A Multicast Routing Protocol with Multiple QoS Constraints

Li Layuan and Li Chunlin*
Department of Computer Science. Wuhan University of Technology,
Wuhan 430063.P. R. China
E-mail: jwtu @ public. wh. hb. cn

Abstract: The next generation Internet and high-performance networks are expected to support multicast routing with QoS constraints. To facilitate this, QoS multicast routing protocols are pivotal in enabling new member to join a multicast group. Multicast routing is establishing a tree which is rooted from the source node and contains all the multicast destinations. A multicast routing tree with multiple QoS constraints may be the tree in which the delay, delay jitter, packet loss and bandwidth should satisfy the pre-specified bounds. This paper discusses the multicast routing problem with multiple QoS constraints, which may deal with the delay, delay jitter, bandwidth and packet loss metrics, and describes a network model for researching the routing problem. It presents a multicast routing protocol with multiple QoS constraints (MRPMQ). The MRPMQ attempts to significantly reduce the overhead of constructing a multicast tree with multiple QoS constraints. In MPRMQ, a multicast group member can join or leave a multicast session dynamically, which should not disrupt the multicast tree. It also attempts to minimize overall cost of the tree, and satisfy the multiple QoS constraints and least cost's (or lower cost) requirements. In this paper, the proof of correctness and complexity analysis of the MRPMQ are also given. Simulation results show that MRPMQ is an available approach to multicast routing decision with multiple QoS constraints.

Keywords: Multicast routing ; protocol ; multiple QoS constraints ; QoS routing ; NP-complete.

*The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

L. Chapin (ed.), *Communication Systems*

1 INTRODUCTION

With the rapid development of Internet, mobile networks and high-performance networking technology, multicast routing has continued to be a very important research issue in the areas of networks and distributed systems. It attracts the interests of many people. Multicast services have been used by various continuous media applications. For example, the multicast backbone (Mbone) of the Internet has been used to transport real time audio/video for news, entertainment, video conferencing, and distance learning. The provision of Quality-of–Service (QoS) guarantees is of utmost importance for the development of the multicast services.

The traditional multicast routing protocols, e.g., CBT and PIM [1-4], were designed for best-effort data traffic. They construct multicast trees primarily based on connectivity. Such trees may be unsatisfactory when QoS is considered due to the lack of resources. Several QoS multicast routing algorithms have been proposed recently. Some algorithms [5-10] provide heuristic solutions to the NP-complete constrained Steiner tree problem, which is to find the delay-constrained least-cost multicast trees. These algorithms however are mot practical in the Internet environment because they have excessive computation overhead, require knowledge about the global network state, and do not handle dynamic group membership. Jia's distributed algorithm [5] does not compute any path or assume the unicast routing table can provide it . However, this algorithm requires excessive message processing overhead. The spanning join protocol by Carlberg and Crowcroft [1] handles dynamic membership and does not require any global network state. However, it has excessive communication and message processing overhead because it relies on flooding to find a feasible tree branch to connect a new member. QoSMIC [6], proposed by Faloutsos et al., alleviates but does not eliminate the flooding behavior. In addition, an extra control element, called Manager router, is introduced to handle the join requests of new members.

Multicast routing and its QoS-driven extension are indispensable components in a QoS-centric network architecture[15-17]. Its main objective is to construct a multicast tree that optimizes a certain objective function (e.g., making effective use of network resources) with respect to performance-related constraints (e.g., end-to-end delay bound, inter-receiver delay jitter bound, minimum bandwidth available, and maximum packet loss probability).

In this paper, we study the delay, delay jitter, bandwidth, and packet loss-constrained low cost QoS multicast routing problem which is known to be NP-complete, describe a network model for researching the routing problem, and present a multicast routing protocol with multiple QoS constraints (MRPMQ). In MRPMQ, a multicast group member can join or leave a multicast session dynamically. The MRPMQ can significantly reduce the overhead of constructing a multicast tree. The join of new member can have minimum overhead to other on-tree nodes or non-tree nodes, and only requires minimum adjustment to original tree and minimum resource reservation process. The protocol may search multiple feasible tree branches, and can select the best branch connecting the new member to the tree. It can also minimized the overall cost of the tree, and satisfy multiple QoS constraints and least cost's (or lower cost) requirements. The protocol can operate on top of some available unicast routing protocol.

The rest of the paper is organized as follows. Section 2 discusses the QoS multicast routing problem and descryibes a network model. Section 3 presents the MRPMQ. Section 4 gives the correctness proof and complexity analysis. Some simulation results are provided in Section 5. The paper concludes with Section 6.

2 NETWORK MODEL

As far as multicast routing is concerned, a network is usually represented as a weighted digraph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of communication links connecting the nodes. $|V|$ and $|E|$ denote the number of nodes and links in the network, respectively, Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes[13]. Associated with each link are parameters that describe the current status of the link.

