PARAMETER TUNING OF RATE-BASED CONGESTION CONTROL ALGORITHMS AND ITS APPLICATION TO TCP OVER ABR

Hiroyuki Ohsaki, Go Hasegawa, Masayuki Murata and Hideo Miyahara

Department of Information and Computer Sciences
Faculty of Engineering Science, Osaka University
Toyonaka, Osaka 560, Japan
E-mail: oosaki@ics.es.osaka-u.ac.jp

Abstract – In this paper, the rate-based congestion control algorithm that has been standardized in the ATM Forum is evaluated. Its behavior is analyzed by utilizing a first-order fluid approximation to provide control parameter tuning. We obtain the maximum queue length at the switch and conditions for avoiding under-utilization. The results are then applied to TCP over ABR service class. More specifically, we evaluate the TCP-level performance through a simulation technique when TCP is applied to ABR service class with the rate-based congestion control. We first compare rate-based control of ABR service and EPD applied to UBR service, and show that rate based congestion control achieves better fairness and higher throughput in most circumstances. However, we demonstrate that rate-based control requires careful tuning of control parameters to obtain its effectiveness.

I. INTRODUCTION

In the ATM Forum, standardization effort has been devoted for the rate-based congestion control, which is aimed at being applied to ABR (Available Bit Rate) service class [1, 2, 3]. The target of the standard is an operation algorithm of both source and destination end systems, and implementation issues regarding an intermediate node (ATM switch) is left to manufactures. However, some example behaviors of intermediate switches are also introduced in the standard [4]. In the current paper, we focus on the simplest switch among those, which is referred to as "EFCI bit setting switch" or "binary switch". In the binary switch, the congestion is detected by the predefined queue threshold at the buffer, and if the queue length exceeds some predefined threshold value, it is recognized as the congestion.

In the standard document [2], several control parameters are defined for controlling cell transmission at the source end system. For instance, envelopes of rate-increase and rate-decrease are controlled by AIR (Additive Increase Rate) and RDF (Rate Decrease Factor), respectively (see Section II-A). At a connection establishment process, the source end system negotiates these control parameters with the network through P-Vector, which is a predefined set of control parameters [2]. The network selects an appropriate P-Vector from the predefined P-Vector sets according to, for example, an estimated round-trip propagation delay or the number of active connections.

Effectiveness of the rate-based congestion control is heavily dependent on an appropriate choice of these control parameters as indicated in [3]. When these parameters are configured properly, the rate-based congestion control can achieve high performance (i.e., no buffer overflow, high link utilization and small cell delay). However, a selection method of control parameters is not specified in the standard, and therefore those should be determined intuitively unless a proper tool is offered. In the current paper, we will show appropriate control parameter set suitable to various operating environments of the rate-based congestion control through the analysis.

There are a lot of research for the rate-based congestion control by a simulation technique. However, the simulation technique cannot offer an effective means when it is applied to parameter tuning. Therefore, in this paper, we take an analytic approach for determination of control parameters. In [3, 5], we have shown performance of a rate-based congestion control algorithm proposed in [4], a former draft of the standard having been discussed by the ATM Forum. In this paper, we modify our previous work to the latest rate-based congestion control standard [2], and show appropriate settings of control parameters of the binary switch. Through the steady state analysis, we will derive two conditions which source control parameters should satisfy; one is for achieving no buffer overflow at the switch buffer, and the other is for full link utilization.

For an application of the rate-based congestion control, we then investigate TCP (Transmission Control Protocol) layer performance since TCP is one of most popular and important protocols as Transport layer protocol and is widely used for today's Internet. For supporting TCP, we have two service classes available at the ATM layer: UBR (Unspecified Bit Rate) and ABR service classes. Since the UBR service class has no means against congestion, it relies on TCP's congestion control mechanism when TCP is applied to the UBR service class (which we will refer to as *TCP over plain UBR* in the below). It has been shown in [6] that this sort of combination cannot provide an effective resource usage. A main reason is that if at least one cell of multiple cells segmented from the upper-layer PDU (Protocol Data Unit: TCP/IP data packet in this case) is lost, the entire packet is treated to be lost since the ATM or AAL layers do not provide the cell retransmission function for error recovery. Then, the authors in [6] have introduced PPD (Partial Packet Discard) and EPD (Early Packet Discard) schemes in the UBR service class. As shown in [6], EPD can provide considerable throughput improvements when compared with TCP over plain UBR.

However, we now have a more sophisticated method: rate-based congestion control mechanism for the ATM layer. Since EPD can be viewed as UBR service class with a minor additional function, it must be cheaper than the rate-based approach in control complexity. However, the EPD's applicability is limited to AAL5 due to its inherent structure. On the other hand, the ABR service class can provide more general data transfer service at the expense of its control cost. Its defect is to require several control parameters and as has been shown in [3], the appropriate choice of the control parameters is a key issue for the rate-based congestion control to work effectively. While in the standard document [2], several parameter sets are provided as a parameter profile, there are no validation for those parameter sets to be applicable to the upper layer protocols.

In the current paper, we investigate the performance of TCP over ABR service by comparing with TCP over plain UBR and UBR with the EPD scheme (referred to as *TCP over EPD* in the below). In Subsection III-B, control parameter values of the ABR service are first drawn from the standard profile in [2], and it will be shown that TCP over ABR sometimes becomes inferior to other methods due to inappropriate control parameters. Then, we will apply the control parameter sets based on the analysis presented in Subsection II to demonstrate that an appropriate choice of control parameters can improve the performance of TCP over ABR in Subsection III-C.

