Stability of Rate-Based Congestion Control Algorithm for ABR Service Class in ATM Networks

Hiroyuki Ohsaki, Masayuki Murata and Hideo Miyahara

Department of Informatics and Mathematical Science
Graduate School of Engineering Science, Osaka University
1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan

oosaki@ics.es.osaka-u.ac.jp

Abstract — A rate-based congestion control algorithm is a feedback-based flow control mechanism for ABR (Available Bit Rate) service class, which is suitable for data transfer applications. The rate-based congestion control algorithm dynamically regulates a cell transmission rate of a source end system according to feedback information from the network. An appropriate choice of control parameters can lead to effective use of network resources. However, once it fails, the rate-based congestion control unexpectedly exhibits unstable operation under some parameter settings, which leads to dramatic degradation of the performance. In this paper, we investigate causes of unstable operation and performance degradation of the rate-based congestion control. We then propose an improvement in the existing rate-based congestion control algorithm. Its effectiveness is demonstrated by providing several simulation experiments.

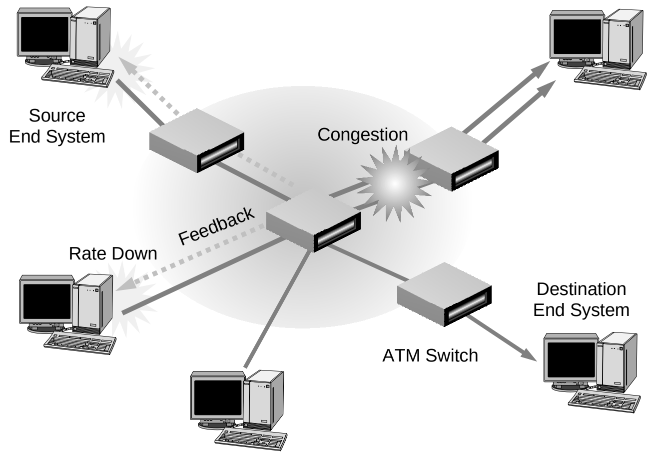


Figure 1: Rate-based congestion control algorithm.

1 Introduction

A rate-based congestion control algorithm is a feedback-based flow control mechanism suitable for data transfer applications. In the rate-based congestion control algorithm, cell emission rate of each source end system is regulated according to congestion information returned by the network. The ATM Forum has adopted it as congestion control mechanism for ABR (Available Bit Rate) service class, and has finished its standardization in 1996 [1, 2, 3]. In the standard, behaviors of source and destination end systems (i.e., terminals) have been specified in detail. A congestion notification mechanism from an intermediate switch to each source end system has also been specified. A source end system periodically sends a forward RM (Resource Management) cell per Nrm data cells, and the corresponding destination end system sends it back to the source end system as a backward RM cell. A switch should notify its congestion to source end systems by setting an EFCI (Explicit Forward Congestion Indication) bit in a data cell header or a CI (Congestion Indication) bit in an RM cell. Since it uses one-bit information for congestion notification, this type of switch is often referred to as a *binary-mode* switch. In the standard, a switch is allowed to explicitly designate a cell transmission rate of a source end system by decreasing an ER (Explicit Rate) value in forward and/or backward RM cell. A switch utilizing the ER value for congestion notification is often referred to as an *explicit-rate switch*.

The congestion notification method from a switch to source end systems has been standardized by the ATM Forum. However, detailed algorithms of binary-mode and explicit-rate switches are not standardized. Namely, their algorithms — when a binary-mode switch sets an EFCI bit or a CI bit, and how an explicit-rate switch changes an ER field — are left to switch manufacturers. Effectiveness of the rate-based congestion control algorithm is therefore heavily dependent on a design of the switch algorithm. In [4, 5], it has been shown that a typical binary-mode switch works effectively in a LAN environment of short propagation delays. On the other hand, an explicit-rate switch has a potential to achieve high performance even in a WAN environment of significant propagation delays due to its ability to directly specify cell transmission rates of source end systems [6, 7]。

There have been a lot of research on the rate-based congestion control algorithm. Since most of commercially available switches support only binary-mode operation (i.e., only an EFCI bit of a data cell header can be changed), many performance evaluations of the rate-based congestion control algorithm with binary-mode switches have been made in the literature. In [5], we have shown that its performance is determined by a choice of control parameters such as RIF (Rate Increase Factor) and RDF (Rate Decrease Factor). Unless these parameters are configured appropriately, performance of the rate-based congestion control algorithm is severely degraded even in a LAN environment. In [5], we have analytically derived the appropriate values of RIF and RDF, which are dependent on various system parameters: the number of active connections, propagation delays, and a buffer size of a switch. It means that control parameters of a source end system should be changed as network condition like the number of active connections changes. However, those parameters are to be negotiated with the network at a connection setup time and cannot be changed afterwards. Thus, initially configured control parameters might become inappropriate and result in performance degradation of the rate-based congestion control algorithm.