Let $s \in V$ be source node of a multicast tree, and $M \subseteq \{V - \{s\}\}$ be a set of end nodes of the multicast tree. Let R be the positive weight and R^+ be the nonnegative weight. For any Link $e \in E$, we can define the some QoS metrics: delay function *delay* (e): $E \rightarrow R$, cost function *cost* (e): $E \rightarrow R$, bandwidth function *bandwidth* (e); $E \rightarrow R$, and delay jitter function *delay-jitter* (e): $E \rightarrow R^+$. Similarly, for any node $n \in V$, one can also define some metrics: delay function *delay* (n): $V \rightarrow R$, cost function *cost* (n): $V \rightarrow R$, delay jitter function *delay-jitter* (n): $V \rightarrow R^+$ and packet loss function *packet-loss* (n): $V \rightarrow R^+$. We also use $T(s,M)$ to denote a multicast tree, which has the following relations:

1) $delay\ (p\ (s,t)) =. \quad \sum_{e \in P(s,t)} delay\,(e) + \sum_{e \in P(s,t)} delay\,(n)$

2) $cost\ (T(s,M))=. \quad \sum_{e \in T(s,M)} cost\ (e) + \sum_{e \in T(s,M)} cost\ (n)$

3) $bandwidth\ (p(s,t)) = \min\{bandwidth\ (e),\ e \in P(s,t)\}$.

4) $delay\text{-}jitter\ (p\ (s,t)) = \sum_{n \in P(s,t)} delay - jitter\,(e)$
$+.\ \sum_{n \in P(s,t)} delay - jitter\,(n)$

5) $packet\text{-}loss\ (p\ (s,t)) = 1 - \prod_{n \in P(s,t)} (1 - packet\ \text{-}\ loss(n))$

where $p\ (s,t)$ denotes the path from source s to end node t of $T\ (s,\ M)$.

Definition 1. QoS-based multicast routing problem deals mainly with some elements: Network $G=(V,E)$, multicast source $s \in V$, the set of end nodes $M \subseteq \{V-\{s\}\}$, $delay(\cdot) \in R$, $delay\text{-}jitter(\cdot) \in R^{+}$, $cost\ (\cdot) \in R$, $bandwidth\ (\cdot) \in R$, and $packet\text{-}loss\ (\cdot) \in R^{+}$. This routing problem is to find the $T\ (s,\ M)$ which satisfies some QoS constraints:

1. Delay constraint: $delay\ (p\ (s,t)) \leq D_t$
2. Bandwidth constraint: $bandwidth\ (p\ (s,t)) \geq B$
3. Delay jitter constraint: $delay\text{-}jitter\ (p\ (s,t)) \leq J$
4. Packet loss constraint: $packet\text{-}loss\ (p\ (s,t)) \leq L$

Meanwhile, the *cost* $(T\ (s,\ M)$ should be minimum. Where D is delay constraint, B is bandwidth constraint, J is delay jitter constraint and L is packet loss constraint. In the above QoS constraints, the bandwidth is concave metric, the delay and delay jitter are additive metrics, and the packet loss is multiplicative metric. In these metrics, the multiplicative metric can be converted to the additive metric. For simplicity, we assume that all nodes have enough resource, i.e., they can satisfy the above QoS constraints. Therefore, we only consider the links' or edges' QoS constraints, because the links and the nodes have equifinality to the routing issue in question. Fig. 1 shows the network model. The characteristics of edge can be described by a fourtuple (D,J,B,C), where D,J,B and C denote delay, delay jitter, bandwidth and cost, respectively.

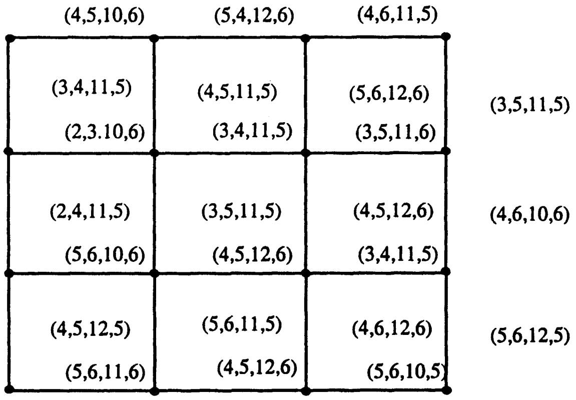

Fig. 1 An example of the nětwork model

3 MRPMQ

3.1 Overview

MRPMQ can operate on top of some available QoS unicast routing protocol that pre-computes QoS paths. QOSPF could be used for the intra-domain routing operation. For inter-domain routing operation, it may require an available inter-domain unicast routing protocol. Though MRPMQ can works on top of OSPF like protocols, it is different from the MOSPF. In MRPMQ, a router (or a node) keeps a routing entry *R* (*G*, *s*, *in*, *out*), where *R.G* is the group address, *R*.s is the source address, *R.in* is the incoming network interface, and *R. out* is the set of outgoing network interfaces. Main control messages of MRPMQ can be described by the following Definition 2.

Definition 2. The set of control message can be defined as follows. JOINreq-a join request message sent towards source *s* by a new member that wishes to join a multicast group (*G*), JOINack-an accept acknowledgment sent downstream towards the new member by some node that accepts the join request, JOINnak-a rejection notification sent downstream towards the new member by some node that rejects the join request, and JOINpend-if the delay of a searching path does not satisfy the constraint *D*, and the next node (t_{j*}) is not the immediate upstream node of t_j, then t_j should add a pending information to the routing entry and marks t_{j*} as a possible upstream node.