The rest of this paper is organized as follows. In Section II, the parameter tuning method of the rate-based congestion control mechanism is presented. The analytic model for the rate-based congestion control is described with definition of parameters, and the behaviors of source and destination end systems and the switch is summarized in Subsection II-A. In Subsection II-B, we derive the dynamical behavior of the cell transmission rate of each source end system and the queue length at the switch buffer. Tuning of control parameters to take the advantage of the rate-based control are then presented in Subsection II-C. In Section III, we will introduce the simulation model, and then provide the comparative performance evaluation of TCP over plain UBR, TCP over UBR with EPD (TCP over EPD), and TCP over ABR. Parameter tuning of the rate-based control scheme is then applied to TCP over ABR. Section IV concludes our paper with some future research topics.

II. PARAMETER TUNING OF RATE-BASED CONGESTION CONTROL MECHANISM

A. ANALYTIC MODEL

Our analytic model consists of homogeneous traffic sources and a single bottleneck ATM link as shown in Fig. 1. The number of active connections that share the bottleneck link is denoted by N_{VC}. The bandwidth of the bottleneck link is denoted by BW, and propagation delays between the source and the switch, and between the switch and the destination are denoted by τ_{sx} and τ_{xd}, respectively. These parameters are varied according to the network configuration (LAN, MAN, or WAN). The round-trip propagation delay between source and destination end systems is denoted by $\tau(= 2\tau_{sx} + 2\tau_{xd})$. We further introduce $\tau_{xds}(= 2\tau_{xd} + \tau_{sx})$ as the propagation delay of congestion indication from the switch to source end systems via destination end systems. Propagation delays τ_{sx} and τ_{xd} are assumed to be identical for all of source and destination pairs.

In the analysis, it is also assumed that each source end system has infinite cells to transmit. Namely, the permitted cell rate ACR (Allowed Cell Rate) is always equal to the actual cell transmission rate CCR (Current Cell Rate). Furthermore, by assuming that they all start cell transmission simultaneously with same rate-control parameters, all of connections behave identically in our model.

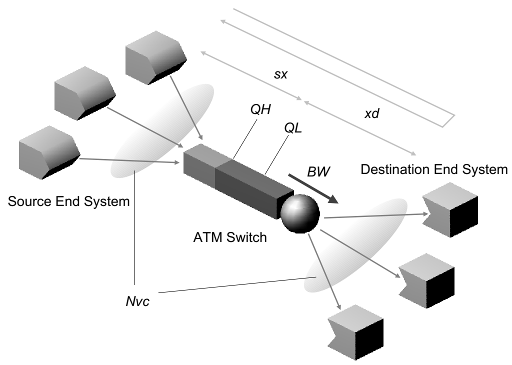


Fig. 1: Analytic Model.

At the switch, congestion occurrence is detected by high and low threshold values associated with the queue length of the buffer. These threshold values are denoted by Q_H and Q_L, respectively. When the queue length exceeds the high threshold value Q_H, the switch detects congestion and notifies it to the source end system via the destination end system by marking the EFCI (Explicit Forward Congestion Indication) bit in the data cells, or by marking the CI (Congestion Indication) bit in the forward RM (Resource Management) cells. After then, when the queue length goes under the low threshold value Q_L, it is regarded that its congestion is terminated. The EFCI bit in data cells or the CI bit in forward RM cells is then cleared to indicate congestion relief to source end systems via the destination end system. This sort of switch operation is often referred to as an EFCI bit marking switch, or simply as a binary switch [3]. As described in the above, we will first assume infinite buffer capacity in Section II-B to examine the steady state behavior. We will then consider the finite case in order to obtain the condition for no buffer overflow in Section II-C.

Behavior of end systems are currently developed in the ATM Forum, and its draft is specified in [2]. Each source end system periodically generates forward RM cells in proportion to its rate; it sends one forward RM cell per N_{RM} data cells. As soon as the receipt of the forward RM cell, the corresponding destination end system sends it back as the backward RM cell to the source end system. Allowed cell rate ACR of the source end system is changed in response to backward RM cells. When it receives the CI bit cleared cell, rate increase is performed as

$$ACR \leftarrow \min(ACR + AIR\, N_{RM}, PCR),$$

where PCR (Peak Cell Rate) is the maximum rate for this connection negotiated at its establishment, and AIR (Additive Increase Rate) is the ratio for proportional rate increase. The CI bit marked cell decreases ACR as

$$ACR \leftarrow \max(ACR - ACR\, N_{RM}/RDF, MCR),$$

where MCR (Minimum Cell Rate) is the minimum rate (or guaranteed rate) for this connection but in usual case MCR may be set to zero. RDF (Rate Decrease Factor) is a ratio for exponential rate decrease.

In the following analysis, forward RM cells are not explicitly considered; congestion indication is performed by the EFCI bit of data cells. However, forward RM cells can easily be taken into account by replacing BW in our analysis by BW' as

$$BW' = BW\, N_{RM}/(N_{RM}+1).$$

When cell queueing takes place at the buffer, the rate at which source end system receives backward RM cells will be bounded by $BW/(N_{VC}\, N_{RM})$. Otherwise it is identical to the cell transmission rate of the source end system. Thus we require an analytical treatment different from the one presented in [7] where the rate change is performed on a timer basis [3].

