TCP Conformance for Network-Based Control

Arata Koike

NTT Information Sharing Platform Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
E-mail: koike.arata@lab.ntt.co.jp

Abstract. In this paper, we first review the potential roles of a network in improving end-to-end TCP control. We discuss problems associated with such network control when a network provider wants to provide a certain degree of service objective based on the control. When we use feedback control mechanism, we assume a specific behavior of end-hosts as a conformance condition. In case of TCP, there are a lot variants of implementation and the differences give big impacts on TCP performance. We evaluate such impacts for Ackadjust method by simulation and point out the necessity to clarify the relationship for the interworking between network-based control and TCP implementation.

1 Introduction

The Transmission Control Protocol (TCP) / Internet Protocol (IP) [1] is a basis for current computer communication networks. Various applications are realized on the top of TCP/IP. TCP assumes no reliability on underlying network. It features all control be done on an end-to-end basis. It deals with flow control and error control on data delivery.

The initial TCP protocol was specified in 19 years ago [2]. We could consider that choosing the window-based control scheme and the assigned values for control parameters for TCP were sufficient for the capabilities of networks and end-hosts at that time. We can see much improvement on the capabilities of network and end-hosts during last 10 years. For example, as for the network capabilities, we now have high-speed networks such as Asynchronous Transfer Mode (ATM) and Ethernet with the speed of 100Mbit/s or Gbit/s. Although we have gotten such great improvement on networks and end-hosts, we are still utilizing TCP as the transport layer protocol. Of course, there are many proposals on TCP algorithm to improve performance of data transmission to adapt the high-speed transmission. These enhancements, however, are mainly focused on the end-to-end control.

Let us again consider the characteristics of the TCP control mechanism:

- TCP is an end-to-end, window-based flow control mechanism.
- TCP indirectly detects packet loss within a network by observing a time-out or arrivals of duplicate Acknowledgement (Ack) packets and then limits the number of packets allowed entering the network.

S. Rao and K.I. Sletta (Eds.): INTERWORKING 2000, LNCS 1938, pp. 243-253, 2000.

Both of these are features of the end-to-end approach and TCP runs independent of network internal structure. With a low-speed network, this kind of end-to-end protocol works well. For high-speed networks, however, these features affect worse both for throughput performance and fairness among TCP connections. Fairness is not an important issue when the goal is merely end-to-end connectivity between different networks for TCP. However, with the evolution of commercial Internet service, many companies are using the IP network as their enterprise network. This raises fairness as an important issue in addition to end-to-end connectivity.

In addition to the improvement of TCP algorithms, several network-based controls for TCP flows are proposed to achieve better utilization of networks. Now, underlying networks, such as ATM, have many rich capabilities regarding to Quality of Service (QoS). End-users, however, only perceive QoS through end-to-end TCP performance and there is no explicit interworking between TCP and underlying networks. One of the objectives of these network-based controls is to provide capabilities of underlying networks to TCP end-users. For example, Available Bit Rate (ABR) service of ATM [3] performs rate-based control at ATM layer and thus we need an interworking mechanism to notify the change of lower layer characteristics to TCP. By providing lower layer capabilities to TCP layer, TCP can truly utilize the performance and QoS that are provided at the lower layer level.

Once a network begins to provide the above-mentioned network-based control as a network service, there arises an issue regarding to the conformity of users against the control. TCP was considered as an end-to-end protocol so that we did not need to consider conformance issues. But when a network provide various controls as a network service, the network must check the user behavior to keep the service level.

This paper is organized as follows. In the next section, we briefly overview interworking mechanisms of network-based control for TCP. In section 3, we describe the relationship between feedback control and associated conformance issues for ABR and TCP. We investigate TCP conformance for network-based control by simulation in section 4. We discuss the results in section 4 from the viewpoint of TCP conformance at section 5. We finally conclude our paper in section 6.

2 Interworking between TCP and Underlying Network

Since TCP is an end-to-end protocol, it means that there is limited room for intermediate network elements, such as routers, to contribute to the TCP control. In this section, we briefly review methods proposed to date in terms of network control. These methods can be categorized as either explicit or implicit. An explicit method is one in which a network element reports something explicitly to a TCP sender. An implicit method requires no modification of the algorithm to TCP senders.

