# Studies on Congestion Control Schemes for Best-Effort Traffic in ATM Networks

Hiroyuki Ohsaki

January 1997

Department of Informatics and Mathematical Science
Graduate School of Engineering Science
Osaka University

# Abstract

An ATM (Asynchronous Transfer Mode) is a promising technology for realizing B-ISDN (Broadband Integrated Services Digital Network) that can transfer many types of multimedia information. In ATM-based networks, all information is transferred by dividing into fixed-size packets (called as *cells*). The ATM network is a connection-oriented network by virtue of a notion of virtual circuits called VPs (Virtual Paths) and VCs (Virtual Channels). The switch capability can be implemented in hardware because all information is transferred as fixed-size cells, and because cell routing at the switch is rather simple due to connection-oriented communication. For these reasons, the ATM technology is able to realize gigabit-class networks, which had been considered to be difficult with conventional packet-switching technology.

To handle multimedia information effectively, five types of service classes are defined by the ATM standard bodies according to QoS (Quality of Services) requirements of various kinds of applications. At a connection setup, an application has to declare its traffic parameters and negotiates its QoS requirements with the network. However, since most of the existing applications such as data communications generate best-effort traffic, it is usually difficult for applications to predict their traffic patters. Thus, it has been an important issue how to accomodate best-effort traffic into ATM networks. Since best-effort traffic manages to utilize available resources in the network, sophisticated congestion control is required to resolve congestion in the network. In the thesis, we therefore focus on congestion control schemes particularly for best-effort traffic.

Congestion control schemes for ATM networks can be classified into two categories: *internal congestion control* that controls short-time congestion in a switch, and *global congestion control* that regulates cell flow from sources into the network. In the thesis, we first evaluate the performance of an input-output buffered type ATM switch with back-pressure function as an internal congestion control scheme. The back-pressure function prohibits cell transmission from input buffers to the corresponding output buffer to avoid buffer overflow due to temporary congestion in the switch. By using an analytic method, we derive the maximum throughput, the packet delay distribution, and the approximate packet loss probability of such an ATM switch with bursty traffic. In addition to a balanced traffic condition, conditions of unbalanced traffic and mixture of bursty and stream traffic are also analyzed. Through numerical examples, we show the effect of the average packet length and the output buffer size on the performance.

We next focus on a rate-based congestion control algorithm that controls cell emission rates of sources according to feedback information from the network. The rate-based congestion control algorithm is applied to the ABR (Available Bit Rate) service class, and has been extensively developed and standardized by many researchers. Although behavior of source and destination are standardized, operation algorithms of ATM switches are left to manufacturers. Several switch algorithms have been proposed in the ATM Forum. While many studies have been devoted for these switch algorithms in the past, only the ABR service class is taken into account; that is, the effect of VBR and CBR service classes, in which multimedia traffic is accommodated, are not considered. In this thesis, we evaluate the performance of the rate-based

congestion control algorithm when not only ABR traffic but also VBR traffic are incorporated into the network. By using simulation technique, we show drawbacks of these algorithms with coexisting multimedia traffic, and give several suggestions to solve these problems.

In the rate-based congestion control algorithm, several control parameters are defined to control the cell emission process of sources. Effectiveness of the rate-based congestion control algorithm heavily depends on a choice of these control parameters, but a method for determining these control parameters is not standardized in the ATM Forum. In the thesis, we analyze the rate-based congestion control algorithm through applying a first-order fluid approximation in order to provide control parameter tuning. In this analysis, we focus on two conditions that control parameters should satisfy; one is the prevention of buffer overflow at a switch, and the other is full link utilization. For this purpose, we first obtain the maximum and the minimum queue lengths at the switch under the condition where all connections are in a steady state. We next analyze the behavior of a newly established connection by assuming that one or more connections start cell emission while other connections are in a steady-state. Based on this analysis, we discuss settings of initial control parameters to avoid buffer overflow at a switch. Through numerical examples, we demonstrate that our parameter set can satisfy two main objectives — prevention of buffer overflow and full link utilization — in both LAN and WAN environments. Furthermore, proper settings of control parameters in various circumstances are investigated. Namely, we analyze the dynamical behavior of the rate-based congestion control algorithm when each connection has a different propagation delay. We also evaluate the effect of CBR traffic on the rate-based congestion control algorithm. Simulation results for a multi-hop network configuration are presented to exhibit the tradeoff among cell loss probability, link utilization and fairness. The selection method of control parameters in the multi-hop network is then proposed based on both analytic and simulation results.

At the end of this thesis, we focus on a more complicated switch algorithm called as an *explicit-rate switch*, which computes an appropriate cell rate for each source instead of simply marking a congestion indication bit. While implementation is rather complex, an explicit-rate switch has a potential to obtain much better performance than simpler switch algorithms. We therefore discuss design criteria of an explicit-rate switch for achieving high performance in terms of throughput, cell loss probability, fairness and so on. We propose our explicit-rate switch algorithm that meets these design criteria, and evaluate its performance through simulation experiments.

# Acknowledgments

I would like to express my sincere appreciation to Prof. Hideo Miyahara, my adviser, for his innumerable help and continuous support. I am heartily grateful to Prof. Mamoru Fujii and Prof. Nobuki Tokura for serving as readers of my thesis committee. Their expertise and insightful comments have been priceless.

I am most grateful to Prof. Masayuki Murata for his enthusiasm in teaching and pursuing congestion control in high-speed networks with me. He has been actual advisor and been opening my eyes toward congestion control. His active interest and encouragement have been of great help in furthering my efforts in this area, and his standards of excellence will stay with me throughput my research career.

I would like to extend my appreciation to Prof. Tohru Kikuno, Dr. Hiroshi Suzuki, Dr. Chinatsu Ikeda for their gracious help and advice with many important things. Their kindness and efforts on my behalf were invaluable and I am forever in debt.

I am heartily thankful to my friends in the department for their inciting discussions, fellowship and underpinning — special thanks to Dr. Kenichi Baba, Dr. Naoki Wakamiya and Hiroaki Harai for their expert suggestions as well as warmheartedness.

I dedicate this thesis to my parents who have loved and supported me thus far.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 ATM Networks

An ATM (Asynchronous Transfer Mode) is a promising technology for realizing B-ISDN (Broadband Integrated Services Digital Network) that can transfer many types of multimedia information [1]. Many efforts of researchers, developments and standardization have been extensively devoted to the ATM technology [2, 3, 4, 5]. In ATM-based networks, all multimedia information is transferred by dividing into fixed-size packets (called as *cells*). Each cell consists of 5 octets of a header and 48 octets of a payload (segmented information). One of the distinctive features of the ATM network in comparison with conventional packet-switching networks such as Ethernet / FDDI / X.25 networks is that all information is packetized into fixed-size cells of a single format. Moreover, the ATM network is a connection-oriented network by virtue of a notion of virtual circuits called VPs (Virtual Path) and VCs (Virtual Channels). Each connection is assigned its unique VP and VC identifiers called as a VPI (Virtual Path Identifier) and a VCI (Virtual Channel Identifier). At the ATM switch, cells are switched based only on VPI / VCI so that an overhead for interpreting cell headers is greatly reduced. Consequently, switching capability can be implemented in hardware because all information is transferred in fixed-size cells, and because cell routing at the switch is rather simple because connection-oriented communication. For these reasons, the ATM technology can realize gigabit-class networks, which had been considered to be difficult with conventional packet-switching technology.

Owing to connection-oriented operation using VP / VC, ATM networks can guarantee different QoS (Quality of Service) requirements of applications (e.g., end-to-end cell delay variation (CDV) and cell loss ratio (CLR)). Actually, different applications require different QoS's. For example, an application like real-time video transmission must be quite sensitive to jitters in cell transmission delays. For such an application, delayed cells are meaningless so that the application requires strict service quality in terms of the end-to-end cell delay. However, it may be tolerable to some cell losses if it adopts video encoding methods like MPEG [6, 7]. On the contrary, an application like data communication is usually not sensitive to cell delays experienced in the network. However, if one ore more cells of an upper-layer PDU (Protocol Data Unit) are lost in the network, these cells (or entire packet) must be retransmitted. Therefore, cell loss ratio (CLR) should be very low for these applications.

To handle many types of multimedia information effectively, five types of service classes are defined by the ATM standard bodies according to QoS (Quality of Services) requirements of various applications [2, 3]. At a connection setup, an application has to declare its traffic parameters and negotiate its QoS requirements with the network. In each service class, different traffic parameters and QoS parameters are used as summarized in Table 1.1. In what follows, we will explain these five service classes more specifically.

<table>
<tr><th>Attribute</th><th>CBR</th><th>rt-VBR</th><th>nrt-VBR</th><th>UBR</th><th>ABR</th></tr>
<tr><td>— Traffic Parameters —</td><td colspan="5"></td></tr>
<tr><td>PCR, CDVT</td><td colspan="5">specified</td></tr>
<tr><td>SCR, MBS, CDVT</td><td>N / A</td><td colspan="2">specified</td><td colspan="2">N / A</td></tr>
<tr><td>MCR</td><td colspan="4">N / A</td><td>specified</td></tr>
<tr><td>— QoS Parameters —</td><td colspan="5"></td></tr>
<tr><td>peak-to-peak CDV</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>maximum CTD</td><td colspan="2">specified</td><td colspan="3">unspecified</td></tr>
<tr><td>CLR</td><td colspan="3">specified</td><td>unspecified</td><td>specified</td></tr>
<tr><td>— Other Parameters —</td><td colspan="5"></td></tr>
<tr><td>Feedback</td><td colspan="4">unspecified</td><td>specified</td></tr>
</table>

Table 1.1: Service Classes in ATM Networks

- Constant Bit Rate (CBR) Service Class

  The CBR service class is expected to guarantees that a static amount of bandwidth is always available to an accepted connection. At a connection setup, an application using the CBR service class (CBR connection) negotiates its QoS with the network. Only PCR (Peak Cell Rate) and CDVT (Cell Delay Variation Tolerance) are used as the traffic parameters. Once this connection is accepted by a CAC (Call Admission Control) of the network, it can always emit cells at the rate not larger than the negotiated PCR. As long as it does not send cells more than PCR, cell delay variation (CDV), a maximum of CTD (Cell Transfer Delay) and CLR (Cell Loss Ratio) must be guaranteed. The CBR service class is, therefore, suitable for applications having strict QoS requirements in cell delay and cell loss.

- Real-Time Variable Bit Rate (rt-VBR) Service Class

  The rt-VBR service class tries to utilize the network resources efficiently by multiplexing several connections generating bursty traffic. In the rt-VBR service class, CDV is also assured. An application using the rt-VBR service class (rt-VBR connection) must declare its traffic parameters in terms of PCR, CDVT, SCR (Sustainable Cell Rate) and MBS (Maximum Burst Size). Any real-time application generating bursty traffic should be classified into this service class. However, it is difficult to perform the call admission control for the rt-VBR service class. More research is still required for implementing the rt-VBR service class in reality.

- Non Real-Time Variable Bit Rate (nrt-VBR) Service Class

  The nrt-VBR service class obliges an application using this service class to declare its traffic parameters in terms of PCR, CDVT, SCR and MBS as with the rt-VBR service class. It differs from the rt-VBR service class in that neither CDV nor the maximum of CTD is guaranteed.

- Unspecified Bit Rate (UBR) Service Class

  No QoS is guaranteed in the UBR service class. Namely, no traffic management is performed in the ATM layer for the UBR service class. A typical application of the UBR service class is an application currently used in conventional networks such as IP networks (e.g., file transfer and remote terminals).

- Available Bit Rate (ABR) Service Class

> The ABR service class guarantees only CLR. An application using the ABR service class has to negotiate its traffic parameters in terms of PCR and MCR (Minimum Cell Rate). After the connection is accepted, its cell emission rate is dynamically controlled by the feedback information from the network. In other words, the ABR service class tries to utilize the unused bandwidth by other service classes. Note that the ABR connection should be able to use the bandwidth of MCR at any time. Similarly to the UBR service class, the ABR service class is suitable for applications like file transfer and computer communications.

Currently, these service classes were well defined, and frameworks for providing these service classes in real networks have been promptly developed and standardized by many researchers. However, there still remains a number of issues to be solved in this area. For example, each service class has been independently developed by different researchers. Hence, the effect of co-existence of different service classes in the network has not been fully discussed. Besides, most of the existing applications are classified as *best-effort traffic*; That is, these applications utilize as much bandwidth as the network can provide. In other words, these applications dynamically share the available bandwidth in the network. Therefore, QoS requirements will be degraded due to congestion in the network unless adequate congestion control for best-effort traffic is provided [8, 9, 10, 11], which is the main objective of this thesis. In what follows, we introduce several congestion control schemes for best-effort traffic in ATM networks.

## 1.2 Congestion Control for Best-Effort Traffic

Congestion control schemes for ATM networks can be classified into two categories: *internal congestion control* that controls short-time congestion in an ATM switch, and *global congestion control* that regulates cell flow from source end systems (terminals) into the network. In this section, we introduce these two types of congestion control schemes.

### 1.2.1 Internal Congestion Control of the Switch

Internal congestion control resolves congestion that occurs at a switch in a short time interval. For example, if multiple cells destined for the same output link arrive at a switch simultaneously, the output link is temporally congested. In this case, cells should be queued at the switch buffer to avoid cell loss. However, the buffer capacity is actually limited so that some mechanism is needed to reduce the opportunity of buffer overflow.

As an internal congestion control scheme, several types of ATM switch architecture have been proposed including output buffer switch, input buffer switch, shared buffer switch, batcher banyan switch [12, 13]. These switches have tradeoffs between performance and implementation complexity. For example, output buffer switch shows better performance than other switches (e.g., high throughput and low cell loss probability) if all switches have a fixed amount of buffer memory. However, since output buffer switch requires memory chips of faster access speed, it cannot be provided with a large amount of memory due to cost or technology limitation. As a cost-effective ATM switch, Fan *et al.* recently proposed a switch architecture that possesses buffers on both sides of input and output ports with a back-pressure function [14]. The key idea of this switch architecture is to provide a large amount of slow-speed (and inexpensive) memory at input ports and a small amount of fast-speed (and expensive) memory at output ports, and to increase its performance by controlling both input and output buffers with the back-pressure mechanism.

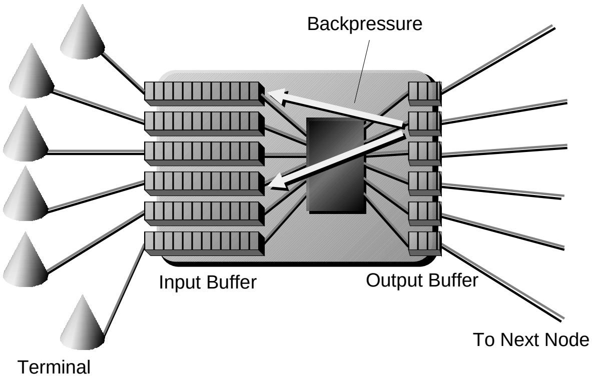

Figure 1.1: ATM LAN Switch with Back-Pressure Function.

The back-pressure function is provided to avoid a temporary congestion in the switch by prohibiting cell transmission from an input buffer to the congested output buffer when the number of cells in the output buffer exceeds a some threshold value (see Fig. 1.1). In this figure, the number of input ports (and output ports) is represented by $N$. This switch is equipped with buffers at both sides of input and output ports. The switching speed of a cell from input buffer to output buffer is $N$ times faster than the link speed; that is, in a time slot, at most one cell at the input buffer is transferred to the output buffer while the output buffer can simultaneously receive $N$ cells from different input buffers. The back-pressure function prohibits transmission of cells from input buffer to output buffer by signaling back from output buffer to input buffer when the number of cells in output buffer exceeds a threshold value [14]. By this control, a cell overflow at output buffer can be avoided. However, it introduces HOL (Head of Line) blocking of cells at input buffer, which results in limitation of the switch performance.

### 1.2.2 Global Congestion Control

On the other hand, global congestion control tries to resolve *network-wide* congestion. Closed-loop rate control is a promising global congestion control mechanism for data communications and is being applied to the ABR (Available Bit Rate) service class in the ATM Forum. Closed-loop control is also called as reactive congestion control, and it dynamically regulates cell emission process of each source end system by using feedback information from the network. It is therefore especially suitable for best-effort traffic. For implementation of closed-loop control, two kinds of schemes was proposed in the ATM Forum: *rate-based* and *credit-based*. The credit-based scheme is based on a link-by-link window flow control mechanism [15, 1]. Independent flow controls are performed on each link for different connections, and each connection must obtain buffer reservations for its cell transmission on each link. This reservation is given in the form of a *credit balance*. A connection is allowed to continue cell transmission as long as it gains credit from the next node. When the connection is starved of credit, it should wait for credit. Owing to this link-by-link fast feedback mechanism, transient congestion can be relieved effectively. In addition, no cell loss occurs because no connection can send cells unless it has credit.

The rate-based scheme, on the other hand, controls the cell emission rate of each connection between end systems [3, 16, 17, 18, 19]. It is simpler than credit-based flow control schemes in which each switch requires complicated queue management for every connection. Typical examples of the rate-based approach are forward explicit congestion notification (FECN) and backward explicit congestion notification (BECN) [20], which are well-known congestion control strategies in conventional packet-switching networks. After long discussions, the rate-based congestion control algorithm has been adopted as the standard mechanism for the ABR service class. In this thesis, we focus on the rate-based congestion control algorithm as a global congestion control for best-effort traffic.

In the following section, we describe a historical overview of development and standardization of the rate-based congestion control algorithm in the ATM Forum. Implementation aspects of FECN-like and BECN-like methods in these control schemes are also described. We will also introduce more intelligent schemes in which the switch controls the rate of connections explicitly.

## 1.3 Rate-Based Congestion Control Algorithm

Several proposals had been contributed in the rate-based congestion control framework to the ATM Forum by the end of 1993: the methods based on FECN [21, 22] and the methods based on BECN [20, 23]. The Rate-Based Traffic Management Ad-Hoc working group was then established to discuss various aspects of rate-based congestion control methods. The result, which will be precisely described in the next subsection, was published as an ATM Forum Contribution [24]. The ATM Forum standard regarding traffic management specifies only the source and destination end systems behaviors; the methods for implementing the switches are left to the manufacturers. We will describe here how the behavior of end systems is standardized and how the various switches proposed in the ATM Forum can cooperate with the *standardized* end systems.

### 1.3.1 Interval-Based Approach

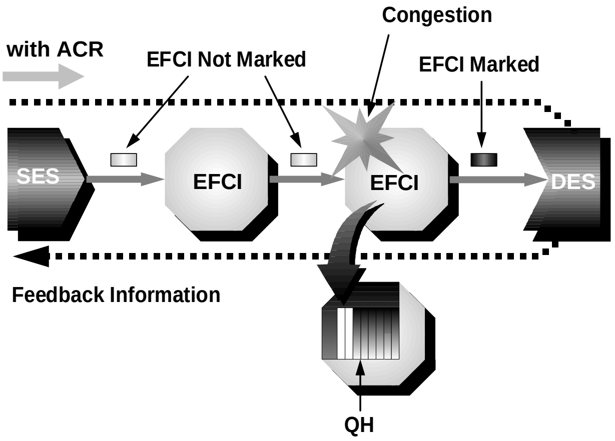

Figure 1.2: Basic Configuration of the Rate-Based Congestion Control.

In this subsection we explain the original rate-based scheme, which was proposed in [24, 25]. Figure 1.2 illustrates a basic configuration of the rate-based congestion control scheme in which the ATM connection is terminated at the source and destination end systems. A permitted cell transmission rate $ACR$ (Allowed Cell Rate) of the source end system is changed according to the congestion status of the network. An initial rate $ICR$, a maximum allowable rate $PCR$, and a minimum cell rate $MCR$ are specified by the network at connection setup time, and the source is then allowed to emit cells at a rate that ranges from 0 to $ACR$. When this scheme is compared with later proposals described in the following subsections, a distinctive point of the original scheme is that the operation of both end systems is based on interval timers. The polarity of the feedback information from the network is *negative*; that is, the source end system receives feedback information only when the network falls into congestion.

An occurrence of congestion is detected at each intermediate switch by monitoring the queue length of the cell buffer. When the queue length exceeds a threshold value ($Q_H$), congestion is signaled to the source by a special cell called an RM (Resource Management) cell, whose Payload Type Identifier (PTI) is "110". The FECN-like signaling mechanism is defined in this scheme. That is, each switch signals its congestion information to its downstream switches by setting an EFCI (Explicit Forward Congestion Indication) bit in the header of passing data cells. When the destination end system receives a data cell in which the EFCI bit is marked, it sends an RM cell back to the source along the backward path. Then the source end system must decrease its $ACR$, multiplicatively according to this feedback information, as

$$ACR \leftarrow max(ACR \times MDF, MCR), \tag{1.1}$$

where $MDF$ is the multiplicative decrease factor and $MCR$ is the minimum cell rate for the $ACR$. A time interval RMI (RM Interval) is defined at the destination end system, and only one RM cell is allowed to be sent in an RMI. The source end system is also provided with an interval timer UI (Update Interval). When the timer expires without an RM cell having been received, the source recognizes no congestion in the network. Then it increases $ACR$ additively as

$$ACR \leftarrow min(ACR + AIR, PCR),$$

where $AIR$ is the additive increase rate and $PCR$ is the peak cell rate of the connection.

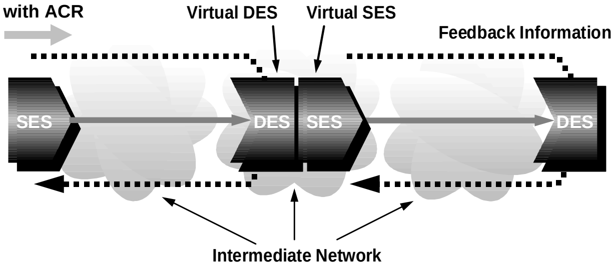

Figure 1.3: Network Segmentation by an Intermediate Network.

As an implementation option, the network can be divided into two or more segments by introducing *intermediate* networks that should act as a *virtual* destination end system for the source and as a virtual source end system for the destination (Fig. 1.3). As a destination end system, an intermediate network has to send RM cells back to the source according to the EFCI status of incoming cells. As a source end system, it is also required to regulate the flow of cells destined for the destination end system.

Since this approach requires interval timers at both end systems, it increases the complexity of implementation and could become expensive. As pointed out in [26], the negative feedback mechanism could cause a collapse of the network in certain conditions. If the network is heavily congested, RM cells can be delayed or lost because of buffer overflow, with the result that timeliness of the congestion information is lost or — in a more serious situation — the source increases its cell emission rate because of the absence of RM cells.

### 1.3.2 Counter-Based Approach

The timer-based approach described in the previous subsection was revised because of its drawbacks, and [27] proposed a proportional rate control algorithm (PRCA) with two major modifications: (1) the polarity of the feedback information is *positive*, and (2) the need for interval timers is eliminated. The origin of the name PRCA is in that opportunities for rate increases are given in proportion to the current sending rate $ACR$. In PRCA the source end system marks the EFCI bit in all data cells except for the first of every $N_{RM}$ cells. The destination end system instantly sends an RM cell back to the source when it receives a cell with the EFCI bit cleared. If the EFCI bit is set by an intermediate switch because of its congestion, the destination takes no action. By this mechanism, a positive congestion signaling is established: receiving the RM cell implies that there is no congestion in the network, and therefore the source end system is given an opportunity to increase its rate.

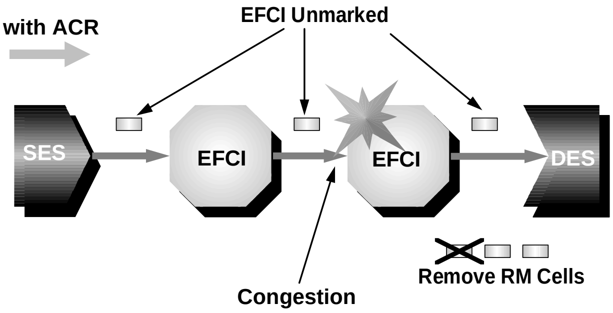

Figure 1.4: BECN-like Congestion Notification in PRCA.

The source end system sends the cells in the following way. Unless receiving an RM cell, the source determines the next cell transmission time at $1/ACR$ after the current time. This implies that the source continuously decreases its $ACR$ (until receiving an RM cell) as

$$ACR \leftarrow \max(ACR - ADR, MCR). \tag{1.2}$$

When the source receives an RM cell, the rate is increased as

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

which compensates the reduced rate since the source received the previous RM cell ($N_{RM}\, ADR$) and increases the rate by $N_{RM}\, AIR$. In an ideal situation with no propagation delay, this should give a linear increase of the cell transmission rate. The network can thus be restored even if heavy congestion results in all RM cells being discarded. This is because the rate is always decreased whenever the source does not receive an RM cell. The above operation provides FECN-like congestion management, but if switches have the ability to discard RM cells in the backward direction, BECN-like operation can also be achieved. This is one of the notable features of PRCA (Fig. 1.4).

Certain problems, however, remain even in PRCA and have been pointed out in [28]. One of them is referred to as an "ACR beat down" problem, and is explained as follows. Each source of active connections that experiences congestion in several switches has less opportunity to receive positive feedback than do sources of other connections with fewer switches. Once one of these relatively feedback-starved sources decreases its transmission rate to the minimum rate $MCR$, it is in some circumstances likely to remain at that rate indefinitely. Thus, fairness among connections cannot be achieved. Another problem is that PRCA requires a considerable amount of buffers when there is a large number of active connections. It is now widely recognized that when the propagation delays are large (as in the WAN environment), the queue length temporarily grows because of the control information delays. This is an intrinsic and unavoidable feature of recent high-speed networks. The problem is, however, that such a long queue length occurs even in a LAN environment. During congestion, the rate is decreased by $ADR$ (Eq. (1.2)). When the number of connections is very large, however, decreasing the aggregated input rate at the switch is too slow. For example, when the control parameters suggested in [27] are used, 241,000 cell buffers are required for assuring no cell loss even in the LAN environment for 1000 active connections [28]. Such a large buffer size is unacceptable given the current memory technology.

