some end users upgrade the computing power frequently while some users persist their old computers. The heterogeneous capability of the end systems will result in the non-uniform processing delay in forwarding the packets to the child nodes in the delivery tree. Therefore, the processing capability of the end systems should be considered in delivery tree

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 1025–1034, 2004.

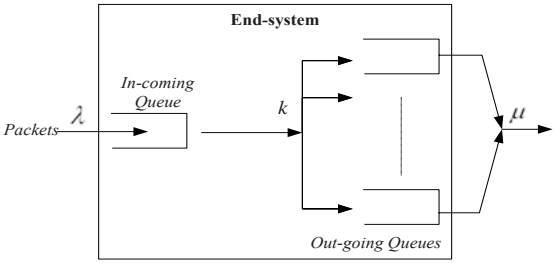


Fig. 1. End system queue model for application layer multicast

construction and reconfiguration as well as the network bandwidth heterogeneity. It will let the end systems to make full use of their high capability and not to incur long processing delay due to low capability.

In this paper, we present a queue model of the end system and, from the equations induced from the queue model, propose a modification to the existing schemes that considers the processing delay, additionally, in the delivery tree construction algorithm. The performance of the original versions and the modified ones are evaluated and compared using the network simulator. The analysis results presented in this paper can be used as the basis to develop a more scalable and adaptable many-to-many application layer multicast schemes.

The organization of this paper is as follows: in Section 2, we describe the queue model and, in Section 3, we describe the modified schemes that incorporate the end system processing delay in delivery tree construction. In Section 4, we show the performance evaluation results of the original schemes and the modified ones. We conclude the paper in Section 5.

2 Modelling of the Impact of the Processing Capability

2.1 Processing Delay Model

The end-to-end delay of data packets in the application layer multicast consists of the delay incurred at the network and the processing delay at the intermediate end systems, which is modelled in this paper. An end system participating in an application layer multicast session maintains an input queue and output queues to relay packets. Incoming packets are arrived at the rate of λ and stored in the input queue, and then copied into k output queues to be forwarded to k child nodes in the delivery tree. We call the number of the child nodes as *fan-out degree* in this paper. Each end system processes the packets at its own processing rate μ from the arrival at its input queue till the disposal out of the output queues. μ depends on the system configuration such as the performance of the CPU, the size of main memory, the access network bandwidth, and the characteristics of the applications running at the end system.

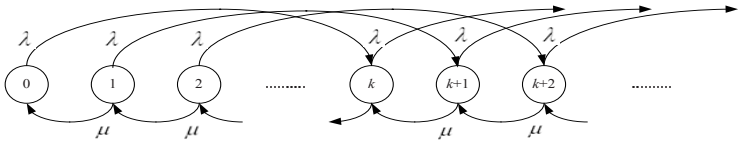


Fig. 2. State Transition Diagram of an end system that has k child nodes

The end system queue model is presented in Fig. 1. We derive the state transition diagram of the end systems as shown in Fig. 2. The state represents the number of packets remaining in the output queue. The processing delay, which is defined as the time from the packet arrival to the completion of the packet forwarding to the child nodes, corresponds to the queue model of $M^{[X]}/M/1$ queue [6], which is a single-server system and allows a bulk input of various batch job sizes.

From the equations in the queue model [6], the processing delay is expressed as (1) in terms of the processing rate, the packet arrival rate, and the number of the child nodes.

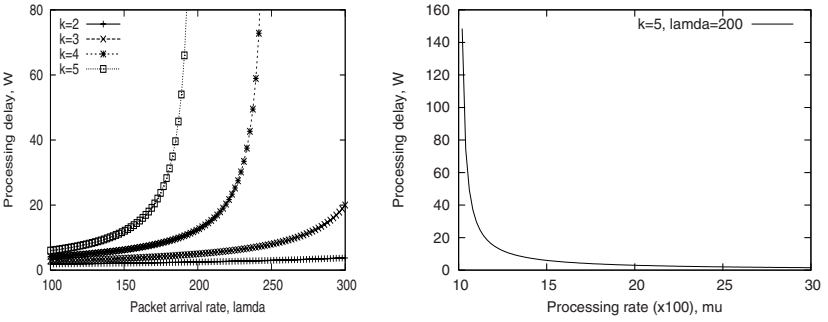
$$W = \frac{(k+1)}{2 \cdot (\mu - \lambda \cdot k)}, \tag{1}$$

where μ is the processing rate of the end system, k is the number of the child nodes in the delivery tree, and λ is the packet arrival rate from the parent node in the delivery tree.

We observe the processing delay induced from the queue model well accords with the delay observed through the simulations, which are presented in our previous work [7].

2.2 The Impact of the Processing Capability

Fig. 3(a) and Fig. 3(b) are derived from (1) to observe the impact of the number of the child nodes, the packet arrival rate, and the processing rate of an end system to the processing delay. Fig. 3(a) shows the impact of the packet arrival rate to the processing delay when the processing rate is fixed. The impact of the arrival rate increases drastically, when it goes over some specific value, since the number of queued packets that are not processed in time is accumulated continuously. Fig. 3(b) shows that, when the packet arrival rate and the number of the child nodes are fixed, the increase of the processing rate decreases the processing delay as well expected. Therefore, in designing the delivery tree construction or reconfiguration algorithm for many-to-many application layer multicast, the number of the child nodes is carefully assigned not to increase the processing delay at a node significantly. On the contrary, if the end system has enough processing capability, it should be allocated with more child nodes than the system of low capability.