In the rate-based congestion control algorithm with binary-mode switches, a cell transmission rate of a source end system oscillates with regularity because a binary-mode switch uses only one-bit information and the propagation delay from the switch to the source end system is non-negligible. In [8], Pecelli et. al have analytically shown the existence of unstable operation in their feedback-based rate control mechanism. Namely, the cell transmission rate does not get stabilized indefinitely in some conditions. Once the operation of the congestion control algorithm becomes unstable, it would cause several problems; fairness among connections is deteriorated, and QoS (Quality of Service) of ABR service class becomes unpredictable. However, their rate control mechanism is different from the rate-based congestion control algorithm standardized by the ATM Forum so that it is unknown whether similar phenomenon occurs in the rate-based congestion control algorithm.

In this paper, we first confirm that unstable operation found in [8] also appears in the rate-based congestion control algorithm specified by the ATM Forum. We find that it actually does by changing some parameters (e.g., control parameters of a source end system, a buffer size at the bottleneck switch, and a propagation delay of each connection), and investigate its causes. We finally propose an improvement in the existing rate-based congestion control algorithm to solve its problems, and demonstrate its effectiveness.

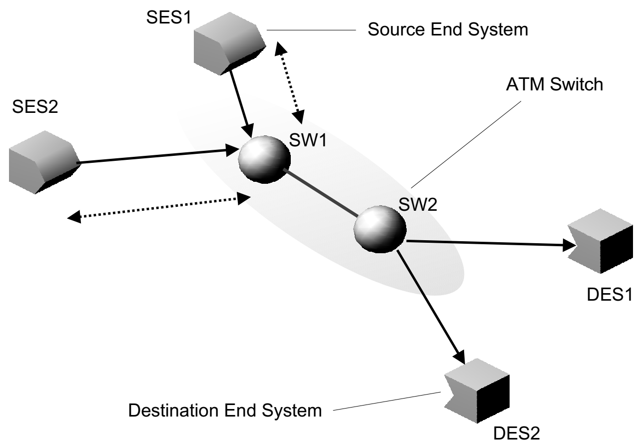


Figure 2: Simulation model.

The rest of this paper is organized as follows. In Section 2, we describe our simulation model and explain performance measures we use throughout this paper. In Section 3, we investigate causes of performance degradation of the rate-based congestion control algorithm. In Section 4, we propose an improvement in the existing rate-based congestion control algorithm to solve its defects, and show its effectiveness through simulation. Finally, we summarize this paper with a few remarks in Section 5.

2 Simulation Model

Figure 2 illustrates our simulation model, which consists of two source–destination end system pairs and two binary-mode switches. The bottleneck resides in the switch 1 (SW1) in this model. To clearly evaluate the effect of a difference in propagation delays, only two connections with different propagation delays are considered. The propagation delay from the source end system 1 (SES1) to the switch 1 (SW1) (denoted by τ_1) is changed from 0.1 ms (about 20 km) to 2.0 ms (about 400 km) while the SES2–SW1 propagation delay (denoted by τ_2) is fixed at 0.1 ms. All of SW1–SW2, SW2–DES1, and SW2–DES2 propagation delays are fixed at 0.1 ms.

In our simulation, a source end systems always has cells to transmit; that is, it transmits cells at its given ACR (Al-

Table 1: Control parameters of source end systems.

PCR (Peak Cell Rate)	BW
MCR (Minimum Cell Rate)	PCR/1000
ICR (Initial Cell Rate)	PCR/10
TCR (minimum rate for data cells)	0.01
RIF (Rate Increase Factor)	1/64 or 1/32
RDF (Rate Decrease Factor)	1/16 or 1/8
Nrm (RM cell opportunity)	32
Mrm (control cell allocation)	2
Trm (minimum interval of RM cells)	100
TBE (Transient Buffer Exposure)	2^{24}
Crm (# of RM cells without control)	32000
CDF (Cutoff Decrease Factor)	1/2
ADTF (ACR Decrease Time Factor)	0.5 ms

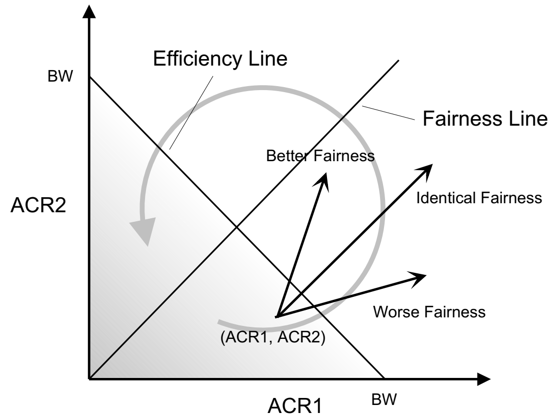


Figure 3: Trajectory of (ACR1, ACR2).

lowed Cell Rate) at any time. A buffer size of a binary-mode switch is denoted by BL, and set to either 30 Kbytes (579 cells) or 300 Kbytes (5,796 cells). A threshold value of the switch buffer, which is used to detect congestion in the switch, is fixed at the half of the buffer capacity (i.e., BL / 2). A bandwidth of each transmission link, BW, is set to 150 Mbps (353.7 cell/ms). Settings of other control parameters are summarized in Table 1.