B. *DYNAMICAL BEHAVIORS*

We first introduce $ACR(t)$ and $Q(t)$ which represent the allowed cell rate ACR of each source and the queue length at the switch observed at time t. In what follows, evolutions of $ACR(t)$ and $Q(t)$ are analyzed by modifying our previous analysis presented in [5]. In those papers, behaviors of source and destination end systems are based on [4], which is one of the rate-based congestion control algorithm proposals. In the current paper, the analysis is modified to reflect the latest standard algorithm described in [2], and is summarized in the below.

Evolutions of $ACR(t)$ and $Q(t)$ have periodicity in steady state as shown in Fig. 2. The behaviors of $ACR(t)$ and $Q(t)$ are divided into four phases. We further introduce $ACR_i(t)$ and $Q_i(t)$, which are defined as

$$\begin{aligned} ACR_i(t) &= ACR(t-t_{i-1}) \quad 0 \le t < t_i, \\ Q_i(t) &= Q(t-t_{i-1}) \quad 0 \le t < t_i, \end{aligned}$$

where t_i is defined as the time when Phase i terminates. Furthermore the length of Phase i is represented by $t_{i-1,i}$ defined as $t_{i-1,i} = t_i - t_{i-1}$.

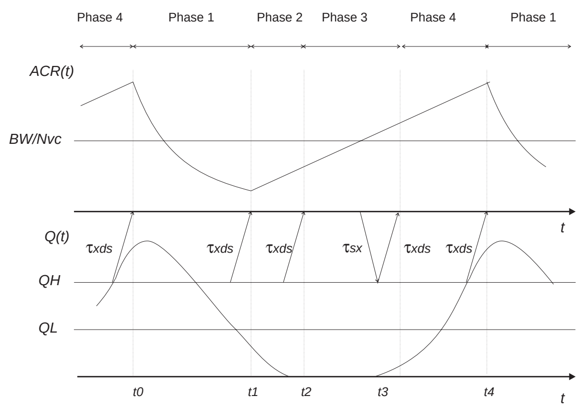


Fig. 2: Pictorial View of $ACR(t)$ and $Q(t)$.

Due to space limitation, we omit the detailed derivation of $ACR_i(t)$, and show only results. See [8] for more precise treatment.

$$\begin{aligned} ACR_1(t) &= ACR_1(0)e^{\frac{BW}{N_{VC}\,RDF}t}. \\ ACR_2(t) &= ACR_2(0) + \frac{AIR\,BW}{N_{VC}}t. \\ ACR_3(t) &\cong ACR_3(0)e^{AIR\,t}. \\ ACR_4(t) &= ACR_4(0) + \frac{AIR\,BW}{N_{VC}}t. \end{aligned} \tag{1}$$

The evolution of $ACR(t)$ and $Q(t)$ is finally obtained by determining the initial value $ACR_i(0)$ and the duration between two phases $t_{i,i+1}$. Given initial rates of Phase i, $Q_i(t)$ is obtained as

$$Q_i(t) = \max(Q_i(\tau_{sx}) + \int_{\tau_{sx}}^{t} (N_{VC}\, ACR(x-\tau_{sx}) - BW)dx, 0), \tag{2}$$

and its duration $t_{i-1,i}$ is defined as

$$t_{i-1,i} = \begin{cases} Q_1^{-1}(Q_L) + \tau_{xds} & i=1 \\ \min(Q_2^{-1}(Q_H) + \tau_{xds}, Q_2^{-1}(0) + \tau_{xds}) & i=2,4 \\ ACR_3^{-1}(BW/N_{VC}) + \tau & i=3 \end{cases}$$

where $ACR_i^{-1}(t)$ and $Q_i^{-1}(t)$ are defined as inverse functions of $ACR_i(t)$ and $Q_i(t)$, respectively. Refer to [8] for more details of these derivations.

C. *PARAMETER TUNING*

In this section, we show appropriate settings of control parameters at source end systems (such as AIR and RDF) and threshold values at the switch (Q_H and Q_L) by utilizing our analysis presented in Section II-B. For this purpose, we first obtain two conditions: conditions for avoiding buffer overflow and for full link utilization in steady state.

1) *Condition for Avoiding Buffer Overflow:*

In Section II-B, the dynamical behavior of $Q(t)$ is derived by assuming the infinite capacity of the cell buffer at the switch. However, its size is limited because of cost or technology limitation in real implementation, which results in the cell loss at the switch buffer. The purpose of this subsection is therefore to derive the condition for avoiding buffer overflow. To this end, we first derive the maximum queue length in steady state in closed-form, which should be bounded by the buffer size for assuring no cell loss.

Let us introduce Q_{max} for the maximum queue length in steady state. While it is possible to obtain Q_{max} by utilizing the analysis presented in Section II-B, it requires an iterative calculation as has been noted. We therefore derive a more tractable condition instead. It is based on the following observation. When the increase/decrease rate becomes large, $Q(t)$ is likely to oscillate to a large extent, i.e., the amplitude of $Q(t)$ becomes large. Then, as the maximum of $Q(t)$ becomes larger, the minimum of $Q(t)$ finally reaches zero. Inversely, the maximum queue length can be bounded by assuming that the minimum of the queue length $Q(t)$ reaches zero. As can be observed in Fig. 2, $Q(t)$ starts growing at the end of Phase 3, that is, at the time τ_{sx} after