(a) Impact of the packet arrival rate (b) Impact of the processing rate

Fig. 3. The impact of the packet arrival rate and the processing rate to the processing delay

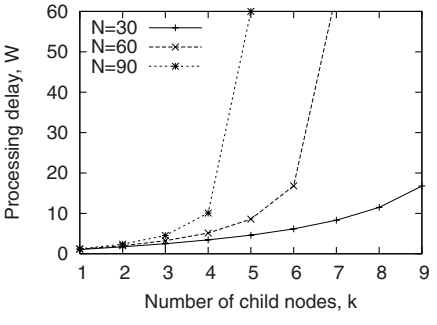


Fig. 4. The impact of the number of the child nodes to the processing delay in a many-to-many session

In a one-to-many session, λ is the same as the sending rate of the source, r, and determined by the source regardless of the number of the participants, n. The increase of the child nodes in the delivery tree will increase the processing delay slightly. However, in a many-to-many session where every participant is a sender, λ, which is simply presented by $r \cdot n$, increases in proportion to the number of the participants. Thus, the impact to the processing delay is more significant than that in a one-to-many session. The resulting W of the existing member is deducted like (2):

$$W(x) = \frac{(k_0 + x) + 1}{2 \cdot (\mu - (n_0 + x) \cdot r \cdot (k_0 + x))}, \tag{2}$$

where x is the number of the newly added child nodes, k_0 and n_0 are the number of the child nodes and that of the members in the session before the new members join, respectively.

Fig. 4 presents four graphs representing the trend of the processing delay change as the number of the child nodes increases when r and λ are fixed as two packet per second (pps) and 1000 packets per second , respectively. As can be expected from the equation 2, the impact of the processing delay increases drastically as the number of the child nodes and the number of the session members increase. It is due to the increase of the packet arrival rates and the number of the forwarding packets at a time.

3 Proposed Modification of the Existing Schemes

Narada [3, 4], TBCP [8], and NICE [1] are considered as representative application layer multicast schemes for many-to-many sessions. From the processing delay model in Section 2, we will induce an equation for processing delay estimation that can be used dynamically during the session. The existing schemes have not taken the processing capability of the end systems into consideration and we will propose a modification of the schemes using the induced equation.

3.1 Processing Delay Estimation

In determining whether to build a new parent-child relationship or not, the processing delay that is expected by the increase of the child nodes should be considered. From (2), if the to-be-child node is a new member, the expected processing delay is induced like (3):

$$W(1) = \frac{(k_0 + 1) + 1}{2 \cdot (\mu - (n_0 + 1) \cdot r \cdot (k_0 + 1))}, \tag{3}$$

where n_0 is the number of the existing members and k_0 is that of the current child nodes. If the to-be-child node is not a new member, the expected processing delay is induced like (4):

$$W(1) = \frac{(k_0 + 1) + 1}{2 \cdot (\mu - n_0 \cdot r \cdot (k_0 + 1))} \tag{4}$$

The difference of (3) and (4) is the packet arrival rate in the denominator. It is because a new comer will incur additional packet arrival as well as the increase of the number of the child nodes while an existing member will increase only the latter one.

3.2 Narada

Narada [3, 4] is designed to support a small-scaled and sparse group for many-to-many sessions, especially multimedia conferencing applications. It forms virtual network as a mesh among the session members and then constructs source-specific delivery tree on the mesh. It assumes that the processing time is zero or the members have the same capability. The maximum fan-out degree is predefined depending on the network bandwidth of the end users, usually from two

to six. When building a mesh and improving the quality of the mesh, a node measures the latency of the link between to-be-neighbor node and evaluate the utility gain of the new link in terms of end-to-end delay reduction and the number of the affected members by the link addition. If the utility gain is larger than the predefined threshold, the link is added.

We propose a modified version of Narada to evaluate the expected processing delay that will be incurred with the neighbor addition using (3) or (4) depending on the situation. The number of the neighbors in the mesh is not same as the number of the child nodes in the delivery tree. However, it works as the upper limitation of the real stress of the end systems. The evaluation of the utility gain is preceded by the expected processing delay evaluation. A threshold of the processing delay is predefined and the addition of the link is allowed only if the expected processing delay is smaller than the threshold. Thus, the modified version can avoid the increase of the child nodes if the processing capability does not allow and, at the same time, can increase as far as the processing capability allows.

3.3 TBCP

TBCP [8] is a simple generic tree configuration algorithm and can be used to build a shared bi-directional delivery tree among the session members. When a new member N joins a session, it requests the would-be parent information to the session leader. The new member determines its parent using the score function value provided by the would-be parent nodes.

$$The score function of member P \equiv max_{\forall A,B \in C_P \cup N} D(P, A, B), \tag{5}$$

where C_P is the set of the child nodes of member P, N is the new member and $D(i, k, j)$is defined as the sum of $d(i, k)$ and $d(k, j)$. $d(i, j)$ is the distance such as RTT between end system i and j. The node of the lowest score is selected as the parent of N. In addition, to control the traffic that flows in the delivery tree, it confines the maximum number of the child nodes of each member into some predefined fixed value.

The proposed modification of TBCP does include the expected processing delay into the modified score function. $D(i, k, j)$ is defined as the sum of $d(i, k)$, $d(k, j)$ and $pd(i)$, where $d(i, j)$ is the same as before and $pd(i)$ is the expected processing delay of end system i calculated using (3). When the tree is reconfigured due to a member leave, (4) will be used for the calculation $pd(i)$. By including the expected processing delay in the score function, the node that is close but overloaded will be excluded from the candidates of the parent of a new member and the node that is far but under-loaded can be included in the candidate parent set.