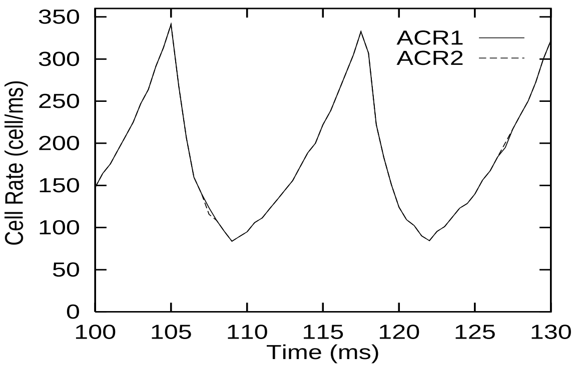
In this paper, we focus on dynamics of cell transmission rates and the queue length (i.e., the number of cells queued in the switch buffer). In addition, we evaluate fluctuation of fairness between two connections (i.e., change of a difference in cell transmission rates of two source end systems). For this purpose, we plot a trajectory of (ACR1, ACR2) where ACRi (i = 1, 2) denotes the cell transmission rate of the source end system i [9] (see Fig. 3). By tracing (ACR1, ACR2) on this graph, fairness between two connections can be observed clearly. When (ACR1, ACR2) is on the line of ACR1 = ACR (*fairness line*), fairness between two source end systems is completely achieved. Namely, if the cell transmission rates of two end systems are changed from (ACR1, ACR2) to (ACR1 + Δ, ACR2 + Δ), the trajectory moves in parallel with the fairness line, and it means that the identical fairness is preserved. If (ACR1, ACR2) is on the line of ACR1 + ACR2 = BW (*efficiency line*), the transmission link is fully utilized.

The ideal operation of the rate-based congestion control algorithm is, therefore, to stabilize (ACR1, ACR2) at the intersection point of the fairness line and the efficiency line, i.e., $(ACR1, ACR2) = (BW/2, BW/2)$. Since the binary-mode switch uses only one-bit information and the propagation delay is not negligible, oscillation of cell transmission rates is unavoidable. Hence, the ideal operation of the rate-based congestion control algorithm with binary-mode switches is actually that values of (ACR1, ACR2) oscillate around the point of (BW/2, BW/2) on the fairness line. In what follows, we will investigate the dynamics of the rate-based congestion control algorithm with binary-mode switches using this trajectory graph.

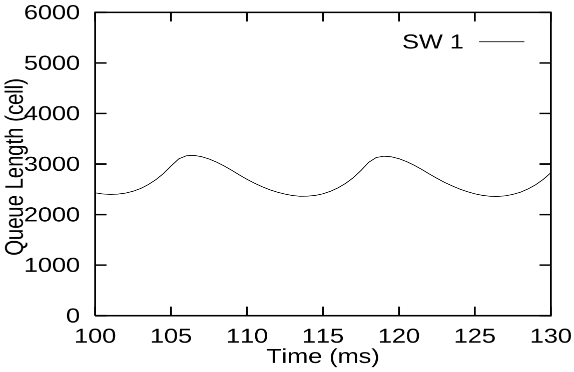
3 Cause of Performance Degradation and Unstableness

In Fig. 4, we first show the simulation result in the case of identical propagation delays of two connections (τ_1 = τ_2 = 0.1 ms). In this figure, the buffer size of each switch, BL, is set to 300 Kbytes, and control parameters of each source end system, RIF and RDF, are set to be appropriate values (1/64 and 1/16, respectively) based on our analytic results in [5]. In this figure, we illustrate (a) fluctuations of cell transmission rates (i.e., ACR1 and ACR2), (b) the fluctuation of the queue length at the SW1's buffer, and (c) the trajectory of (ACR1, ACR2). In the figure, we depict simulation results from 100 ms to 130 ms to eliminate the effect of the initial state. In Fig. 4(c), (ACR1, ACR2) is plotted for each 0.5 ms, and (ACR1, ACR2) moves in a counterclockwise direction in all cases.

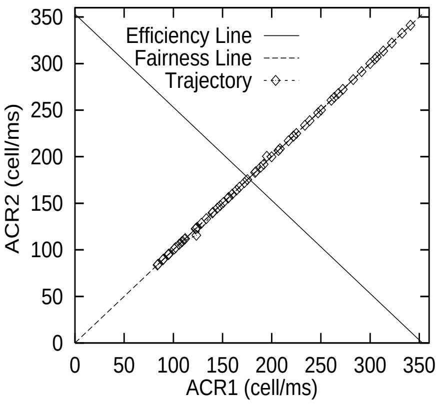
It can be found from Fig. 4(a) that cell transmission rates of source end systems are almost equivalent, and oscillate regularly, showing a stable operation. Figure 4(b) shows the queue length is also in stability around the congestion-detection threshold value (2898 cells in this case), indicating the transmission link is fully utilized. Figure 4(c) tells that the trajectory of (ACR1, ACR2) lies on the fairness line so that the operation of the rate-based congestion control algorithm is mostly ideal. Therefore, when propagation de-



(a) Cell rate



(b) Queue length



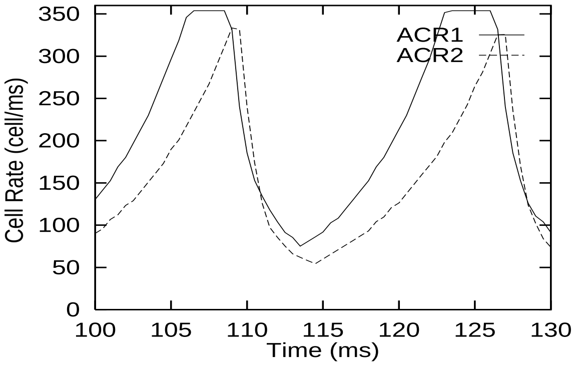
(c) Trajectory of (ACR1, ACR2)

Figure 4: Case of identical propagation delays for τ_2 = 0.1 ms, RIF = 1/64, and RDF = 1/16.