### 1.3.3 Enhanced PRCA Method

An improved version of PRCA — called EPRCA (Enhanced Proportional Rate Control Algorithm) — was then proposed in [29, 30]. New functionalities are added as implementation options in two ways. One, to achieve better fairness among connections, is a capability to send a congestion indication to particular sources rather than all sources. The fairness could be achieved if each connection is maintained separately at the switch, which is called *per-VC accounting* (see Subsection 1.3.4 for more detail). However, since it requires an additional control complexity, EPRCA adopts another method “intelligent marking”, which is originated from the work in [31]. The other is the means for reducing the rate of each connection explicitly; that is, the switch can have a responsibility for determining the cell transmission rate of selected connections. While some modifications were required in order to incorporate these new features, EPRCA preserves a backward compatibility with PRCA. A switch supporting only PRCA can thus also be used in an EPRCA-based network. For distinguishing this switch from other new switches, it is called an EFCI bit setting switch.

EPRCA requires forward RM cells as well as backward RM cells. RM cells contain a CI (Congestion Indication) bit that is used to carry congestion information to the source. Instead of unmarking an EFCI bit of data cells as PRCA does, the source end system periodically sends a forward RM cell every $N_{RM}$ data cells. When the destination end system receives the forward RM cell, it returns the RM cell to the source as a backward RM cell. When doing this, the destination end system sets the CI bit of the backward RM cell according to the EFCI status of the last incoming data cell. The source end system can thus be notified of the congestion detected at the intermediate switches by marking the EFCI bit of data cells in the forward path. This is a FECN-like implementation of congestion notification. Furthermore, the switch can be allowed to set a CI bit of backward RM cells as a BECN-like implementation.

The two major enhancements of EPRCA — *intelligent marking* and *explicit rate setting* — require additional information fields in each RM cell: $CCR$ (Current Cell Rate) and $ER$ (Explicit Rate) fields. An $ER$ element is used to decrease the source rate explicitly, and is initially set to $PCR$ by the source. One or more congested intermediate switches can change it to a lower value so that the rate of the source end system is rapidly decreased for quick congestion relief. The $CCR$ element is set to the current $ACR$ of the source in effect, and a fair distribution of the bandwidth can be achieved with these two values. The intermediate switch selectively signals congestion indication to the sources with larger $ACR$ values. A more impartial sharing of the available bandwidth can be achieved by using this intelligent marking mechanism in conjunction with the explicit rate setting mechanism.

Three types of switch architecture with different functions are suggested in the form of pseudo-code in [30]; EFCI bit setting switches (EFCI), Binary Enhanced Switches (BES), and Explicit Down Switches (EDS). EFCI switches, which are already on the market, are same as switches supporting PRCA and are expected to be the least expensive.

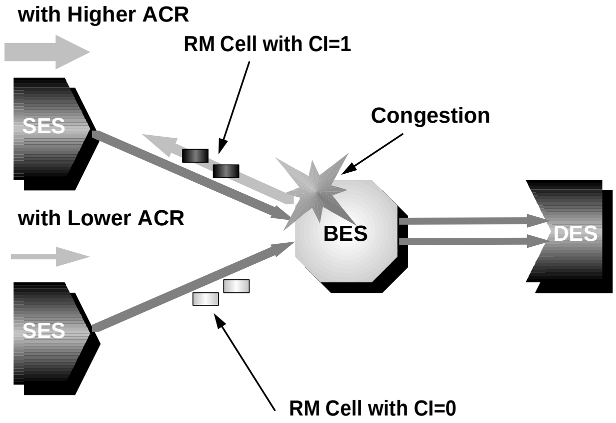

Figure 1.5: Intelligent Marking in a BES Switch.

In BES switches, two threshold values for indicating congestion are defined: $QT$ and $DQT$ (on behalf of $Q_H$ in the EFCI switch). When a BES switch is congested (i.e., when the queue length in the cell buffer of a BES switch exceeds $QT$), the switch performs intelligent marking. It selectively reduces the rate of sources with larger $ACR$ (Fig. 1.5), by which the ACR beat down problem can be avoided. For implementing this mechanism, the switch maintains a control parameter $MACR$ (Mean ACR) that should ideally be the mean of the $ACR$'s of all active connections. When the rate of all connections is equal to $MACR$, the bandwidth is shared equally and the switch can be fully used without falling into congestion. The key to this is obtaining an accurate $MACR$. The BES switch updates its $MACR$ according to the $CCR$ field of forward RM cells. For example, $MACR$ is calculated as

$$MACR \quad \leftarrow \quad MACR(1 - AV) + CCR \times AV,$$

where $AV$ is used as an averaging factor [30]. When the switch becomes congested, it indicates its congestion to the sources having higher rates. More specifically, the switch marks the CI bit of the backward RM cells if its $CCR$ value exceeds $MACR \times DPF$ (Down Pressure Factor), where a typical value of $DPR$ is 7/8 for safe operation. The switch may remain congested, however, if only intelligent marking is used. Therefore when a BES switch becomes *very* congested (such that the queue length exceeds $DQT$), all backward RM cells are marked irrespective of their $CCR$ values. Note that it is evident from the above description that a BECN-like quick congestion notification can be accomplished in BES switches.

The EDS switches are provided with an *explicit rate setting* capability in addition to intelligent marking (Fig. 1.6). These maintain $MACR$ as BES switches do, and they control the transmission rate of sources by setting the $ER$ field of backward RM cells according to a degree of congestion. When a backward RM cell with $CCR$ larger than $MACR$ passes through the congested EDS switch, the value of its ER element is set to $MACR \times ERF$ (Explicit Reduction Factor). If the switch becomes very congested, $MACR \times MRF$ (Major Reduction Factor) is set in all backward RM cells to achieve quick congestion relief, which is called *major reduction*. Typical values of $ERF$ and $MRF$ shown in [30] are 7/8 and 1/4. A quantitative evaluation of these three types of switches will be presented in the next section.

Not only can three types of switches (EFCI, BES and EDS switches) coexist in EPRCA, but the operation of EFCI switches can also be enhanced by locating BES or EDS switches downstream. BES and EDS switches interpret the EFCI bit of forward data cells. When a BES or EDS

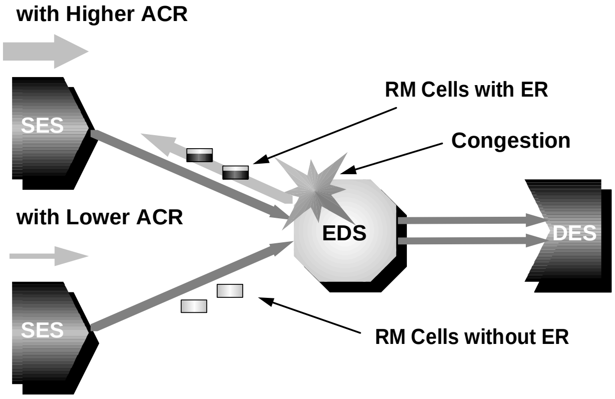

Figure 1.6: Intelligent Marking and Explicit Rate Setting in an EDS Switch.

switch is congested, entries in the VC table are marked according to the EFCI status of forward RM cells, and the EFCI bit is cleared. Then the CI bit of backward RM cells is set to notify the source of the switch's congestion if its associated entry in the VC table is marked. This mechanism enables a BECN-like quick congestion notification even if there are EFCI switches in the network: the enhanced switches can behave as virtual destination end systems for the EFCI switch.

The behavior of the source end system is simplified in [32], which will be included as an example source code in the standard. In that proposal, the source end system decreases its $ACR$ by $ACR/RDF$ every $N_{RM}$ cells sent until reaching $MCR$. While this modification on EPRCA would degrade its performance to some extent, the complexity at the source end system can be decreased considerably.

### 1.3.4 Recent Proposals for Enhancement of EPRCA

This subsection introduces several proposals that enhance the switch capabilities. As mentioned in the introduction to Section 1.3, since the standard does not specify the switch's behavior, the methods shown in this subsection will not be reflected in the standard. We think, however, that these methods will help us understand how the EPRCA can be improved for more effective congestion control.

#### Adaptive Proportional Rate Control

Although EPRCA shares resources more fairly than PRCA does and uses link bandwidth more efficiently, it still has a fault in that fairness is not assured in some configurations [33]. When a BES or EDS switch is very congested, it forces all connections to decrease their rates equally but not selectively. Thus, when the switch is very congested for a long time, *intelligent marking* does not work well. While it has been suggested that this problem can be avoided by eliminating this operation in *very congested* states, this results in excessive queue length rather than fairness [33].

Adaptive Proportional Rate Control (APRC), which is an originator of *intelligent marking* [31], is modified to solve this problem in [34]. Congestion in the switch is detected by evaluating the change of queue length in a fixed time interval rather than by comparing queue length with a threshold value. If the queue length increases in $N$ cell times, the switch is expected to fall into congestion. Then each connection that has a higher rate than $MACR$ is adjusted to a lower rate by setting $MACR$ to the $ER$ field of the backward RM cells. When the number of cells in the buffer exceeds $DQT$, the switch selectively sets $MACR \times DPF$ to $ER$ only for connections with higher rates. This modification improves the responsiveness to congestion and therefore can reduce the maximum queue length. It also improves the fairness among connections.

This *intelligent marking* capability of APRC was incorporated into EPRCA and a newer scheme called APRC2 was introduced in [35]. In EPRCA, the $CCR$ value in the forward RM cell is used to compute $MACR$. At one switch, however, the effective rate of some connections that experience congestion at another switch may be quite different from the $CCR$ values contained in the RM cells. This leads to misbehavior of the explicit rate control. This problem can be avoided by introducing $UCR$ value at the switch in order to establish stable operation [35], where $UCR$ is defined as a mean of $CCR$'s only for connections with larger $CCR$ than $MACR$. The value of $UCR$ is updated as

$$UCR \leftarrow UCR + a(CCR - UCR)$$

only when $CCR$ is greater than $MACR$. Thus $UCR$ is used to determine the explicit rate $ER$ effectively.

The operation of the source end system and switches can also be improved by shortening *ramp-up* time, which is the time between when a connection begins/resumes its transmission and when the network settles into steady state. The ramp-up time is of importance because (1) since each connection starts its transmission with rate $ICR$ regardless of the network status, it might cause a large queue buildup or under-utilization of the link unless $ICR$ is set properly, and (2) most networks operate in a transient state since many connections are established but idle because of the bursty nature of the ABR traffic. Simulation experiments in [36] show that APRC2 results in better ramp-up time, link utilization, and maximum queue length than do EPRCA and APRC. Excellent arguments on limitations of all existing rate control schemes can be found in [35]. Refer to it for understanding various trade offs: “intelligent marking vs. non-selective binary marking” and “counter-based vs. interval-based”.

### EPRCA+ and EPRCA++ Methods

Implementing *explicit rate setting* requires the number of active connections to be known by the switch. One way of assuring this is *per-VC accounting*, which can be implemented in several ways with additional hardware complexity. For instance, each switch can have a VC table to record the number of active connections. Each VC entry is marked or unmarked according to the status of its corresponding VC (active or inactive), and the number of marked entries represents the number of active connections. In this way, the rates of all sources are adjusted through RM cells in one round-trip time when there is one congested switch in the network. EPRCA+, proposed in [37], also uses this kind of scheme, and its simplified version can be found in [38].

In EPRCA+, congestion is detected by estimating the traffic load at the switch rather than by using a threshold value in the cell buffer. For this, the switch is provided with an interval timer and counts the number of cells received during a fixed time interval. The source end system is also equipped more expensively with an interval timer instead of a counter for sending RM cells. The rate of the source is kept unchanged until it receives a backward RM cell in which the explicit rate $ER$ determined by the switch is contained.

One attractive feature of EPRCA+ is its small number of control parameters, which can be set easily by a network manager. Many control parameters required in EPRCA are eliminated in EPRCA+. Furthermore, in EPRCA+ the target utilization band (TUB) around which the switch is utilized, can be set freely. One may set the TUB of the switch under 95% link utilization, and then the queue size at the switch is smaller and cell delays are shorter. Although there is an additional expense for timers and the VC table, EPRCA+ can provide better fairness and responsiveness than EPRCA can [37].

Redundant complexities in the latest EPRCA are pointed out in [39]. For example, an active source end system should decrease its rate by $ACR/RDF$ every $N_{RM}$ cell sent until reaching $MCR$. Its necessity is, however, not well justified, and it might be unnecessary in stable environments. A new scheme called EPRCA++ proposed in [40] uses a counter at the source end system for forward RM cells instead of a timer as in EPRCA+. Furthermore, the source end system decreases its $ACR$ only if no backward RM cell is received in $k \times N_{RM}$ cell times (where $k$ is set to a rather large value). These modifications enable EPRCA++ to perform better than EPRCA+, especially in transient state.

## 1.4 Outline of Thesis

As we have discussed before, congestion control is an essential part of ATM networks for fulfilling stable and efficient operation. Especially, the impact of best-effort traffic with bursty nature on the system should be throughly evaluated since most of existing applications are classified into this category. In the thesis, therefore, two promising congestion control schemes for best-effort traffic — the ATM switch with the back-pressure function and the rate-based congestion control algorithm — are evaluated using both analytic and simulation methods. In the rest of this section, we summarize the objectives of this thesis and refer to other related works in the literature.

### Analysis of ATM Switch with Back-Pressure Function

First, in Chapter 2, we mathematically analyze the performance of the ATM switch with the back-pressure function, which is an inventive switch architecture for minimizing internal congestion as explained in Section 1.2.1. The performance of the ATM switch with the back-pressure function has been analyzed by Iliadis in [41, 42, 43]. However, he assumed that cell inter-arrival times at each input port follow a geometric distribution. Especially when the above switch is applied to ATM networks for supporting data transfer service, its performance should be evaluated by taking into account the bursty nature of arriving traffic — packets coming from the upper protocol layers. On the contrary, we will explicitly model such a bursty nature of traffic by assuming that cells (forming a packet) continuously arrive at the input port and are destined for the same output port. More recently, Elwalid *et al.* have analyzed the performance of multistage switching networks with the back-pressure function for bursty traffic in [44], and Gianatti *et. al* have analyzed the shared-buffered banyan networks for arbitrary traffic patterns in [45]. However, their target switch architecture is different from ours, and they have treated only cell level performance such as average cell delay and cell loss probability. When an upper layer protocol such as TCP (Transmission Control Protocol) is applied on ATM-based networks, packet (or burst) level performance becomes more important. In this chapter, therefore, we analytically derive the packet delay distribution and the approximate packet loss probability in addition to the maximum throughput. In addition to a balanced traffic condition, an unbalanced traffic and a mixture of bursty and stream traffic are also analyzed. Through several numerical examples, we quantitatively show the effects of the average packet length and the output buffer size on its performance.

### Performance of Rate-Based Congestion Control Algorithm

Second, in Chapter 4, our interest turns to the performance evaluation of the rate-based congestion control algorithm, which is a promising global congestion control scheme applied to the ABR service class. In this chapter, we evaluate and compare performance of various switch algorithms of the rate-based congestion control. Particularly, we focus on two representative schemes: EPRCA and EPRCA++. EPRCA is a basis of standard traffic management mechanism adopted by the ATM Forum [30, 3]. In the standard, only the behaviors of the source and destination end systems are described, and the implementation issues regarding the ATM switches are left to manufacturers. In [30], however, they suggest three types of switches, EFCI bit setting switch (EFCI), binary enhanced switch (BES), and explicit down switch (EDS), which have different processing capabilities against congestion as summarized in Section 1.3.3. On the other hand, EPRCA++ is a lately proposed algorithm for improving the network performance but needs more complex functions at the switch as summarized in Section 1.3.4.

While a lot of studies have been devoted to evaluate these schemes, only ABR traffic is taken into account; the effect of VBR and CBR service classes, in which multimedia traffic is accommodated, are not considered. In this chapter, we evaluate performance of rate-based congestion control schemes when not only ABR traffic but also VBR traffic is incorporated into the network. We assume that CBR service class is used for video traffic. Namely, when the video traffic is generated, the call setup process is performed before its actual cell transmissions (i.e., call admission control based on closed-loop control is invoked). Since we consider the CBR service class, only the peak rate is required for the traffic descriptor in this case. Furthermore, in the network, CBR service class cells are assumed to be given higher priority than ABR service class cells for assuring QoS of CBR service class. See, e.g., [46] for the switch architecture to provide such priority services.

The problem is, however, that video traffic essentially has a bursty nature. That is, the cell generation rate per frame is varied if the compression technique as MPEG is applied to the video sources. Note that we here distinguish a traffic class and a service class; CBR traffic generates cells in the constant bit rate while CBR service class is the class related to CAC. Therefore, the CBR service class may accept VBR traffic that generates cells in the variable bit rate. Then the available bandwidth (i.e., residual bandwidth) to the ABR service class is changed dependent on cell generation of the VBR traffic. It was never considered in the past studies in which the available bandwidth to the ABR service class is fixed. In this chapter, we treat such a case that the VBR traffic is applied to the CBR service class, which is most likely to be realized in the ATM network by its simplicity since VBR service class still has difficulties for implementation in CAC and UPC.

When we consider both ABR and CBR service classes in the network, the rate-based control algorithms for ABR service class must be affected by the characteristics of video traffic, which has never been considered. In this chapter, we will use sampled data taken from MPEG streams. Then, we investigate the performance of ABR traffic class. For this purpose, EPRCA and EPRCA++, the rate-based control algorithms discussed in the ATM Forum, are used and drawbacks of these algorithms are demonstrated through simulation experiments.

### Parameter Tuning of Rate-Based Congestion Control Algorithm

Third, in Chapter 4, a determination scheme of control parameters of the rate-based congestion control algorithm is analytically obtained. As we have explained in Section 1.3, the target of the standard is an operation algorithm of both source and destination end systems, and implementation issues regarding intermediate switches are left to manufacturers. However, some example behaviors of intermediate switches are also introduced in the standard [30, 34, 47, 3]. In this chapter, we will focus on the simplest switch among these, which is referred to as an “EFCI bit setting switch” or a “binary switch”.

In the standard document [3], several control parameters for controlling cell transmission at the source end system are defined. These include the $RIF$ (Rate Increase Factor) and $RDF$ (Rate Decrease Factor) which control the envelopes of rate increase and decrease, respectively. During the process to establish a connection, the source end system negotiates the control parameters with the network. The effectiveness of rate-based congestion control is heavily dependent on the choice of control parameters as shown in [48, 49]. If these parameters are configured properly, rate-based congestion control can achieve high performance (i.e., no buffer overflow, high link utilization, and short cell delay). However, a method of selecting the control parameters is not specified in the standard, and the parameters need to be determined intuitively unless a proper tool is provided. Through steady-state analysis, we will derive two conditions that the source control parameters should satisfy; one is the prevention of buffer overflow at the switch buffer, and the other is full link utilization. For this purpose, we will analytically obtain values close to the maximum and minimum queue lengths observed at the switch buffer. While the maximum queue length is separately obtained in [50], which is a slight extension of our paper [16], our objective in this chapter is to study the control aspect in more depth. That is, we will first focus on how our analysis results can be applied to the control parameter settings of the rate-based congestion control method. Another key issue in dealing with rate-based congestion is how to choose the initial values of control parameters, by which the initial behavior of source end systems is determined. So we will also focus on the initial transient state, and show which settings of the initial control parameters of source end systems are desirable based on our analysis.

### Robustness of Rate-Based Congestion Control Algorithm

Fourth, in Chapter 5, we further investigate proper settings of control parameters for more generic network configurations. In reality, each connection may have a different round-trip delay according to the network configuration. In such a case, fairness among connections may be degraded due to the different feedback delays. When another ABR connection is newly established in the network, the ramp-up time of this connection is also important. Furthermore, we need further investigation to apply our analysis to more general network configurations with multiple switches where each connection has the different number of hops and the different propagation delay.

We further need to consider existence of real-time applications such as audio and video in a multimedia network environment. Since these applications use CBR (Constant Bit Rate) or VBR (Variable Bit Rate) service class, multiple service classes co-exist in the network. For ABR service class to utilize the available bandwidth unused by CBR/VBR service class, CBR/VBR traffic should be given higher priority than ABR traffic at the switch to guarantee QoS (Quality of Service) requirements of CBR/VBR traffic. Namely, cells of ABR traffic are awaited in the switch buffer if a CBR/VBR cell exists in the switch buffer in the case that the switch has two logically independent buffers — one for CBR/VBR service class and the other for ABR service class [51]. In other words, the bandwidth available to the ABR service class is limited by CBR/VBR traffic. Therefore, when a CBR/VBR connection is newly added into the network, the bandwidth available to the ABR service class is suddenly decreased, which would give a serious effect on the performance of the ABR connections; the switch buffer for ABR cells may become overloaded for a while leading to a large queue buildup and eventually cell losses due to the buffer overflow.

In this chapter, we focus on the two subjects. We first analyze the behavior of the rate-based congestion control for a single-hop network but each group of connections is allowed to have the different propagation delay. In [52], Blot et al. have analyzed a dynamical behavior of a rate-based congestion control for connections with different propagation delays. However, their analytic model was quite simple and different from the rate-based congestion control standard [3]. Through numerical examples, we show the effect of control parameters on the ramp-up time of a new ABR connection. We also derive the maximum queue length at the switch after a new CBR connection is established in the network. In [53], we have investigated some aspects of the performance of the rate-based congestion control algorithm in the multimedia network environment through simulation experiments. The author in [54] has analyzed the effect of CBR traffic on the ABR service class. However, he has considered only the case where the CBR connection requires the bandwidth close to the link capacity, and his analytic model was different from the rate-based congestion control standard [3]. On the contrary, in this chapter, we treat a more general and realistic model where the CBR connection requires any portion of the link bandwidth, and derive the maximum queue length via the worst case analysis.

We next investigate a proper setting of control parameters for a multi-hop network configuration by simulation. In the simulation, we use the model with multiple connections with different numbers of hops. The main purpose is to evaluate the effect of two rate-control parameters ($RIF$ and $RDF$) on the performance. In [55, 56], the authors have provided simulation results for several combinations of control parameters. In this chapter, control parameters are chosen based on our analytic results. As performance measures, cell loss possibility, link utilization and fairness among connections are considered. We also validate how our analytic results of the single-hop model can be applied to generic network models.

### Designing Efficient Explicit-Rate Switch Algorithm

Fifth, Chapter 6 is devoted to design an efficient explicit-rate switch algorithm. While implementation is rather complex, an explicit-rate switch has a potential to obtain much better performance even in WAN environments. A typical operation of an explicit-rate switch is to compute an appropriate bandwidth allocation for every connection based on, for example, the bandwidth available to ABR connections and the degree of congestion. The switch then updates the ER value of forward and / or backward RM cells as

$$ER \leftarrow \min(FS, ER). \tag{1.3}$$

In the above equation, $ER$ is the ER value in the RM cell written by some other switch, and $FS$ is a computed bandwidth allocation for the connection. When the source end system receives the backward RM cell, it updates its $ACR$ as

$$ACR \leftarrow \min(ACR + PCR \times RIF, PCR, ER) \tag{1.4}$$

Thus, bandwidth allocation for all connections can be finished within one round-trip time only if $RIF$ is set to be a large value; that is, if $RIF$ is small, the source end system needs more RM cells to increase its $ACR$ to $ER$. The brightness of the above equation is that the source end system does not necessarily know the switch type (i.e., binary-mode or explicit-rate switch). In other words, an effectiveness of explicit-rate switches is fully dependent on the determination method of the ER value.

In the ATM Forum, several switch algorithms with explicit-rate marking have been proposed through standardization process of the rate-based congestion control algorithm [3, 18]. These include EPRCA (Enhanced Proportional Rate Control Algorithm) [30], CAPC (Congestion Avoidance using Proportional Rate Control) [57], APRC2 (Adaptive Proportional Rate

Control) [35] and ERICA (Explicit Rate Indication for Congestion Avoidance) [47]. Each algorithm has its own advantages and disadvantages in terms of, for example, effectiveness, robustness, fairness and configuration simplicity. However, tradeoffs among these objectives have not been discussed in detail by any researcher. Therefore, in this chapter, we first discuss design goals of an efficient explicit-rate switch algorithm. We then investigate existing switch algorithms based on the discussion about design goals. For this, we first summarize a recently proposed switch algorithm called as the max-min scheme [58]. A strong point of this algorithm compared with others is that it can satisfy *max-min fairness* for any network configuration; that is, total throughput of the network is maximized and fairness among connections is maintained [59]. However, its defect is in lack of adaptability to changes in the network (e.g., connection addition / disconnection) as will be demonstrated through simulation experiments. Thus, we propose our enhancements to the max-min scheme to improve its stability and efficiency. We also evaluate its performance by comparing with other explicit-rate switch algorithms.

Finally, in Chapter 6.4, we conclude this thesis.

# Chapter 2

# Analysis of ATM Switch with Back-Pressure Function

The ATM switch with both input and output buffers provided with a back-pressure function has been proposed as a cost-effective switch architecture. The back-pressure function prohibits cell transmission from the input buffer to the corresponding output buffer to avoid cell loss at the output buffer due to a temporary congestion. Especially when this switch is applied to ATM LANs for data transfer services, its performance should be evaluated by taking into account bursty traffic. In this chapter, we show the maximum throughput, the packet delay distribution, and the approximate packet loss probability of such an ATM switch for bursty traffic through an analytic method. In addition to a balanced traffic condition, an unbalanced traffic and a mixture of bursty and stream traffic are also analyzed. Through several numerical examples, we show the effects of the average packet length and the output buffer size on its performance.