3.4 NICE

NICE [1] is a representative application layer multicast for a many-to-many session, which arranges the members into hierarchically layered clusters. The size

of each cluster is managed to be within m and $3m-1$, where m is a predefined basic cluster size. A leader of a cluster is selected to reside at the center of the cluster and belongs to multiple clusters and is in charge of the data relaying between the clusters. Due to the hierarchical clustering, it is considered scalable to the number of the participants. It alleviates the routing information overhead significantly by localizing the membership management in a cluster except of the cluster leader. However, since a leader is in charge of the data delivery to the members in the clusters where it belong, the traffic is concentrated to the leaders, especially the leader of the top layer and it contributes to the increase of the end-to-end delivery.

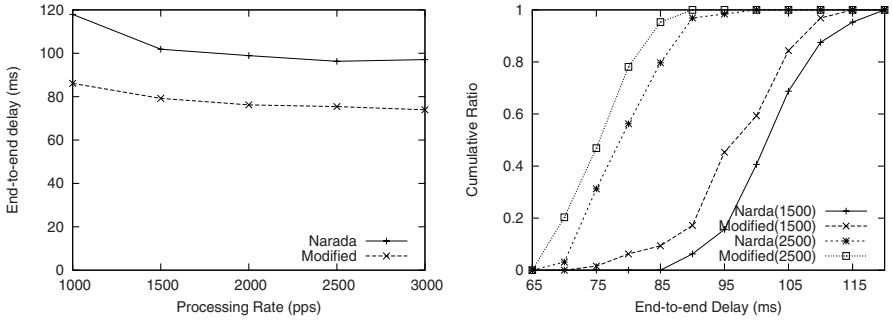
NICE can be modified to consider the processing delay at the end systems by including the processing capability in cluster leader election algorithm. NICE concerns location vicinity most important and the modified scheme will compare the expected processing delay of the candidate leaders when their location vicinity is within small range.

4 Evaluation of the Impact of the Processing Delay

We have simulated the original and modified schemes of Narada, TBCP, and NICE with the network simulator ns-2 [10] on Intel XEON(R) 2.0GHz dual machine running RedHat(R) linux 8.0. The network topology is generated by GT-ITM [2] with 100 routers and 100 end systems. All the link delay is assigned as 20 ms for the links that connect the center nodes of the topology and 5 ms for the other links except those that connect a router and an end system. An end system is attached to a router with a link of 2 ms delay. The bandwidth of each link is assigned as 100 Mbps. We provide the network bandwidth large enough not to lose the packets but the processing rate of the end systems is not high enough to fully process the packets from the network bandwidth.The number of the session participants is fixed as 30 and the participants are randomly selected from the 100 end systems. The members join the session at random time in predefined interval, which may result in different delivery tree run by run. They do not leave during the session. The session duration is 1000 seconds. The data packet size is one kilo bytes. The basic cluster size, m is set as three in NICE simulation. The number of the neighbors is confined from three to six in the original Narada, according to the experiments presented in [3].

4.1 Narada

Increasing the processing rate from 1000 packets per second (pps) to 3000, the average end-to-end delay and the cumulative distribution of the end-to-end delay are observed. With the end systems of low capability, the original Narada and the modified scheme may have the fan-out degree to be in a similar bound. However, as shown in Fig. 5(a), the modified scheme shows about ten percent 50 percent lower end-to-end delay. The delay reduction is achieved by the increase of the child nodes of each end system. The modified scheme expands the breadth of



(a) Impact of the processing rate

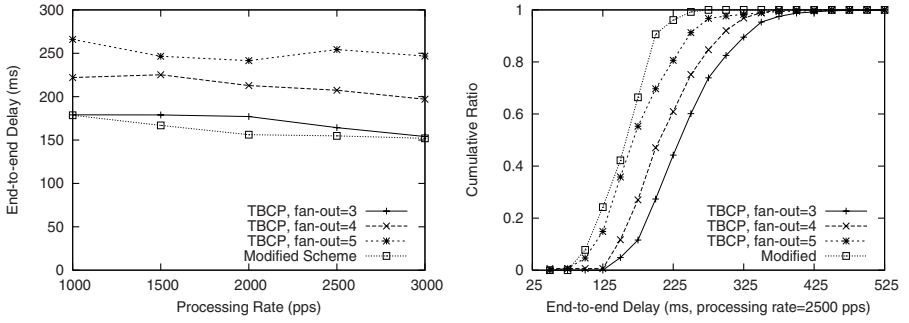
(b) Cumulative distribution of end-to-end delay

Fig. 5. Narada: The average end-to-end delay and cumulative distribution of the end-to-end delay in Narada and the modified version

the tree according to the processing rate while the original Narada builds a slim delivery tree regardless of the processing rate. The modified scheme allows each end system to accept as many child nodes as its processing capability allows. As a result, as shown in Fig. 5(b), when the processing rate is fixed as 1500 and 2500, about twenty percent more members have the average end-to-end delay lower than a given end-to-end delay with the modified scheme, compared those with the original scheme.