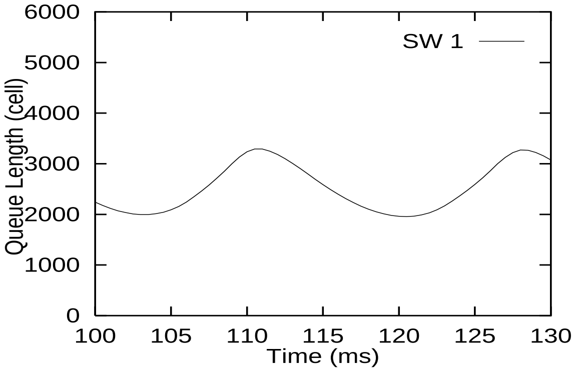
lays of connections are equivalent and control parameters are chosen appropriately, the rate-based congestion control algorithm works quite effectively.

As a difference in propagation delays becomes large, however, the rate-based congestion control algorithm achieves less fairness as shown in Fig. 5. In this figure, all parameters are same with those of Fig. 4 except τ_2 is changed from 0.1 ms to 1.0 ms. Figures 5(a) and 5(c) show degradation of fairness between two connections; ACR1 and ACR2 are different and the trajectory of (ACR1, ACR2) is not on the fairness line. In particular, Fig. 5(c) clearly illustrates that the difference in cell transmission rates becomes large (i.e., fairness is degraded) when source end systems increase their cell transmission rates. This problem is caused by the fact that a source end system sends a forward RM cell per Nrm data cells; that is, a cell transmission rate is increased exponentially rather than linearly as originally expected [1]. In the standard algorithm, when a switch detects relief of congestion (i.e., the queue length goes below the threshold value), it does not set an EFCI bit of a data cell, which is then returned to the source end system as a CI-bit-cleared backward RM cell by the corresponding destination end system. Since a connection with a short propagation delay can respond to this feedback more quickly than one with a long propagation delay, its cell transmission rate becomes slightly larger. On the other hand, a receiving rate of backward RM cells at a source end system is basically determined by its previous generation rate of forward RM cells. A source end system transmits a forward RM cell per Nrm data cells, which means its receiving rate of backward RM cells is approximatelly proportional to its cell transmission rate. Hence, as can be seen in Fig. 5(a), the cell transmission rate is increased exponentially rather than linearly so that the difference in cell transmission rates of two source end systems with different propagation delays become much larger.

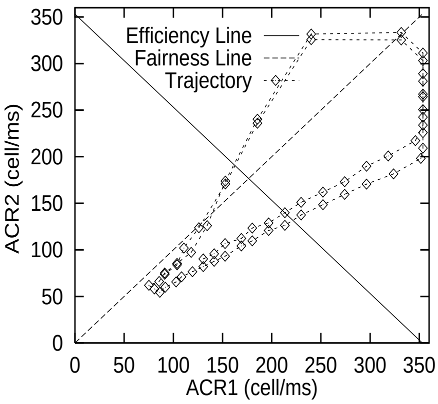
When the capacity of the switch buffer becomes small, fairness among connections is further degraded as shown in Fig. 6. In this figure, the buffer size is changed from 300 Kbytes to 30 Kbytes. As can be found from Figs. 6(a) and 6(c), ACR1 is *always* larger than ACR2. The reason of fairness degration casued by a small buffer size can be explained as follows. When the buffer size is small, the switch buffer sometimes becomes empty (see Fig. 6(b)). It results in a shorter cycle of the queue length oscillation, and therefore a shorter cycle of the cell transmission rate as well. Consequently, as shown in Fig. 6(a), the cell transmission rate of the source end system 1 with a short propagation delay (ACR1) does not reach its peak cell rate (353.7 cell/ms in this case, see Table 1), On the other hand, in the previous case with a large buffer size (Fig. 5), the cell transmission



(a) Cell rate



(b) Queue length



(c) Trajectory of (ACR1, ACR2)

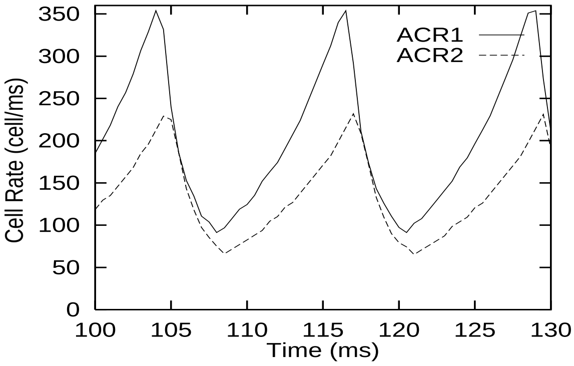
Figure 5: Case of different propagation delays for τ_2 = 1.0 ms, RIF = 1/64, and RDF = 1/32.

rate ACR1 was restricted by PCR. Then, the trajectory of (ACR1, ACR2) moves across the fairness line, which means that ACR2 sometimes becomes larger than ACR1 during ACR1 is restricted by PCR, and that the fairness degradation is limited to some extent on average. In other words, it would be possible that fairness degradation can be improved to some extent by setting PCR for each connection properly even if the buffer size is small. However, it is beyond our scope of the current paper to seek the approprite values of PCR dependenet on propagation delays and the buffer size.