2.1 Explicit TCP Control by Network

An explicit method can indicate the status of a network element directly. One such explicit contribution could be achieved by sending an ICMP (Internet Control Message Protocol) Source Quench message when a network element suffers severe

congestion. According to the TCP/IP implementation, this will cause a TCP slow start. However, the use of a Source Quench message causes additional side effects. These disadvantages are summarized in [4].

A more explicit scheme is the study of Explicit Congestion Notification (ECN) [4][5]. ECN uses the TOS field of the TCP header and explicitly reports the network status to the TCP source. In contrast to the existing notification scheme, which utilizes packet loss, this scheme should work effectively by reporting network status directly without discarding packets. But this scheme requires modification of TCP software to accommodate ECN. As discussed in [4] and [5], in introducing a new scheme that affects the end-host algorithm, the coexisting environments of the existing and the new terminals must be taken into account.

2.2 Implicit TCP Control by Network

The other way to influence TCP control by a network is by implicit contribution. This kind of method does not require modifications of TCP software, yet the network has the capability of influencing TCP control, which effectively utilizes various TCP properties. One such idea that is widely employed is packet discarding. Random Early Detection (RED) [6] is the most well-known. It intentionally discards packets in a probabilistic manner when the number of stored packets in the buffer exceeds a certain threshold, indicating buffer congestion. It is reported that probabilistic discarding discards more packets of connections that share more bandwidths and is effective for recovering fairness. This scheme is based on packet discarding, but it will be a problem for high-speed networks due to the wait for the retransmission timeout.

In order to obviate packet discarding for high-speed networks, [7] proposes TCP-GATEWAY, which divides the TCP control segments. It maintains connectivity of the TCP connection at a LAN-to-WAN gateway point but it terminates the control loop at the gateway and uses a higher-speed native protocol for backbone segments. In terms of flow control, this method is especially effective for a network that has a large RTT, by reducing the length of each control loop and tuning control parameters appropriately for each segment. TCP, however, involves not only flow control but also error control in its feedback. Thus, in order to assure the reliability of TCP, huge amount of data must be maintained at the gateway.

In [8], [9] and [10], we proposed a rate adjustment function at a router to prevent packet loss and to influence TCP control (Fig. 1). This could be considered as a realization of transport middleware for TCP control. The key idea of this approach is based on the fact that TCP is a window-based control. This rate adjustment function could be done by scheduling forward and/or backward packets according to the current network status. Such scheduling causes a delay in response and limits the transmission of new packets from the TCP source, since the number of packets traveling without being acknowledged is determined by the TCP window size. We applied this scheme for interworking between legacy (e.g., Ethernet) networks and an ATM backbone network using ABR service. This approach enables us to relate rate-based control at the ATM layer and window-based control at the transmission layer. We translated the ATM layer rate information to an Ack packet interval, which

controls the transmission speed of TCP packets. We want to emphasize that our approach does not require any modification of TCP end-hosts. This scheme is further explained in the next section.