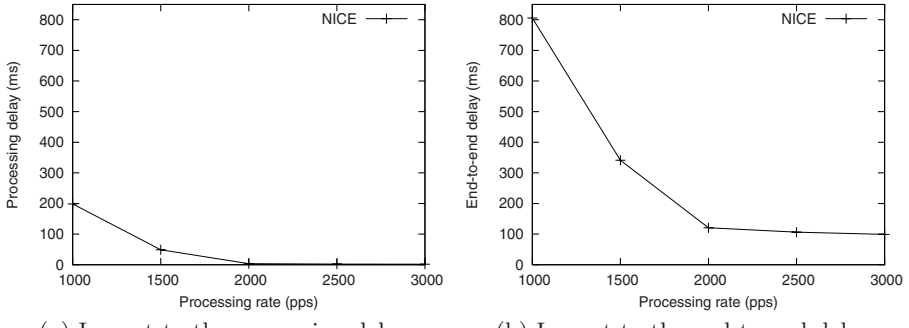
4.2 TBCP

The average end-to-end delay and the cumulative distribution of the end-to-end delay are measured like the case of Narada, which are presented in Fig. 6(a) and 6(b), respectively. The behavior of the original TBCP is observed by increasing the fan-out constraint from three to five. As well expected, the average end-to-end delay decreases as the processing rate increases and the modified scheme shows the lowest delay in Fig. 6(a). By the way, when the processing rate is 2500 pps and the fan-out constraint is three, the delay of the original TBCP scheme increases about 10 ms. It is due to the increase of the network transmission delay, which is resulted from the increased number of the network hops between the end systems. The high processing rate results in small delay and the increased network hop counts have an visible affect on the end-to-end delay. Thus, the end-to-end delay increases even with a high processing rate. However, as Fig. 6(b) shows, the end systems running the modified TBCP have smaller end-to-end delay than the original TBCP schemes regardless of the processing rate.



(a) Impact of the processing rate (b) Cumulative distribution of end-to-end delay

Fig. 6. TBCP: Average end-to-end delay and cumulative distribution of the end-to-end delay in TBCP and the modified version



(a) Impact to the processing delay (b) Impact to the end-to-end delay

Fig. 7. NICE: The processing delay of the top layer members and its impact to the average end-to-end delay

4.3 NICE

Fig. 7(a) and 7(b) show the performance of the NICE in terms of processing delay of the top layer members and the average end-to-end delay of the session members, respectively. The long processing delay of the top layer members occupies about 25 percent of the average end-to-end delay when the given processing rate is 1000 pps. Since the members that reside at the opposite side of the delivery tree centering the top layer members exchange data packets through the top layer members, the delay at the top layer contributes to the long end-to-end delay. The processing delay of the top layer members decreases as the processing rate of each end system increases but occupies the similar ratio of the end-to-end delay. We tried the modification of NICE but the modified scheme did not show visible change in the end-to-end delay. The locality is the most important

factor in cluster configuration of the original scheme and there is little room to introduce the processing capability. Moreover, the hierarchical clustering of NICE cannot avoid the concentration of the overhead at the top layer members, inherently.

5 Concluding Remarks

In this paper, the impact of the processing capability of the end systems is investigated starting from the end system queue model and the existing schemes have been evaluated from that viewpoint with the simulation. Based on the observation and the proposed model in this paper, we will devise more scalable many-to-many application layer multicast schemes that are also adaptable to the end system capability as well as the network bandwidth heterogeneity. We will progress the research, in parallel, to devise the mechanism to estimate the processing capability from the practical measurements.

Acknowledgements

This work was supported in part by the National Research Laboratory Program funded by Ministry of Science and Technology, Republic of Korea under Grant M1-0104-00-0130.

References

[1] Banerjee, S., et al.: Scalable Application Layer Multicast. Proc. of ACM SIGCOMM 2002, ACM, New York (2002) 205-217 1025, 1029, 1030
[2] Calvert, K. et al.: Modelling Internet Topology. IEEE Communications 6(1997) 160-163 1031
[3] Chu Y.: A Case for End System Multicast. Proc. Of ACM SIGMETRICS 2000. ACM Press, New York (2000) 1-12 1025, 1029, 1031
[4] Chu, Y., et al.: Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. Proc. of ACM SIGCOMM 2001. ACM Press, New York (2001) 55-67 1029
[5] Deering, S.: Host Extensions for IP Multicasting, Internet RFC 1112. Virginia (1989) 1025
[6] Gross, D. and Harris, C. M.: Fundamentals of Queuing Theory. 3rd edn. John Wiley & Sons, Inc., New Jersey (1998) 116-122. 1027
[7] Kim, S.: An Analysis of Impact of Processing Delay in Many-to-many Application Layer Multicast Protocols. Proc. of KISS Fall Conference, KISS, Seoul (2003) 289-291 1027
[8] Mathy, L. et al.: An Overlay Tree Building Control Protocol. Lecture Notes in Computer Science, Vol. 2233, Springer-Verlag, Berlin Heidelberg New York (2001) 76-87 1025, 1029, 1030
[9] Quinn, B. and Almeroth, K.: IP Multicast Applications: Challenges and Solutions. RFC3170, Internet Society, Virginia (2001) 1025
[10] NS-2, Network Simulator 2, http://www.isi.edu/nsnam/ns/ 1031

MARE: A Fault-Tolerant Mobile Agent System

Kyeongmo Park[1] and Arun Sood[2]

[1] School of Computer Science and Information Engineering
The Catholic University of Korea
Bucheon-si, Gyeonggi-do, 420-743, ROK
kpark@catholic.ac.kr
[2] Department of Computer Science, George Mason University
Fairfax, VA, 22030-4444, USA
asood@cs.gmu.edu

Abstract. The replication and voting are of great importance to achieve fault tolerance and security in open distributed systems. We present our replication system with voting, called Mobile Agent Replication Extension (MARE), and evaluate the performance. The system makes mobile agents fault-tolerant and also detects attacks by malicious hosts. The reliability and performance issues involved in mobile agents for Internet applications are explored. As a part of our experimental studies, the effects of varying the degree of replication, passive and active replication methods, and voting frequencies are examined. We quantify our prototype's performance and conclude with the future directions of our work.