## 2.1 Analytic Model

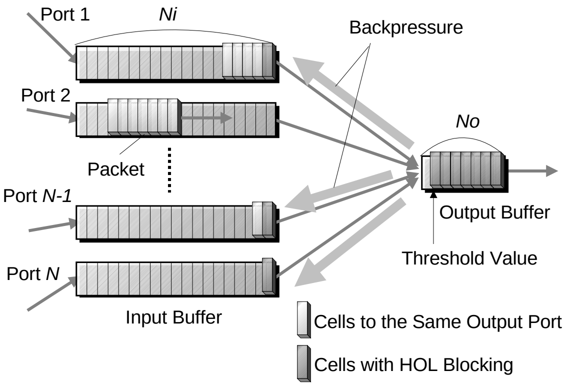

Figure 2.1: Analytic Model.

In this section, we describe our analytic model. We assume that a stream of successively arriving cells forms a packet, and the number of cells in the packet follows a geometric distribution with mean $\overline{BL}$. Let $p$ denote the probability that at the input port, a newly arriving cell belongs to the same packet. Thus, we have a relation;

$$\overline{BL} = \sum_{i=1}^{\infty}(1-p)ip^{i-1} = \frac{1}{1-p}$$

We assume that all cells are stored under first-in-and-first-out (FIFO) discipline at input buffer.

The practical threshold value at output buffer would be $(N_O - N)$ [14]. However, as an ideal case, we assume that the HOL cells are randomly transferred from input buffer to output buffer until the output buffer becomes full. Then, when the output buffer becomes fully occupied, input buffers that have HOL cells destined for this output buffer receives a back-pressure signal to stop cell transmission. Thus, all HOL cells are awaited at the head of input buffers. As soon as the cell in output buffer is transmitted onto the output link, one of HOL cells is selected at random and transmitted to the output buffer. Therefore, it is considered that HOL cells destined for the same output port form a virtual queue, which we will call a HOL queue. While HOL cells are actually stored at the HOL queue, it can be regarded that HOL packets form the HOL queue [41, 42, 43]. Therefore, in what follows, we will use “HOL cell” and “HOL packet” without discrimination.

The switch size $N$ is assumed to be infinity in the following analysis. By introducing this assumption, we can focus on one single output port and its associated HOL queue. The infinite switch size gives the performance limitation as shown in [42, 60]. That is, when compared with the finite case, the maximum throughput with the infinite case gives an upper bound. It is also known that the close values are obtained when $N$ reaches 16 or 32 when the cell interarrivals follow a geometric distribution [42]. In this chapter, we will examine this fact even in the case of bursty traffic in Section 2.3.

In this chapter, we will first assume the infinite capacity of the input buffer $(N_I = \infty)$ to obtain the maximum throughput (Section 2.3) and the packet delay distribution (Section 2.4). Because the memory speed of the output buffer should be $N$ times faster than the link speed, the capacity of the output buffer is limited. On the contrary, the input buffer can operate at the same speed with the input link, i.e., the input buffer can be equipped with large capacity. This assumption is then relaxed to derive the packet loss probability in Section 2.5. Although the analysis is approximate, it is accurate in the case of the large buffer size as will be validated by comparing with simulation results in Section 2.5.

## 2.2 Derivation of Steady State Probability

We focus on a single output buffer and its associated HOL queue by assuming the infinite number of input and output ports. We consider a discrete-time system where its slot time equals a cell transmission time on the input and output link. Under the assumptions described in Section 2.1, the system state is represented by two random variables, $Q_k$ and $H_k$. $Q_k$ is the number of cells at an output buffer at $k$th slot and $H_k$ is the number of HOL cells at the input buffers associated with that output buffer. In what follows, the steady state probability of the doublet of two random variables, ($Q_k$, $H_k$), will be derived. For this purpose, we further introduce $A_k$ as a random variable representing the number of HOL packets newly arriving at the HOL queue at the beginning of $k$th slot. By defining a symbol $(x)^+ = \max(0, x)$, we have the following possibilities.

1. $H_{k-1} + A_k \le N_O - (Q_{k-1} - 1)^+$; that is, all HOL cells can be transferred to the output port. At first, we have

$$Q_k = (Q_{k-1} - 1)^+ + H_{k-1} + A_k. \tag{2.1}$$

Let $B_k$ be the number of the HOL packets that further generate HOL cells at the next $(k+1)$th slot. When there exist $i$ HOL packets in the HOL queue, the probability that $B_k$ becomes $j$ is

$$b_{i,j} = \binom{i}{j} p^j (1-p)^{i-j}, \tag{2.2}$$

and we have

$$H_k = B_k.$$

2. $H_{k-1}+A_k > N_O-(Q_{k-1}-1)^+$; that is, some HOL cells cannot be transferred to the output port at $k$th slot.

   $(N_O-(Q_{k-1}-1)^+)$ HOL cells are transferred to the output buffer, and $C_k$ cells of them further generate HOL cells in the next $(k+1)$th slot. Therefore, $(H_{k-1}+A_k-(N_O-(Q_{k-1}-1)^+))$ cells are kept waiting at the HOL queue. Hence, we have

$$\begin{aligned} Q_k &= N_O \\ H_k &= H_{k-1}+A_k-(N_O-(Q_{k-1}-1)^+)+C_k. \end{aligned}$$

As explained in Section 2.1, we assume that arrivals of packets at input ports in time slot follow a Poisson distribution since the switch size $N$ is assumed to be infinity. Therefore,

$$a_j \equiv P[A=j] = P[A_k=j] = \frac{\lambda_p^j e^{-\lambda_p}}{j!},$$

where $\lambda_p$ is the mean arrival rate of packets at each input port. By defining $\lambda_c$ as the mean arrival rate of cells at input ports, we have

$$\lambda_c = \lambda_p \overline{BL}. \tag{2.3}$$

We consider $s_{n,m,n',m'}$, which is a transition probability from a state $[Q_{k-1}=n, H_{k-1}=m]$ to $[Q_k=n', H_k=m']$. $s_{n,m,n',m'}$ is obtained as follows.

1. When $n' < N_O$; that is, when the back-pressure function does not work.

   From Eq. (2.1), we have

$$A_k = Q_k-(Q_{k-1}-1)^+-H_{k-1}.$$

   When $m'$ packets of $(Q_k-(Q_{k-1}-1)^+)$ HOL packets further generate cells at the next time slot, we have a relation

$$s_{n,m,n',m'} = a_{n'-(n-1)^+-m} b_{n'-(n-1)^+,m'}. \tag{2.4}$$

2. When $n' = N_O$; that is, when the back-pressure function works.

   From Eq. (2.3), we have

$$A_k = N_O-(Q_{k-1}-1)^+-H_{k-1}+(H_k-C_k).$$

   Since $C_k$ packets of $(N_O-(Q_{k-1}-1)^+)$ HOL packets further generate cells at the next time slot, we have

$$s_{n,m,n',m'} = \sum_{i=0}^{m'} a_{n'-(n-1)^+-m+i} b_{n'-(n-1)^+,m'-i}. \tag{2.5}$$

Let $r_{n,m}$ be the steady state probability defined as

$$r_{n,m} = \lim_{k\to\infty} P[Q_k = n, H_k = m] = P[Q = n, H = m].$$

In what follows, we will obtain $r_{n,m}$ from Eqs. (2.4) and (2.5).

1. When the state is $[Q = 0, H = 0]$, the output port becomes idle. Thus, we have

   $$r_{0,0} = 1 - \rho,$$

   where $\rho$ is defined as the maximum throughput normalized by the link capacity. By our assumption of the infinite input buffer size, the maximum throughput $\rho$ is equivalent to the cell arrival rate $\lambda_c$ in steady state if it exists.

2. By considering all states that may change to state $[Q = n-1, H = 0]$, we have $r_{n,0}$ as follows (see Fig. 2.2).

   $$r_{n,0} \quad = \quad \frac{1}{s_{n,0,n-1,0}}\left\{r_{n-1,0} - \sum_{i=0}^{n-1}\sum_{j=0}^{i} s_{i,j,n-1,0}r_{i,j}\right\} \quad (0 < n \le N_O)$$

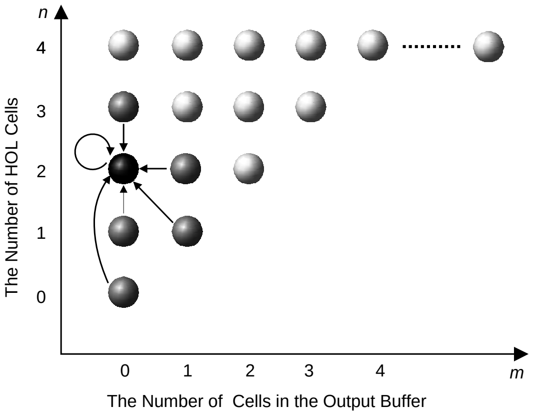

Figure 2.2: State Transition Diagram in the Case of $m = 0$ and $0 < n \le N_O$.

3. By considering all states that may change to state $[Q = n, H = m]$, we have $r_{n,m}$ as follows (see Fig. 2.3).

   $$r_{n,m} \quad = \quad \frac{1}{1 - s_{n,m,n,m}}\left\{\sum_{i=0}^{n-1}\sum_{j=0}^{i} s_{i,j,n,m}r_{i,j} + \sum_{k=0}^{m-1} s_{n,k,n,m}r_{n,k}\right\} \quad (0 < m, n < N_O)$$

4. By considering all states that may change to the state $[Q = N_O, H = m-1]$, we have $r_{N_O,m}$ as follows (see Fig. 2.4).

   $$r_{N_O,m} = \frac{1}{s_{N_O,m,N_O,m-1}}\left\{r_{N_O,m-1} - \sum_{i=0}^{N_O-1}\sum_{j=0}^{i} s_{i,j,N_O,m-1}r_{i,j} - \sum_{k=0}^{m-1} r_{N_O,k}\right\} \quad (0 < m)$$

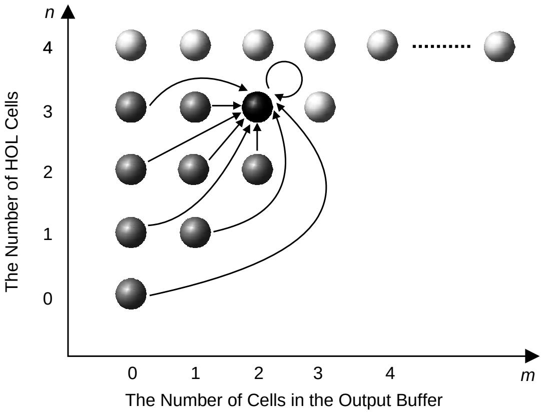

Figure 2.3: State Transition Diagram in the Case of $0 < m$ and $n < N_O$.

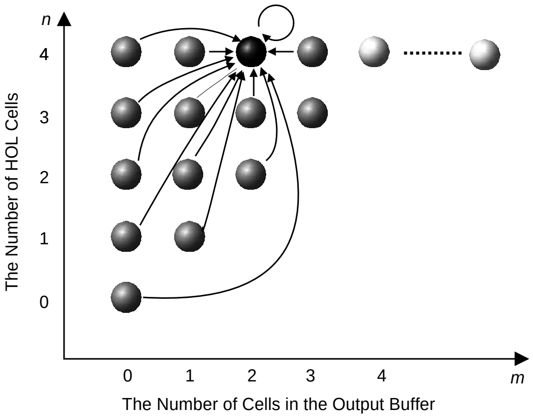

Figure 2.4: State Transition Diagram in the Case of $0 < m$ and $n = N_O$.

## 2.3 Maximum Throughput Analysis

By using the steady state probabilities derived in Section 2.2, we obtain the maximum throughput under a balanced traffic condition in Subsection 2.3.1, under an output-unbalanced traffic condition in Subsection 2.3.2, and under an input-unbalanced traffic condition in Subsection 2.3.3. The case of a mixture of bursty and stream traffic is also considered in Subsection 2.3.4.

### 2.3.1 Case of Balanced Traffic condition

In this subsection, a balanced traffic condition is assumed; that is, the mean packet arrival rate at every input port is identical and each packet determines its output port with an equal probability $1/N$.

In order to obtain the maximum throughput, we consider the case where all input ports are saturated so that packets are always waiting in HOL queues. In this case, we have a relation:

$$\sum_{i=1}^{N} A^i \quad = \quad N - \sum_{i=1}^{N} H^i,$$

where $A^i$ is the random variable which represents the number of arriving packets destined for the output port $i$ in a slot and $H^i$ is the random variable for the number of HOL cells destined for the output port $i$. By dividing the above equation by $N$ and letting $N$ to be infinity, we have

$$\lambda_p = 1 - \overline{H}, \tag{2.6}$$

where $\overline{H}$ is the average number of HOL cells. $\overline{H}$ is expressed with $r_{n,m}$ derived in Section 2.2 as

$$\overline{H} = \sum_{n=0}^{N_O} \sum_{m=1}^{\infty} m\, r_{n,m}.$$

From Eqs. (2.3) and (2.6), we have

$$\lambda_c = (1 - \overline{H})\overline{BL}. \tag{2.7}$$

The maximum throughput $\rho$ can be obtained by substituting $\lambda_c$ in the above equation with $\rho$ and solving it for $\rho$. Since $\overline{H}$ depends on $\rho$, $\rho$ is solved iteratively by virtue of a standard iteration technique such as a bisection method [61].

In Figs. 2.5 and 2.6, the maximum throughput $\rho$ is plotted for the average packet length $\overline{BL}$ and the output buffer size $N_O$, respectively. These figures show that the packet length drastically degrades the maximum throughput. Furthermore, we may observe that the size of output buffers must be larger than the average packet length to gain a sufficient throughput. We note that the maximum throughput for $N_O = 1$ is exactly same as the well known value of the input queuing, 0.585 [60].

Figure 2.7 compares the analytic results (the switch size $N = \infty$) with simulation results ($N = 8, 16$ and 32) for $N_O = 1$ and $N_O = 50$ dependent on the average packet length $\overline{BL}$. The case of $\overline{BL} = 1$ in the figure corresponds to results obtained in [41, 42, 43], and it can be found that the analytic results become close to simulation results as the switch size gets large even in the case of $BL > 1$. We note that 95% confidence intervals of all simulation results for maximum throughput are within 2% of mean values, and are not shown in the figure.

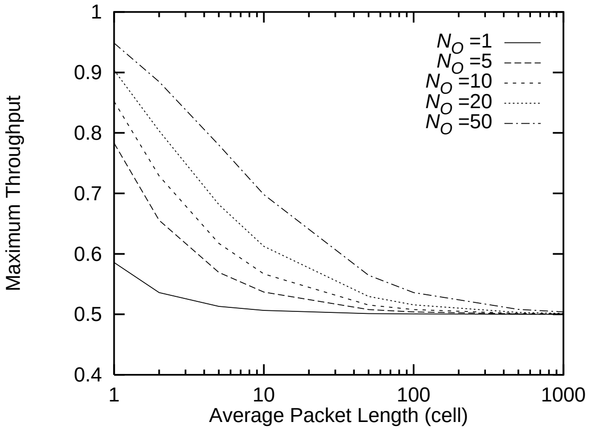

Figure 2.5: Maximum Throughput vs. Average Packet Length.

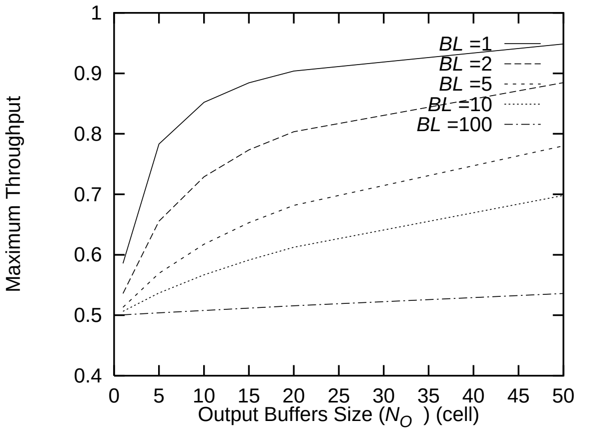

Figure 2.6: Maximum Throughput vs. Output Buffer Size.

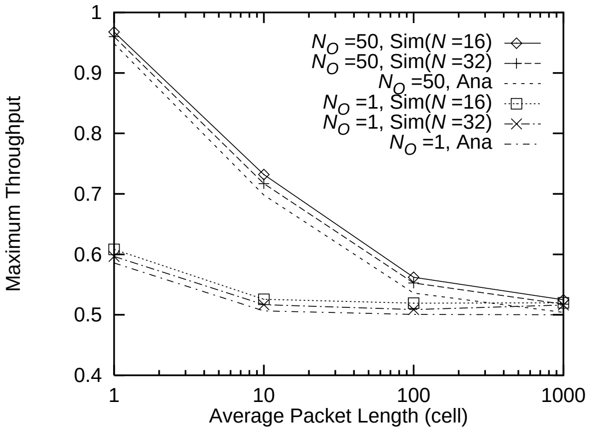

Figure 2.7: Comparison with Simulation Results.

### 2.3.2 Case of Unbalanced Traffic at Output Ports

In this section, output unbalanced traffic is treated following the approach presented in [41]. Output buffers are divided into two groups called $O_1$ and $O_2$. Let $q_O$ be a ratio of the number of output ports belonging to the group $O_1$ as

$$q_O \equiv \frac{|O_1|}{N}. \tag{2.8}$$

The packet arrival rate at each input port is identical. However, each packet arriving at the input port selects one of output ports in group $O_1$ with probability $P_{G1}$ or one of output ports in group $O_2$ with probability $P_{G2}$. By assuming $P_{G1} \geq P_{G2}$ without loss of generality, the relative probability $r_O$ is denoted as

$$r_O \equiv \frac{P_{G1}}{P_{G1} + P_{G2}} \geq 0.5. \tag{2.9}$$

It is noted that the balanced traffic case is a special case by setting $q_O = 0$, $q_O = 1$ or $r_O = 0.5$. Let $P_1$ and $P_2$ be the probabilities that an arriving packet is destined to output ports belonging to the $O_1$ and $O_2$, respectively. From Eqs. (2.8) and (2.9), we have

$$\begin{aligned}
P_1 &= \frac{|O_1|P_{G1}}{|O_1|P_{G1} + (N - |O_1|)P_{G2}} \\
&= \frac{q_O r_O}{1 - q_O - r_O + 2q_O r_O} \\
P_2 &= \frac{(N - |O_1|)P_{G2}}{|O_1|P_{G1} + (N - |O_1|)P_{G2}} \\
&= \frac{1 - q_O - r_O + q_O r_O}{1 - q_O - r_O + 2q_O r_O}.
\end{aligned}$$

We define $\lambda_p$ as the packet arrival rate at each input port, and $\lambda_{p1}$ and $\lambda_{p2}$ as the packet arrival rates at output ports belonging to the group $O_1$ and $O_2$, respectively. We then obtain

$$\begin{aligned}
\lambda_{p1} &= \frac{r_O \lambda_p}{1 - q_O - r_O + 2q_O r_O} \\
\lambda_{p2} &= \frac{(1 - r_O)\lambda_p}{1 - q_O - r_O + 2q_O r_O}.
\end{aligned}$$

For deriving the maximum throughput, we consider a relation

$$\sum_{i=1}^{N} A^i = N - \left( \sum_{i=1}^{|O_1|} H_1^i + \sum_{i=1}^{|O_2|} H_2^i \right),$$

where random variables $H_1^i$ ($H_2^i$) is the number of HOL cells destined for the output port belonging to the group $O_1$ ($O_2$). By dividing the above equation by $N$ and letting $N$ to be infinity, we have

$$\lambda_p = 1 - \{q_O \overline{H}_1 + (1 - q_O)\overline{H}_2\},$$

where $\overline{H}_1$ and $\overline{H}_2$ are the average number of HOL cells destined for the group $O_1$ and $O_2$, respectively. From Eq. (2.3), we have

$$\lambda_c = \left[1 - \{q_O \overline{H}_1 + (1 - q_O)\overline{H}_2\}\right] \overline{BL}.$$

The maximum throughput $\rho$ can be obtained by substituting $\lambda_c$ with $\rho$ in the above equation and solving for $\rho$ in the same manner presented in Subsection 2.3.1.

In Figs. 2.8 and 2.9, the relations between $q_O$ and the maximum throughput are plotted for $\overline{BL} = 1$ and $\overline{BL} = 10$, respectively. These figures show that an unbalanced traffic and a larger packet size cause degradation of the maximum throughput.

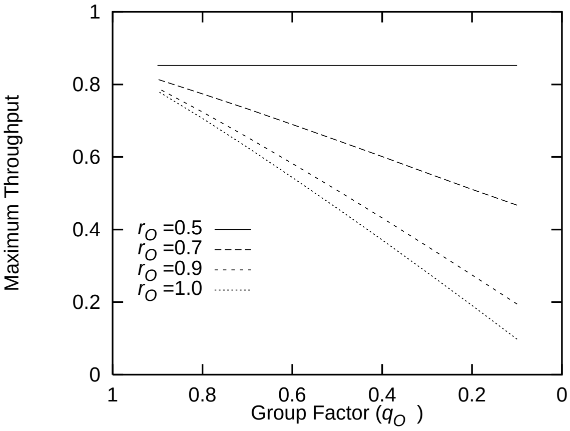

Figure 2.8: Unbalanced Traffic at Output Ports ($N_O = 10$ and $\overline{BL} = 1$).

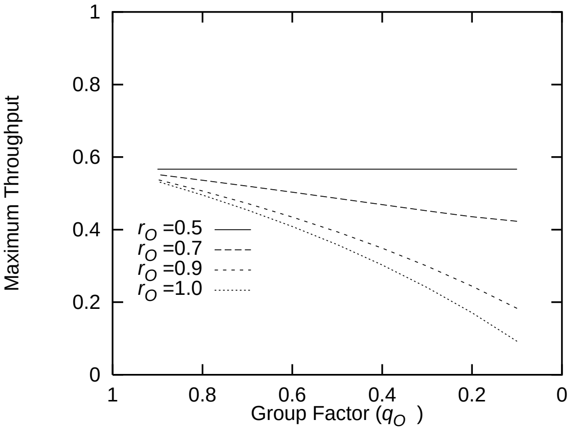

Figure 2.9: Unbalanced Traffic at Output Ports ($N_O = 10$ and $\overline{BL} = 10$).

### 2.3.3 Case of Unbalanced Traffic at Input Ports

In this subsection, we evaluate the performance of the switch under an unbalanced traffic condition at the input ports. Similar to the previous subsection, input ports are divided into two groups $I_1$ and $I_2$. Let $q_I$ be a ratio of the number of input ports belonging to the group $I_1$ defined as

$$q_I \equiv \frac{|I_1|}{N}. \tag{2.10}$$

We further introduce $\lambda_{p1}$ and $\lambda_{p2}$ as mean packet arrival rates at the groups $I_1$ and $I_2$, respectively. Assuming that $\lambda_{p1} \geq \lambda_{p2}$ without loss of generality, we introduce $r_I$ as

$$r_I \equiv \frac{\lambda_{p1}}{\lambda_{p1} + \lambda_{p2}} \geq 0.5. \tag{2.11}$$

It is noted that the balanced traffic case is the special case by setting $q_I = 0$, $q_I = 1$ or $r_I = 0.5$. We assume that each packet arriving at the input port chooses the output port with a same probability $1/N$. By letting $\lambda_p$ denote the packet arrival rate at each output port, $\lambda_{p1}$ and $\lambda_{p2}$ are given as

$$\begin{aligned} \lambda_{p1} &= \frac{\lambda_p r_I}{1 - q_I - r_I + 2q_I r_I} \\ \lambda_{p2} &= \frac{\lambda_p (1 - r_I)}{1 - q_I - r_I + 2q_I r_I}. \end{aligned}$$

To obtain the maximum throughput, we consider the case where input ports are saturated. Recalling that we assume $\lambda_{p1} \geq \lambda_{p2}$, the input buffers belonging to the group $I_1$ is saturated first. Thus, we have

$$\sum_{i=1}^{|I_1|} A_1^i = |I_1| - \sum_{i=1}^{N} \frac{\lambda_{p1}}{\lambda_p} \frac{|I_1|}{N} H^i,$$

where the random variable $A_1^i$ is the number of packets arriving at the input port $i$ belonging to the group $I_1$. By dividing the above equation by $N$ and letting $N$ to be infinity, we have

$$\lambda_{p1} = 1 - \frac{r_I \overline{H}}{1 - q_I - r_I + 2q_I r_I}.$$

From Eq. (2.3), we obtain

$$\lambda_{c1} = (1 - \frac{r_I \overline{H}}{1 - q_I - r_I + 2q_I r_I}) \overline{BL},$$

where $\lambda_{c1}$ is the mean packet arrival rate at each input port belonging to the group $I_1$. The maximum throughput $\rho$ can be obtained by substituting $\lambda_{c1}$ in the above equation with $\rho$ and solving for $\rho$ as in the same manner presented in Subsection 2.3.1.

Figures 2.10 and 2.11 show the maximum throughput dependent on $q_I$ for $\overline{BL} = 1$ and $\overline{BL} = 10$, respectively. These figures show that an unbalanced traffic condition and a larger packet size degrade the maximum throughput. The result for $\overline{BL} = 1$ is almost same as that for the output unbalanced traffic (Fig. 2.8). On the other hand, the result for $\overline{BL} = 10$ shows higher performance than that of output unbalanced traffic (Fig. 2.9). This is because unbalanced traffic at input ports causes less HOL blocking than at output ports.

### 2.3.4 Case of Mixture with Stream Traffic

Finally, we derive the maximum throughput in the case where the bursty traffic and the stream traffic coexist. We assume that the stream traffic occupies some portion of the link with constant peak rate. For example, this class of traffic can support an uncompressed video transfer service.