A multicast group member may join or leave a multicast session dynamically. It is thus important to ensure that member join/leave will not disrupt the on-going multicast, and the multicast tree after member join/leave still satisfies the QoS requirements of all the on-tree receivers and remains near optimal. If a multicast tree is re-constructed from the source each time a member joins or leaves, on-tree nodes may not switch to the new tree simultaneously and the packets that are originally routed on the old multicast tree may be lost and have to be retransmitted. A seamless transition may thus not be achieved. Hence, in the case of member join/leave, a method that incrementally changes the *T*(*s*,*M*) can be used. When a receiver leaves a multicast session, if the receiver is a leaf node of a multicast tree, it sends a leave message upstream. The leave message travels upstream along the on-tree branch until it reaches a fork node (i.e.,a node with more than one downstream on-tree interface and/or with receivers on its directly attached subnet). Upon receipt of a leave message, an intermediate node releases the network resources. The rest of the multicast tree remains unchanged. On the other hand, if the leaving receiver is not a leaf node, it is not disconnected from the multicast tree. From this point onwards, this node simply relays incoming messages to the outgoing downstream interfaces. The ongoing session will not be affected.

When a receiver intends to join a multicast session, it sends a join request JOINreq to the source of the session with the information of its delay, delay jitter, bandwidth and cost. When a node receives a JOINreq from other node with IGMP, it will computes a path from s to itself, if the node is not the on-tree node. If a join request is accepted, a JOINack is sent downstream towards the new member. Otherwise if a join request is rejected, a JOINnak is also sent to the receiver. In MRPMQ, the source node forwards the QoS probing message to all receivers every ten seconds, so that QoS constraints information can be carried in the message, which can update current QoS requirements on time. In addition, MRPMQ also assumes each receiver knows all Link state information and the address of the multicast group in its intra-domain.

3.2 Detailed description

This section will describe some details of MRPMQ, the emphasis is laid on the join process, multicast tree construction and multiple constraints.

In MRPMQ, a searching tree is formed incrementally. When a receiver, such as the leaf node t_i (see Fig.2), wants to join the multicast group, it send a JOINreq to intermediate node (router) t_l. When t_l receives the JOINreq message from t_i, and it is already the on-tree node, it will make an eligibility test. It check whether or not the QoS requirement of the new member as well as the existing QoS guarantees to the other on-tree members, can be fulfilled. Suppose the path delay and delay jitter from t_1 to t_0 are $d(1,0)$ and $d_j(1,0)$, respectively. Similarly, the path delay and delay jitter from t_i to t_1 should also be $d(i,1)$ and $d_j(i,1)$, respectively. Let $bw(t_1,t_0)$ be the bandwidth of lind from t_1 to t_0, and $bw(t_i,t_l)$ be the bandwidth of link from t_i to t_1. Recall that the delay, delay jitter and bandwidth constraints are D, J and B, respectively. The t_1 will check if

$$(d(i,1)+d(1,0)\leq D)\wedge(dj(i,1)+dj(1,0)\leq J)\wedge(bw(t_i,t_1)\wedge bw(t_1, t_0)\geq B)$$

it will add the corresponding *out* to the routing entry, and transfers a JOINack message to t_i. Otherwise, t_1 will transfer JOINnak message to t_i. In Fig.2, other nodes are the on-tree nodes of the multicast group.

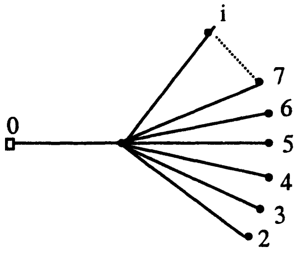

Fig. 2 t_i requires to join multicast group

If $(d(i,1)+d(1,0)>D)\wedge(dj(i,1)+dj(1,0)\le J)$ then there are two possible cases.

Casel If next hop in the JOINreq is just the immediate upstream node $t_{1'}$ (see Fig.3), t_1 can transfer the JOINreq from the new member t_i to $t_{1'}$, and add *out* to the corresponding routing entry and mark it as pending. In the case, $t_{1'}$ may accept the join request or transfer the join request upstream.

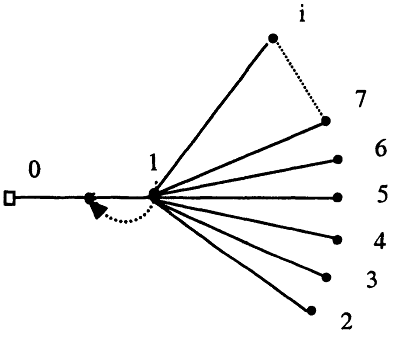

Fig. 3 $t_{1'}$ is the immediate upstream node

Case 2 If next hop is not the immediate upstream node, such as t_{i*} shown in Fig.4, and if

$$(d(0.1')+d(1',1)\ge D-d(1,i))\wedge(dj(0,1')+dj(1',1)\ge J-dj(1,i))$$

then t_1 will transfer the JOINreq to t_{1*}, and add a pending routing entry and mark t_{1*} as upstream rode. If t_{1*} transfer JOINack message to t_1, t_1 will forward a pruning message to $t_{1'}$. This case will produce a switch from the original path $(t_0\to t_{1'}\to t_1)$ to the new path $(t_0\to t_{1*}\to t_1)$.

Under this situation, the t_{1*} should send a QoS probing message with the new QoS requirements to all downstream nodes.