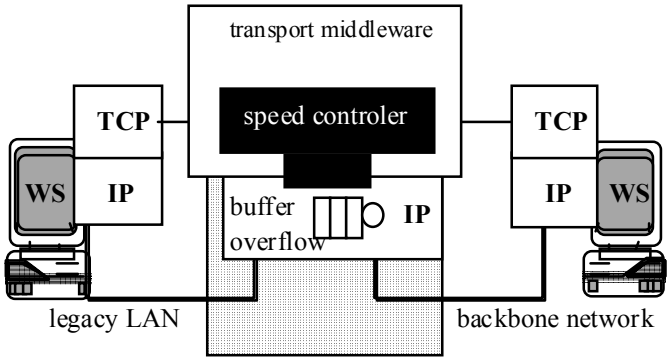

Fig. 1. Transport middleware for TCP rate adjustment

2.3 Interworking between TCP and ABR

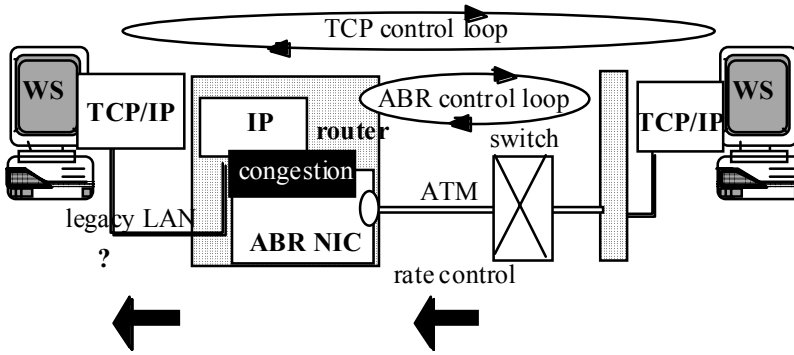

Fig. 2. Interworking between ABR backbone network and legacy LAN

We focus on the interworking between end-to-end TCP and backbone ABR network. ABR is a rate-based ATM layer protocol. It features a closed feedback control loop by using a special cell, called Resource Management (RM) cell. The allowed source transmission rate, ACR, is determined by feedback information that reflects the status of intermediate network nodes or the destination. By this nature, traffic characteristics of ABR services vary time to time. In case of backbone ABR network, the rate-based control is only done within the ATM-ABR segment. It can avoid congestion at the ATM network. On the other hand, when other networks are connected to the backbone ABR network through a router, congestion in ATM switches is simply moved to the router. And since the router has a limited buffer

capacity, this congestion may cause buffer overflow at the router. This results in poor TCP performance even if we achieve good ATM layer performance (Fig. 2). As we described above, we proposed an interworking method to relate ACR information to intervals of Ack packets.

Our proposal achieves self-clocking of the interworking based on the queue length information of forward direction at a router. And we control intervals of Ack packet by creating buffers for Ack packets in a router. We call this scheme as 'AckAdjust' method (Fig. 3). The algorithm of our scheme is as follows. If the queue length l of a buffer for the forward direction in a router exceeds a certain threshold, say t, we schedule the Ack packet of the backward direction based on the rate ACR/H, where H is a parameter. If the queue length is less than or equal to the threshold t but greater than 0, we schedule the Ack packet based on the rate ACR/L, where L is also a parameter.

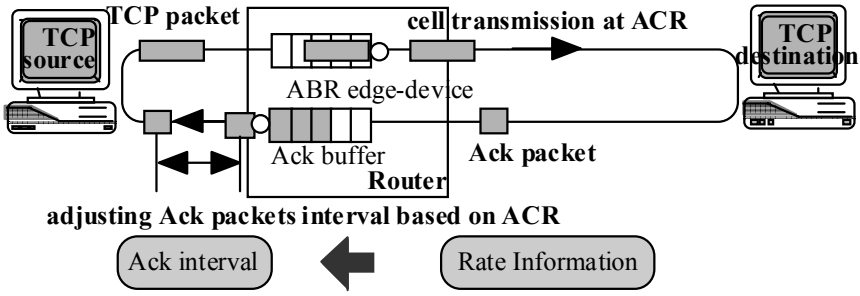

Fig. 3. Illustration of AckAdjust scheme

3 TCP Conformance

As described in the previous section, there have been made a lot of proposals on network-based TCP control mechanisms. An objective of these mechanisms includes providing network capabilities to end-users. Recent advanced networking technologies have potentials to provide high-performance communication. Unfortunately, TCP has no capability to utilize them directly. TCP end-users only perceive network performance and QoS on an end-to-end basis. Thus, using network-based controls makes it possible to implicitly enhance end-to-end TCP performance. TCP uses a feedback control. Therefore, when we initiate a network-based control, we assume a specific reaction of TCP end-hosts to the control. Let us consider two examples. In case of Source Quench, as an example of explicit control, a router that sent ICMP Source Quench packets expects initiation of slow-start at corresponding end-hosts. In case of RED, as an example of implicit control, source end-hosts are

expected to shrink their congestion window size according to the packet discarding by a network.

If we use the network-based control only for an advisory purpose, we do not require any specific reaction of TCP end-hosts. But if we use it as a method to provide guaranteed performance and QoS to end-users, the correctness of a specific reaction of TCP end-hosts shall be one of the important issues. When we relay on a closed-loop feedback control, we always encounter this kind of requirement on the conformance of the behavior of end-users.

For example, ABR service of ATM also uses closed-loop feedback control. Switches can indicate allowed cell transmission rate (ACR) to an ABR source end-terminal by using resource management (RM) cells. Since ABR specification requires fairness among connections and minimum cell rate guarantee, it strictly specifies conformance for the feedback. The conformance requirement for feedback of ABR is a delay bound regarding to the processing delay of an end-terminal to react the indicated transmission cell rate by RM cells. If an end-terminal fails to meet the delay bound to adapt rate change, it is considered as non-compliant to the ABR conformance.