Let $R$ denote the peak rate of stream traffic normalized by the link capacity. The switch can simultaneously accept $m(\leq \lfloor 1/R \rfloor)$ calls of stream traffic. We assume that call arrivals of the stream traffic follow a Poisson distribution with mean $\lambda_{CBR}$, and its service time (call

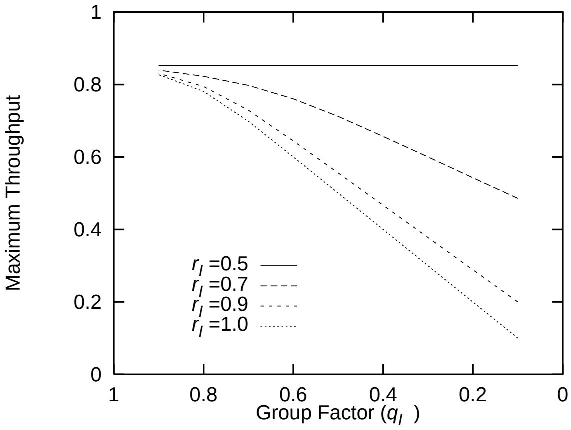

Figure 2.10: Unbalanced Traffic at Input Ports ($N_O = 10$ and $\overline{BL} = 1$).

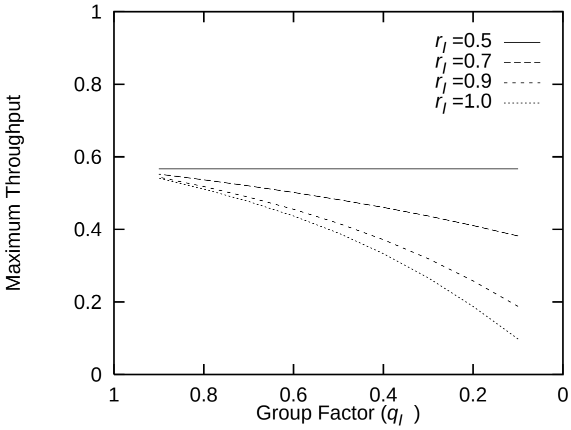

Figure 2.11: Unbalanced Traffic at Input Ports ($N_O = 10$ and $\overline{BL} = 10$).

holding time) has an exponential distribution with mean $1/\mu_{CBR}$. While both bursty and stream traffic share a link, cells of the stream traffic are given higher priority. Namely, cells of stream traffic arriving at the input port are transferred to its destination output port prior to cells of bursty traffic [14]. By this control mechanism, it can be considered that bursty traffic can utilize $1 - nR$ of the link capacity when $n$ calls of stream traffic are accepted. We note that if compressed video transfer service is accommodated as stream traffic, more capacity can be utilized by bursty traffic. Thus, the maximum throughput derived in the below should be regarded as the "minimum" guaranteed throughput for the bursty traffic.

Since the stream traffic is given high priority, it can be modeled by an M / M / m / m queuing system. By letting $\pi_n$ be the probability that $n$ calls of stream traffic are accepted in steady state, $\pi_n$ is given as follows (e.g., [62]).

$$\pi_n = \left[\sum_{n=0}^{m} \left(\frac{\lambda_{CBR}}{\mu_{CBR}}\right)^n \frac{1}{n!}\right]^{-1} \left(\frac{\lambda_{CBR}}{\mu_{CBR}}\right)^n \frac{1}{n!}$$

Since the service time of steam traffic can be assumed to be much longer than cell or the packet transmission time of bursty traffic, the available link capacity for bursty traffic is regarded to be constant when the number of accepted calls of stream traffic is fixed. By letting $\rho_n$ be the maximum throughput for bursty traffic when $n$ calls of the stream traffic are accepted, we have [14]

$$\rho_n = (1 - nR)\rho,$$

where $\rho$ is defined as the maximum throughput of bursty traffic when all link capacity is allocated to bursty traffic, and has been already derived in Subsection 2.3.1. Consequently, the "averaged" maximum throughput $\rho'$ is obtained as follows.

$$\rho' = \sum_{n=0}^{m} \pi_n \rho_n$$

Figure 2.12 shows the maximum throughput of bursty traffic and throughput of stream traffic dependent on an offered traffic load for stream traffic for $N_O = 50$, $\mu_{CBR} = 0.1$, $R = 0.2$ and $m = 5$. From this figure, we can observe the natural idea that the larger the average packet length is, the smaller the maximum allowable throughput of bursty traffic is. Therefore, the available bandwidth allocated to the stream traffic should be limited in some way to avoid a degradation of bursty traffic efficiency. One possible approach is to decrease $m$, which is the maximum number of calls of stream traffic that the switch can accept. In an actual situation, it can be implemented in CAC (Call Admission Control) so that an acceptable number of calls of stream traffic is limited. Figure 2.13 shows the maximum throughput of both bursty traffic and stream traffic dependent on the offered traffic load for stream traffic for $\overline{BL} = 1$ and several values of $m$. It shows that the performance degradation of bursty traffic can be avoided to some extent by limiting $m$.

## 2.4 Derivation of Packet Delay Distribution

In this section, we derive the packet delay experienced at both input and output buffer. The packet delay is defined as the time duration from when the first cell of the packet arrives at the input port of the switch to when the last cell is transmitted onto the output link. We divide the packet delay into the following three elements.

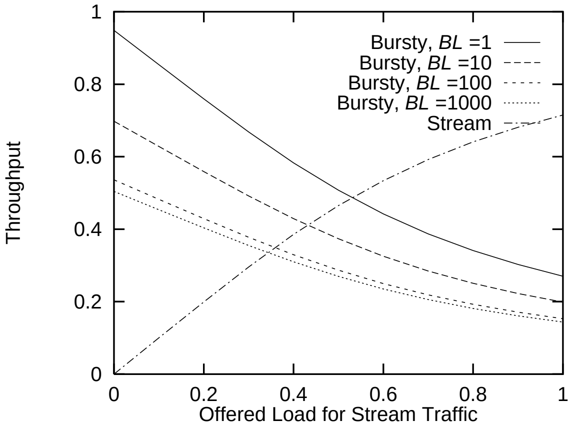

Figure 2.12: Throughput vs. Offered Load for Stream Traffic.

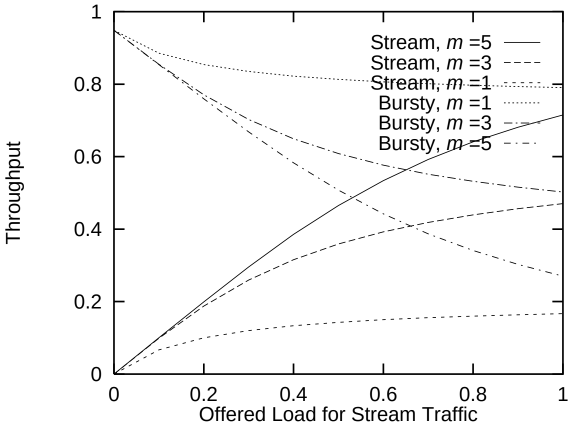

Figure 2.13: Effect of Available Link Capacity Limitation on Stream Traffic.

1. $W_I$ : The packet waiting time at the input buffer from the arrival time of the first cell of the packet at the input buffer to its arrival time at the HOL queue.
2. $W_S$ : The switching delay from the HOL queue to its destination output port; that is, the time duration from the arrival time of the first cell at the HOL queue to the departure time of the last cell from the HOL queue.
3. $W_O$ : The packet waiting time at the output buffer from the arrival time of the first cell of the packet at the output buffer to its departure time from the output buffer.

It is assumed that the cell transmission from the HOL queue is performed by a random discipline for cells arriving in the same slot, and by a FIFO discipline for cells arriving in different slots. In Subsections 2.4.1, 2.4.2 and 2.4.3, we will derive the above three elements.

### 2.4.1 Switching Delay

For obtaining the switching delay $W_S$, we examine the cell transmission behavior of the tagged packet arriving at the HOL queue. Let $u_m$ be the probability that the number of packets waiting in the HOL queue including the just arriving tagged packet equals $m$, which is obtained as

$$u_m = \sum_{n=0}^{N_O} \sum_{j=1}^{m} r_{n,m-j} {a'}_j,$$

where ${a'}_j$ is the probability that the tagged packet arrives with $j$ packets in the same slot; that is,

$${a'}_j = \frac{j a_j}{\sum_{k=1}^{\infty} k a_k} = \frac{j a_j}{\lambda_p}.$$

In what follows, we will refer a cycle to the time to transfer all cells of the tagged packet from the HOL queue to the output buffer.

Suppose now that there are $m$ packets including the tagged one in the HOL queue at the beginning of the cycle, that $j$ packets of them have more cells to transfer, and that $(m' - 1 - j)$ packets newly arrive at the HOL queue during the cycle. In this case, the transition probability $t_{m,m'}$ is given as

$$t_{m,m'} = \sum_{j=0}^{m'-1} b_{m-1,j} a^m_{m'-1-j},$$

where $a^m_k$ is defined as the probability that $k$ packets arrives at the HOL queue during $m$ slots; that is,

$$a^m_k = \frac{(\lambda_p m)^k e^{-\lambda_p m}}{k!}.$$

Let $T_{m,m'}(k)$ be the cycle time distribution when $m$ HOL cells exist at the beginning of the cycle, and when there are $m'$ HOL cells at the beginning of the next cycle. Using the above probability $t_{m,m'}$, $T_{m,m'}(k)$ is expressed as follows.

$$T_{m,m'}(k) = \begin{cases} t_{m,m'}, & \text{if } k = m \\ 0, & \text{otherwise} \end{cases} .$$

By letting $T^l_{m,m'}(k)$ be the distribution over $l$ cycles, we have

$$T^l_{m,m'}(k) = \sum_{j=1}^{\infty}[T^{l-1}_{m,j} \otimes T_{j,m'}](k),$$

where the symbol $\otimes$ is the convolution operator of two probability distributions; that is, for two probability distributions $y_1(k)$ and $y_2(k)$, it is defined as

$$[y_1 \otimes y_2](k) \equiv \sum_{j=0}^{k} y_1(j)y_2(k-j).$$

Next, let $U_m(k)$ represents the delay distribution of the last cell of the tagged packet. Because of our assumption that the cell transmissions are done by a random discipline among cells arriving at the HOL queue in the same slot, we have

$$U_m(k) = \begin{cases} 1/m, & \text{if } 0 \leq k \leq m-1 \\ 0, & \text{otherwise} \end{cases}.$$

We further introduce $W_m(k)$ that is denoted as the transmission time distribution of the tagged packet conditioned on $m$, which is the number of HOL packets when the tagged packet arrives at the HOL queue. Recalling that the packet length (the number of cells in the packet) follows a geometric distribution with parameter $p$, $W_m(k)$ is given by

$$W_m(k) \quad = \quad (1-p)U_m(k) + \sum_{l=1}^{\infty} p^l(1-p)\sum_{j=1}^{\infty}[T^l_{m,j} \otimes U_j](k).$$

Hence, the mean switching delay $W_S$ is obtained as

$$W_S = \sum_{m=1}^{\infty}\sum_{k=1}^{\infty} kW_m(k)u_m.$$

### 2.4.2 Packet Waiting Time at Input Buffer

In order to obtain $W_I$, we first consider the random variable $W_H$, the time from when the first cell of the packet arrives at the HOL queue to when all cells belonging to the same packet are transferred to the output buffer. The derivation of distribution for $W_H$ is similar to that of $W_S$, but in addition to the state of the HOL queue, the state of the output buffer should be taken into account. Let $u_{n,m}$ be the probability that there are $m$ packets in the HOL queue and $n$ cells in the output buffer at the arriving instant of the tagged packet. It is determined as

$$u_{n,m} = \begin{cases} \sum_{j=1}^{m}(r_{0,m-j} + r_{1,m-j})a'_j, & \text{if } n = 0 \\ \sum_{j=1}^{m} r_{n+1,m-j}a'{}_j, & \text{otherwise} \end{cases}$$

We define $C_{n,m,n',m'}(k)$ as the probability distribution of a cycle time that the state was $(n,m)$ at the beginning of a cycle, and that the state becomes $(n',m')$ at the beginning of the next cycle. It is noted that the current definition of the cycle is different from that in the previous subsection in the sense that it is observed at the HOL queue. More precisely, when the output buffer has space to accept, say three cells, three cells can be transmitted simultaneously in one slot from the HOL queue if these exist, and in the current definition of the cycle, it is counted as one slot. On the other hand, in the previous subsection, it is counted as three slots to derive the switching delay. $C_{n,m,n',m'}(k)$ is obtained dependent on $m$ and $n$ as follows.

- $m \leq N_O - n$

  Since all HOL cells can be transferred to the output buffer, the cycle time is just one slot. The state of the output buffer then becomes $n' = n+m$. On the other hand, the number of HOL packets becomes $m' = j+k+1$ when $j$ of HOL packets (except the tagged one) have more cells to transfer and when $k$ packets newly arrive in the current cycle. Consequently, we have

$$C_{n,m,n',m'}(k) \quad = \quad \begin{cases} \sum_{j=0}^{m'} b_{m-1,j} a_{m'-1-j}, & \text{if } k = 1 \text{ and } n' = n+m \\ 0, & \text{otherwise} \end{cases} .$$

- $m > N_O - n$

  $(N_O - n)$ cells are transferred to the output buffer in one slot, and the other $(m-(N_O-n))$ cells are transferred continuously in the following slots. Therefore, the cycle time is $(1 + m - (N_O - n))$, and the state of the output buffer becomes $n' = N_O$. When $j$ packets of $(m-1)$ HOL packets have more cells to transfer and when $k$ packets arrive at the current cycle, the number of HOL packets becomes $m' = k+1$. Therefore, we have

$$C_{n,m,n',m'}(k) \quad = \quad \begin{cases} \sum_{j=0}^{m'} b_{m-1,j} a_{m'-1-j}^{m-(N_O-n)+1}, & \text{if } k = m - (N_O - n) \text{ and } n' = N_O \\ 0, & \text{otherwise} \end{cases} .$$

The cycle time distribution over $l$ cycles is then obtained as

$$C_{n,m,n',m'}^{l}(k) = \sum_{n''=0}^{N_O} \sum_{m''=1}^{\infty} [C_{n,m,n'',m''}^{l-1} \otimes C_{n'',m'',n',m'}](k).$$

Let $U_{n,m}(k)$ be the delay distribution of the last cell of the packet in the cycle. Because of our assumption that the cell transmission is done by a FIFO discipline among cells arriving in distinct slots, $U_{n,m}(k)$ is given as follows.

- $m \leq N_O - n$

$$U_{n,m}(k) = \begin{cases} 1, & \text{if } k = 0 \\ 0, & \text{otherwise} \end{cases}$$

- $m > N_O - n$

$$U_{n,m}(k) = \begin{cases} (N_O - n)/m, & \text{if } k = 0 \\ 1/m, & \text{if } k \leq m - (N_O - n) \\ 0, & \text{otherwise} \end{cases}$$

Probability distribution of $W_H$ is obtained as

$$W_H(k) = \sum_{n=0}^{N_O} \sum_{m=1}^{\infty} u_{n,m} \Big[ (1-p) U_{n,m}(k) + \sum_{l=1}^{\infty} p^l (1-p) \sum_{n'=0}^{N_O} \sum_{m'=1}^{\infty} [C_{n,m,n',m'}^{l} \otimes U_{n',m'}](k) \Big].$$

The corresponding $n$th moment $W_H^{(n)}$ is then given by

$$W_H^{(n)} \quad = \quad \sum_{k=1}^{\infty} k^n W_H(k).$$

Finally, by considering a Geom/G/1 queuing system where the first and second moments of the service time are given by $W_H^{(1)}$ and $W_H^{(2)}$, respectively, we have (see, e.g., [63])

$$W_I = \frac{\lambda_p W_H^{(2)}}{2(1 - \lambda_p W_H^{(1)})}.$$

### 2.4.3 Packet Waiting Time at Output Buffer

Since $W_O$ means the delay of the first cell of the packet in the output buffer, we simply have

$$W_O = 1 + \sum_{n=1}^{N_O} \sum_{m=0}^{\infty} nr_{n,m},$$

which includes the transmission time of the last cell.

### 2.4.4 Numerical Examples

Figures 2.14 and 2.15 show relations between the offered load and the average packet delay for $\overline{BL} = 1$ and $\overline{BL} = 3$, respectively, for various values of output buffer size $N_O$. In Fig. 2.15, simulation results for the switch size $N = 16$ are also provided due to computational complexity of our analytic approach. In simulation, we have set the switch size $N$ to 16 in obtaining the results for larger $N_O$'s. These figures show that the high offered load suddenly increases the average packet delays, which becomes saturated at the point where the offered load reaches the maximum throughput. Inversely, if we use an appropriate size of the output buffer, it would be possible to sustain increase of the average packet delay as having shown in Subsection 2.3 (see Fig. 2.5), but it is limited as the mean packet length becomes large. To validate our analytic method, we provide simulation results as well as analytic ones in Fig. 2.16 for $N_O = 5$ and $\overline{BL} = 1$ and $\overline{BL} = 3$. It can be found that our analysis gives slightly larger value than simulation. It is just because the switch size $N$ is assumed to be infinite in our analysis.

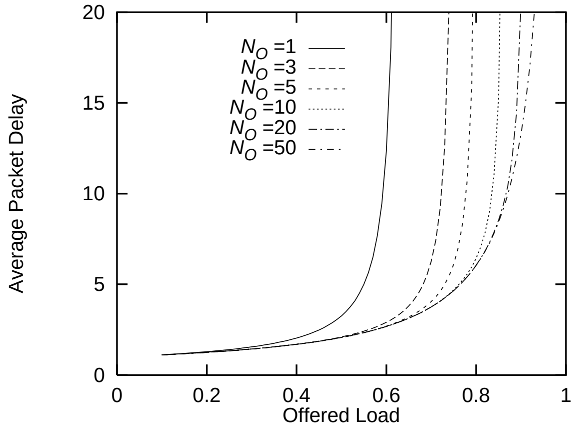

Figure 2.14: Average Packet Delay vs. Offered Load for $\overline{BL} = 1$.

## 2.5 Approximate Analysis of Packet Loss Probability

In this section, the packet loss probability is derived by utilizing a Gaussian approximation. In addition to the FIFO switch considered above, the RIRO (Random-In-Random-Out) switch [14] is also considered for comparison. In the RIRO switch, in order to avoid the HOL blocking, all cells at each input buffer are stored in logically separated buffers, each of which is associated with the destination output port. The packet loss probabilities for these two switches are approximately derived in the followings.

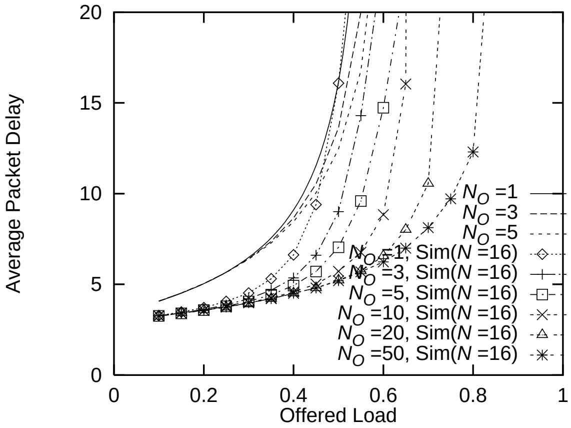

Figure 2.15: Average Packet Delay vs. Offered Load for $\overline{BL} = 3$.

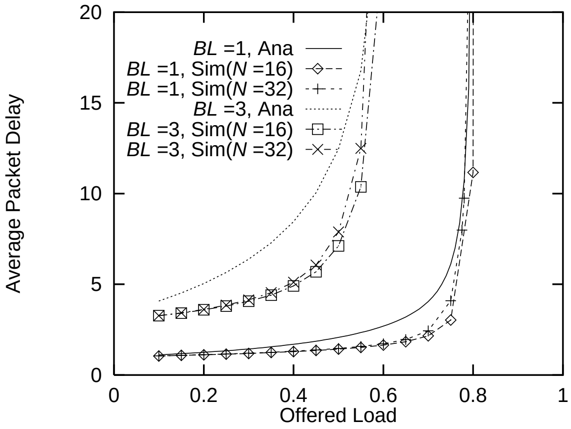

Figure 2.16: Comparison with Simulation Results.

### 2.5.1 Case of FIFO Switch

At first, we consider a discrete time Geom / G / 1 queuing system where packet interarrival times follow a geometric distribution with parameter $\lambda_p$. We define $\Lambda(z)$ as the probability generation function (PGF) for the distribution of the number of packets arriving in a slot, which is given by

$$\Lambda(z) = 1 - \lambda_p + \lambda_p z.$$

Furthermore, we let $B(z)$ be the PGF of probability distribution of the service time of the customers. Its $i$th derivative is defined by $b^{(i)}$; that is,

$$b^{(i)} \equiv \left. \frac{d^i B(z)}{dz^i} \right|_{z=1}.$$

The PGF of the unfinished work for this system is given by (see, e.g., [63])

$$U(z) = \frac{(1-\rho)(1-z)\Lambda[B(z)]}{\Lambda[B(z)] - z},$$

where $\rho$ is the utilization obtained as

$$\rho = \lambda_p b^{(1)}.$$

The average and the variance of $U(z)$ is derived as

$$\begin{aligned} E[U] &= \left. \frac{dU(z)}{dz} \right|_{z=1} \\ &= \frac{\lambda_p b^{(2)} + \lambda_p^{(2)} (b^{(1)})^2 + \rho(1-2\rho)}{2(1-\rho)} \\ V[U] &= E[U^2] - E[U]^2, \end{aligned}$$

where $E[U^2]$ is given by

$$E[U^2] = \left. \frac{d^2 U(z)}{dz^2} \right|_{z=1} + E[U]$$

In the FIFO switch, we can view the number of cells in the input buffer as the unfinished work. Therefore, the packet loss probability $P_L$ is approximately given as

$$P_L(FIFO) \cong Pr[U > N_I] = \int_{N_I}^{\infty} \frac{1}{\sqrt{2\pi V[U]}} e^{-\frac{(y-E[U])^2}{2V[U]}} dy, \qquad (2.12)$$

where $N_I$ represents the buffer size. The probability distribution of $W_H$ obtained in Section 2.4 can be applied to Eq. (2.12) for the moments of the service time distribution. Namely, $b^{(i)}$'s $(1 \le i \le 3)$ are given by

$$\begin{aligned} b^{(1)} &= W_H^{(1)} & (2.13) \\ b^{(2)} &= W_H^{(2)} - W_H^{(1)} & (2.14) \\ b^{(3)} &= W_H^{(3)} - 3W_H^{(2)} + 2W_H^{(1)}. & (2.15) \end{aligned}$$

### 2.5.2 Case of RIRO Switch

We assume that each input buffer is composed of $N$ Geom / G / 1 queues, each of which is associated with the output port. We further assume that each queue is served independently. This assumption is realistic if the switch performs an appropriate cell transmission scheduling [14]. Furthermore, by assuming balanced traffic load condition, the mean packet arrival rate at the $j$th queue at the input buffer (dedicated to the output port $j$) is given as

$$\lambda_j = \frac{\lambda_p}{N}.$$

By letting $\Lambda_j(z)$ be the $z$-transform for the number of packets arriving in a slot, we have

$$\Lambda_j(z) = 1 - \lambda_j + \lambda_j z.$$

We define $V_j$ as a random variable for the number of cells waiting at the $j$th queue in the input buffer. To prevent a single queue from occupying the whole input buffer, the threshold value $T_h$ is introduced for all queues, and the packet loss probability due to this threshold value $T_h$ is given by

$$P(T_h) \cong Pr[V_j > T_h]$$

The packet service time distributions for each queue are obtained from Eq. (2.15) by letting $\lambda$ be $\lambda_j$.

Next, let $U_N$ be the random variable to represent the unfinished work defined as

$$U_N = \sum_{j=1}^{N} V_j.$$

By introducing $U_N(z) = V_j(z)^N$ for the PGF of $U_N$, the average and the variance of $U_N$ are obtained as follows.

$$\begin{aligned} E[U_N] &= \left.\frac{dV_j^N(z)}{dz}\right|_{z=1} \\ V[U_N] &= E[{U_N}^2] - E[U_N]^2 \end{aligned}$$

Let $P_L$ denote the probability that the number of cells at the input buffer exceeds the physical buffer size $N_I$, we have

$$P_L(RIRO) \cong Pr[\lim_{N\to\infty} \sum_{j=1}^{N} V_j > N_I] = Pr[\lim_{N\to\infty} U_N > N_I].$$

Consequently, the packet loss probability for the RIRO switch, $P(RIRO)$, is obtained as

$$P(RIRO) \cong \max(P(T_h), P_L(RIRO)).$$

### 2.5.3 Numerical Examples

In Figs. 2.17 and 2.18, packet loss probabilities dependent on the offered load are plotted for $\overline{BL} = 1$ and $\overline{BL} = 3$, respectively. For comparison purposes, we also provide the result of the output buffer switch [60]. Here, we set $N_I + N_O = 30$ in the cases of FIFO and RIRO switches and $N_O = 30$ in the case of the output buffer switch. In both cases of FIFO and RIRO switches, the higher offered load results in sudden degradation of the packet loss probability. The FIFO switch gives the larger packet loss probability than both of the RIRO switch and the output buffer switch for the same buffer size. However, the performance of the FIFO switch can be further improved by a large capacity of the input buffer with low speed memory while the output buffer switch requires high speed buffers at the output ports. It should be noted from Fig. 2.17 that FIFO switch with $N_O = 5$ shows better performance than RIRO switch with $N_O = 1$ when the offered load is rather low. This is explained as follows. Performance degradation of FIFO switch is mainly caused by HOL blocking. However, when the offered load is much lower than the maximum throughput, HOL blocking rarely occurs. Thus, FIFO switch with larger output buffer ($N_O = 5$) gains lower packet loss probability than RIRO switch with smaller output buffer ($N_O = 1$). Of course, if the same amounts of buffers are equipped at input/output buffers, RIRO switch gives higher performance than FIFO switch at the expense of more complicated hardware implementation. Furthermore, RIRO switch gives lower packet loss probabilities than the output buffer switch even though it requires a less amount of high-sped output buffer memory.