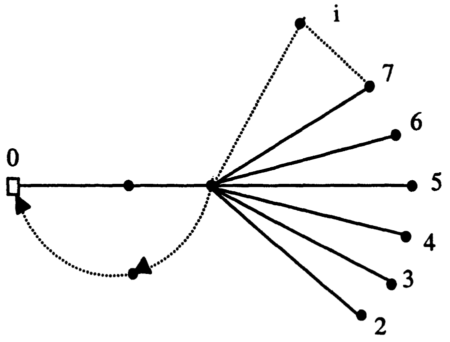

Fig. 4 Switch to new path ($t_0 \rightarrow t_{1*} \rightarrow t_1$)

If $d(0,1')+d(1',1) \leq D$ or $dj(0,1')+dj(1',1) \leq J$, t_1 will compute a new path. The new path should satisfy delay constraint $\min[d(0,1), D-d(1,i)]$ and delay jitter constraint $\min[dj(0,1), J-dj(1,i)]$. If so, t_1 will receive the JOINack message, otherwise it will receive the JOINnak message. The situation can be shown in Fig.5.

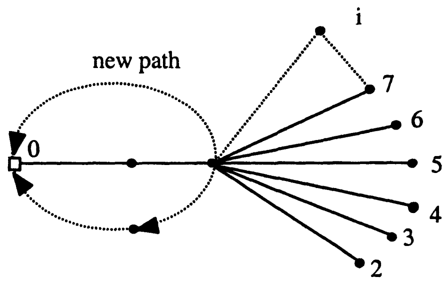

Fig. 5 Compute the new path

The Join procedure of MRPMQ can be formally described as follows.

1. if a new member (*ti*) wishes to join a $T\ (s,M)$ the new member sends JOINreq to some neighbor t_j
2. if $(d(s,*)+d(j,i) \leq D) \square (d_j(s,*)+dj(j,i) \leq J) \square (bw(t_u,t_v) \geq B)$; {where $d(s,*)$ and $d_j(s,*)$ are the delay sum and the delay jitter sum from the source s to all downstream nodes of a path, respectively, but except for the last a pair of nodes. The u and v are the sequence numbers between two adjacent nodes on path from source to the new member}

```
 →$t_j$ transfers JOINack to $t_i$
fi
```

```
  if $bw\ (t_u,t_v)<B\rightarrow$
    remove $e\ (u,v)$ from $G$
  fi
3. if $(d\ (s,*)+d\ (j,i)>D)\square(dj\ (s,*)+dj\ (j,i)>J)\rightarrow$
    if the next hop is the immediate upstream node ($t_{j'}$ of $t_j$)
   $t_j$ transfers JOINreq to $t_{j'}\rightarrow$
   $t_j$ adds JOINpend for $t_i$ to the forwarding entry
   $t_{j'}$ transfers JOINack(or JOINnak ) to $t_j$
   fi
    if the next hop is not the immediate upstream node.
    if $(d\ (s,*)\geq\ D-d\ (j,i))\wedge(dj\ (s,*)\geq\ J-dj\ (j,i))\rightarrow$
   $t_j$ transfers JOIN reg to $t_{j*}$
   $t_j$ adds the routing entry
    marks $t_{j*}$ as upstream node
    $t_{j*}$ transfers JOINack(or JOINnak)to $t_j$
    if $t_j$ receives JOINack$\rightarrow$
      $t_j$ forwards a pruning msg to $t_{j'}$
        fi
   fi
   fi
4. if $(d(s,*)+d(j,i)\leq D)\square(dj(s,*)+dj(j,i)\leq J)\rightarrow$
    $t_j$ computes a new path
    if $(d\ (p\ (s,i))=\min[d\ (s,*),\ D-d\ (j,i)])\wedge$
     $(dj(p\ (s,j))=\min[dj\ (s,*),\ J-dj\ (j,i)])\rightarrow$
     $t_j$ receives JOINack
    fi
     $t_j$ receives JOINnak
    fi
```