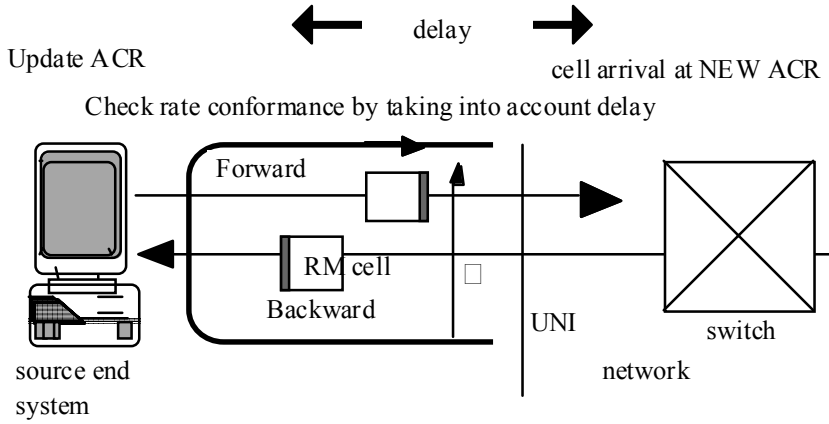

Fig. 4. ABR conformance for feedback

We consider the case for Explicit Rate (ER) switches (Fig. 4). In this case, we observe the following procedures. A backward RM cell passes through an ER switch with a new ER information. It reaches a source end system and ACR at the source was updated by the ER value contained in the backward RM cell. The source sends a new forward RM cell with the new ACR. Then the RM cell reaches the switch. A certain time, □, is required in the course of these procedures. If the new ER value is less than the old one, a UPC function of the switch uses old ER value until a predetermined time □ for the policing threshold. After expiration of □, the switch examine the cell arrival rate using the new ER value. If an end system continues sending cells at the old ER after the time □, the UPC identifies the source as a violator

of the conformance for feedback. We need this kind of requirement for end system when we provide services using feedback control. The requirement for the above example is delay in control. By using such requirement and the corresponding checking mechanisms, a network can provide services fairly among each end system.

In case of TCP, the time scale required to show the effect of TCP flow control is much longer than that of ABR control. Therefore, in practice, it has little meaning as a conformance requirement that takes into account delay bound regarding to the processing delay of an end-host. On the other hand, in case of TCP, differences in end-host mechanisms have impacts on the effectiveness of network-based control. In case ABR, end system behavior for the feedback is strictly specified. Therefore, we could expect the same reaction for all end systems for the same feedback. On the other hand, there exist various kinds of TCP implementations, variants and options according to the evolution of TCP. If we only consider connectivity as the role of TCP, we can assure interoperability among all different versions of TCPs. When we introduce network-based control to provide better performance for TCP end-users, however, such differences in TCP cause problems. This is because if we apply the same control mechanisms for different TCP flows, TCP end-hosts behave differently depending on their implementations. For example, in case of RED, we could expect different results for TCP with go-back-N and TCP with SACK even if a network discards packets in the same way. A network-based control assumes a specific end-host behavior but the control functions only when end-hosts support the expected behavior. Therefore, we must determine conforming TCP mechanisms to meet the requirement of network-based control.

4 Simulation for TCP Conformance

Here, we investigate how TCP implementations affect the outcome of network-based controls. As an example of network-based control, we use Ackadjust scheme and consider interworking between end-to-end TCP and backbone ABR network [10]. This scheme translates ABR rate information to the interval of TCP's Ack packets at the edge of ABR connection (Fig. 3). As a result, data transmission rate of a TCP end-host is accommodated to that of ABR without changing the TCP source code. In this paper, we consider the effect of TCP implementation differences based on the AckAdjust scheme.

As an example of TCP implementation difference, we investigate the differences in the number of Ack packets sent from a destination. Since the Ackadjust scheme causes delay in Ack packets of each TCP flows based on ACR information, the differences in the number of Ack packets shall affect the outcome of the scheme. Some TCP implements so called 'delayed-Ack' method. The 'delayed-Ack' method allows acknowledging multiple data packets by a single Ack packet. Here, we assume for 'delayed-Ack' TCP that a destination end-host transmits an Ack packet when it receives five data packets or when 100 ms has passed after transmitting the last Ack packet.