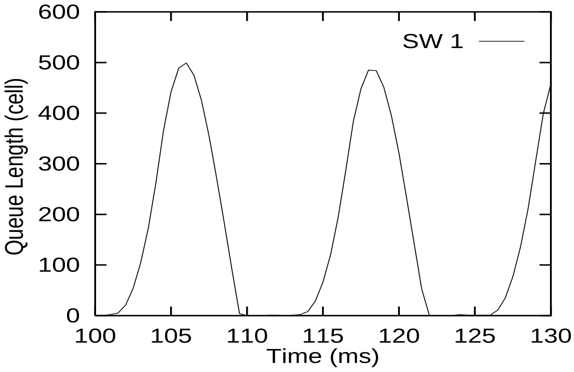
We next show the simulation result for the case of inappropriate control parameters in Fig. 7. By *inappropriate control parameters*, we mean that full link-utilization never be expected by using those parameters for given network parameters [5]. In obtaining this figure, we use control parameters of RIF = 1/32 (fast increase) and RDF = 1/8 (fast decrease) for evaluating the effect of inappropriate control parameters. Other parameters are unchanged from the previous case (Fig. 6). It can be found from Fig. 7(c) that the trajectory of (ACR1, ACR2) becomes larger, and that worse fairness between connections is obtained. It is because rate increase is faster in the current case (i.e., RIF is too large) so that the difference in cell transmission rates becomes much larger. We should note here that since the increase of ACR1 is limited by PCR in this case, fairness degradation is restricted as have explained above. However, fairness would be further deteriorated if ACR1 is not limited by PCR, which would take place when, for example, the number of connections is large. We further note that in the current case, we have intentionally used inappropriate control parameters. However, it is not a unlikely situation in a real network; that is, even if control parameters of a source end system is initially chosen appropriately, it would become inappropriate as network parameters such as the number of active connections changes.

As a difference in propagation delays becomes large, the operation of the rate-based congestion control algorithm becomes unstable as pointed out by [8] where the authors have used a different congestion control algorithm. We show the simulation result for τ_2 = 2.0 ms in Fig. 8. Figure 8(a) shows that the cell transmission rate does not exibit a cyclic behavior as in previous cases and seems to change randomly. Moreover, the trajectory of (ACR1, ACR2) in Fig. 8(c) seems to be irregular.

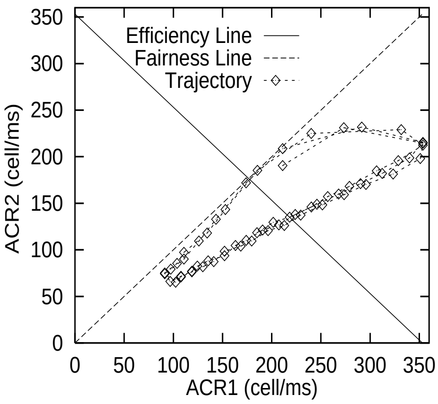
To exhibit unstability of the rate-based congestion control algorithm in Fig. 8, we show the trajectory of (ACR1, ACR2) for a simulation time of 10 s in Fig. 9. It can be found from this figure that some area is completely filled with the trajectory of (ACR1, ACR2), which implies a chaotic behavior of the rate-based congestion control algorithm.



(a) Cell rate

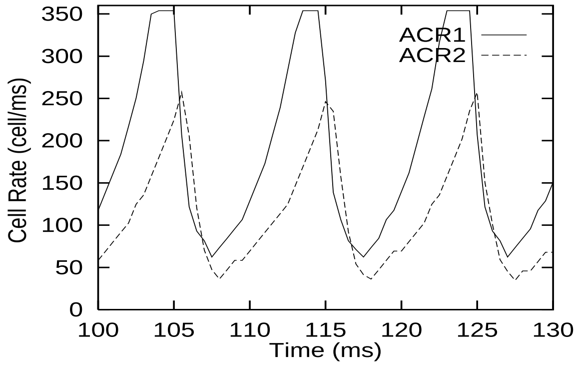


(b) Queue length

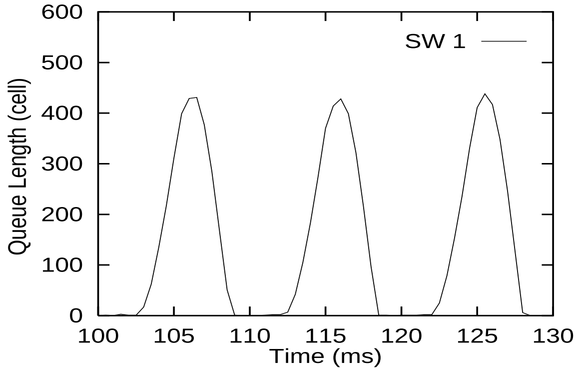


(c) Trajectory of (ACR1, ACR2)