We can use the following example to show how the MRPMQ works and how the multicast tree is constructed in a distributed fashion. Fig, 6 is a network graph. In this example, node t_0 is the multicast source. The t_4, t_9, t_{14}, t_{19} and t_{24} are the joining nodes. Recall that characteristics of network's edge can be described by a fourtuple (D,J,B,C). In this example shown in Fig 6, suppose delay constraint D =20, delay jitter J =30 and bandwidth constraint B =40, The t_4 wishes to join the group, it computes the paths according to the multiple QoS constraints: the path $(t_0\rightarrow t_1\rightarrow t_2\rightarrow t_3\rightarrow t_4)$. Path $(t_0\rightarrow t_1\rightarrow t_2\rightarrow t_3\rightarrow t_8\rightarrow t_4)$ and path $(t_0\rightarrow t_1\rightarrow t_2\rightarrow t_7\rightarrow t_8\rightarrow t_4)$ do not satisfy the delay constraint. The path $(t_0\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_4)$, path $(t_0\rightarrow t_5\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_4)$ and path $(t_0\rightarrow t_1\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_4)$ satisfy the delay constraint, the delay jitter constraint and the bandwidth constraint. Furthermore, the path $(t_0\rightarrow t_1\rightarrow t_6\rightarrow t_7\rightarrow t_8\rightarrow t_4)$ has minimum cost among these paths. Therefore, the join path should be the path $(t_0 \to t_1 \to t_6 \to t_7 \to t_8 \to t_4)$. The bold lines of Fig.7(a) show the tree when t_4 has joined the group. When t_9 joins the group, it computes a path $(t_0 \to t_1 \to t_6 \to t_7 \to t_8 \to t_9)$which should satisfy delay, delay jitter and bandwidth constraints, and also have minimum cost. The JOINreq is accepted at t_8. The bold lines of Fig.7(b) show the tree when t_9 has joined the group. When t_{14} join the group, it computes the paths with multiple QoS constraints. The path $(t_0 \to t_1 \to t_6 \to t_7 \to t_{12} \to t_{13} \to t_{14})$ does not satisfy the delay jitter constraint. The path $(t_0 \to t_5 \to t_6 \to t_7 \to t_8 \to t_{13} \to t_{14})$ does not satisfy delay and delay jitter constraints. The path $(t_0 \to t_6 \to t_7 \to t_{12} \to t_{13} \to t_{14})$ and path $(t_0 \to t_5 \to t_6 \to t_7 \to t_{12} \to t_{13} \to t_{14})$ satisfy delay, delay jitter and bandwidth constraints. The later has the lower cost. Therefore, the join path should be the path $(t_0 \to t_5 \to t_6 \to t_7 \to t_{12} \to t_{13} \to t_{14})$. Meanwhile, t_6 should prune off from original parent t_1, the resulted tree is shown in Fig.7(c) (see the bold lines of Fig.7(c)). The tree after t_{19} joins the group is also shown in Fig.7(d). When t_{24}joins the group, it computes the join paths. If t_{18} receives JOINreg from t_{24}, it find out that the existing path $(t_0 \to t_5 \to t_6 \to t_7 \to t_{12} \to t_{13} \to t_{18})$ does not satisfy the delay constraint for new member t_{24}, whilethe new path $(t_0 \to t_5 \to t_6 \to t_7 \to t_{12} \to t_{17} \to t_{18})$ does not satisfy the delay jitter constraint for t_{24}. The t_{18} computes a new feasible path with delay constraint, which is given by

$$\begin{aligned} d\,(p\,(s,j)) &= \min[d\,(s,*),\, D{-}d\,(j,i)] \\ &= \min[(d\,(0,5)+d\,(5,6)+d\,(6,7)+d\,(7,12)+d \\ &\quad (12,13)+d\,(13,18)),\, D{-}d\,(18,24)] \\ &= \min[19,18] \\ &=18 \end{aligned}$$

and delay jitter constraint, which can be given by

$$\begin{aligned} dj\,(p\,(s,j)) &= \min[dj\,(s,*),\, J{-}dj\,(j,i)] \\ &= \min[(dj\,(0,5)+dj\,(5,6)+dj\,(6,7)+dj\,(7,12)+ \\ &\quad dj\,(12,13)+dj\,(13,18)),\, J{-}dj\,(18,24)] \\ &= \min[28,28] \\ &=28 \end{aligned}$$

Thus, this new feasible path should be path$(t_0 \to t_6 \to t_7 \to t_{12} \to t_{13} \to t_{18})$. The t_6 should prune off from old parent t_5, and the final tree can be shown in Fig.7(e)(see the bold lines of Fig.7(e)).