4.1 Simulation Configuration

The simulation model is described in Fig. 5.

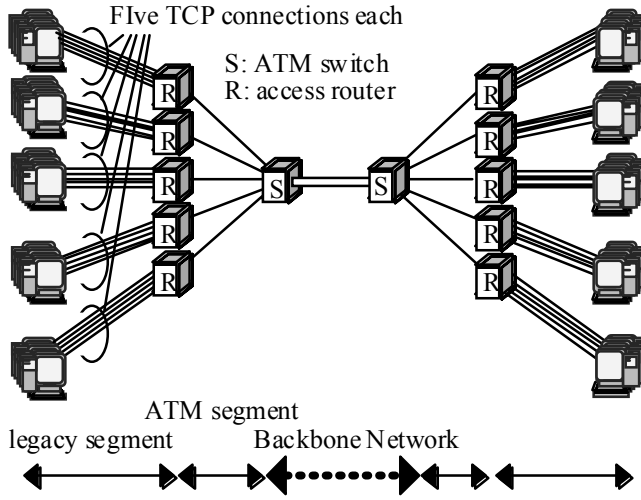

Fig. 5. Simulation model

There are five routers and five TCP sources at the source side. Each TCP source is connected to a router. The same configuration appears at the destination side. We call these segments legacy segments. Each router is assumed to be an ABR edge-device. Routers are connected to a switch and are aggregated to a backbone link. Then they are distributed to destination side routers at the opposite side switch. Router-switch (switch-router) segments are ATM segments. Each router terminates a Virtual Circuit (VC) and the five TCP connections are aggregated on a VC.

4.2 Simulation Parameters

We use the following simulation parameters.

4.2.1 Network Parameters

We assume Minimum Cell Rate (MCR) is 0 Mbit/s and Peak Cell Rates (PCR) of each VC is 150Mbit/s. Link rates for legacy segments are 100Mbit/s. Distances in both legacy segments and router-switch segments are 1 km. The backbone link length is 10 km. The light velocity is assumed to be 200,000 km/s. The ABR switch algorithm we used is CAPC2 [11] and we assume 95% utilization. Buffer size at ATM switches are 4096 cells and we did not observe any cell loss events at the ATM switch.

4.2.2 TCP Parameters

We use TCP-Reno, which includes fast retransmission and fast recovery. The TCP timer granularity is 200 ms. We assume the use of TCP window scale option and the maximum TCP window sizes is 128 Kbytes. Each packet size is 9188 bytes. We also assume the TCP destination packet processing delay obeys an exponential distribution with a 1-ms mean. Each TCP source transmits data until the end of simulation, and all sources start to transmit simultaneously.

4.2.3 Router Parameters

Data buffer size is 15 packets. The threshold of data queue is 3 packets and H (when $l > t$) = 2.08 and L (when $0 < l \bullet t$) = 0.5 which is the same as in [10]. The Ack buffer size is unlimited. This final assumption is validated because Ack packets only contain a sequence number in their payload so the storage requirement is much smaller than for TCP datagram packets.

5 Discussion

We first describe the comparison of the TCP performance with and without the use of AckAdjust scheme for a non-delayed-Ack environment. In Fig. 6, we show the results for 25 TCP connections. Without using AckAdjust scheme, we can see the performance degradation both in throughput and fairness due to the packet discarding. In general, ABR achieves fairness among VCs but it is not expected to contribute fairness among TCP flows. But from this figure, by using AckAdjust scheme, we can see the realization of fairness among TCP flows in a VC.