1 Introduction

Today, one of the most exciting areas of growth in the computer industry is the mobile computing device. Everything from laptops to palmtops, from cars to cellular phones, accesses Internet services to accomplish user tasks. Typically, these mobile devices have unreliable, low-bandwidth, high-latency wireless network connections. With the rapid growth of the services and information on the Internet, a great number of people have ubiquitous access to an astonishing amount of information from anywhere or everywhere. Internet terminals become commonplace in public spaces, such as government offices, school libraries, airports, and hotels. Web email services make it convenient that users are able to access their email from any terminal.

Java already offers support for Web access on cell phones, but surfing has not taken off due to relatively slow connection speeds and cramped screens that make it hard to display Web pages designed for PCs. Java software technology for wireless mobile users, e.g., the Sun MIDP (Mobile Information Device Profile)[11], is designed to address some of those shortcomings and is expected to offer a big leap in support for Internet data services. Ultimately, wireless mobile users will have full access to their files and applications from any computer or cell-phone.

Mobile agents are autonomous computer programs that can migrate from host to host in a heterogeneous network at times and to places of their own

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 1035–1044, 2004.

choosing. The state of the running program among a set of networked hosts, is saved, transported to the new host, and restored, allowing the program to continue where it left off. Mobile agents have been proposed for a variety of applications in the Internet and other large distributed systems [2,4,7]. Almost all major Internet sites are capable of hosting and willing to host some form of mobile agents.

A mobile agent differs from a traditional operating system process. Unlike a process, a mobile agent knows where it is executing in a distributed system at any point in time. A mobile agent is aware of the communication network and makes an informed decision to move asynchronously and independently from one node to another during execution. Mobile agent systems differ from process migration systems. In a process-migration system the system decides when and where to move the running to balance workload, whereas the agents move when they choose through a jump or go statement. It has been shown in [6] that agent migration is much faster than migration of traditional processes.

Although many system design schemes to build mobile agent systems have been proposed, they share a common layered infrastructure, as shown in Fig. 1. An application creates mobile agents running over the agent platform that supports the possible action of mobile gents. The platform utilizes the resources and communication channels provided by the operating system and network of the underlying architecture.

Mobile agents are a very attractive paradigm for distributed computing over the Internet, for several reasons, including reducing vulnerability to network disconnection and improvements in latency and bandwidth of client-server applications [4]. Mobile agents carry the application code with them from the client to the server, instead of transferring data between a client and a server. Since the size of the code is often less than the amount of data interchanged between the client and the server, mobile agent system provide considerable improvement in performance over client-server computing. Thus, the use of mobile agents is expanding rapidly in many Internet applications [2,4].

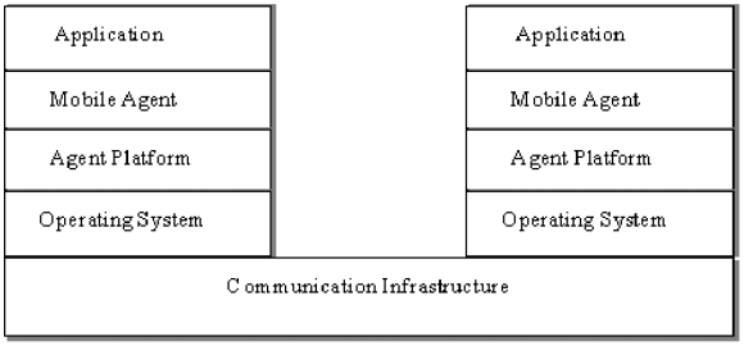


Fig. 1. General Mobile Agent Infrastructure

Computing over the Internet is not dependable. Hosts connected via the Internet constantly fail and recover. The communication links go down at any time. Due to high communication load, link failures, or software bugs, transient communication and node failures are common in the Internet. Information transferred over the Internet is insecure and the security of an agent is not guaranteed. The Internet is unreliable. Therefore, reliability is an important issue for Internet applications [7,10].

Fault tolerance guarantees the uninterrupted operation of a distributed software system, despite network node failure. So, it is important to make mobile agents fault-tolerant. Errors should be detected and recovered. In this paper, we address the fault tolerance and performance issues for mobile agent systems running across the Internet. We present our replication system with voting, called *Mobile Agent Replication Extension (MARE)*. The system makes mobile agents fault-tolerant and also detects attacks by malicious hosts.

The rest of this paper is organized as follows. Section 2 provides basic concepts for a fault-tolerant model for mobile agents. We describe a replication approach that is devised to design reliable Internet applications. We present an architectural framework making mobile agent fault-tolerant and discuss the replication schemes. Section 3 describes experiments we ran to explore performance of replication and voting in a network setting. The effects of varying the degree of replication, the active and passive replication methods, and the voting frequencies are experimentally examined. Finally, in Section 4, our conclusions are presented.

2 MARE: A Fault-Tolerant Mobile Agent System

While a mobile agent model overcomes limitations of the traditional client-server model of distributed computing, there are several fundamental research issues in the design, implementation, and deployment of mobile agent systems running over the Internet. These include agent fault tolerance, agent security, and inter-agent communication and synchronization. In this work, we focus our study on the agent fault tolerance with replication and voting to build a reliable mobile agent system.

2.1 Approach

Our approach offers a user-transparent fault tolerance in agent environments. The user can select a single application given to the environment and can decide for every application whether it has to be treated fault-tolerant or not. That is, the user or the application itself can decide individually, if and when fault tolerance is to be activated. The execution of fault-tolerant and non-fault-tolerant applications is possible. Thus, to enable fault-tolerant execution, it is not necessary to change the application code. The separation between application and agent kernel platform facilitates user transparency. Once mobile agents are injected into the network, the users do not have much control over their execution.