Finally, we assess the accuracy of our analytic results by comparing with simulation results. Figures 2.19 and 2.20 illustrate the comparison results for the packet length $\overline{BL} = 1$ and $\overline{BL} = 3$, respectively, for the FIFO switch. Since our approach is based on the Gaussian approximation method, only the small packet loss probabilities are meaningful as indicated in the figures.

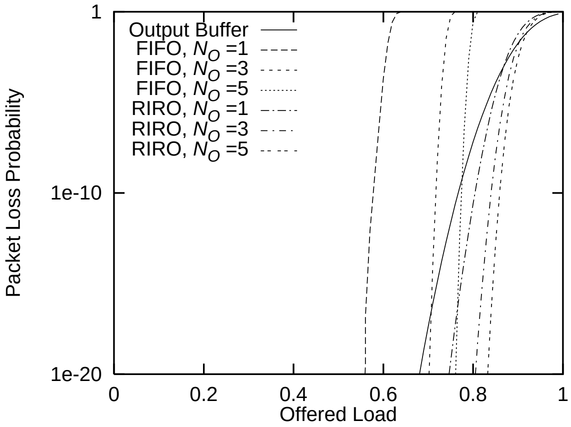

Figure 2.17: Packet Loss Probability vs. Offered Load for $\overline{BL} = 1$.

## 2.6 Conclusion

In this chapter, an ATM switch with input and output buffers equipped with the back-pressure function was treated. We have analyzed its performance under bursty traffic condition for applying it to data communications. We have derived the maximum throughput and the packet delay distribution as well as the approximate packet loss probability under the assumption that the switch size is infinite. Consequently, we have shown that larger packet lengths drastically degrade the performance of the switch. However, it is possible to sustain such a degradation to some extent by providing large output buffers. At least, the output buffer size comparable to the average packet length is necessary to gain a sufficient performance.

Recently, congestion control schemes such as the rate-based congestion control for ABR service class and EPD (Early Packet Discard) for UBR service class have been actively studied

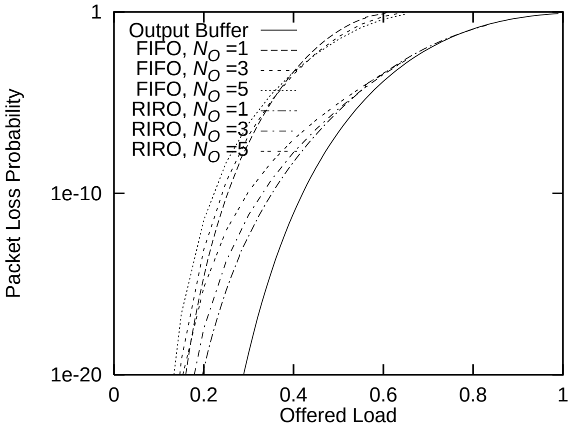

Figure 2.18: Packet Loss Probability vs. Offered Load for $\overline{BL} = 3$.

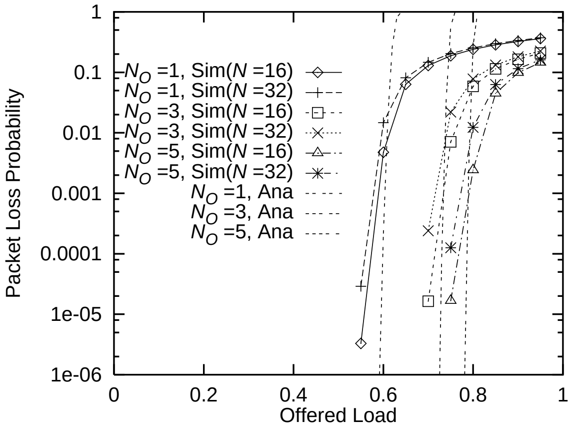

Figure 2.19: Comparison with Simulation Results for $\overline{BL} = 1$.

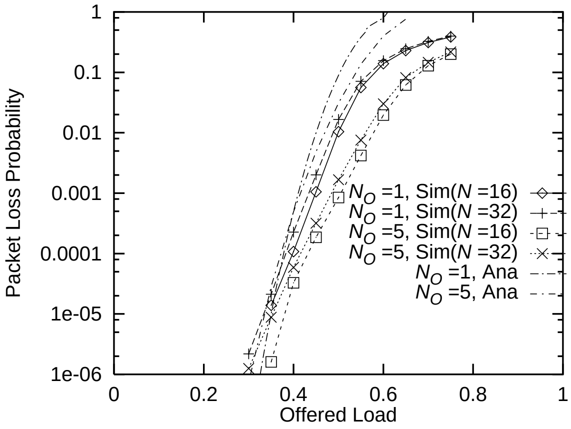

Figure 2.20: Comparison with Simulation Results for $\overline{BL} = 3$.

by many researchers [16, 64]. In most of their studies, the switch architecture is assumed to be ideal. That is, the internal switch speed is enough high so that congestion occurs at the output buffer. Thus, a threshold value associated with a single queue is considered to detect congestion. However, to implement these congestion control schemes in actual, performance limitations caused by the switch architecture should be taken into account as we have discussed in this chapter. For this purpose, our analytic results obtained in this chapter can give the basis to investigate the congestion control mechanism in the ATM layer.

We further note that our analytic approach described in this chapter can be applied to the other cases, for example, the case where the switching speed is $L$ $(1 \leq L \leq N)$ times faster than the link speed (see, e.g., [65]), or the case where when $L'(> L)$ cells are simultaneously destined for the same output buffer, $(L' - L)$ cells are lost or kept awaiting at the input buffer.

For further works, we should evaluate the performance of the network in which two or more ATM switches are interconnected. In such a network, even when a long term congestion introduces large queue length at the input buffer, cell losses may be avoided to send back-pressure signals to the upper adjacent switches.

# Chapter 3

# Performance of Rate-Based Congestion Control Algorithm

The rate-based congestion control promises effective traffic management for the ABR service class suitable to data communications in ATM networks. There have been several switch algorithms of the rate-based congestion control algorithm proposed in the ATM Forum, including EPRCA and EPRCA++ methods. While many studies have been devoted for these schemes in the past, only ABR traffic is taken into account; the effect of VBR and CBR service classes, in which multimedia traffic are accommodated, are not considered. In this chapter, we evaluate performance of these switch algorithms when not only ABR traffic but also VBR traffic is incorporated into the ATM network. Through simulation experiments, we show drawbacks of these switch algorithms for multimedia traffic, and give several suggestions to overcome these problems.

## 3.1 Rate-based control schemes

In this section, we summarize EPRCA and EPRCA++. The former is adopted as a standard mechanism for traffic management scheme for ABR service class in the ATM Forum while the latter is proposed in [40] as an improved version of EPRCA. For more detail, refer to [16].

### 3.1.1 Enhanced Proportional Rate Control Algorithm

We first introduce a basic feature of EPRCA, which is based on a positive feedback mechanism. A source end system periodically sends an RM (Resource Management) cell every $N_{RM}$ data cells to check the congestion status of the network. The RM cell received at the destination end system is returned to the source along the backward path if congestion does not occur in the network. The switch can notify its congestion to the destination end system by marking an EFCI (Explicit Forward Congestion Indication) bit in the header of data cells. As presented in [21], the basic operation of the rate-based congestion control is that the source end system normally increases its allowable cell transmission rate called $ACR$ (allowed cell rate) while it is decreased when the network falls into congestion. A notable feature of EPRCA is, however, that the source end system always decreases $ACR$ until it receives the RM cell from the network. It can increase the rate only when the RM cell is received. The RM cell is discarded at the destination end system if congestion is indicated by the EFCI bit. It results in that the source end system continues to decrease its $ACR$. This *positive feedback* mechanism accomplishes a safer rate reduction at the source end system even if RM cells are lost at the switch because of buffer overflow.

More specifically, the source end system determines the cell transmission rate in the following way. Unless receiving an RM cell, the source determines the next cell transmission time at $1/ACR$ after the current time. This implies that the source continuously decreases its $ACR$ (until receiving an RM cell) as

$$ACR \leftarrow \max(ACR - ADR, MCR). \tag{3.1}$$

When the source receives an RM cell, the rate is increased as

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

which compensates the reduced rate since the source received the previous RM cell ($N_{RM}\,ADR$) and increases the rate by $N_{RM}\,AIR$. In an ideal situation with no propagation delay, this should give a linear increase and an negatively exponential decrease of the cell transmission rate.

In EPRCA, three types of switch architectures with different functions are suggested in the form of pseudo-code. The first one is an EFCI bit setting switch (EFCI) whose operation is described in the above, and it is originally proposed in [27]. The EFCI switch, however, has a fault that it produces unfairness among connections as pointed out in [26].

A capability to select source end systems having large $ACR$ is then added in the second switch called binary enhanced switch (BES). The BES switch maintains a control parameter $MACR$ (Mean ACR) that should ideally be the mean of the $ACR$'s of all active connections. When the rate of all connections is equal to $MACR$, the bandwidth is shared equally and the switch can be fully used without falling into congestion. The key to this is obtaining an accurate $MACR$. The BES switch updates its $MACR$ according to the $CCR$ (Current Cell Rate) field of forward RM cells. For example, $MACR$ is calculated as

$$MACR \quad \leftarrow \quad MACR\,(1 - AV) + CCR \times AV,$$

where $AV$ is used as an averaging factor [30]. When the switch becomes congested, it indicates its congestion to the sources with higher rates. More specifically, the switch marks the CI (Congestion Indication) bit of the backward RM cells if its $CCR$ value exceeds $MACR \times DPF$ (Down Pressure Factor), where a typical value of $DPR$ is 7/8 for safe operation. The switch may remain congested if only the above operation is applied. Therefore when a BES switch becomes *very* congested (such that the queue length exceeds $DQT$), all backward RM cells are marked irrespective of their $CCR$ values. Note that it is evident from the above description that a BECN-like quick congestion notification can be accomplished in BES switches.

A more drastic rate reduction mechanism is adopted by the last one, the explicit down switch (EDS) switch. The EDS switch maintains $MACR$ as the BES switch does, and it controls the transmission rate of sources by setting the $ER$ field of backward RM cells according to a degree of congestion. When a backward RM cell with $CCR$ larger than $MACR$ passes through the congested EDS switch, the value of its ER element is set to $MACR \times ERF$ (Explicit Reduction Factor). If the switch becomes very congested, $MACR \times MRF$ (Major Reduction Factor) is set in all backward RM cells to achieve quick congestion relief, which is called *major reduction*. Typical values of $ERF$ and $MRF$ shown in [30] are 7/8 and 1/4.

### 3.1.2 EPRCA++

EPRCA++ provides a more effective (but expensive) *explicit rate setting*/ mechanism than EPRCA does [40]; the $ER$ value of backward RM cells are directly computed based on the number of active connections and the traffic load at the switch. The number of active connections is kept to be known at the switch by using *per-VC accounting*, which can be implemented in several ways with additional hardware complexity. For instance, each switch can have a VC table to record the number of active connections. Each VC entry is marked or unmarked according to the status of its corresponding VC (active or inactive), and the number of marked entries represents the number of active connections. Furthermore, the switch is provided with an interval timer and it counts the number of cells received during every fixed time interval to monitor the traffic load. The $ER$ value in the RM cell is computed as

$$\begin{aligned}\text{Overload} &= \text{(Input rate)/(Target utilization)}\\ \text{Fair share} &= f_1(\text{Available rate}, \text{\# of active VC's})\\ \text{This VC's share} &= f_2(CCR, \text{Overload})\\ ER &= \max(\text{Fair share}, \text{This VC's share})\\ ER \text{ in Cell} &= \min(ER \text{ in Cell}, ER),\end{aligned}$$

where $f_1$ and $f_2$ are some appropriate functions, e.g., typical functions are;

$$\begin{aligned}f_1(\text{Available rate}, \text{\# of active VC's}) &= \text{(Available rate)/(\# of active VC's)},\\ f_2(CCR, \text{Overload}) &= CCR/\text{Overload}.\end{aligned}$$

By this way, rates of all sources are adjusted through RM cells in one round-trip time when there is one congested switch in the network. The rate of the source is kept unchanged until it receives a backward RM cell in which the explicit rate $ER$ determined by the switch is contained.

One attractive feature of EPRCA++ is its small number of control parameters, which can be set easily by a network manager. Many control parameters required in EPRCA (see Appendix3.4) are eliminated in EPRCA++. Furthermore, in EPRCA+ the target utilization around which the switch is utilized, can be set freely. One may set the target utilization of the switch under 95% link utilization, and then the queue size at the switch is smaller and cell delays are shorter. Although there is an additional expense for timers and the VC table at the switch, EPRCA+ can provide better fairness and responsiveness than EPRCA can [40].

### 3.1.3 Comparison of EPRCA and EPRCA++

In this subsection, performance of EPRCA and EPRCA++ are evaluated for the network model illustrated in Fig. 3.1 as a preliminary study the video traffic is not added to investigate the basic features of the above two algorithms. The propagation delay between the source and destination end systems $\tau$ are set at 0.01 ms, 1.00 ms as typical values for LAN and WAN environments, respectively. The link speed at the switch is set to 156 Mbit/s. There are the number $N_{VC}$ of connections, and the establishment of connections is staggered by 5 ms; that is, $n$th connection starts its cell transmission at $(n-1) \times 5$ ms. Then all connections continue cell transmissions until the end of simulation runs. For control parameters of EPRCA and EPRCA++, we use the values suggested in [30] and [40], respectively (see 3.4). Each simulation is executed during 300 ms.

Figures 3.2 through 3.4 show the allowed cell rates $ACR$ of selected connections, the aggregate rate of all connections, and the queue length at the switch for EPRCA with EFCI, BES and EDS switches. The number of connections, $N_{VC}$, and the the propagation delay $\tau$ are set to 10 and 0.01 ms (as LAN environment), respectively. The case of EPRCA++ is also shown in Fig. 3.5. From these figures, one can easily find that the EFCI switch shows slower rate increase

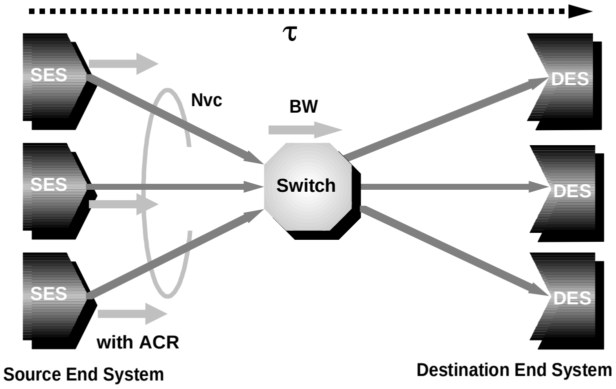

Figure 3.1: Configuration of Simulation Model.

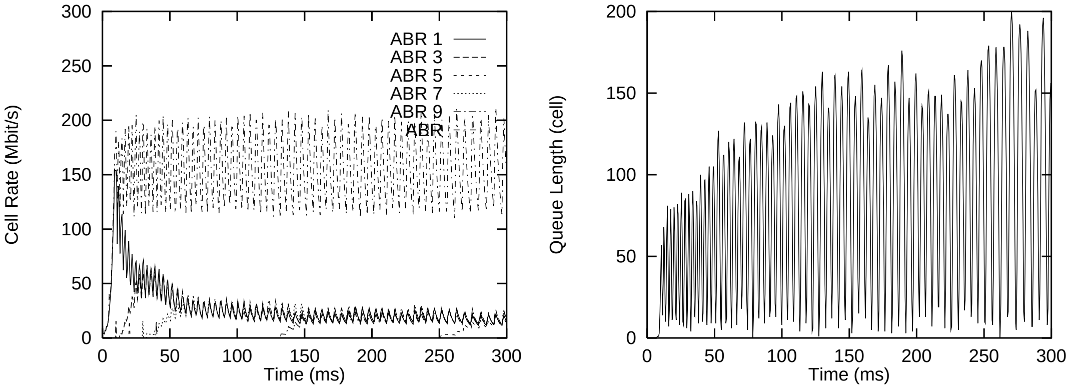

Figure 3.2: EPRCA with an EFCI Switch ($N_{VC} = 10, \tau = 0.01$).

than others. This is because the EFCI switch uses non-selective feedback; connections with lower rate as well as other connections with higher rate are forced to decrease its rate in congestion in the same manner. In addition, since the increase rate is proportional to the current cell rate, the rate is increased slower than others (in this simulation, we use $ICR = PCR/10$. Furthermore, among three switches of EPRCA, the magnitude of amplitudes in the EFCI switch is much larger than other two switches. It is also found that EPRCA++ gives an extremely stable operation compared with EPRCA, i.e., the queue length in the case of EPRCA++ is very small. It is owing to a precise explicit rate setting capability of EPRCA++.

EPRCA++, however, causes a problem as the propagation delay becomes large. In Figs. 3.6 through 3.9, simulation results for EPRCA and EPRCA++ are plotted for $\tau = 1.00$ ms. It can be observed that the maximum queue length of EPRCA++ grows continuously with no limitation. Even with the EFCI switch, the maximum queue length in the simulation run is 976 (in cells). The reason is that EPRCA++ determines the explicit rate $ER$ based on the old information (i.e., $CCR$ in the RM cells). As has been described in Subsection 3.1.2, the $ER$ value is computed at the switch as

$$ER = \max(\text{Fair share}, \text{This VC's share}),$$

and “This VC's share” is obtained as

$$\text{This VC's share} = f_2(CCR, \text{Overload}).$$

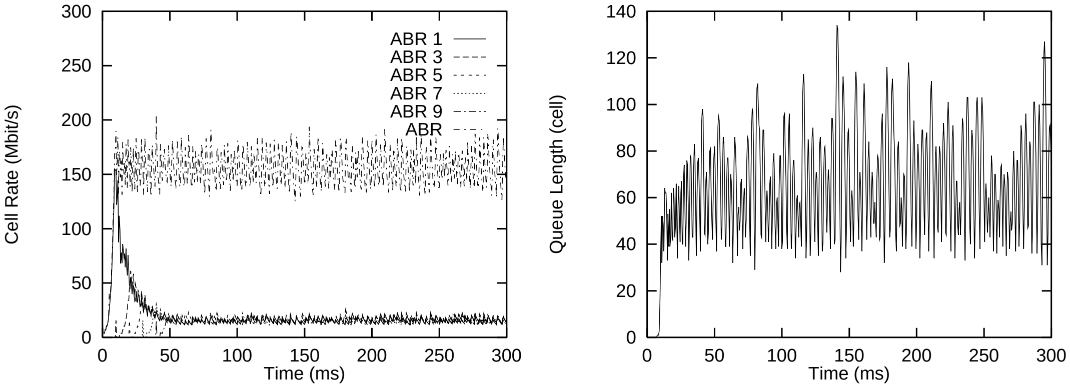

Figure 3.3: EPRCA with a BES Switch ($N_{VC} = 10$, $\tau = 0.01$).

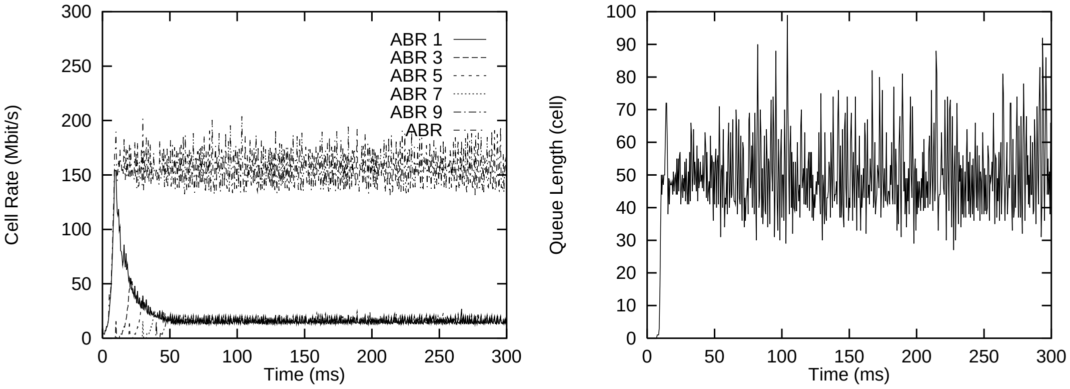

Figure 3.4: EPRCA with an EDS Switch ($N_{VC} = 10$, $\tau = 0.01$).

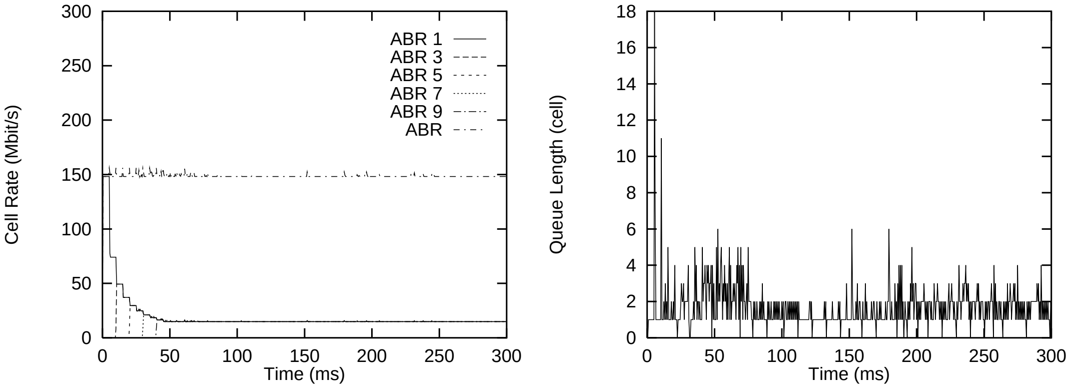

Figure 3.5: EPRCA++ ($N_{VC} = 10$, $\tau = 0.01$).

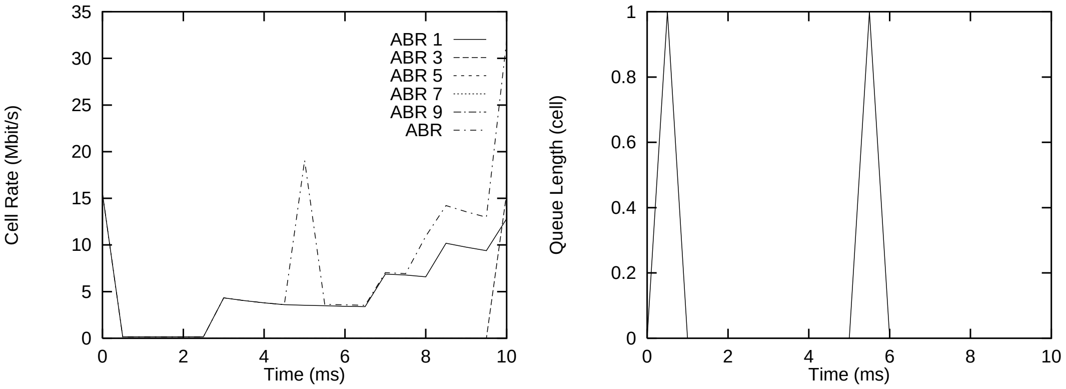

Figure 3.6: EPRCA with an EFCI Switch ($N_{VC} = 10$, $\tau = 1.00$).

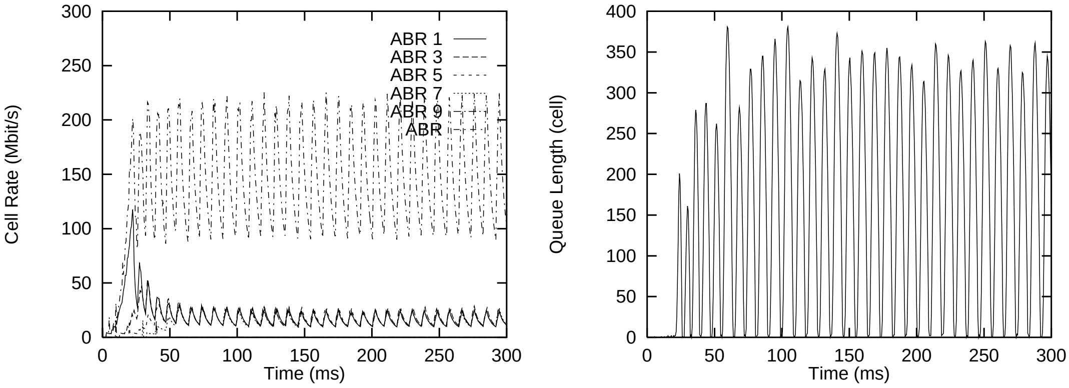

Figure 3.7: EPRCA with a BES Switch ($N_{VC} = 10$, $\tau = 1.00$).

Since the $CCR$ value in the backward RM cell received at the source end system is too old when $tau$ is large, $ER$ value is likely to be set incorrectly. The operation of EPRCA++, however, might be stabilized by, for example, decreasing the target utilization. In Fig. 3.10, we set the target utilization to 0.9 instead of 0.95, which is the value suggested in [40]. EPRCA++ then works well except the initial transient state at the expense of lower link utilization. In general, however, it is not an easy task to optimize such a parameter dependent on the propagation delay. We should also note that we have assumed the cell transmission for every connections is continued during the simulation run. It is only an illustrative purpose, and in actual situation, the active period of connections should depend on characteristics of the traffic source. Although the queue length in EPRCA++ is much smaller than the other schemes if we exclude the initial transient state, the smaller maximum queue length is important to avoid the cell loss from the viewpoint of the buffer dimensioning. Therefore, we may conclude that the EDS switch of EPRCA is still superior to EPRCA++ in the case of large propagation delays.