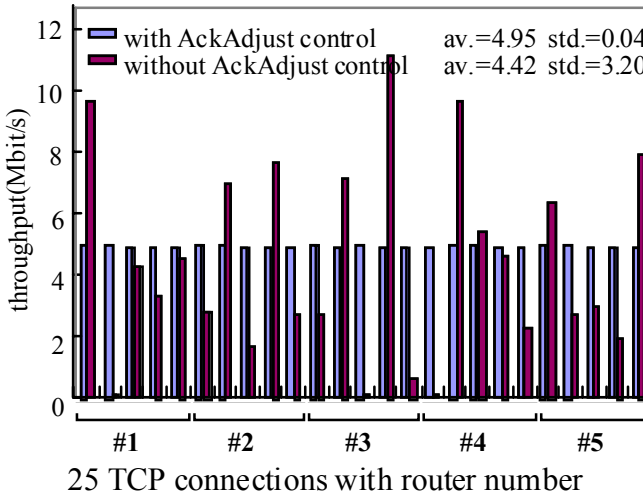

Fig. 6. Throughput characteristics with and without AckAdjust scheme

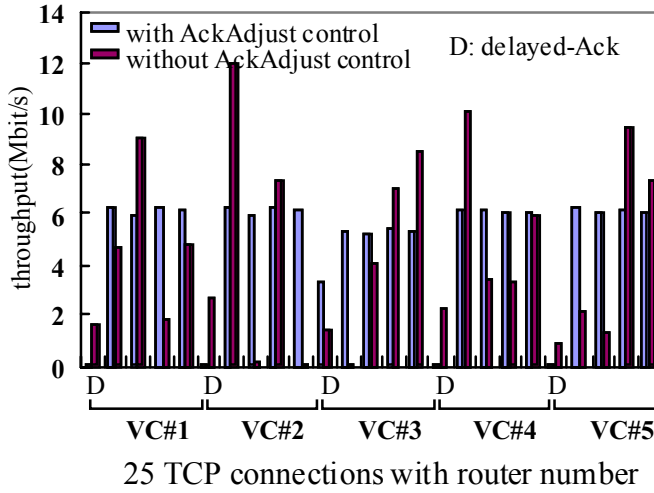

Fig. 7. Throughput characteristics with delayed-Ack TCP connections

We next discuss the case for delayed-Ack environment. For each router, we assume one of the five TCP connections uses 'delayed-Ack' method. The other four TCP connections' destinations always return an Ack packet when they receive a data packet. Using this configuration, we show a result in Fig. 7. 'Delayed-Ack' TCP connections are indicated by a symbol 'D'. This figure shows that fairness among TCP connections are kept except for the connections using 'delayed-Ack' mechanism. 'Delayed-Ack' connections suffer unfairness in throughput since the mechanism causes delay in control and thus throughput degradation occurs. As a result, when we adopt the network-based control, i.e., Ackadjust scheme, to improve fairness, we can conclude that TCP implementation with 'delayed-Ack' is not a conforming implementation for that control.

6 Conclusion

In this paper, we briefly overview network-based control for TCP. We consider problems associated with such network-based control to achieve service objectives. Networks assume a certain end-hosts behavior as a conformance condition for feedback control. Differences in TCP implementation are an important conformance condition regarding to a network-based control. We investigate such example based on AckAdjust scheme, which controls Ack packets intervals and achieves TCP and ABR interworking. As a conclusion, we need to clarify the relationship between the performance and TCP implementation when we have performance objectives based on a network-based control.

Acknowledgement

Thanks are due to Ms. Kiyoka Takeishi and Ms. Mika Ishizuka for her support in the simulation and for useful discussions.

References

1. Stevens, W.R.: TCP/IP Illustrated Vol. 1, Addison Wesley (1994)
2. Postel, J. B.: Transmission Control Protocol, IETF RFC793, September (1981)
3. The ATM Forum: Traffic Management Specification 4.0, af-tm-0056.000, April (1996)
4. Floyd, S.: TCP and Explicit Congestion Notification, ACM Computer Communication Review, V. 24 N. 5, October (1994) 10-23
5. Ramakrishnan, K. K., Floyd, S.: A Proposal to add Explicit Congestion Notification (ECN) to IP, IETF RFC 2481, January (1999)
6. Floyd, S., Jacobson, V.: Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, Vol. 1 No. 4, August (1993) 397-413
7. Hasegawa T., Hasegawa, T., Kato, T., Suzuki, K.: Implementation and Performance Evaluation of TCP Gateway for LAN Interconnection through Wide Area ATM Network, Trans. IEICE, B-I, Vol. J79-B-I, No. 5, (1996) 262-270 (in Japanese)
8. Koike, A.: TCP flow control with ACR information, ATM Forum/97-0758, September (1997)
9. Koike, A.: TCP/ABR interworking, ATM Forum/97-0998, December (1997)
10. Koike, A.: Interworking between end-to-end TCP flow control and ABR rate-based control for backbone network, Proc. of Interworking98, July (1998)
11. Barnhart, W.: Example Switch Algorithm for Section 5.4 of TM Spec.: ATM Forum/95-0195, February (1995)