If the action for fault tolerance was dictated by a monitor instance, the autonomy was limited. All decisions that are made by an autonomous agent would need to be coordinated with the monitor. To enable activation of fault tolerance during runtime, the complete agent is replaced with one that carries those functionalities with it. This would increase demands for memory and computing time. So a modular exchangeable composition of mobile agents is required. The required modularity and separation between the application and agent platform imply that the functional modules should wok independently and in parallel. The application can influence the agent behavior. It is possible to affect the behavior of a mobile agent during runtime.

2.2 Agent Fault Tolerance

The main goal of this work is to provide fault tolerance to mobile multi-agents through selective agent replication. Multi-agent applications reply on the collaboration among agents. Replication is a technique used widely for enhancing Internet services. Replication of agents and data at multiple computers is a key to providing fault tolerance in distributed systems. The motivations for replication are to improve a service's performance, to increase its availability, and to make it fault-tolerant. If one of the involved agents fails, the whole computation can get damaged. The solution to this problem is replicating specific agents. One must keep the solution as independent and portable as possible from the underlying agent platform, so as to be still valid even in case of drastic changes of the platform. This offers interoperability between agent systems. The properties of agent systems are dynamic and flexible. This increases the agent's degree of proactivity and reactivity. Note that replication may often be expensive in both computation and communication. A software element of the application may loose at any point in progress. It is important to be able to go back to the previous choices and replicate other elements.

In the passive model of replication, there is a single 'primary' or 'master' replica manager (RM) at any time and one or more secondary RMs - 'backups (slaves)'. Front-ends communicate only with the primary RM to obtain the service. The primary RM executes the operations and sends copies of the updated data to the backups. If the primary fails, one of the backups is promoted to act to the primary.

The passive replication system implements linearizability if the primary is correct, since the primary sequences all the operations upon the shared objects. If the primary fails, then a backup becomes the new primary and the new system configuration takes over: the primary is replaced a unique backup and the RMs that survive agree on which operations had been performed when the replacement primary takes over.

The passive model is used in the Sun NIS (Network Information Service)[11], where the replicated data is updated at a master server and propagated from the master to slave servers using one-to-one rather than group communication. In NIS, clients communicate with either a master or slave server but they may not request updates. Updates are made to the master's files.

In the active model, the RMs are state machines that play equivalent roles and are organized as a group. Front-ends multicast their requests to the group of RMs and all the RMs process the request independently but identically and reply. If any RM crashes, then this need have no impact upon the performance of the service, because the remaining RMs continue to respond in the normal way.

Schneider [9] has proposed active replication with majority voting to obtain a consensus on the computation performed by a set of replicated nodes. This active replication system achieves sequential consistency. All correct RMs process the same sequence of requests. The reliability of multicast ensures that they process them in the same order. Since they are state machine, they all end up with the same state as one another after each request. Front end's requests are served in FIFO order, which is the same as program order. The active system does not achieve linearizability. This is because the total order the RMs process requests is not necessarily the same as the real-time order the clients made their requests.

A simple agent computation might visit a succession of hosts, delivering its result messages to an actuator. The difficulties here arise in making such a computation fault-tolerant. The agent computation of interest can be viewed as a pipeline, depicted in the shaded box of Fig. 2. Nodes represent hosts and edges represent movement of an agent from one host to another. Each node corresponds to a *stage* of the pipeline. ***S*** is the *source* of the pipeline; ***A*** is the *actuator*. The computation is not fault-tolerant. The correctness of a stage depends on the correctness of its predecessor, so a single malicious failure can propagate to the actuator. Even if there are no faulty hosts, some other malicious host could disrupt the computation by sending an agent to the actuator first.

One step needed to make fault-tolerant is replication of each stage. We assume that execution of each stage is deterministic, but the components of each stage are not known *a priori* and they depend on results computed at previous stages. A node m in stage k takes as its input the majority of the inputs it receives from the nodes comprising stage $k-1$. And then, m sends its output to all of the nodes that it determines consisting of $k+1$. Fig. 2 illustrates such a fault-tolerant

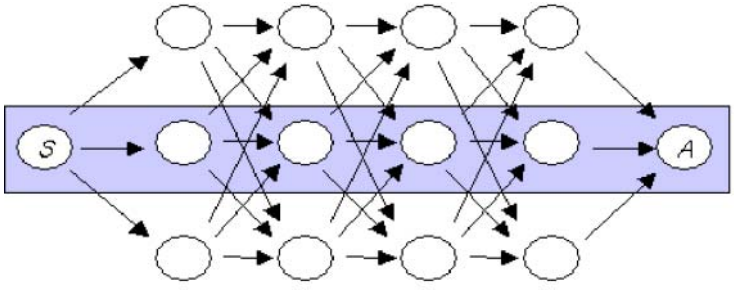


Fig. 2. Replicated agent computation with voting

execution. The replicated agent computation with voting tolerates more failures than an architecture where the only voting occurred just before the actuator. The voting at each stage makes it possible for the computation to recover by limiting the impact of a faulty host in one stage on hosts in subsequent stages.

2.3 The MARE Architecture

We describe our fault-tolerant agent architecture for reliable Internet applications, called the *Mobile Agent Replication Extension (MARE)*. The architecture of MARE system (Fig. 3) is similar to several other agent systems including Agent-Tcl [1], DaAgent [7], DarX [3], and FATOMAS [8]. The focus in the design of the MARE system has been on modularity and reusability to facilitate experimentation. A modular composition of mobile agents is possible in MARE and so it is convenient to reuse existing function units contained in specific modules in developing new applications.