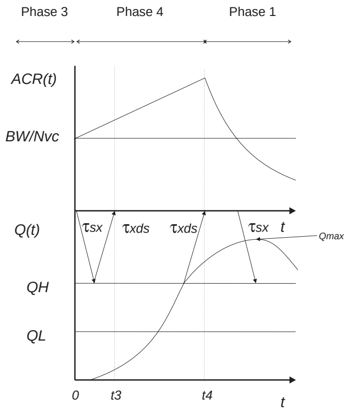


Fig. 3: Pictorial View of $ACR(t)$ and $Q(t)$.

from when aggregate cell rate ($ACR\,N_{VC}$) exceeds the bottleneck link bandwidth BW. Namely, in our analysis, initial values of $ACR(t)$ and $Q(t)$ are set as follows.

$$ACR(0) = BW/N_{VC} \quad (3)$$
$$Q(\tau_{sx}) = 0 \quad (4)$$

Evolutions of $ACR(t)$ and $Q(t)$ are then obtained for Phase 3 followed by Phases 4 and 1. As shown in Fig. 3, $Q(t)$ reaches its maximum in Phase 1, after τ_{sx} from when ACR is decreased to BW/N_{VC}. Hence Q_{max} is obtained as

$$Q_{max} = Q_1(t_{max}),$$

where t_{max} is the time when $Q(t)$ reaches its maximum in Phase 1 and obtained as

$$t_{max} = t_4 + ACR_1^{-1}(BW/N_{VC}) + \tau_{sx}.$$

Note that $Q_1(0)(= Q_4(t_4))$ is required for obtaining $Q_1(t)$ as the initial value, and $Q_1(0)$ and t_4 can be obtained from Eqs. (1), (2), (3) and (4). To simplify the following analysis, Phase 3 is not taken into account; that is, Eqs. (3) and (4) is used as the initial values of Phase 4. Finally, Q_{max} is obtained in the closed-form equation as

$$\begin{aligned} Q_{max} = Q_H + RDF\sqrt{\frac{2AIR\,N_{VC}\,Q_H}{BW}} \\ -RDF\,N_{VC}\,log\left(1 + \frac{2AIR\,Q_H}{BW}\right) + \frac{\tau^2\,AIR\,BW}{2} \\ +\tau\left(\sqrt{2AIR\,BW\,Q_H} + AIR\,RDF\,N_{VC}\right). \end{aligned}$$

The condition for avoiding buffer overflow is finally obtained as

$$BL \geq Q_{max}, \quad (5)$$

where BL denotes the buffer size at the switch. As stated before, Q_{max} takes a smaller value if $Q(t)$ is always greater than zero. (Actually, we will derive such a condition in the next subsection since the case where the queue length occasionally becomes zero implies to not fully utilize the output link.) Hence, Eq. (5) is a strict condition for control parameters to achieve no cell loss.

2) Condition for Full Link Utilization:
The bottleneck link is fully utilized when the buffer at the switch never becomes empty. In other words, full link utilization is achieved when the following condition is satisfied.

$$Q_{min} > 0, \quad (6)$$

where Q_{min} is the minimum queue length in steady state. As explained before, evolution of $Q(t)$ has a periodicity in steady state. $Q(t)$ takes its minimum at the beginning of Phase 2, that is, after τ_{sx} from when ACR reaches BW/N_{VC}. Thus initial values of $ACR(t)$ and $Q(t)$ are decided as follows.

$$ACR(0) = BW/N_{VC}$$
$$Q(\tau_{sx}) = Q_{min} \quad (7)$$

Similar to the approach taken in Subsection II-C.1, evolutions of $ACR(t)$ and $Q(t)$ are obtained from Phase 1 followed by Phases 2, 1, and 2. $Q(t)$ takes again its minimum (Q_{min}) during Phase 2. Hence Q_{min} should satisfy the following equation.

$$Q_{min} = Q_2(t_{min}), \quad (8)$$

where t_{min} is the time when $Q(t)$ reaches its minimum in Phase 2, and obtained as

$$t_{min} = t_1 + ACR_2^{-1}(BW/N_{VC}) + \tau_{sx}.$$

Since t_1 and initial values $ACR_2(0)$ and $Q_2(0)$ depends on Q_{min}, iterative calculation is required to find Q_{min}.

The optimum set of source and switch control parameters exists in the region where both of Eqs. (5) and (6) are satisfied. However, in some condition like extremely large propagation delays, it becomes impossible to satisfy both conditions as will be shown in the next section.

III. APPLICATION TO TCP OVER ABR

In this section, we apply the results obtained in the previous section to TCP over ABR service class.

A. SIMULATION MODEL

We have added the TCP layer on ATM layer in Fig. 1. In the ATM layer of each source end system, cells segmented from the upper layer PDU (TCP packet with IP and AAL5 overhead) are transmitted to the switch. The actual cell emission rate may be limited by the TCP window size, and by the rate of ABR when congestion occurs. As soon as the destination end system receives cells, these cells are reassembled into the packet, and then passed to the TCP layer. Corrupted packets are discarded if at least one cell of the packet is lost. Finally, the TCP ACK (Acknowledgment) packet is returned to the source end system in order to inform the latest number of the successfully received packet.

For comparison purposes, we conduct the simulation experiments for UBR and EPD in addition to ABR for the ATM layer.

1. Plain UBR
   No congestion control mechanism is provided at both of the switch and end systems. When the cell buffer at the switch becomes full, all incoming cells are discarded.

2. EPD mechanism [6]
   In EPD, when at least one cell of the packet is lost, it would be impossible to reassemble the packet at the destination end system. Therefore, to avoid the waste of bandwidth by transferring incomplete packets, EPD discards all cells of the newly arriving packet when the queue length at the switch buffer exceeds some threshold value. Following [6], we set the threshold at half the buffer size in simulation.