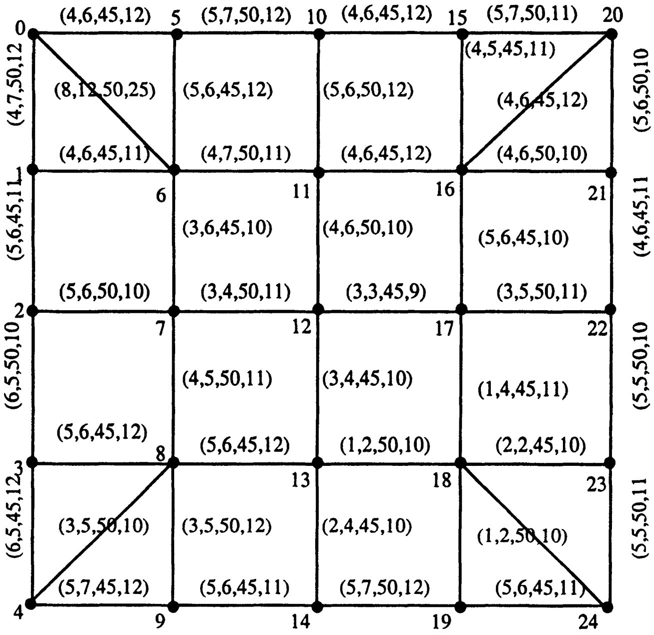

Fig. 6 An example network graph

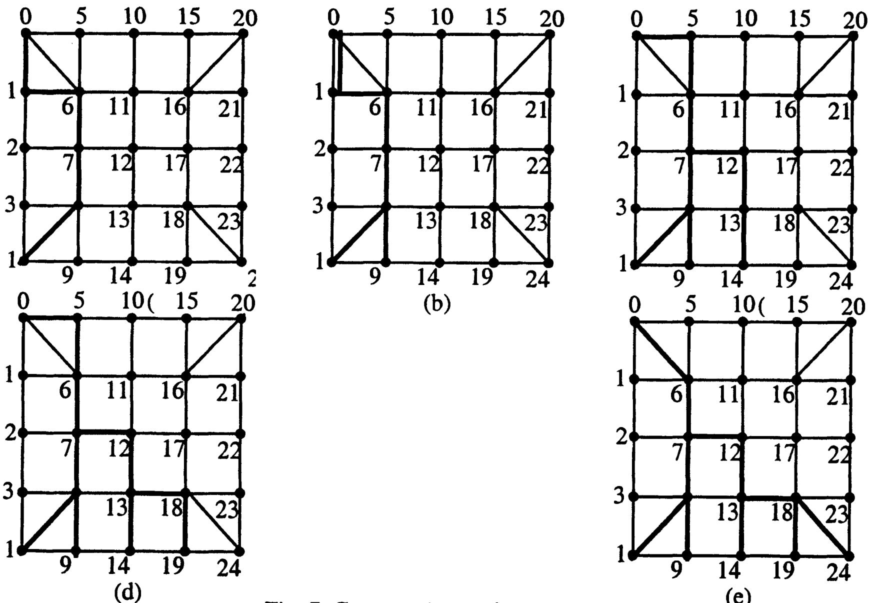

Fig. 7 Constructing multicast tree

The loop-free routing for the above protocol can be achieved by maintaining a searching tree at any time.

4 THE CORRECTNESS AND COMPLEXITY ANALYSIS

4.1 The correctness proof

Theorem 1. If a path from a new member to *T*(*s*,*M*) has sufficient resources to satisfy the QoS constraints and has minimum cost, it searches only one path.

Proof. Note that a necessary condition for multiple paths to be searched is a single path does not satisfy the QoS constraints, such as $(d(p(s,j)) \neq \min[d(s,*),\ D\text{-}d(j,i)]) \wedge (dj(p(s,j)) \neq \min[dj(s,*),\ J\text{-}dj(j,i)])$. However, if sufficient resources are available on every link and node of the path, no node forwarding JOINreg will ever enter the multiple paths search state. Thus, the above theorem holds.

Lemma 1. Whenever during the routing process, all paths being searched form a *T*(*s*,*M*) structure .

Proof. The paths being searched will be marked by the routing entries at the nodes. In MRPMQ, any routing entry has a single out interface and one or multiple in interfaces. Hence, the nodes will form a searching tree structure. This tree is just a *T*(*s*,*M*).

Theorem 2. An available and feasible path found by MRPMQ is loop-free.

Proof. This Theorem follows directly from the above Lemma 1.

Lemma 2. If MRPMQ terminates without an available and feasible path, all nodes out of *T*(*s*,*M*) are ether initial state or in failure state.

Proof. MRPMQ terminates without success only when the new member's JOINreg is rejected, i.e., it changes into the failure state. Since the new member is leaf node of the searching tree, when it changes into the failure state, all nodes in the searching tree must be in the failure state. The nodes outside the searching tree remain in the initial state.

Theorem 3. MRPMQ can find an available and feasible path if one exists.

Proof. This theorem can be proved by contradiction. Suppose MRPMQ fails while an available and feasible path does exist. Let *e*(*i*,*j*) be the first link in the path that the protocol did not explore. Since *e*(*i*,*j*) is the first unexplored link of the path, t_i must have received a request message from the previous link or t_i is the new member issuing the request message. In either case, t_i is not in the initial state. Therefore, t_i is in the failure state by Lemma 2. which requires t_i to explore all outgoing links including $e(i,j)$. It contradicts the assumption that $e(i,j)$ is not explored.

4.2 Complexity Analysis

The complexity of the QoS-based multicast routing protocol can be analyzed in terms of the computation complexity and the number of messages needed to construct a multicast tree. The former mainly concerns the computation overhead needed to find paths and construct the multicast tree. The later mainly involves the overhead of message exchange needed to build the multicast tree. In MRPMQ, route computation can generally be made by the end node. If the join path is computed on-demand, the complexity depends on the unicast protocol. If QoS metrics are delay and bandwidth, there exist QoS routing heuristics which are O ($|V|$ x $|E|$), where $|V|$ is the number of nodes and $|E|$ is the number of edges in a network. For most networks, $|E|=O|V|$, hence the complexity is $O(|V|^2)$. For a multicast group with $|M|$ members, the computation overhead is $O(|V|^2|M|)$. The study shows that computation complexities of CSPT and BSMA [8] are O ($|E|$ log $|V|$) and O($|V|^3$ log$|V|$), respectively. The computation complexity of MRPMQ is $O(|V|^2$ $|M|)$. In overhead of message exchange, MRPMQ requires two messages, i.e., JOINreq and JOINack or JOINnak. This means that a multicast group with $|M|$ members deals with $2|M|$ message. A JOINreg will be processed by all K hops along the way up to the node where it is accepted or rejected, hence the overhead of message processing for joining $|M|$ members is $K.2|M|$. Jia's algorithm [5] also requires two control messages, however, each message actually includes full information of a multicast tree, and thus its message processing is more complex than in MRPMQ. QoSMIC[6] requites a considerable number of search of BID-ORDER messages to join a new member, all messages need processing at both the new member as well as other nodes. The study shows that the average message processing overheads to construct the multicast tree of MRPMQ, Jia's algorithm, and QoSMIC (centralized or distributed) are $K.2|M|$, $K.2|M|$, $|M|$ $(w \cdot (w\text{-}1)^{(y\text{-}1)}+c\text{-}k) \cdot x$ (centralized QoSMIC) and $|\mathrm{M}|$ $(w \cdot (w\text{-}1)^{(y\text{-}1)}+|T| \cdot x$ (distributed QoSMIC) , respectively, where the x factor is added to reflect the fact that messages have to be processed at more than one node, w is the average degree of a node, y is maximum TTL used for search, $|T|$ is tree size, c is number of candidates for BID-ORDER session and x depends on the topology and y , while $2\leq$ $x\leq$ $1+K$.