The MARE system is a replication extension system with voting that makes mobile agents fault-tolerant and reliable. The Replication Group (RG) consists of multiple Agent Replication Tasks (ARTs). MARE provides group membership management to add or remove replicas. The number of replicas and the internal details of a specific task are hidden from the other tasks. Each RG has exactly one master communicating with the other ART tasks. The master acts as a fixed sequencer, providing totally ordered multicast within its RG.

Agents are allowed to inherit the functionalities (variables and methods) of other ART objects, enabling the underlying system to handle the agent computation and communication. Therefore, it is possible for MARE to act as a middleware for agents. In Fig.3, each ART is wrapped in an Application Task Shell (ATS) that acts as a Replication Group Manager (RGM), and is responsible for delivering messages to all the members of the RG. RGM is associated each agent. It keeps track of all the replicas in the group, and of the current replication method in use. RGM can change the replication policy and tune its parameters,

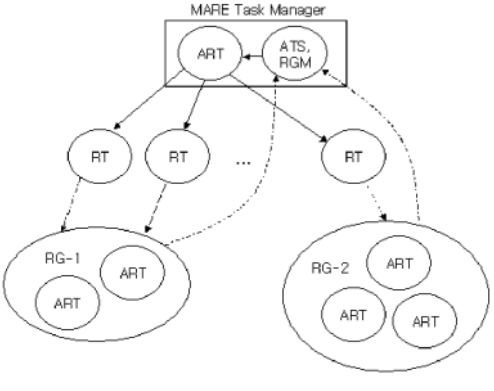


Fig. 3. The MARE system architecture

such as the number of replicas or the periods between backups in case of passive replication. ATS intercepts input messages and enables caching. All messages are processed in the same order within a RG. When an agent is replicated, its RG is suspended and the corresponding ART is copied to a new ATS on the requested host system. A task can communicate with a Remote Task (RT) by using a local proxy with the RT interface. Each RT references a distinct remote entity considered as the master of its RG.

MARE uses both passive and active replication schemes. It is possible to switch to any user-defined replication method. MARE is implemented in Java with Remote Method Invocation (RMI) as a communication layer, and it provides a global name service. Each application task corresponds to a generic name that is independent of the current location of the RG elements.

MARE also uses a fault tolerance mechanism to detect attacks by malicious hosts. It is assumed for every stage, i.e., an execution session on one host, a set of independent replicated hosts, i.e., hosts that offer the same set of resources, but do not share the same interest in attacking a host, because they are operated by different organizations. Every execution step is processed in parallel by all replicated hosts. After execution, the hosts vote about the result of the step. At all hosts of the next step, the votes, the resulting agent states, are collected. The execution with the most votes wins, and the next step is executed.

3 Implementation and Performance Study

In this section, we describe our MARE implementation and test runs. The implementation uses the Condor checkpointing packages [5] to store the state of a process in a file and has been done in C and Java. The MARE has been implemented on a network of Sun Ultra SparcStation running Solaris connected by a Fast Ethernet. As a part of the experimental study, the effects of varying the degree of replication, the active and passive replication methods and the voting frequencies are examined.

To further explore these performance issues, we run more experiments. The system we tested consists of 4 Sun SparcStation workstations connected by a 10Mbit Ethernet. An agent moving from node to node was simulated by sending a message between these node. In the experiment of Subsec. 3.1, we look at the behavior for 1, 4, and 8 replicas.

3.1 Voting Performance Effects

This experiment examined the cost of voting in the case that host speeds are uniform. We are interested in how synchronization delay can be amortized by voting less frequently. In this experiment, agents visited a sequence of N hosts before voting, rather than voting at the end of each stage. We found remarkable improvements as N advanced from 1 to 8. For N greater than 8, the further improvements were not significant.

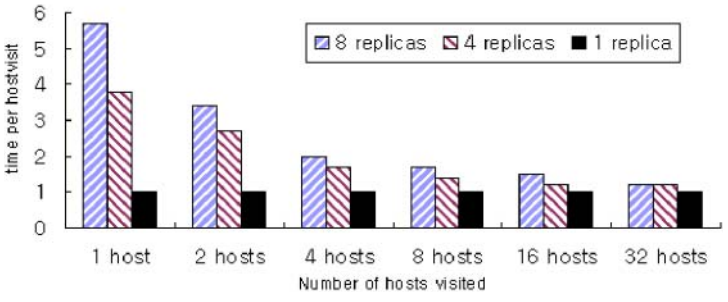


Fig. 4. Voting performance with various voting frequencies; The graph of the average time per host visit when N ranges from 1 to 32. The data depicted reports averages from runs of 320 rounds. The time spent per host had a variance 0.1

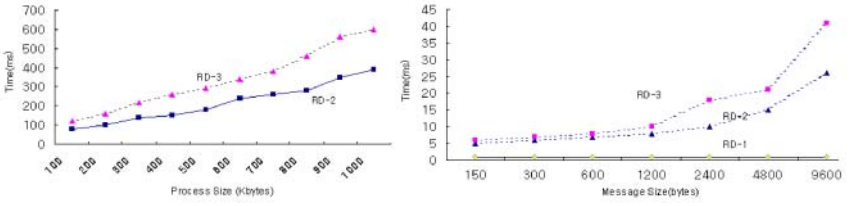
It is interesting to note synchronization delay versus voting tradeoff. When voting is infrequent, replica completion time drift apart, so the synchronization delay increases. A voter needs to wait for a correct majority, so a vote-delimited stage will complete as soon as the median correct replica votes. Therefore, the completion time for a replicated computation that votes infrequently should approximate the completion time when there is a single replica. The results of Fig. 4 show this behavior.