For TCP, we have implemented 4.3-Tahoe BSD release [9, 10]. We have also implemented the fast retransmit procedure [9] to discover the possible packet loss quickly. As suggested in [6], the clock granularity for measuring the round-trip time is set to 0.1 ms to avoid the coarse-grained clock. Furthermore, the random traffic is added to the switch to avoid the traffic phase effect of TCP [6]. The mean arrival rate of such traffic is set at 1% of the link capacity. The TCP sender continues to send the fixed-size packet (or segment) as long as the window size is allowed. We assume that IP is irrelevant and it only determines the size of the packet as in [6]. In addition, the processing time of TCP is assumed to be neglected. This assumption may be unrealistic since the processing overhead affects performance. Nevertheless, we assume zero processing time for investigating our objective, i.e., potential capabilities of the above-mentioned congestion control mechanisms for supporting high speed data transfer.

B. COMPARATIVE EVALUATION

In this subsection, we provide comparative results for TCP over UBR, TCP over EPD and TCP over ABR in terms of the throughput and the fairness among connections. The run time of each simulation experiment is 50 s, which is reasonably long to obtain steady state statistics. The start time of connections are staggered by 1 ms: i th connection starts its packet transmission at $(i-1)$ ms to avoid the traffic phase effect as in [6]. The TCP packet size is set to 4,352 byte. It corresponds to about 90 ATM cells including overheads of IP and AAL5. In this section, control parameter values for the ABR service are fixed based on the standard profile [2];

- N_{RM}: 32
- PCR (Peak Cell Rate): 353.2 cells/msec
- MCR (Minimum Cell Rate): PCR/1000
- $AIRF$ $(= AIR\,N_{RM}/PCR)$: 1/64
- $RDFF$ $(= RDF/N_{RM})$: 16
- ICR (Initial Cell Rate): $PCR/20$

Parameter tuning for improving the performance will be presented in the next subsection.

In what follows, we first compare the performance of three methods mainly in terms of the effective throughput, which is defined as the total numbers of successfully transmitted packets (i.e., the sum of latest sequence number of the received acknowledgments at sources) in cells normalized by the link capacity.

1) Comparisons of Effective Throughput:

The first result in Fig. 4 shows the effective throughput and the number of lost packets during the simulation run dependent on the buffer size. The round-trip propagation delay between sources and destinations, τ, is set at 0.02 ms, approximately 4 km. The number of connections, N_{VC}, is set to 10. We note that the similar result can be found in [6] for TCP over UBR and TCP over EPD cases. From the figure, we can see the effectiveness of TCP over ABR irrespective of the buffer size.

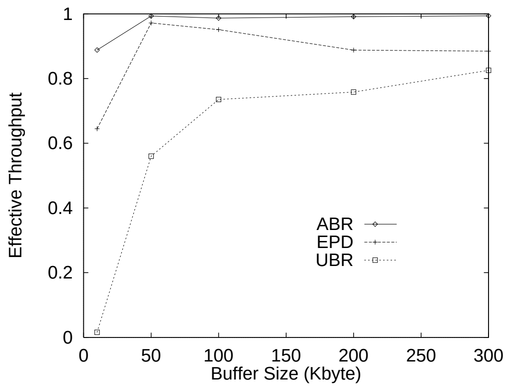


Fig. 4: Comparisons of effective throughput among three methods as a function of buffer size: $\tau = 0.02$ ms, $N_{VC} = 10$.

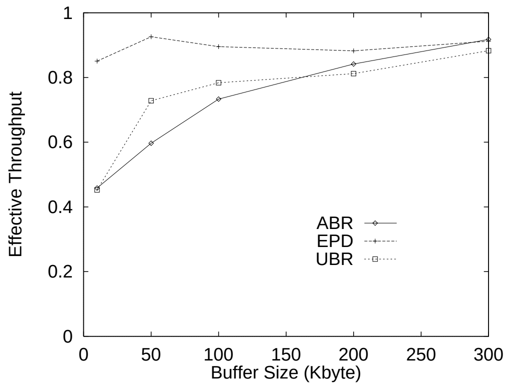


Fig. 5: Comparisons of effective throughput among three methods as a function of buffer size: $\tau = 2.0$ ms, $N_{VC} = 10$.

When the propagation delay becomes much larger, however, the order of the performance gain in three methods is reversed. The case of 2.0 ms propagation delay is shown in Fig. 5. The throughput of EPD and UBR cases are slightly decreased, but throughput degradation of ABR is drastically except the case of the large buffer size. In UBR and EPD, the congestion management relies on the TCP layer. Thus, the effect of the propagation delay is not so large if our concern is the throughput.

The longer propagation delay degrades performance of ABR, which can be observed by comparing Figs. 4 and 5. In the current paper, we use the binary switch for the rate-based congestion control. The long propagation delay causes control delay on congestion occurrence. Then, the rate decrease at the source end systems is also delayed. It introduces a long congestion period, which is one of main problems of the rate-based congestion control algorithm [5]. When the packets are corrupted due to cell losses during the long congestion period, the time out at the TCP layer tends to occur frequently. Then, the TCP window size is decreased to one segment according to its congestion control strategy. If the cell losses take place continuously for some time, window sizes of all TCP connections are decreased. It is just the ABR case with inappropriate control parameter settings. In the case of EPD, on the other hand, such a case can be avoided. Once a packet is accepted at the buffer, the whole packet consisting of multiple cells can be transfered onto the output link. This mechanism can assure the randomness of packet dropping to some extent, and the packet loss is likely to be detected by the first retransmit policy, which avoids the throughput degradation.