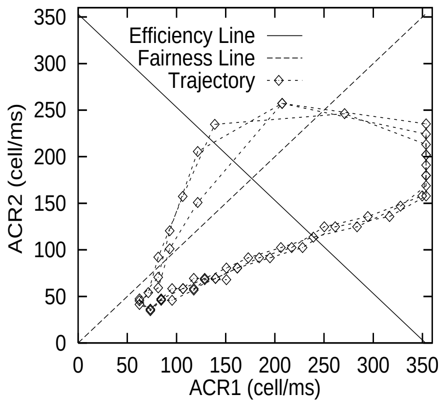
Figure 6: Case of small buffer size for τ_2 = 1.0 ms, RIF = 1/64, and RDF = 1/16.



(a) Cell rate

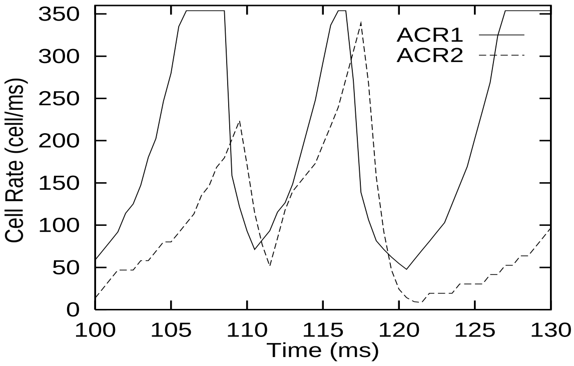


(b) Queue length

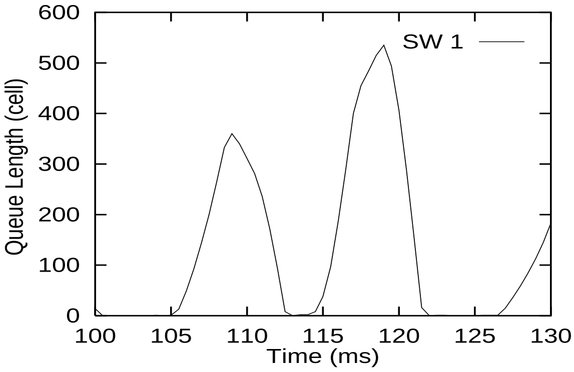


(c) Trajectory of (ACR1, ACR2)

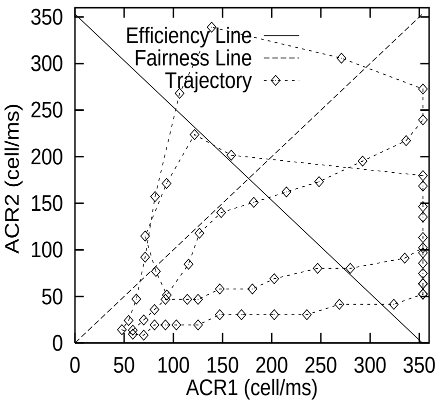
Figure 7: Case of inappropriate control parameters for τ_2 = 1.0 ms, RIF = 1/32, and RDF = 1/8.



(a) Cell rate



(b) Queue length



(c) Trajectory of (ACR1, ACR2)

Figure 8: Case of big difference in propagation delays for τ_2 = 2.0 ms, RIF = 1/32, and RDF = 1/8.

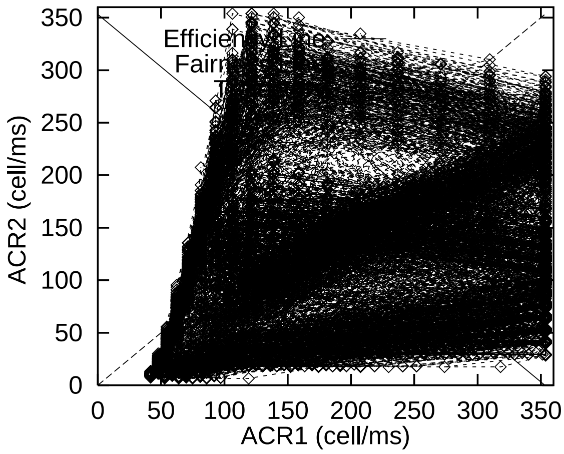


Figure 9: Chaotic behavior of (ACR1, ACR2) for τ_2 = 2.0 ms, RIF = 1/32, and RDF = 1/8.

4 Improvement in the Rate-based Congestion Control Algorithm

As we have shown in the previous section, performance of the rate-based congestion control algorithm with binary-mode switches is significantly degraded when each connection has a different propagation delay and/or a source end system has inappropriate control parameters. It is because (1) an interval of successively transmitted forward RM cells is dependent on the current cell transmission rate, and (2) timing when a source end system changes its cell transmission rate is not synchronized with those of other source end systems due to different feedback delays. The first problem could be solved by using a timer at the source end system to regulate the interval of forward RM cell transmission. The second could be solved by synchronizing rate-change timing at all source end systems by, for example, carrying a delay information, after which the source end system changes its rate, in a backward RM cell [10].