Voting can lead to a replicated computation being faster than the corresponding non-replicated one. Suppose there is a small probability that any given host will be slow. Over a long non-replicated execution, an agent is bound to encounter a slow host. Accordingly, the computation will be slowed. However, with replication and periodic voting, it is likely that a majority of the agents reaching a voter will have encountered no slow hosts. Because the voter waits for this majority, the replicated system's execution time will be independent of the speed of the slow hosts.

3.2 Comparison of Passive and Active Methods with Different RDs

We quantify our MARE prototype's performance, communication and update cost as a function of the replication degree. In this experiment, we measured the time needed to synchronously send a message to a replication group using the active replication method. The communication cost here is the time needed to send a message to a processor and to receive a reply message from the processor.

Fig. 5(b) shows the communication cost as a function of the replication degree. There are three different Replication Degree (RD) configurations. In the first configuration, denoted by RD-1, the process on the local host system is not replicated. In the second configuration, RD-2, the process is replicated on the remote host system. In the third RD-3 configuration, there are three replicas: one master of the local host and the two replicas residing in two remote hosts. We also measured the time to update remote replicas using the passive replication. The time to update a local replica was not significant and so it was ignored.



(a) (b)

Fig. 5. (a)The time needed to update a replication group in RD-2 and RD-3 configurations. The update cost for remote replicas using the passive method (b)Communication cost as a function of RD, The time needed to send a message synchronously to a replication group using the active method

Fig. 5(a) shows the update cost as a function of the replication degree, i.e., the cost to change a replication group using different RD-2 and RD-3 replication degrees. Our findings from Fig. 5 indicate that active methods tend toward fast recovery but high communication overhead, while passive methods present slow recovery but low overhead.

4 Conclusion

In this paper, we have presented MARE, a replication system with voting that makes mobile agents fault-tolerant and also detects attacks by malicious hosts. The proposed system represents a fault-tolerant mobile agent model for reliable Internet applications. The performance of the MARE prototype has been evaluated. As a part of the experimental study, the effects of varying the degree of replication, the active and passive replication methods, and the voting frequencies were examined. Our findings show that replication and voting improve performance by ensuring that slow hosts do not affect the progress of computation. Making voting less frequent makes voting delays insignificant.

There are several works needed to further explore. First, we can compare the MARE system the other popular agent systems. We measure the time needed to move an agent from one node to another, to determine the efficiency of MARE and the overhead of agent mobility. The migration time of MARE compares favorably with Agent-Tcl. Second, we are developing test applications to validate the MARE work. Those applications we considered include a distributed agenda and an e-commerce system to test survivability of MARE. Further enhancements of our system are the possibility of a specific duplication of agent, which is supported by distributed transaction processing mechanism. Finally, we need to integrate security policies of different agent languages, such as Java, Tcl, and Agent-Tcl.

Acknowledgements

The first author wishes to thank the anonymous reviewers for their helpful comments. This work was supported by the Catholic University of Korea Research Fund 2004.

References

1. R. S. Gray: Agent Tcl: A Transportable Agent System. In *Proc. CIKM Workshop Intelligent Information Agents*, Baltimore, MD, Dec. 1995. 1040
2. D. Kotz and R. S. Gray: Mobile Agents and the Future of the Internet. *ACM Operating Systems Review*, 33(3): (Aug. 1999) 7–13. 1036
3. G. Lacote et al.: Towards and Fault-Tolerant Agents. In *Proc. ECOOP'2000 Workshop Distributed Objects Programming Paradigms*, Cannes, Italy. 1040
4. D. B. Lange and M. Oshima: Seven Good Reasons for Mobile Agents. *Communications of the ACM*, 42(3): (Mar. 1999) 88–89. 1036
5. M Lizkow et al.: Checkpoint and Migration of Unix Process in the Condor Distributed System. Tech. Report 1346, Computer Science Dept. Univ. Wisconsin, Madison, 1997. 1041
6. D. S. Milojicic, S. Guday, and R. Wheeler: Old Wine in New Bottles Applying OS Process Migration Technology to Mobile Agents. In *Proc. 3rd ECOOP Workshop Mobile Object Systems*, Jul. 1998. 1036
7. S. Mishra et al.: DaAgent: A Dependable Mobile Agent System. In *Proc. of the 29th IEEE Int. Symp. Fault-Tolerant Computing*, Madison, WI, June 1999. 1036, 1037, 1040
8. S. Pleisch and A. Schiper: FATOMAS: A Fault-Tolerant Mobile Agent System Based on the Agent-Dependent Approach. In *Proc. IEEE Int. Conf. Dependable Systems and Networks*, Jul. 2001, Goteborg, Sweden, 215–224. 1040
9. F. Schneider: Toward Fault-Tolerant and Secure Agentry. In *Proc. 11th Int. Workshop Distributed Algorithms*, Sep. 1997, Saarbrucken, Germany, 1–14. 1039
10. M. Strasser and K. Rothermel: Reliability Concepts for Mobile Agent. *Int. J. Cooperative Information Systems*, 7(4): (Dec. 1998) 355–382. 1037
11. Sun Microsystems, Inc. Java Technology: MIDP (Mobile Information Device Profile) for Palm OS, NIS (Network Information Service). *http://java.sun.com* and *http://www.sun.com* 1035, 1038

Author Index