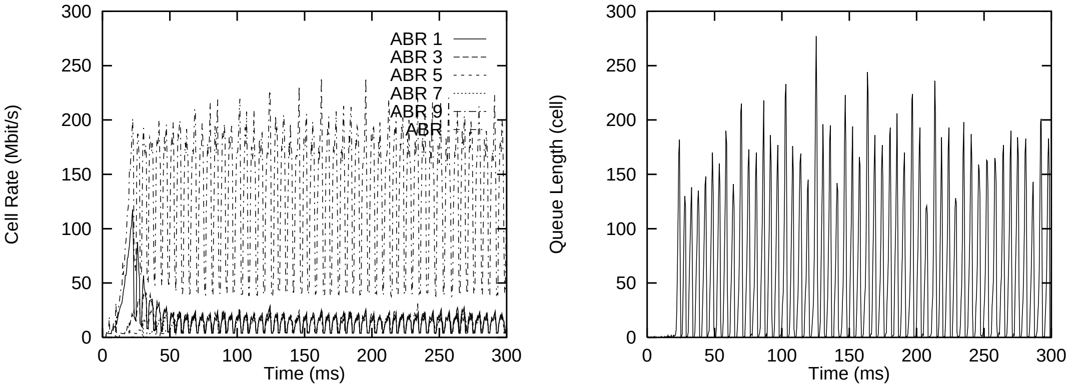

Figure 3.8: EPRCA with an EDS Switch ($N_{VC} = 10$, $\tau = 1.00$).

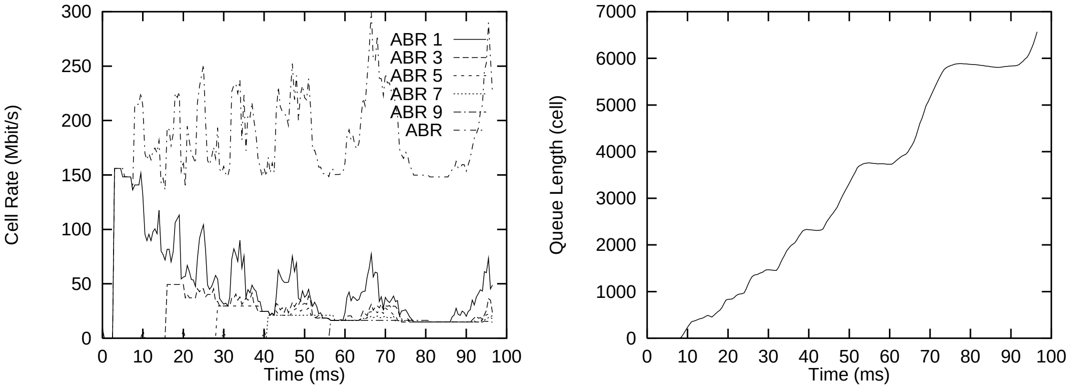

Figure 3.9: EPRCA++ ($N_{VC} = 10$, $\tau = 1.00$, target utilization $= 0.95$).

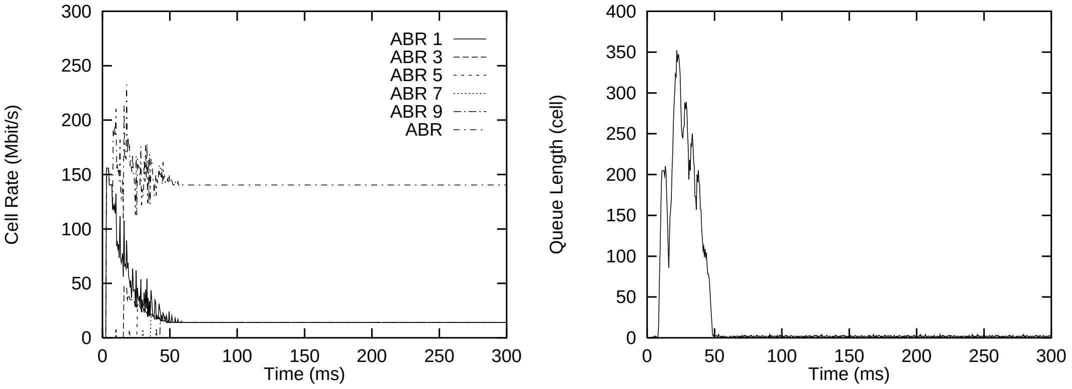

Figure 3.10: EPRCA++ ($N_{VC} = 10$, $\tau = 1.00$, target utilization $= 0.90$).

## 3.2 Effect of VBR traffic

One of attractive capabilities of ATM networks is that it can support multiple QOS's according to requirements of each traffic. As described before, rate-based congestion control has been developed to be applied to the ABR service class, which is suitable for data communications and most of existing applications. Real-time traffic such as voice and motion video is, on the other hand, accommodated into the CBR or VBR service class when multimedia traffic is supported in the ATM network. In spite of this fact, rate-based congestion control methods have been developed without considering these real time traffic classes.

In this section, the effect of VBR traffic on rate-based congestion control schemes is evaluated by a simulation technique using the same model presented in Section 3.1.3. It is assumed that VBR traffic is assigned higher priority than ABR traffic; i.e., VBR traffic cells are transmitted prior to ABR cells at the switch if VBR cells exist in the buffer. Therefore, the bandwidth available to ABR traffic should be affected by the cell generation rate of VBR traffic, which is varied dependent on the time. As a typical example of VBR traffic, we adopt MPEG-1 encoded video stream of 30 frame/s, 352 × 240 pixels with average rate 4.5 Mbit/s and peak rate 14.84 Mbit/s. It means that up to ten video streams can be multiplexed since we assume that the CBR service class is used to transport video streams. In our simulation, ten identical VBR sources are multiplexed with different starting points.

As described in Subsection 3.1.2, EPRCA++ requires information about the bandwidth available to the ABR traffic. If we only consider the ABR traffic, it is identical to the VP capacity, being equal to the physical capacity of the link in most cases. When VBR traffic is also accommodated onto the link, however, we should introduce some method to measure the bandwidth available to the ABR traffic because it is dynamically changed due to the bursty nature of VBR traffic. Since such a method is not described in the original EPRCA++ method [40], we assume that the switch counts incoming VBR cells in a fixed time interval besides input traffic monitoring; That is, the available bandwidth for ABR traffic, $BW'$, is estimated as

$$BW' = BW \times (T - N_{VBR})/T,$$

where $T$ is the averaging interval, $BW$ is the link speed, and $N_{VBR}$ is the number of incoming VBR cells during $T$. In our simulation, $T$ is set to 30 cell times. We note that in the case of EPRCA, such a mechanism is not necessary since the status on the bandwidth utilization is guessed from the queue length.

Simulation results for $N_{VC} = 10$ and $\tau = 0.01$ (as LAN environment) are first presented in Figs. 3.11 through 3.14. The target utilization of EPRCA++ is set to 0.95 as suggested in [40]. $ACR$ of selected connections and the aggregate rate for both of ABR and VBR connections are plotted in these figures. In the figures of EPRCA (Figs. 3.11 through 3.13), the maximum queue length becomes much larger than the ones in the previous case with no VBR traffic (Figs. 3.2 through 3.4) because of reduction of the bandwidth available to ABR connections. We can further observe that the frequency of the rate increase and decrease is directly influenced by the aggregate generation rate of VBR traffic as can be expected. When comparing EPRCA and EPRCA++, it may conclude that the overall performance of ABR connections are not very bad even in the existence of VBR connections, and that EPRCA++ method outperforms other EPRCA methods in LAN environment as in the previous case without VBR traffic.

When the propagation delay becomes large, however, VBR traffic gives a different impact on each scheme. In Figs. 3.15 through 3.18, we show cell rates and the queue length for $\tau = 1.00$ ms as WAN environment. We also illustrate the aggregate throughput of ABR and VBR traffic in Figs. 3.19 through 3.22 to see that under-utilization occurs in these cases. In these figures, the BES switch gives better utilization since (1) the EFCI switch uses FECN-like slower

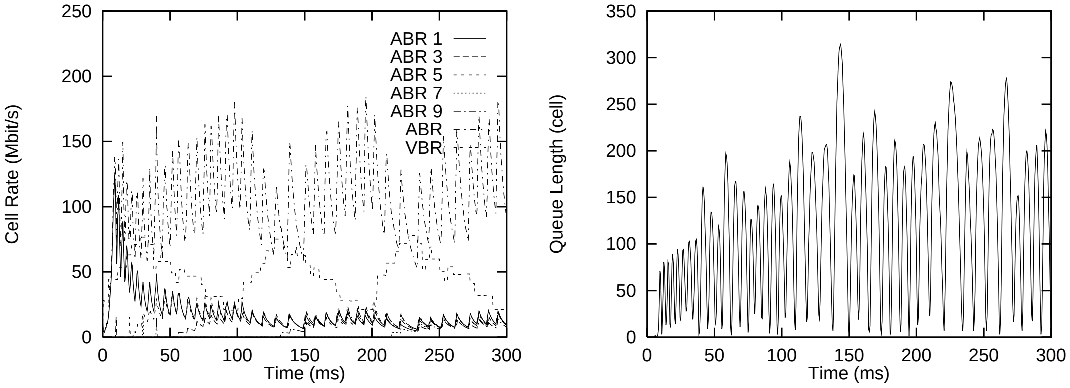

Figure 3.11: EPRCA with an EFCI Switch ($N_{VC} = 10$, $\tau = 0.01$).

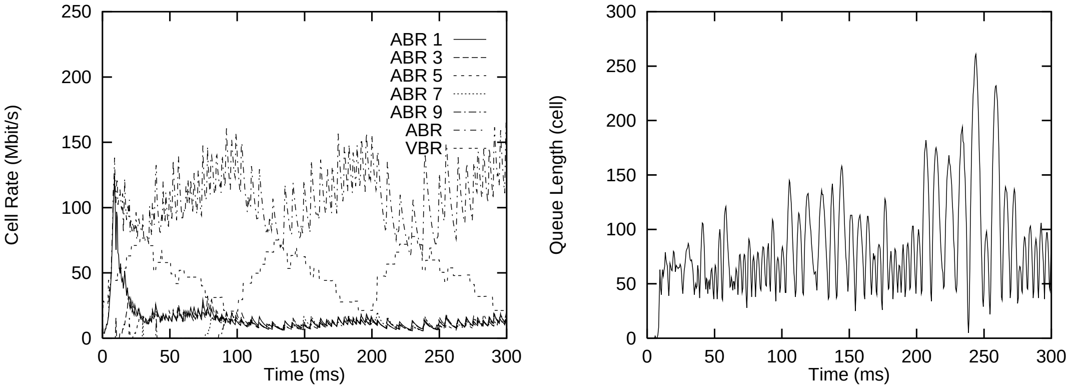

Figure 3.12: EPRCA with a BES Switch ($N_{VC} = 10$, $\tau = 0.01$).

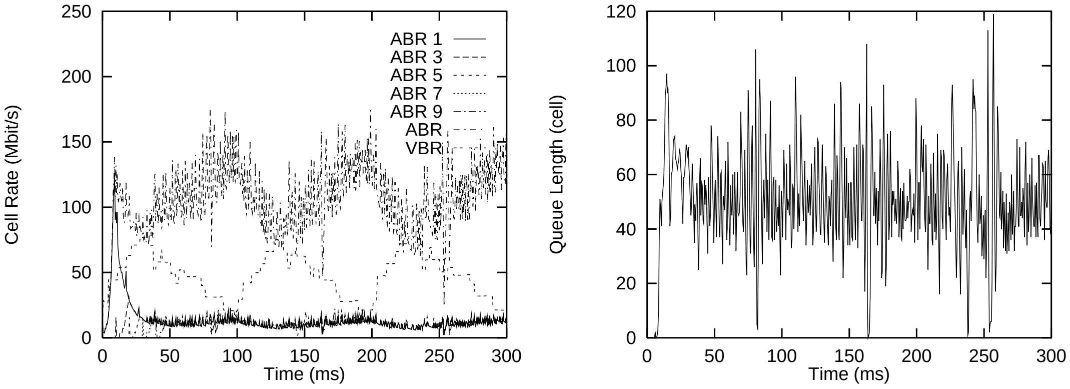

Figure 3.13: EPRCA with an EDS Switch ($N_{VC} = 10$, $\tau = 0.01$).

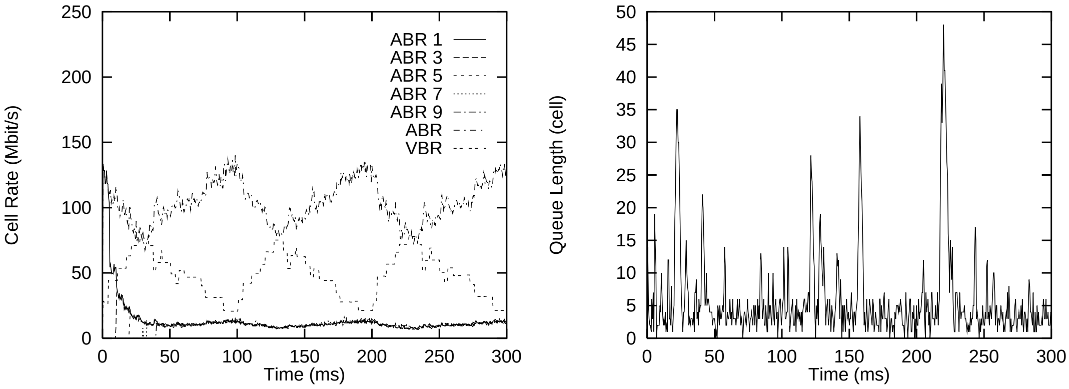

Figure 3.14: EPRCA++ ($N_{VC} = 10$, $\tau = 0.01$).

congestion notification and (2) the EDS switch frequently does major reduction. The performance of the EDS switch may be improved by a proper use of control parameters. However, it implies that too intelligent scheme causes unexpected results unless the careful parameter tuning is performed before applying such a scheme to real systems.

This tendency becomes more apparent when we see the results of EPRCA++. As can be observed in Fig. 3.18, the queue length explosion is unacceptable in the case of EPRCA++ when the target utilization is set to 0.95. The reason why EPRCA++ shows worst performance can be explained as follows. EPRCA++ determines the explicit rate of source end systems ($ER$) by observing the usable bandwidth for the ABR traffic so that it tries to fully utilize the link at the target utilization load. However, since it becomes too old when the RM cell containing the $ER$ value arrives at the source end system in the case of large propagation delays. Therefore, when the cell arriving rate of VBR traffic at the switch grows (around at time 20 ms), the switch becomes overloaded more and more. Recalling that EPRCA++ uses FECN-like congestion notification, the larger $\tau$ introduces more overloaded switch. This problem can be avoided by setting target utilization properly (see Fig. 3.23 in the case of target utilization $= 0.85$). To make the effect of the target utilization clear, we show the maximum queue length of EPRCA++ with (and without) VBR traffic for different values of the target utilization ($\tau = 1.00$, $N_{VC} = 10$) in Fig. 3.24. From this figure, it can be found that the queue length increase rapidly unless the target utilization is set to a proper value, and that a slightly larger value of the target utilization causes a serious effect on the network performance.

In summary, we show the effect of the propagation delay $\tau$ on the maximum queue length in Fig. 3.25 for $N_{VC} = 10$, and effects of the number of connections $N_{VC}$ on the maximum queue length in Figs. 3.26 and 3.27 for $\tau = 0.01$ and $\tau = 1.00$, respectively. From these figures, we may conclude that the EDS switch of EPRCA is of good performance regardless of the network scale, and that EPRCA++ gives almost optimal performance in the LAN environment. Furthermore, it seems to be difficult to apply EPRCA++ to the WAN environment unless control parameters are set carefully.

## 3.3 Conclusion

Rate-based congestion control schemes have been developed in the ATM Forum as means of simple and effective traffic management scheme for ABR traffic. However, there has been lit-

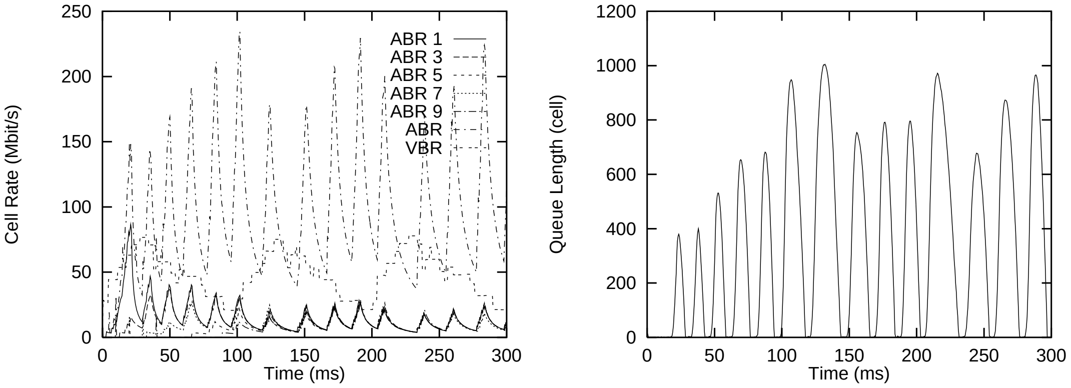

Figure 3.15: Effect of VBR Traffic on EPRCA with an EFCI Switch ($N_{VC} = 10$, $\tau = 1.00$).

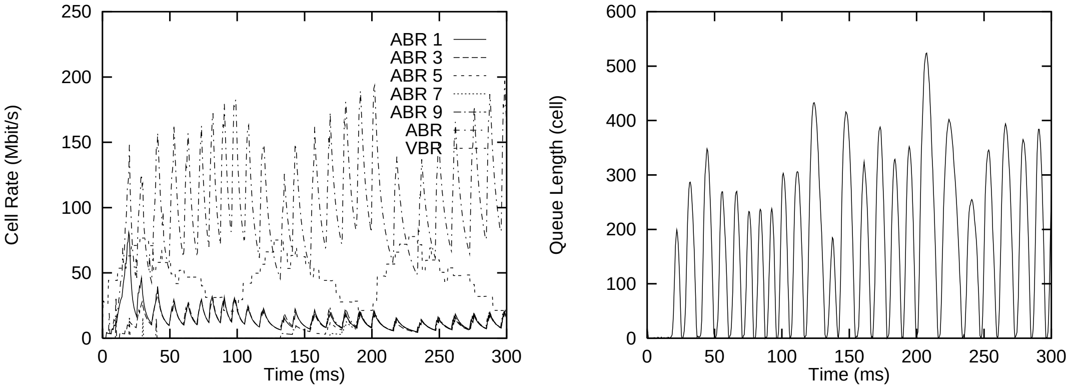

Figure 3.16: Effect of VBR Traffic on EPRCA with a BES Switch ($N_{VC} = 10$, $\tau = 1.00$).

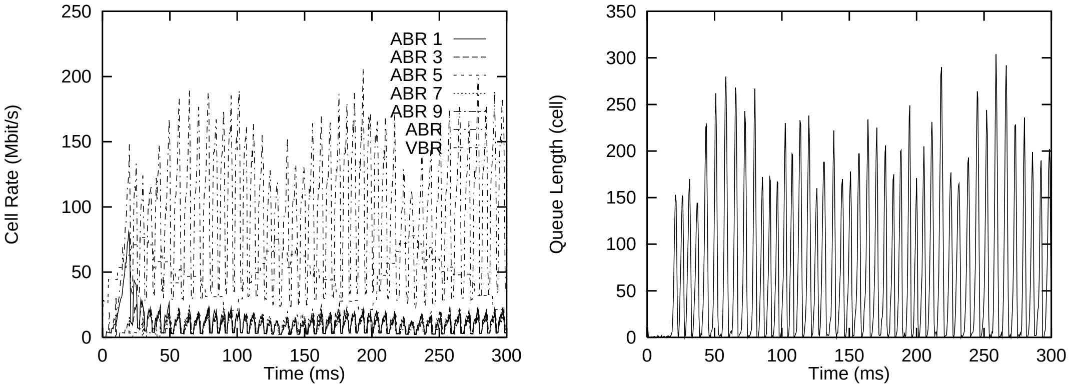

Figure 3.17: Effect of VBR Traffic on EPRCA with an EDS Switch ($N_{VC} = 10$, $\tau = 1.00$).

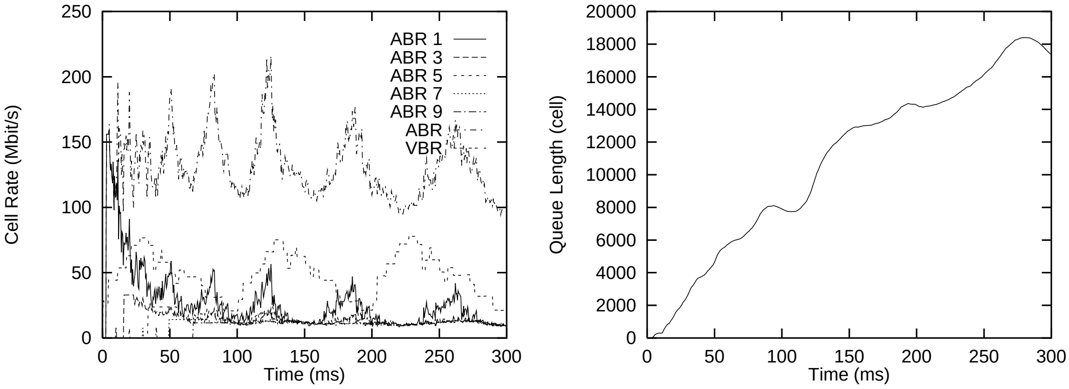

Figure 3.18: Effect of VBR Traffic on EPRCA++ ($N_{VC} = 10$, $\tau = 1.00$, target utilization $= 0.95$).

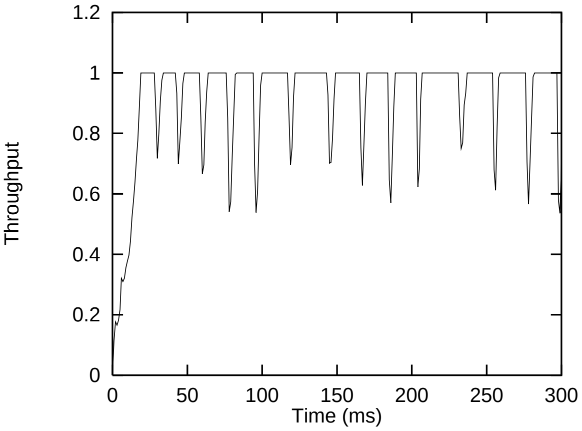

Figure 3.19: Aggregate Throughput of EFCI Switch ($N_{VC} = 10$, $\tau = 1.00$).

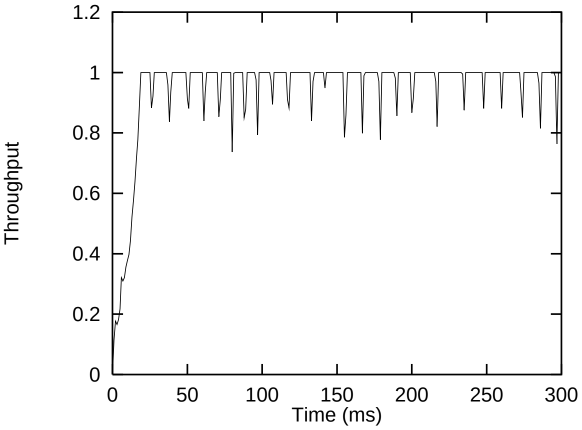

Figure 3.20: Aggregate Throughput of BES Switch ($N_{VC} = 10$, $\tau = 1.00$).

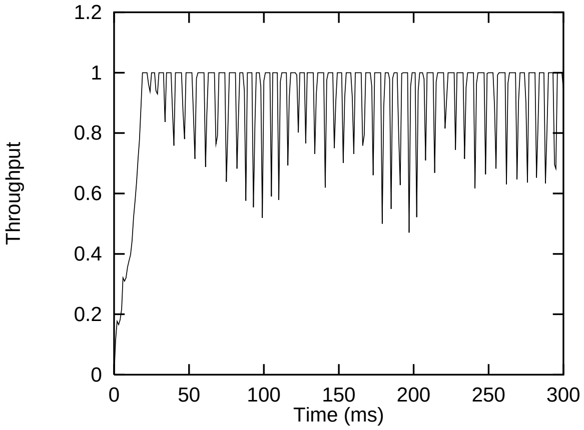

Figure 3.21: Aggregate Throughput of EDS Switch ($N_{VC} = 10$, $\tau = 1.00$).

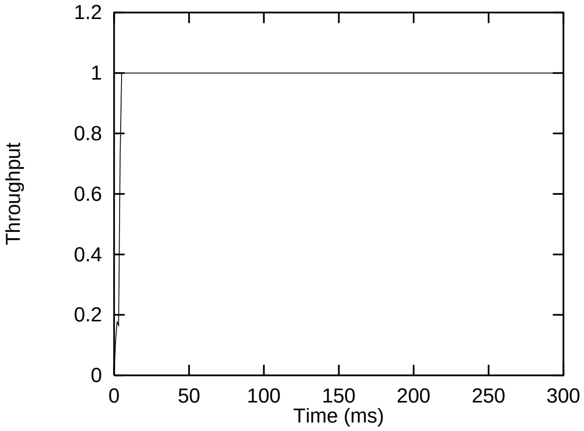

Figure 3.22: Aggregate Throughput of EPRCA++ ($N_{VC} = 10$, $\tau = 1.00$).

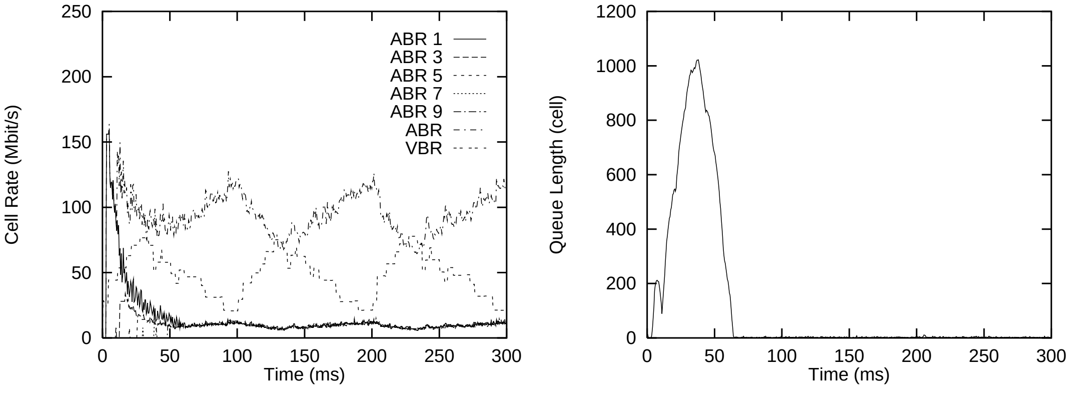

Figure 3.23: EPRCA++ ($N_{VC} = 10$, $\tau = 1.00$, target utilization $= 0.85$).