Last, we illustrate the effect of the number of connections on the effective throughput in Fig. 6. The propagation delays, τ, are set to 0.02 ms (Fig. 6(a)) and 2.0 ms (Fig. 6(b)), and the buffer size is 300 Kbyte. When the propagation delay is small (0.02 ms), the high throughput can be kept even with the larger number of connections in both of ABR and EPD. On the other hand, when the propagation delay becomes large, the throughput is degraded by the larger number of connections in ABR. It is another problem of ABR that the maximum queue length is increased almost linearly by the number of connections [5]. Then, cell losses and resulting packet losses are increased in the case of ABR. The parameter tuning of ABR in this case will be also discussed in the next subsection.

2) Fairness Comparison:
We next examine a fairness aspect in terms of throughput. In Fig. 7, we present the number of successfully transmitted packets for each connection as a function of time in ABR and EPD. The propagation delay and the buffer size are set to 0.02 ms and 300 Kbyte, respectively. From this figure, we can see an excellent fairness property of ABR. An exception is observed during very early time of simulation since the connection with the larger number experiences the larger round trip time at the start up time. Then the increase of the TCP window size is delayed. On the other hand, in the case of EPD, the fairness cannot be provided; a close glance exhibits that only a limited number of connections transmit packets at time. See, for example, around 300 ms in Fig. 7(a). One connection transmits about 40 packets during 10 ms while other connections does only a few packets. This tendency can be explained as follows. In EPD, when the queue length exceeds the threshold, the newly arriving packets are blocked as long as the cells belonging to the same accepted packet arrive. After the EOP cell arrives, another newly arriving packet can be accepted. The problem is that such a new packet (actually the first cell of the packet) is likely to come from the same connection. That is, the link tends to be possessed by the limited connections for some time.

The next figure shows the case of the larger propagation delay, 2.0 ms (Fig. 8). While the degree of fairness among connections is not very different from the previous case in EPD, it is degraded in ABR. This indicates that the large propagation delay heavily affects the behavior of ABR in throughput as well as in the fairness. The fairness among connections in ABR can be further improved, which will be presented in the next subsection.

C. PARAMETER TUNING FOR TCP OVER ABR

In the previous subsection, we have shown that the ABR service can provide an effective use of the bandwidth compared with EPD and UBR except the cases of (1) the small buffer size, (2) the large number of connections and (3) the large propagation delay. As will be shown in the below, performance degradation of ABR is mostly caused by inappropriate setting of control parameters. We then modify the control parameters by applying two conditions obtained in the previous section. Those imply opposite directions regarding RDF and AIR. Then, there exist cases where no control parameter set is met for the above conditions. Such a case occurs when the propagation delay is large as will be shown in the below.

1) Effective Throughput:
Our first investigation is to improve the effective throughput. We then show that as a side effect, the fairness among connections can also be improved. We examine the values of increase and decrease rates of ACR under the condition that the propagation delay can be estimated a priori, which is implemented by the ABR service [2].

We first use 300 Kbyte as large buffer size, and the propagation delay is set to 2.0 ms. In this case, we have the following parameters for satisfying two conditions: $AIRF = 1/128$ and $RDFF = 2$. $AIRF$ is set to be large compared with the previous case. The throughput is increased from 0.9178 (Fig. 5(a)) to 0.9588. The reason is that the corresponding number of lost packets is decreased to 0 from 8,464. Furthermore, the fairness, which has not been achieved in the case without tuning, can be improved. In Fig. 9, we show the time dependent behavior of the queue length at the switch buffer and the number of successfully transmitted packets. Compare with Fig. 8 for EPD and ABR without parameter tuning.

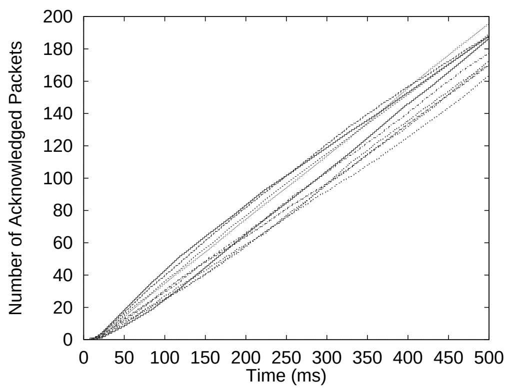
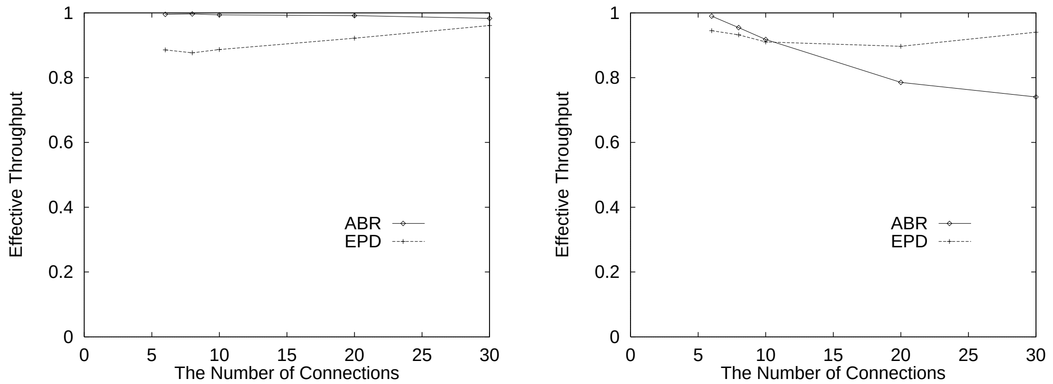