Actually, in the early development process of the rate-based congestion control algorithm in the ATM Forum, the timer-based approach, which uses an interval timer to periodically generate forward RM cell, had been considered. However, a counter-based approach was finally adopted as standard because of its implementation simplicity [11]. In what follows, we propose an improvement in the existing rate-based congestion control algorithm to regulate the interval of forward RM cells without using an interval timer.

The following pseudo-codes explain the basic operation algorithm of a source end system regarding transmission of forward RM cells (sender side) and receipt of backward RM cells (receiver side). For detailed algorithm of a source end system, refer to [1].

Sender Side

```
if now >= time-to-send and data-in-queue
    if (count >= Nrm)
        # send forward RM cell
        send RM (DIR=forward, ...)
        # reset the counter
        count = 0
    else
        # send a data cell
        send data (DIR=forward, ...)
        # increment counter
        count = count + 1
    time-to-send += now + 1/ACR
```

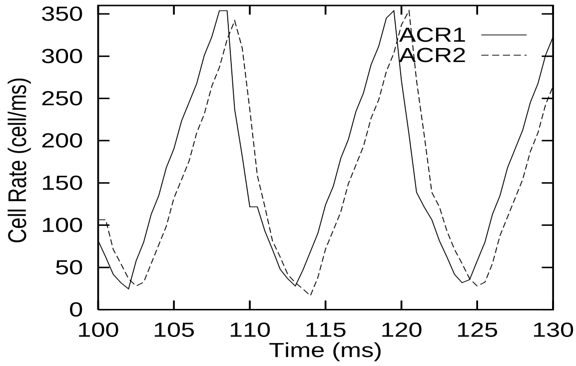
Receiver Side

```
# if backward RM cell is received
if receive RM (DIR=backward, ...)
    # congestion
    if CI = 1
        ACR = ACR - ACR * RDF
    # no congestion
    else
        ACR = ACR + RIF * PCR
        ACR = min(ACR, PCR)
        ACR = min(ACR, ER)
        ACR = max(ACR, MCR)
```

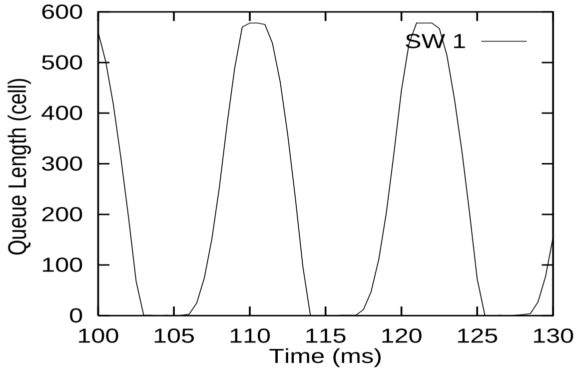
The key idea for regulating the emission interval of forward RM cells is to recompute `count` and `Nrm` at the receiver side whenever `ACR` is changed by receipt of a backward RM cell. By letting `Ts` denote a desired interval of successive forward RM cells, the algorithm of the receiver side is improved as follows.

Receiver Side (Improved)

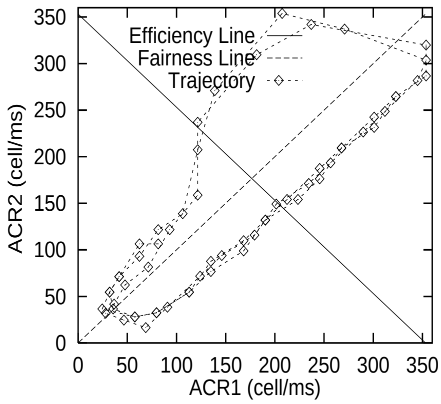
```
# If backward RM cell is received
if receive RM (DIR=backward, ...)
    # Elapsed time from last RM cell
    T = count * (1 / ACR)
    # congestion
    if CI = 1
        ACR = ACR - ACR * RDF
    # no congestion
    else
        ACR = ACR + RIF * PCR
        ACR = min(ACR, PCR)
        ACR = min(ACR, ER)
        ACR = max(ACR, MCR)
    # Recompute Nrm
    Nrm = Ts / (1 / ACR)
    # Adjust timing by changing counter
    count = Nrm - (Ts - T)/(1 / ACR)
```



(a) Cell rate



(b) Queue length



(c) Trajectory of (ACR1, ACR2)

Figure 10: Case of our improved scheme for τ_2 = 2.0 ms, RIF = 1/32, and RDF = 1/8.

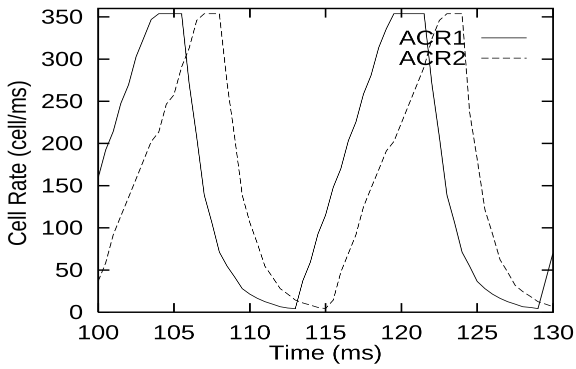
We demonstrate effectiveness of our improvement in Fig. 10, where all parameters are identical to those of Fig. 8 (τ_2 = 2.0 ms, BL = 30 Kbytes, RIF = 1/32, and RDF = 1/8). The unstableness of the rate-based congestion control algorithm observed in Fig. 8 disappears by introducing our improvement. Figure 10(a) indicates that the cell transmission rate increases almost linearly rather than exponentially, by which fairness between connection is significantly improved. Moreover, as shown in Fig. 10(c), the trajectory of (ACR1, ACR2) is parallel to the fairness line when the switch is not congested; that is, fairness between connections is preserved.