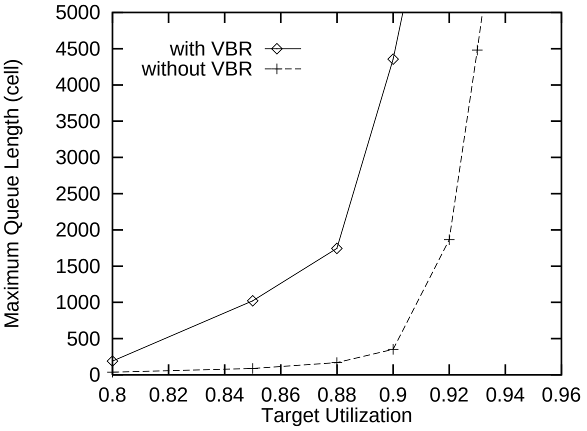

Figure 3.24: Effect of the Target Utilization ($N_{VC} = 10$, $\tau = 1.00$).

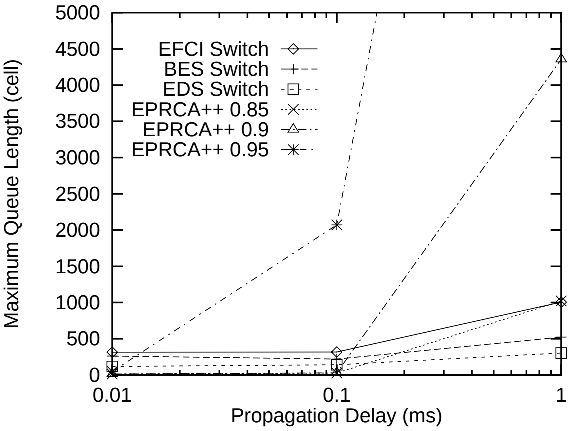

Figure 3.25: Effect of $\tau$ on the Maximum Queue Length with VBR Traffic ($N_{VC} = 10$).

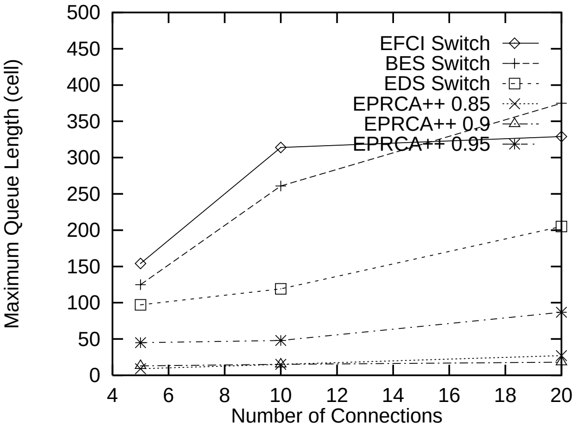

Figure 3.26: Effect of $N_{VC}$ on the Maximum Queue Length with VBR Traffic ($\tau = 0.01$).

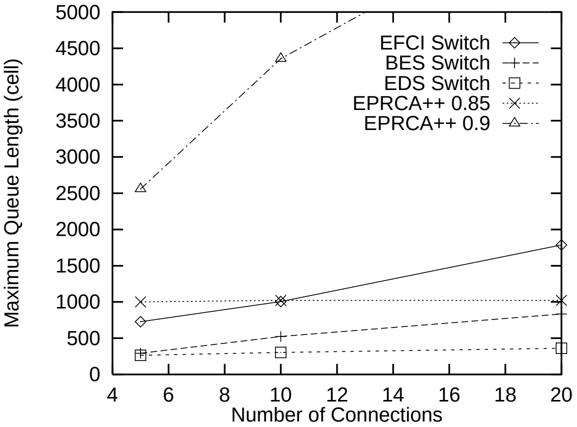

Figure 3.27: Effect of $N_{VC}$ on the Maximum Queue Length with VBR Traffic ($\tau = 1.00$).

tle consideration on VBR and CBR service classes, which are applied for real time traffic. In this chapter, we have evaluated performance of two representative rate-based control schemes — EPRCA, which is a standard scheme, and EPRCA++, a more intelligent scheme — by simulation technique. We first compared these schemes for only ABR traffic, and pointed out a problem that EPRCA++ causes serious queue explosion in WAN environment unless a careful parameter setting is applied. As a typical application of VBR traffic, multiplexed MPEG streams were added on the switch to exhibit how VBR traffic influences the performance of theses schemes. We have shown the effect of VBR traffic on cell emission rates of ABR connections, the maximum queue length, and the throughput at the switch. It should be emphasized that control parameters of complicated schemes should be set carefully in order to achieve effective and stable operation.

## 3.4 Control Parameters

In this section, control parameters used in our simulations are listed.

### Source End System Parameters

| | | |
|---|---|---|
| $PCR$ | 156 Mbit / s | Peak Cell Rate; a maximum rate which $ACR$ can be set |
| $MCR$ | $PCR/1000$ | Minimum Cell Rate; a minimum rate of $ACR$ |
| $ICR$ | $PCR/10$ | Initial Cell Rate; an initial / reset value for $ACR$ |
| $AIR$ | 0.5 | Additive Increase Rate; rate increase permitted |
| $MDF$ | 256 | Multiplicative Decrease Factor; $MDF = 2^{MD}$ |
| $N_{RM}$ | 16 | Number of Cells / RM; $N_{RM} = 2^N$ |

### Switch Parameters

| | | |
|---|---|---|
| — EPRCA — | | |
| $QH$ | 50 | High Threshold |
| $QL$ | 50 | Low Threshold |
| $DQT$ | 100 | High queue limit to determine very congested |
| $VCS$ | 7/8 | VC Separator |
| $AV$ | 1/16 | Exponential Averaging Factor; for averaging $ACR$'s |
| $MRF$ | 1/4 | Major Reduction Factor; for major reduction |
| $DPF$ | 7/8 | Down Pressure Factor |
| $ERF$ | 15/16 | Explicit Reduction Factor |
| — EPRCA++ — | | |
| Averaging Interval | 30 cell | |
| Target Utilization | 0.95 | |

# Chapter 4

# Parameter Tuning of Rate-Based Congestion Control Algorithm

The rate-based congestion control algorithm defines several control parameters for controlling cell emission process of the source end system. The effectiveness of the rate-based congestion control algorithm heavily depends on a choice of these control parameters, but a method for selecting these control parameters has not been specified in the standard.

In this chapter, we analyze the performance of the rate-based congestion control algorithm through applying a first-order fluid approximation to provide control parameter tuning. In the analysis, we focus on two conditions that control parameters should satisfy; one is the prevention of buffer overflow at the switch buffer, and the other is full link utilization. For this purpose, we first obtain the maximum and minimum queue lengths at the switch under the condition that all connections are in a steady state. We next analyze the behavior of a newly established connection by assuming that one or more connections start cell emission while other connections are in a steady-state. Based on this analysis, we discuss proper settings of initial control parameters such as $ICR$ (Initial Cell Rate) to avoid buffer overflow at a switch. Through numerical examples, we demonstrate that our parameter set can satisfy two main objectives — prevention of buffer overflow and full link utilization — in both LAN and WAN environments.

## 4.1 Analytic Model

Our analytic model consists of homogeneous traffic sources and a single bottleneck ATM link as shown in Fig. 4.1. 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 $\tau_{sx}$ and $\tau_{xd}$, respectively. These parameters are varied according to the network configuration (LAN, MAN, or WAN). The round-trip propagation delay between the 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 from the switch to the source end systems via the destination end systems. Propagation delays $\tau_{sx}$ and $\tau_{xd}$ are assumed to be identical for all source and destination pairs.

In the analysis, we assume that each source end system has an infinite number of cells to transmit. That is, the permitted cell rate $ACR$ (Allowed Cell Rate) is always equal to the actual cell transmission rate $CCR$ (Current Cell Rate). Furthermore, since we assume that all connections start cell transmission simultaneously with the same rate-control parameters, all

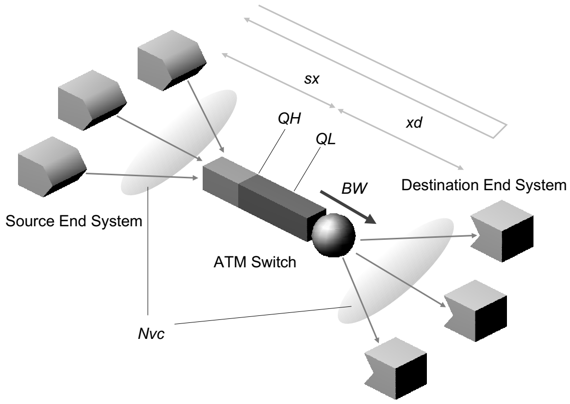

Figure 4.1: Analytic Model.

connections behave identically in our model. By analyzing this model, we will investigate the steady state behavior of the system. Then, the assumptions are relaxed so that a single connection can be added to the network (see Section 4.4).

At the switch, congestion is detected by using 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 each source end system via the corresponding 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 that, when the queue length goes below the low threshold value $Q_L$, the congestion is considered to be relieved. The EFCI bit in the data cells or the CI bit in the forward RM cells is then cleared to indicate to the source end systems via the destination end system that the congestion has been relieved. This sort of switch operation is often referred to as an EFCI bit marking switch, or simply as a binary switch [16]. As mentioned above, we will first assume an infinite buffer capacity in Section 4.2 to examine the steady state behavior. We will then consider the finite case to obtain the condition of no buffer overflow in Sections 4.3 and 4.4. Note that as an optional operation, the CI bit of the backward RM cells may also be marked by the switch when it is congested. While this possibility is not treated in this chapter, the analysis can be easily extended for this case as in [48, 49].

The behavior of end systems has been standardized by the ATM Forum, and it is specified in [3]. In what follows, we will briefly summarize the standard. Each source end system maintains an $ACR$ (Allowed Cell Rate), which is the allowable maximum rate at which the source can emit cells. To adjust the $ACR$ depending on the network congestion status, each source periodically generates forward RM cells in proportion to its rate; it sends one forward RM cell per $(N_{RM} - 1)$ data cells. Upon the receipt of a forward RM cell, the destination end system sends it back as a backward RM cell to the corresponding source end system. The $ACR$ of the source end system is changed in response to the backward RM cells. When the source end system receives the CI bit cleared cell, a rate increase is performed as

$$ACR \leftarrow \min(ACR + RIF \times PCR, PCR), \tag{4.1}$$

where $PCR$ (Peak Cell Rate) is the maximum rate of the connection, which is negotiated when the connection is established. The $RIF$ (Rate Increase Factor) is a ratio for a proportional rate increase. On the other hand, the CI bit marked cell decreases $ACR$ as

$$ACR \leftarrow \max(ACR - ACR \times RDF, MCR), \tag{4.2}$$

where $MCR$ (Minimum Cell Rate) is the minimum rate (or guaranteed rate) for this connection but $MCR$ is usually set to zero. The $RDF$ (Rate Decrease Factor) is a ratio for an exponential rate decrease.

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

$$BW' = BW \frac{N_{RM} - 1}{N_{RM}}, \tag{4.3}$$

which shows the control overhead specified by the ATM Forum. In addition, queuing and / or processing delays of backward RM cells are assumed to be zero in our analysis. That is, these cells are given higher priority than data cells at the switch. Therefore, it is assumed that a backward RM cell is transfered from the destination to the source with a fixed delay $\tau/2$.

When cell queuing takes place at the switch buffer in the forward direction, the rate at which the source end system receives backward RM cells is bounded by $BW/(N_{VC}\, N_{RM})$. Otherwise it is identical to the cell transmission rate at the source end system. Thus we require an analytical treatment different from the one presented in [21] where the rate change was regarded as being performed on a timer basis [16].

## 4.2 Dynamical Behaviors

We first introduce $ACR(t)$ and $Q(t)$ that represent, respectively, the allowed cell rate $ACR$ of each source and the queue length at the switch observed at time $t$. In what follows, the evolution of $ACR(t)$ and $Q(t)$ is analyzed by modifying our previous analysis presented in [48, 49]. In these papers, the behavior of the source and destination end systems were based on [30], which is a rate-based congestion control algorithm proposal. In this chapter, the analysis is modified to reflect the latest standard algorithm described in [3], and we summarize this analysis below.

### 4.2.1 Determination of $ACR(t)$

The evolution of $ACR(t)$ and $Q(t)$ have periodicity in the steady state as shown in Fig. 4.2. The behavior of $ACR(t)$ and $Q(t)$ can be divided into four phases, and we introduce $ACR_i(t)$ and $Q_i(t)$, which are defined as

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

where $t_i$ is defined as the time when Phase $i$ terminates. We further introduce $t_{i-1,i}$ as the interval of Phase $i$, which is defined as $t_i - t_{i-1}$. In each phase, $ACR_i(t)$ and $Q_i(t)$ are determined as follows.

- Phase 1: $ACR_1(t)$

  In this phase, $ACR$ is continuously decreased by receiving backward RM cells with $CI = 1$ from the network. On the receipt of a backward RM cell, $ACR$ is decreased according

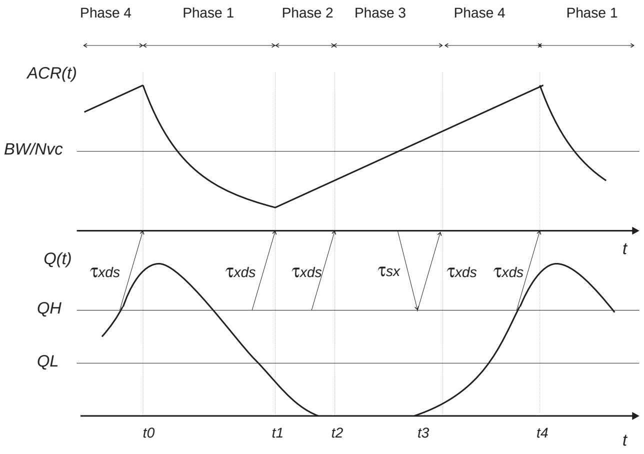

Figure 4.2: Pictorial View of $ACR(t)$ and $Q(t)$.

to Eq. (4.2). Since the bottleneck link is fully utilized, backward RM cells are sent to each source end system at a fixed interval. By letting $x_1$ be the interval of two backward RM cells, we have a relation

$$x_1 = \frac{N_{VC}\, N_{RM}}{BW}. \tag{4.4}$$

From Eqs. (4.2) and (4.4), the differential equation of $ACR_1(t)$ is obtained as

$$\frac{dACR_1(t)}{dt} = \frac{-ACR_1(t)\, RDF}{x_1}. \tag{4.5}$$

By assuming $MCR = 0$, $ACR_1(t)$ is obtained as

$$ACR_1(t) = ACR_1(0) e^{-\frac{BW \times RDF}{N_{VC}\, N_{RM}} t}. \tag{4.6}$$

- Phase 2: $ACR_2(t)$

  Since the switch is not congested in this phase, backward RM cells with $CI = 0$ are transferred to the source end system. Therefore, $ACR$ is increased linearly in the form of Eq. (4.1). The interval of two successive RM cells, $x_2$, is also constant as with the case of Phase 1 (see Eq. (4.4)), and is given as

$$x_2 = \frac{N_{VC}\, N_{RM}}{BW}. \tag{4.7}$$

  Thus, the differential equation of $ACR_2(t)$ is given by

$$\frac{dACR_2(t)}{dt} = \frac{RIF \times PCR}{x_2}. \tag{4.8}$$

  Solving this equation leads to

$$ACR_2(t) = ACR_2(0) + \frac{BW \times RIF \times PCR}{N_{VC}\, N_{RM}} t. \tag{4.9}$$

- Phase 3: $ACR_3(t)$

  Though $ACR$ is increased linearly as in Phase 2, the interval of two successive RM cells $x_3$ varies according to the previous $ACR$. More specifically, $x_3$ is given by the root of the following equation.

$$\int_0^{x_3} ACR_3(y - \tau)dy = N_{RM}. \tag{4.10}$$

  Since it is difficult to solve this equation, we use the following equation as an approximation of $x_3$.

$$x_3 \cong \frac{N_{RM}}{ACR_3(t)} \tag{4.11}$$

  The differential equation of $ACR_3(t)$ is obtained as

$$\frac{dACR_3(t)}{dt} = \frac{RIF \times PCR}{x_3}. \tag{4.12}$$

  Hence $ACR_3(t)$ is approximately given by

$$ACR_3(t) \cong ACR_3(0)e^{\frac{RIF \times PCR}{N_{RM}}t}. \tag{4.13}$$

- Phase 4: $ACR_4(t)$

  $ACR_4(t)$ is given in an equivalent form to $ACR_2(t)$ since the arrival rate of backward RM cells is identical to the case in Phase 2.

### 4.2.2 Evolution of $ACR(t)$ and $Q(t)$

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 the initial rates of Phase $i$, $Q_i(t)$ is obtained as

$$Q_i(t) = \max(Q_i(\tau_{sx}) + \int_{\tau_{sx}}^{t} (N_{VC}\, ACR_i(x - \tau_{sx}) - BW)dx, 0), \quad \tau_{sx} \leq t < \tau_{sx} + t_{i-1,i}. \tag{4.14}$$

The duration of Phase $i$, $t_{i-1,i}$, is defined as

$$t_{i-1,i} = \begin{cases} Q_1^{-1}(Q_L) + \tau_{xds} & i = 1 \\ \min(Q_2^{-1}(Q_H) + \tau_{xds}, Q_2^{-1}(0) + \tau_{xds}) & i = 2, 4 \\ ACR_3^{-1}(BW/N_{VC}) + \tau & i = 3 \end{cases} \tag{4.15}$$

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 [48, 49] for further details on these derivations.

## 4.3 Parameter Tuning in The Steady State

In this section, we determine proper settings of the control parameters for source end systems (such as $RIF$ and $RDF$) and threshold values at the switch ($Q_H$ and $Q_L$) by applying our analysis explained in Section 4.2. In Subsections 4.3.1 and 4.3.2, we analytically obtain two conditions for preventing buffer overflow, and attaining full link utilization, respectively. In Subsection 4.3.3, we show several numerical examples to demonstrate the applicability of binary-mode switches in various network configurations.

### 4.3.1 Condition for Avoiding Buffer Overflow

In Section 4.2, the dynamical behavior of $Q(t)$ was derived by assuming the infinite capacity of the cell buffer at the switch. However, in an actual implementation, its size is limited due to cost or technological limitations, which results in cell loss at the switch buffer. The purpose of this subsection is to derive the condition that lets us avoid buffer overflow at a finite switch buffer. To this end, we first derive the maximum queue length in a closed-form equation, which should be bounded by the buffer size to ensure there is no cell loss.

Let us introduce $Q_{max}$ as the maximum queue length in the steady state. While it is possible to obtain $Q_{max}$ by utilizing the analysis in Section 4.2, it requires an iterative calculation as has been presented in [48]. Instead, we derive a more tractable condition, which is based on the following observation. When the increase / decrease rate is high, $Q(t)$ is likely to oscillate to a large extent; that is, the amplitude of $Q(t)$ becomes large. Thus, as the maximum of $Q(t)$ becomes larger, the minimum of $Q(t)$ eventually 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 seen in Fig. 4.2, $Q(t)$ starts growing at the end of Phase 3, i.e., after $\tau_{sx}$ from when aggregate cell rate ($ACR \times N_{VC}$) exceeds the bottleneck link bandwidth $BW$. In our analysis, the initial values of $ACR(t)$ and $Q(t)$ are set as follows.

$$\begin{aligned} ACR(0) &= \frac{BW}{N_{VC}} && (4.16) \\ Q(\tau_{sx}) &= 0 && (4.17) \end{aligned}$$

The evolution of $ACR(t)$ and $Q(t)$ is then obtained for Phase 3, then for Phase 4 and 1. As shown in Fig. 4.3, $Q(t)$ reaches its maximum in Phase 1, after $\tau_{sx}$ from when $ACR$ falls to $BW/N_{VC}$. Hence, $Q_{max}$ is obtained as

$$Q_{max} = Q_1(t_{max}), \tag{4.18}$$

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

$$t_{max} = t_4 + ACR_1^{-1}(\frac{BW}{N_{VC}}) + \tau_{sx}. \tag{4.19}$$

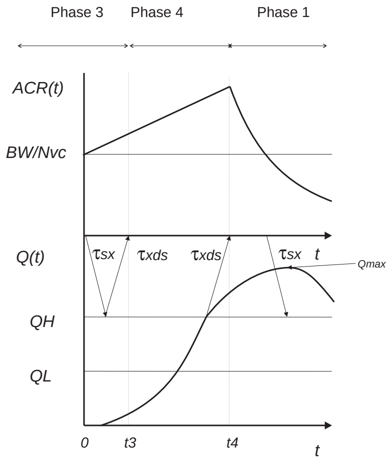

Figure 4.3: Pictorial View of $ACR(t)$ and $Q(t)$.

The initial value of $Q_1(t)$ is required to be $Q_1(0)(= Q_4(t_4))$, and $Q_1(0)$ and $t_4$ can be obtained from Eqs. (4.9), (4.14), (4.16), and (4.17). To simplify the following analysis, Phase 3 is not taken into account; that is, Eqs. (4.16) and (4.17) are used as the initial values of Phase 4. Finally, $Q_{max}$ is obtained in a closed-form equation as (after some manipulation)

$$\begin{aligned} Q_{max} &= Q_H + \frac{N_{VC}}{RDF}\sqrt{\frac{2N_{RM}\,Q_H\,RIF \times PCR}{BW}} \\ &\quad -\frac{N_{RM}\,N_{VC}}{RDF}\log\left(1+\sqrt{\frac{2Q_H\,RIF \times PCR}{BW\,N_{RM}}}+\frac{\tau\,RIF \times PCR}{N_{RM}}\right) \\ &\quad +\tau\left(\sqrt{\frac{2BW\,Q_H\,RIF \times PCR}{N_{RM}}}+\frac{N_{VC}\,RIF \times PCR}{RDF}\right) \\ &\quad +\frac{\tau^2\,BW \times RIF \times PCR}{N_{RM}}. \end{aligned} \tag{4.20}$$

This equation shows that $Q_{max}$ increases as the propagation delay $\tau$ or the number of connections $N_{VC}$ increases. Note that the low threshold value $Q_L$ does not appear in the equation because of our assumption that $Q(t)$ reaches zero; $Q_{max}$ is obtained regardless of the interval of Phase 1.

The condition for avoiding buffer overflow is finally obtained as

$$BL \geq Q_{max}, \tag{4.21}$$

where $BL$ denotes the buffer size at the switch. As stated above, the actual maximum queue length may take a smaller value than the $Q_{max}$ obtained in the above if $Q(t)$ never reaches zero. Actually, a condition that $Q(t)$ does not reach zero, on a similar condition, is necessary to fully utilize the output link, as will be derived in the next subsection. The applicability of our approximation method will be validated in the next section by a comparison with simulation results.

### 4.3.2 Condition for Full Link Utilization

The bottleneck link is fully utilized when the buffer at the switch is never empty. In other words, full link utilization is achieved when the following condition is satisfied.

$$Q_{min} > 0, \tag{4.22}$$

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

$$\begin{aligned} ACR(0) &= \frac{BW}{N_{VC}} && (4.23) \\ Q(\tau_{sx}) &= Q_{min} && (4.24) \end{aligned}$$

In a way similar to the approach taken in Subsection 4.3.1, the evolution of $ACR(t)$ and $Q(t)$ is obtained from Phase 1 followed by Phases 2, 1, and 2 as shown in Fig. 4.4. Again, $Q(t)$ reaches its minimum, $Q_{min}$, during Phase 2. Therefore, $Q_{min}$ should satisfy the following equation.

$$Q_{min} = Q_2(t_{min}), \tag{4.25}$$

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

$$t_{min} = t_1 + ACR_2^{-1}\left(\frac{BW}{N_{VC}}\right) + \tau_{sx}. \tag{4.26}$$

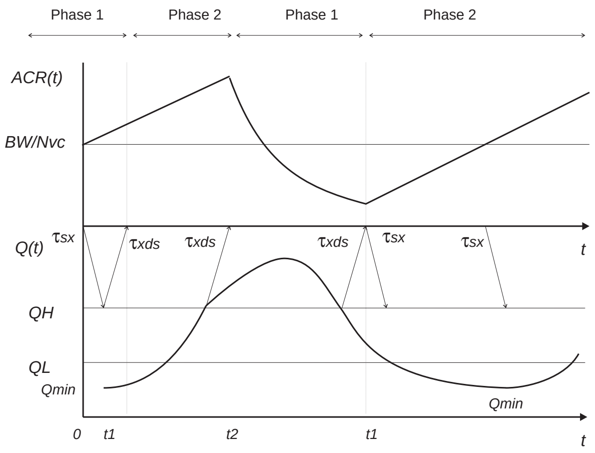

Figure 4.4: Pictorial View of $ACR(t)$ and $Q(t)$.

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

The optimum set of source and switch control parameters exists in the region where Eqs. (4.21) and (4.22) are both satisfied. However, under some conditions, for example, extremely long propagation delays, it may become impossible to satisfy both conditions as will be shown in the following numerical examples.