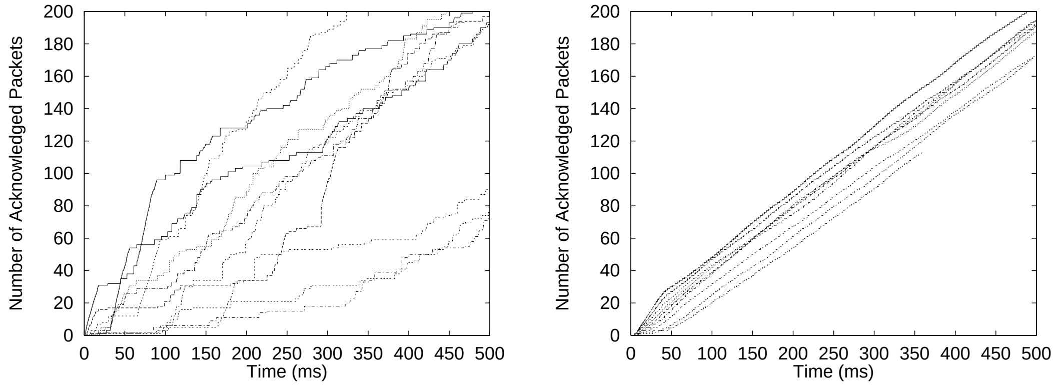
Fig. 9: Time dependent behavior of ABR with parameter tuning: $\tau = 2.0$ ms, $N_{VC} = 10$, buffer size = 300 Kbyte.

One problem is that we could not find the parameters which satisfy both conditions in the case of the small buffer size (10 Kbyte). Allowing "under utilization", but inhibiting the cell loss leads to parameters $AIRF = 1/256$ and $RDFF = 2$. The result was that while the number of lost packets is decreased to 1,163 from 62,953, the throughput was unexpectedly decreased to 0.435 from 0.458. (EPD achieved the throughput of 0.853 in Fig. 5(a).) One favorable point is that it can achieve the fairness among connections as shown in Fig. 10. In the figure, the case of EPD is also shown for comparison purpose. We next change the parameters so that the condition of no under utilization is fulfilled, but the cell loss is allowed. By this settings, we expect the ability of TCP packet retransmissions to improve



(a) Effective throughput for $\tau = 0.02$ ms. (b) Effective throughput for $\tau = 2.0$ ms.

Fig. 6: Comparisons of EPD and ABR methods as a function of the number of connections N_{VC}: buffer size $= 300$ Kbyte.



(a) EPD. (b) ABR.

Fig. 7: The number of successfully transmitted packets as a function of time: $\tau = 0.02$ ms, $N_{VC} = 10$, buffer size $= 300$ Kbyte.

the performance. If we use parameters $AIRF = 1/256$ and $RDFF = 4$, the throughput is improved to 0.542 at the expense of the fairness degree as shown in Fig. 11.

The last experiment in this subsection is parameter tuning dependent on the number of connections N_{VC}. As has been shown in Fig. 6(b), the throughput of ABR is degraded by the larger number of connections when we use fixed values of control parameters. However, it can be avoided. For example, when $N_{VC} = 30$, the parameters should be $AIRF = 1/128$ and $RDFF = 4$ for the case of 2.0 ms propagation delay and 300 Kbyte buffer size. That is, the total increase rate is reduced. Then, the throughput is dramatically improved from 0.740 (Fig. 6(b)) to 0.995, which outperforms even the EPD case (0.931). The reason is that the number of lost packets is decreased from 34,862 to only 59 during simulation run.

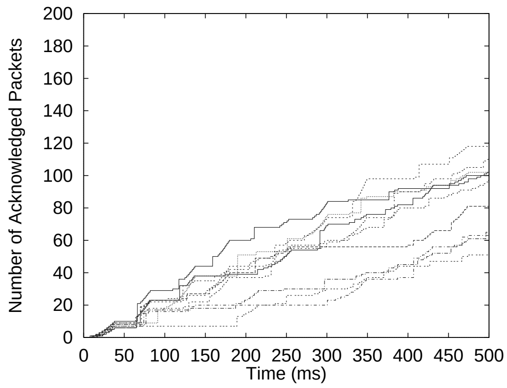
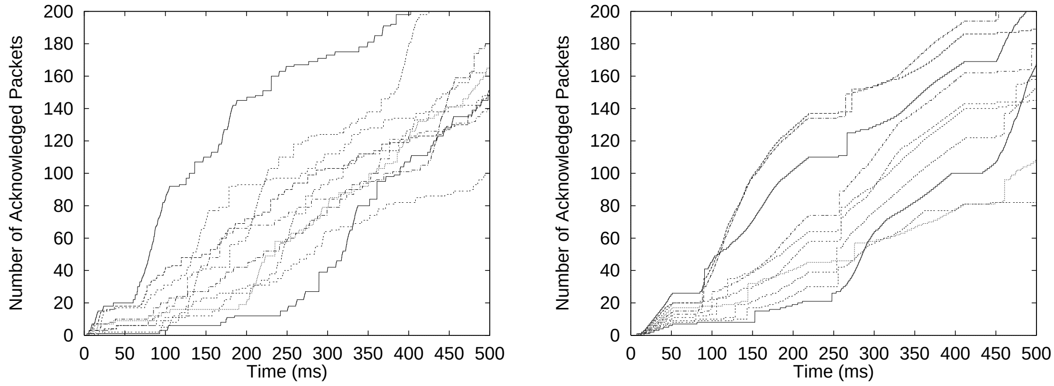


Fig. 11: The number of successfully transmitted packets as a function of time in ABR with parameter tuning: $\tau = 2.0$ ms, $N_{VC} = 10$, buffer size $= 10$ Kbyte, $AIRF = 1/256$ and $RDFF = 4$.