Finally, the simulation result for the case of the extremely large propagation delay (τ_2 = 5.0 ms) are shown in Fig. 11. It can be found from this figure that our improvement on the rate-based congestion control algorithm can avoid unfairness between connections even with such a large propagation delay. It can also be found that oscillations of the cell transmission rates and the queue length of the switch buffer are quite stable.

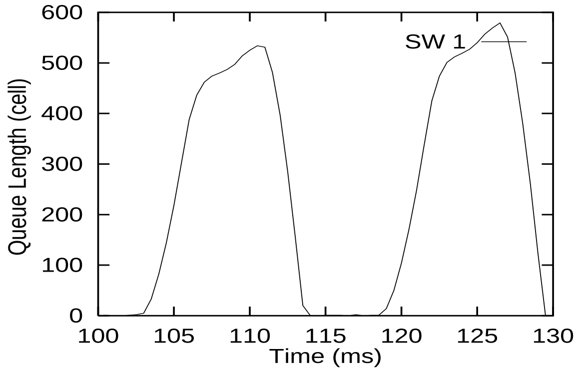
5 Conclusion

In this paper, we have studied performance of the rate-based congestion control algorithm by focusing on the cause of performance degradation. In particular, we have shown the effect of changing control parameters of the source end system, the buffer size at the switch, and the difference in propagation delays of connections. We have also shown that operation of the rate-based congestion control algorithm becomes unstable when the difference in propagation delays of connections is very large. We have then proposed an improvement in the existing rate-based congestion control algorithm to solve its defects. The key idea was to regulate an interval of generating forward RM cells at the source end system. The effectiveness of our improvement was demonstrated in terms of good fairness and stable operation.

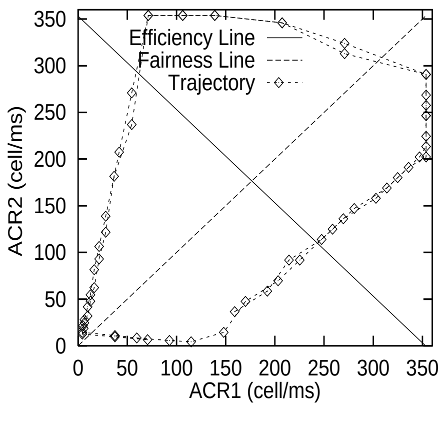
For future works, we should confirm effectiveness of our scheme in more generic network configurations (e.g., with many connections, or with generic network topology). Furthermore, it must be valuable to obtain a condition that the rate-based congestion control algorithm becomes unstable by an analytic approach.



(a) Cell rate



(b) Queue length



(c) Trajectory of (ACR1, ACR2)

Figure 11: Case of our improved scheme for τ_2 = 5.0 ms, RIF = 1/32, and RDF = 1/8.

References

[1] S. S. Sathaye, "ATM forum traffic management specification version 4.0," *ATM Forum Contribution 0056.000*, April 1996.

[2] K. W. Fendick, “Evolution of controls for the available bit rate service,” *IEEE Communications Magazine*, vol. 34, pp. 35–39, November 1996.

[3] Thomas M. Chen, Steve S. Liu, and Vijay K. Samalam, “The available bit rate service for data in ATM networks,” *IEEE Communications Magazine*, pp. 56–71, May 1996.

[4] C. Fang and A. Lin, “A simulation study of ABR robustness with binary-mode swiches: part II,” *ATM Forum Contribution 95-1328R1*, October 1995.

[5] H. Ohsaki, M. Murata, and H. Miyahara, “Parameter tuning of rate-based congestion control algorithm for ABR service class in ATM networks,” to appear in *International Journal of Communication Systems*, 1998.

[6] A. Arulambalam and X. Chen, “Allocating fair rates for available bit rate service in ATM networks,” *IEEE Communications Magazine*, vol. 34, pp. 92–100, November 1996.

[7] H. Ohsaki, M. Murata, and H. Miyahara, “Designing efficient explicit-rate switch algorithm with max-min fairness for ABR service class in ATM networks,” in *Proceedings of IEEE ICC ’97*, pp. 182–186, June 1997.

[8] G. Pecelli and B. G. Kim, “Dynamic behavior of feedback congestion control schemes,” in *Proceedings of IEEE INFOCOM ’95*, pp. 253–260, April 1995.

[9] D.-M. Chiu and R. Jain, “Analysis of the increase and decrease algorithm for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems 17*, pp. 1–14, 1989.

[10] D. H. K. Tsang and W. H. F. Wong, “A new rate-based switch algorithm for ABR traffic to achieve max-min fairness with analytical approximation and delay adjustment,” in *Proceedings of IEEE INFOCOM ’96*, pp. 1174–1181, March 1996.

[11] 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.