5 SIMULATIONS

The effectiveness and availability of MRPMQ are studied by simulations. In the simulations experiment, four algorithms are simulated : MRPMQ, CSPT, BSMA and KMB [20]. KMB applies Prim's minimum spanning tree algorithm to the complete distance graph. This heuristic find a tree whose cost is within twice the cost of the corresponding Steiner tree.

Network topologies used in the simulations are deliberately manipulated to simulate wide area sparse networks. A large network is likely to be loosely interconnected [11,14]. An *n*-node graph is considered to be sparse when less than 5% of the possible edges are present in the graph. The network graphs used in the simulations are constructed by the Waxman's random graph model[12].

In this random graph, the edge's probability can be

$$P_e\,(u,v)=\beta\, exp\,(-\frac{d(u,v)}{\alpha L})$$

where $d\ (u,v)$ is geometric distance from node u to node v, L is maximum distance between two nodes, parameter α can be used to control short edge and long edge of the random graph, and parameter β can be used to control the value of average degree of the random graph.

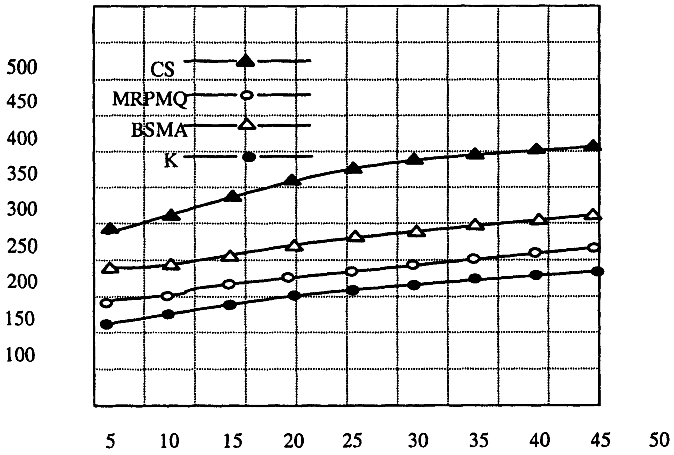

Fig. 8 Network cost vs. group size

In the simulations, we compare the quality of routing trees by their network cost[18]. The network cost is measured by the mean value of the total number of simulation runs. At each simulation point, the simulation runs 80 times. Each time the nodes in the group G are randomly picked out from the network graph. The network cost is simulated against two parameters: delay bound D and group size. In order to simulate the real situations, group size are always made less than 20% of the total nodes, because multicast applications running in a wide area network usually involve only a small number of nodes in the network, such as video conference systems, distance learning, co-operative editing systems, etc[19,21,22].

Fig. 8 shows the network cost versus group size. In this round of simulations, the network size is set to 300 and D is $d_{\max}$+3/8$d_{\max}$. From Fig. 8, we can see wthen group size grows, the network cost produced by MRPMQ, BSMA and KMB increases at a rate much lower than CSPT. The MRPMQ performs between BSMA and KMB. BSMA, KMB and the proposed MRPMQ can produce trees of comparable costs.

Fig. 9 is the network cost versus D. During this round of simulations, the network size is fixed at 300 nodes, group size is 20. We define the smallest meaningful value of D as $d_{\max}$=max({d_u|for any $u \in G$: d_u is the delay on the shortest path from s to u}). D starts from $d_{\max}$. With an even smaller D, there does not exist such a routing tree which satisfies the delay bound D. D is incremented by $d_{\max}/8$ each time. The increment of $d_{\max}/8$ is selected to maximally capture the trend of network cost against the change of D after many simulation runs. Since for each simulation run, G is different, thus $d_{\max}$ is different. The D values on the X-axis are the mean values of D in all runs.

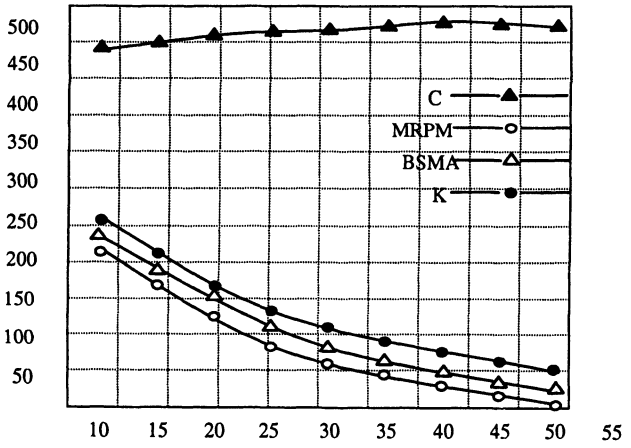

Fig. 9 Network cost vs. delay bound

From Fig.9, it can be seen that the network cost of the CSPT algorithm is on the top and almost does not change as D increases. This is because the generation of the shortest path tree does not depend on D. Of the remaining three algorithms, the proposed MRPMQ has the lowest cost. From Fig.9, we can also see that tree costs decrease for MRPMQ, BSMA and KMB algorithms as the delay bound is relaxed. This shows all three schemes indeed can reduce the cost when delay bound is relaxed. From Fig.8 and Fig.9, one can see that MRPMQ, BSMA and KMB algorithms can produce trees of comparable costs. However, compared with BSMA and KMB algorithms, the proposed MRPMQ has the advantage of being fully distributed and allowing incremental tree build-up to accommodate dynamic joining new members. Furthermore, the MRPMQ is much less costly in terms of computation cost and in terms of cooperation needed from other network nodes compared with other schemes.