IV. CONCLUDING REMARKS

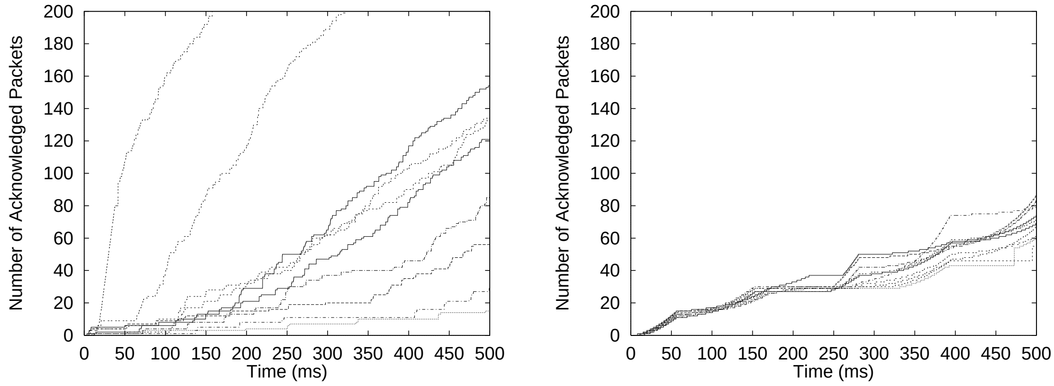
In this paper, we have analyzed the rate-based congestion control by utilizing a first-order fluid approximation. We have also derived appropriate settings of control parameters — including source end system parameters and switch parameters — in order to fulfill two objectives: no buffer overflow at the switch and full link utilization of the bottleneck link. The source end system negotiates with the network at connection establishment time. Those includes N_{RM}, AIR and RDF in a form of P-Vector as well as ICR, X_{RM}. The latter two parameters are necessary for initial cell transmission. We have already examined the necessary conditions that those parameters should fulfill [8] while it is not included in the current paper due to space limitation.

Furthermore, we have investigated performance of three methods: TCP over UBR, TCP over EPD and TCP over ABR. Through simulation experiments, we have shown that TCP over ABR can outperform the other two methods if the control parameters of rate-based congestion control algorithm are chosen carefully. While the duality prob-



(a) EPD. (b) ABR.

Fig. 8: The number of successfully transmitted packets as a function of time: $\tau = 2.0$ ms, $N_{VC} = 10$, buffer size $= 300$ Kbyte.



(a) EPD. (b) ABR.

Fig. 10: The number of successfully transmitted packets as a function of time in EPD and ABR: $\tau = 2.0$ ms, $N_{VC} = 10$, buffer size $= 10$ Kbyte. ABR uses $AIRF = 1/256$ and $RDFF = 2$.

lem of two congestion control mechanisms was observed in some parameter settings, we have also shown that it can be avoided when we can choose control parameters appropriately. Other simulation experiments including investigations on the initial packet transmission have already been available in [11].

For future works, we should evaluate models that multiple switches exist in the network, and/or that each connection have a different propagation delay as a more realistic model.

ACKNOWLEDGMENT

The authors would like to thank Dr. Hiroshi Suzuki and Ms. Chinatsu Ikeda with NEC corporation for their helpful discussions.

REFERENCES

[1] P. Newman, “Traffic management for ATM local area networks,” *IEEE Communications Magazine*, pp. 44–50, August 1994.

[2] The ATM Forum Technical Committee, “Traffic management specification version 4.0,” *ATM Forum Contribution 95-0013R7*, 1995.

[3] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, “Rate-based congestion control for ATM networks,” *ACM SIGCOMM Computer Communication Review*, vol. 25, pp. 60–72, April 1995.

[4] L. Roberts, “Enhanced PRCA (proportional rate-control algorithm),” *ATM Forum Contribution 94-0735R1*, August 1994.

[5] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, “Analysis of rate-based congestion control methods in ATM networks, Part 1: steady state analysis,” *to be presented at GLOBECOM ’95*, 1995.

[6] A. Romanow and S. Floyd, “Dynamics of TCP over ATM networks,” *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 633–641, May 1995.

[7] N. Yin and M. G. Hluchyj, “On closed-loop rate control for ATM cell relay networks,” *IEEE INFOCOM ’94*, pp. 99–109, June 1994.

[8] H. Ohsaki, M. Murata, and H. Miyahara, “Parameter tuning of rate-based congestion control algorithms for ATM networks,” in preparation for submission, 1995.

[9] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Massachusetts: Addison-Wesley, 1994.

[10] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Massachusetts: Addison-Wesley, 1995.

[11] G. Hasegawa, H. Ohsaki, M. Murata, and H. Miyahara, “Performance evaluation and parameter tuning of tcp over abr service in atm networks,” submitted to *IEICE Transactions on Communications*, 1995.