6 CONCLUSION

In this paper, we discuss the multicast routing problem with multiple QoS constraints, which may deal with the delay, delay jitter, bandwidth and packet loss metrics, and describe a network model for researching the routing problem. We have presented a multicast routing protocol with multiple QoS constraints (MRPMQ). The MRPMQ can significantly reduce the overhead of establishing a multicast tree. In MRPMQ, a multicast group member can join or leave a multicast session dynamically, which should not disrupt the multicast tree. The MRPMQ also attempts to minimize overall cost of the tree. This protocol may search multiple feasible tree branches in distributed fashion, and can select the best branch connecting the new member to the tree. The join of new member can have minimum overhead to on-tree or non-tree nodes. The correctness proof and complexity analysis have been made. Some simulation results are also given. The study shows that MRPMQ is an available and feasible approach to multicast routing with multiple QoS constraints. Further work will investigate the protocol's suitability for inter-domain multicast and hierarchical network environment.

7 ACKNOWLEDGMENT

The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

8 REFERENCES

[1] K. Carberg and J. Crowcroft. "Building shared trees using a one-to-many joining mechanism". *ACM Computer Communication Review.* Jan. 1997, pp. 5-11.

[2] T. Ballardie, P. Francis and J. Crowcroft, "An architecture for scalable inter-domain multicast routing," *ACM SIGCOMM,* Sept. 1993, pp. 85-95.

[3] Li Layuan and Li ChunLin, "The QoS routing algorithm for ATM networks," *Computer Communications*, NO.3-4, Vo1.24,2001, pp. 416-421.

[4] Li Layuan. "A formal specification technique for communication protocol". *Proc of IEEE INFOCOM*, April. 1989, pp. 7481.

[5] X. Jia. "A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks," *IEEE/ACM Transactions on Networking*, No.6, Vo1.6, Dec. 1998, pp. 828-837.

[6] M. Faloutsos, A. Banerjea, and R. Pankaj, "QoSMIC: Quality of Service sensitive multicast internet protocol," *SIGCOMM'*98. September 1998.

[7] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new Resource ReSerVation Protocol," *IEEE Network*. Sept. 1993.

[8] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum -cost multicasting," *Proc. IEEE INFOCOM 95, Boston, MA*, April 1995.

[9] Y. Xiong and L.G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE Trans on Networks*, Vol. 7, No. 1, Feb, 1999, pp. 98-110.

[10] S. Chen and K. Nahrstedt, "Distributed QoS routing in ad-hoc networks," *IEEE JSAC*, special issue on ad-hoc networks, Aug. 1999.

[11] Li Laywan. "The routing protocol for dynamic and large computer networks," Journal of computers, No.2, Vol.11, 1998, PP. 137-144.

[12] B. M. Waxman. "Routing of multipoint connections," *IEEE Journal of Selected Area in Communications*, Dec. 1998, pp. 1617-1622.

[13] R. G. Busacker and T. L. Saaty, Finite Graphs and Networks: An introduction with applications, McGraw-Hill, 1965.

[14] Li Layuan and Li Chunlin, "A routing protocol for dynamic and large computer networks with clustering topology," *Computer Communication*, No.2, Vol. 23, 2000, pp. 171-176.

[15] D. G. Thaler and C. V. Ravishankar, "Distributed center-location algorithms," *IEEE JSAC*, Vol. 15, April 1997, pp. 291-303.

[16] I. Cidon, R. Rom, and Y. Shavitt, "Multi-path routing combined with resource reservation," IEEE INFOCOM'97. April 1997,pp. 92-100.

[17] J. Mog. "Multicast routing exeensions to OSPF," RFC 1584. march. 1994.

[18] Li Layuan and Li Chunlin, "The QoS-based routing algorithms for high-speed networks." *Proc of WCC*, Aug. 2000, pp-1623-1628.

[19] Roch A. Guerin and Ariel Orda, "QoS routing in networks with inaccurate information: Theory and algorithms," *IEEE/ACM Trans. On Networking*, No.3, Vol.7. June. 1999. pp. 350-363.

[20] Bin Wang and Jennifer C. Hou, "Multicast routing and its QoS extension: Problems, algorithms, and protocols," *IEEE Network*, Jan/Feb, 2000. pp. 22-36.

[21] Li layuan and Li Chunlin, Computer Networking, National Defence Industry Press, Beijing, 2001.

[22] Moses Charikar, Joseph Naor and Baruch Schieber, Resoutce optimization in QoS multicast routing of real-time multimedia, *Proc of JEEE INFOCOM*. 2000. pp. 1518-1527.