### 4.3.3 Numerical Examples

In the following numerical examples, the bandwidth of the bottleneck link, $BW$, is set to 353.7 cell / ms assuming a 150 Mbit / s ATM link. At the switch, the buffer size, $BL$, is set to either 300 Kbyte (5,796 cells) or 10 Kbyte (193 cells). Although 10 Kbyte is fairly small for an ATM switch, we use it as an extreme case of buffer shortage. High and low threshold values, $Q_H$ and $Q_L$, are first fixed at the one-half of $BW$, and are then varied to investigate their effects. $N_{RM}$ is fixed at 32.

Figure 4.5 shows the maximum queue length $Q_{max}$ as functions of $RIF$ and $RDF$ for $N_{VC} = 10$ and $\tau = 0.02$ms as in a LAN environment. The $RIF$ is changed from $1/2^0$ to $1/2^{20}$, and the $RDF$ from $1/2^0$ to $1/2^8$. $Q_{max}$ rises from 0 to $BL$ (5,796 cells). This means that the buffer overflow occurs in the upper flat region. The larger $RIF$ is (i.e., the faster the rate increase) or the smaller $RDF$ is (i.e., the slower the rate decrease), the larger $Q_{max}$ becomes. In other words, a small $RIF$ value and a large $RDF$ value are desired to avoid buffer overflow at the switch. However, a smaller value of $RIF$ increases the convergence time of the congestion due to the slow rate increase.

As can be seen from Eq. (4.20), $Q_{max}$ increases as the propagation delay, $\tau$, or the number of connections, $N_{VC}$, increases. In Figs. 4.6 and 4.7, we change $\tau$ from 0.02 ms to 2.00 ms as in a WAN environment, and $N_{VC} = 10$ to $N_{VC} = 50$, respectively. These figures show that $Q_{max}$ exceeds $BL$ more easily than in the case of $\tau = 0.02$ and $N_{VC} = 10$ (Fig. 4.5). Thus, $RIF$ and $RDF$ need to be determined more carefully when $\tau$ and / or $N_{VC}$ is large.

In Fig. 4.8, we show the boundary line of the condition for no cell loss obtained in Eq. (4.21) for $RIF$ and $RDF$. The high and low threshold values, $Q_H$ and $Q_L$, are set to an identical value (denoted by $Q_T$) that is varied between $BL/4$, $BL/2$, and $3BL/4$. The lower region of the boundary line satisfies Eq. (4.21). Figure 4.8 shows that a large $Q_T$ limits the proper parameter region for zero cell loss. Therefore, we conclude that threshold values should be set

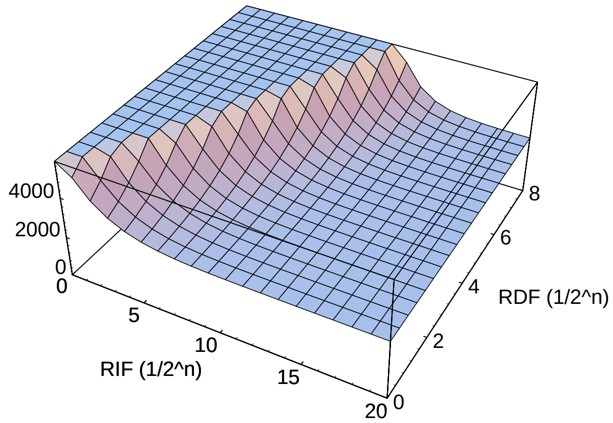

Figure 4.5: Maximum Queue Length for $BL = 300$ Kbyte, $N_{VC} = 10$, and $\tau = 0.02$ ms.

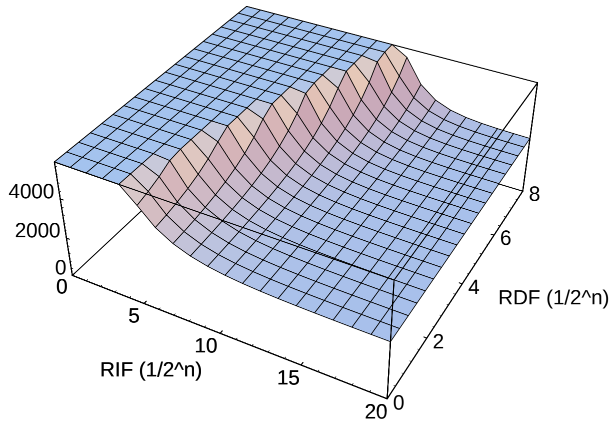

Figure 4.6: Maximum Queue Length for $BL = 300$ KByte, $N_{VC} = 10$, and $\tau = 2.00$ ms.

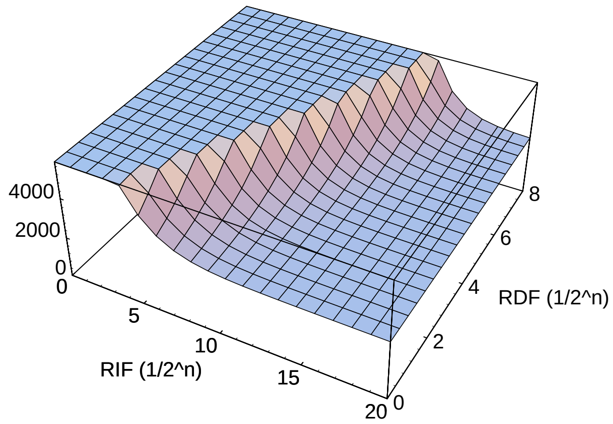

Figure 4.7: Maximum Queue Length for $BL = 300$ KByte, $N_{VC} = 50$, and $\tau = 2.00$ ms.

to a small value to avoid buffer overflow.

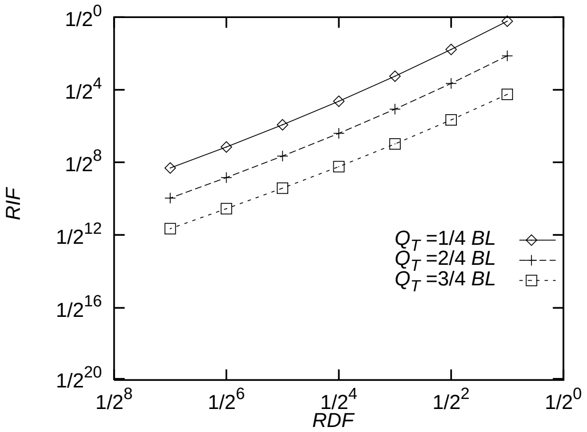

Figure 4.8: Condition for No Buffer Overflow for $BL = 300$ Kbyte, $N_{VC} = 10$, and $\tau = 0.02$ms.

To investigate the effect of the number of connections, $N_{VC}$, we plotted the boundary lines for different values of $N_{VC}$ where $Q_T$ is fixed at $BL/2$ (Fig. 4.9). This figure shows that a small $RIF$ value and a large $RDF$ value are needed to avoid buffer overflow as $N_{VC}$ increases. However, the change in the boundary line is small as $N_{VC}$ increases. Thus, $RIF$ and $RDF$ can be determined regardless of $N_{VC}$ when many connections are constantly active.

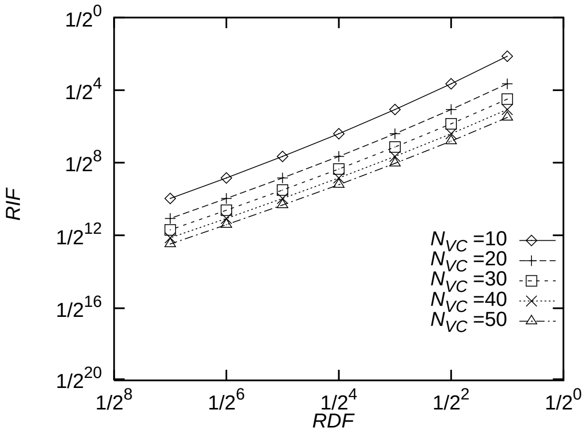

Figure 4.9: Condition for No Buffer Overflow for $BL = 300$ Kbyte, $Q_T = BL/2$, and $\tau = 0.02$ ms.

Next, the effect of the propagation delay, $\tau$, is shown in Figs. 4.10 and 4.11. In both figures, $\tau$ is changed from 0.02 ms to 2.00 ms. The main difference from Figs. 4.8 and 4.9 is found in the region of large $RIF$ and $RDF$; the slope of the boundary line for $\tau = 2.00$ ms is not as steep as that for $\tau = 0.02$ ms. In this case, the queue length tends to increase because of slow notification of congestion from the switch. Thus, even with a large $RDF$ value (i.e., fast rate reduction), $RIF$ should be set to a small value to restrain the queue build-up. We conclude from these observations that a small $RDF$ value should be used for a WAN environment.

Buffer dimensioning is also an important issue. To show the effect of buffer size, we decrease the buffer size, $BL$, to 10 Kbyte (only 193 cells) and show the boundary line of Eq. (4.21) for $\tau = 0.02$ ms and $\tau = 2.00$ ms in Figs. 4.12 and 4.13, respectively. Figure 4.12 shows little dif-

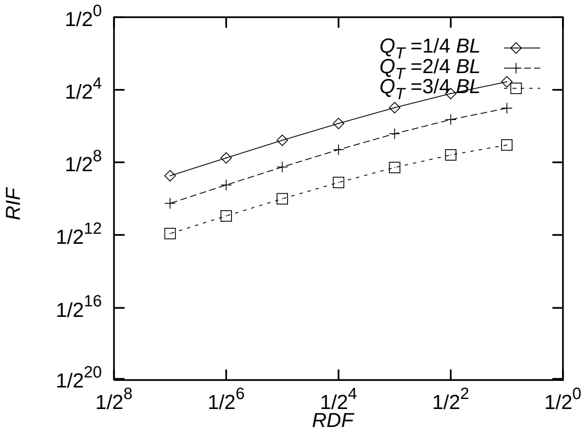

Figure 4.10: Condition for No Buffer Overflow for $BL = 300$ Kbyte, $N_{VC} = 10$, and $\tau = 2.00$ ms.

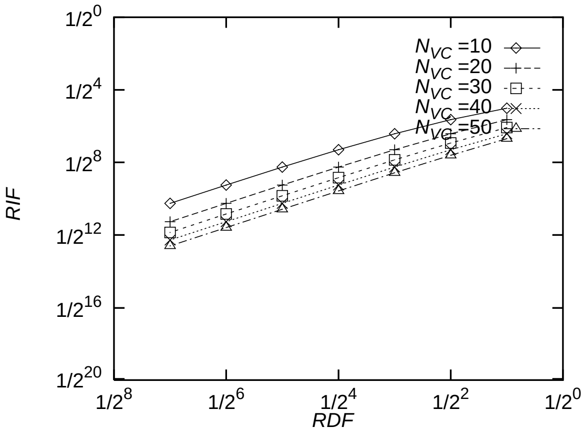

Figure 4.11: Condition for No Buffer Overflow for $BL = 300$ Kbyte, $Q_T = BL/2$, and $\tau = 2.00$ ms.

ference compared with the case where $BL = 300$ Kbyte (see Fig. 4.8) except when $RDF$ is large. This can be explained as follows. When $\tau$ is small (0.02 ms in these cases), the queue length never grows rapidly because of quick notification of congestion. However, when $\tau$ becomes large, a lack of buffer capacity has a different impact as shown in Fig. 4.13. The applicable region of $RIF$ and $RDF$ is much narrowed than in Fig. 4.10. However, we should note that buffer overflow can be still avoided even with a small buffer.

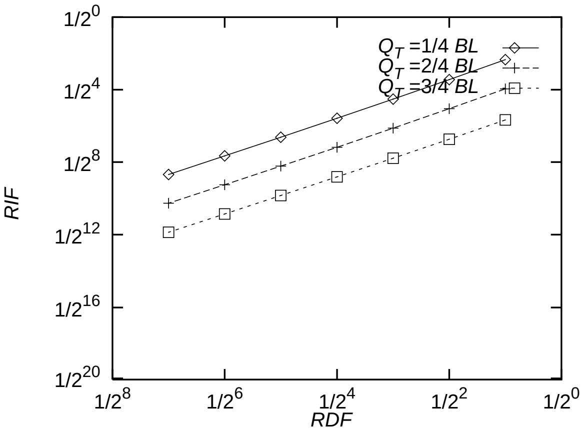

Figure 4.12: Condition for No Buffer Overflow for $BL = 10$ Kbyte, $N_{VC} = 10$, and $\tau = 0.02$ ms.

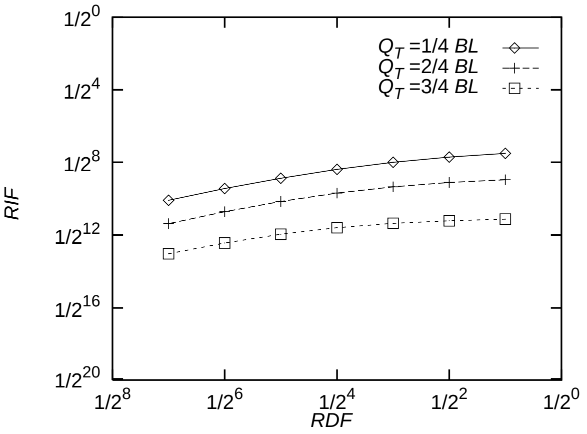

Figure 4.13: Condition for No Buffer Overflow for $BL = 10$ Kbyte, $Q_T = BL/2$, and $\tau = 2.00$ ms.

We next investigate the applicable region of $RIF$ and $RDF$ for achieving full link utilization. Figures 4.14 and 4.15 show boundary lines of the condition for full link utilization obtained from Eq. (4.22) for $\tau = 0.02$ ms and $\tau = 2.00$ ms, respectively. In these figures, $Q_T$ is varied between $BL/4$, $BL/2$, and $3BL/4$ while the other parameters are set to the same values used in Figs. 4.8 and 4.10. In Figs. 4.8 and 4.10, the bottleneck link can always be utilized in the upper region of the boundary line. As we have explained, only the values of $RIF$ and $RDF$ taken from the region between two boundary lines, Eqs. (4.21) and (4.22), can satisfy both the no buffer overflow objective, and the full link utilization objective. For example, in the case of $\tau = 0.02$ ms, $RIF$ and $RDF$ should be chosen from the region bounded by Figs. 4.8 and 4.14. Note that both objectives can be attained for $\tau = 2.00$ ms even though the applicable region of

$RIF$ and $RDF$ is rather small. Unlike the condition for no buffer overflow, a large value of $Q_T$ is needed for full link utilization.

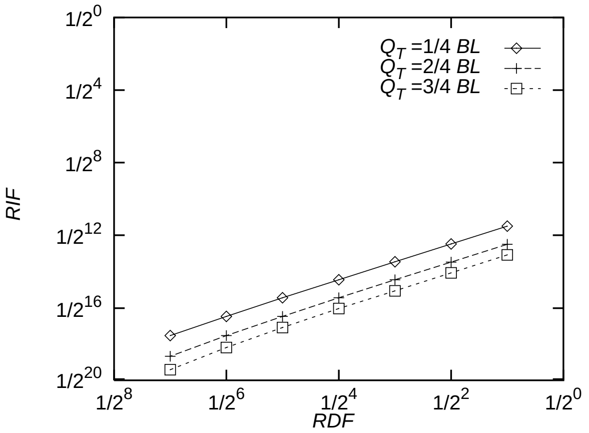

Figure 4.14: Condition for Full Link Utilization for $BL = 300$ KByte, $N_{VC} = 10$, and $\tau = 0.02$ ms.

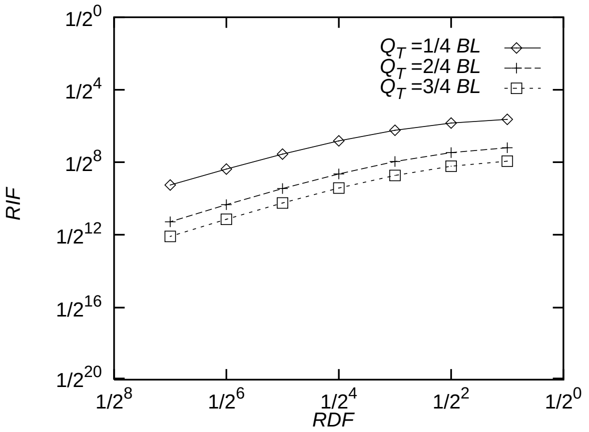

Figure 4.15: Condition for Full Link Utilization for $BL = 300$ KByte, $N_{VC} = 10$, and $\tau = 2.00$ ms.

The effect of the number of connections, $N_{VC}$, is shown in Figs. 4.16 and 4.17 for $\tau = 0.02$ ms and $\tau = 2.00$ ms, respectively, while $Q_T$ is fixed at $BL/2$. As with the prevention of buffer overflow condition, a large value of $N_{VC}$ does not greatly reduce the size of the applicable region for full link utilization. Hence, we anticipate that the prevention of buffer overflow and full link utilization can both be attained even when $N_{VC}$ is large. This is true in the case of a sufficient buffer size and/or a short propagation delay. However, it is impossible to achieve both objectives under some conditions. In Fig. 4.18, the buffer size $BL$ is changed to 10 KByte while the propagation delay $\tau$ is unchanged (0.02 ms). Figures 4.12 and 4.18 shows that both Eqs. (4.21) and (4.22) can still be satisfied. However, when $BL = 10$ Kbyte and $\tau = 2.00$ ms, we find that no solution of Eq. (4.22) exists for $RDF$ larger than $1/2^8$. That is, full link utilization while avoiding buffer overflow cannot be obtained in this case.

In what follows, we validate our analysis through a comparison with simulation results. We ran simulations for the same model as our analytic model (see Fig. 4.1), and used the same

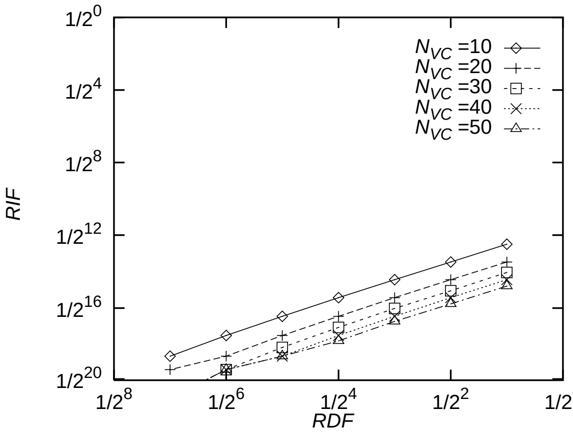

Figure 4.16: Condition for Full Link Utilization for $BL = 300$ KByte, $Q_T = BL/2$, and $\tau = 0.02$ ms.

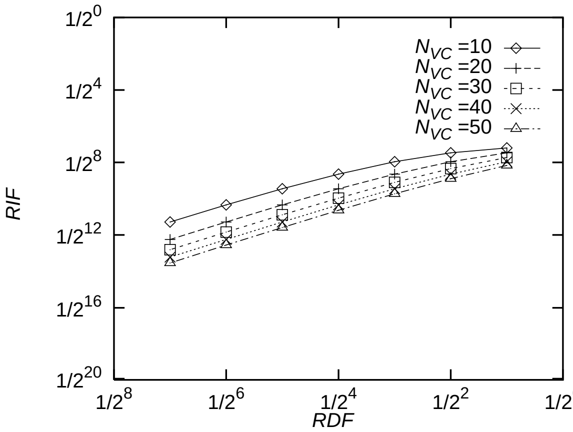

Figure 4.17: Condition for Full Link Utilization for $BL = 300$ KByte, $Q_T = BL/2$, and $\tau = 2.00$ ms.

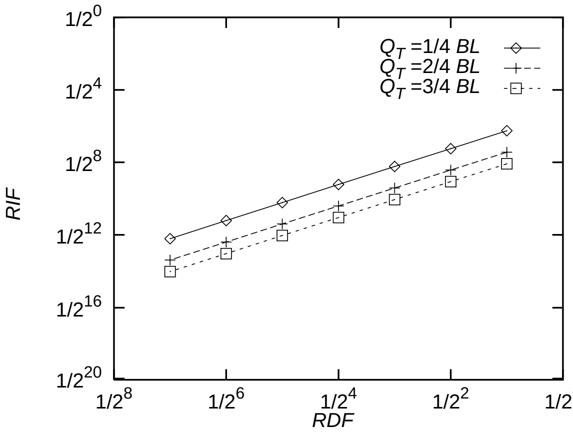

Figure 4.18: Condition for Full Link Utilization for $BL = 10$ KByte, $Q_T = BL/2$, and $\tau = 0.02$ ms.

control parameters as were used in Fig. 4.5. Each source end system was given $ICR = BW/20$. In Figs. 4.19 and 4.20, we show the maximum queue length obtained by analysis (see Eq. (4.20)) and by simulation for the propagation delays, $\tau = 0.02$ ms and $\tau = 2.00$ ms. As can be seen, Eq. (4.20) provided values close to the simulation results regardless of the large propagation delay, and the simulation results were upper-bounded as in the analysis.

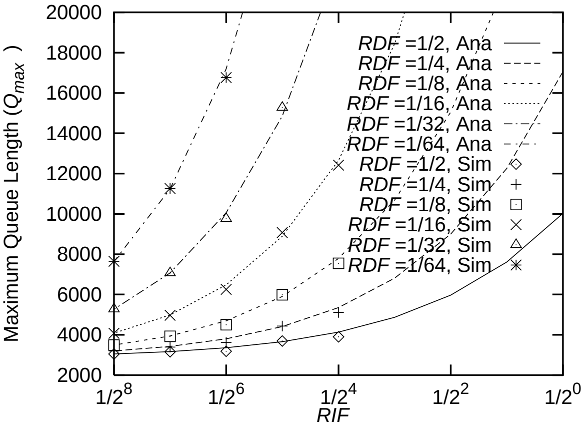

Figure 4.19: Comparison of $Q_{max}$ with Simulation Results for $\tau = 0.02$ ms.

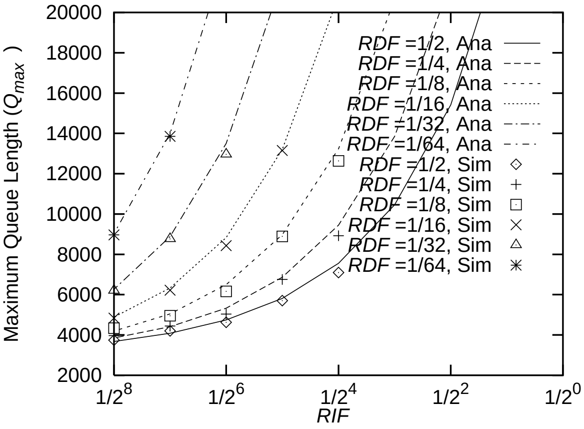

Figure 4.20: Comparison of $Q_{max}$ with Simulation Results for $\tau = 2.00$ ms.

## 4.4 Parameter Tuning in The Initial Transient State

In this section, we focus on determining the proper setting of the control parameters, especially for achieving good performance in the initial transient state. In Subsection 4.4.1, we explain the behavior of rate-based congestion control when one or more new connections start transmitting cells, and explore the proper setting of initial control parameters for the source end system. In Subsection 4.4.2, we show several numerical examples.

### 4.4.1 Appropriate Setting of Initial Control Parameters

During the process to establish a connection, the source end system negotiates several control parameters with the network. These control parameters include $ICR$ (Initial Cell Rate) and $C_{RM}$ in addition to $N_{RM}$, $RIF$, and $RDF$ that we have discussed in Section 4.2. The first two parameters are needed for the initial cell transmission. After the connection is established, the $ACR$ is initially set to the $ICR$, and cells are transmitted as a series of one forward RM cell followed by $(N_{RM} - 1)$ data cells. Once a backward RM cell is received by the source via the destination, its $ACR$ is controlled by the status of the subsequent backward RM cells as explained in Section 4.1. In other words, no rate-control is performed until the first backward RM cell is received by the source. To limit the number of cells drained without rate-control, the source end system uses $C_{RM}$. That is, when the number of forward RM cells sent without receiving any backward RM cells exceeds $C_{RM}$, the source independently decreases its $ACR$ at each forward RM cell emission according to the following equation.

$$ACR \leftarrow \max(ACR - ACR \times CDF, MCR) \tag{4.27}$$

In this equation, $CDF$ (Cutoff Decrease Factor) is a ratio of the exponential rate decrease.

If new connections start their cell transmission while other connections are active and settled, the queue length at the switch buffer grows rapidly unless both of $ICR$ and $C_{RM}$ are chosen properly. For safer operation, $ICR$ should be given a small value to avoid a queue build-up, which would eventually lead to cell loss due to the limited buffer capacity. On the other hand, a large $ICR$ value is desirable to minimize burst (packet) transmission delays. In what follows, we investigate a proper setting of $ICR$ and $C_{RM}$, taking into account the need to maximize $ICR$ while avoiding buffer overflow at the switch.

$ICR$ and $C_{RM}$ are negotiated through two parameters called $TBE$ (Transient Buffer Exposure) and $FRTT$ (Fixed Round-Trip Time) [3]. The $TBE$ is the number of cells that the source can transmit before receiving backward RM cells. In other words, $TBE$ is the buffer capacity reserved for that connection at the intermediate switches. Therefore, cell loss can be prevented as long as the number of cells emitted by the source does not exceed $TBE$. $FRTT$ is the actual round-trip delay of that connection including processing delays and propagation delays. Based on $TBE$ and $FRTT$, the values of $ICR$ and $C_{RM}$ at the switch are computed as follows.

$$ICR = \min\left(ICR, \frac{TBE}{FRTT}\right) \tag{4.28}$$

$$C_{RM} = \left\lceil \frac{TBE}{N_{RM}} \right\rceil \tag{4.29}$$

In these equations, the time needed to receive the first backward RM cell (i.e., $FRTT$) is equalized to the time after which the rate begins to decrease after sending $C_{RM}$ forward RM cells [66]. In the remainder of this section, we investigate the proper setting of $TBE$ and $FRTT$, instead of $ICR$ and $C_{RM}$, using the same model described in Section 4.1. We will first obtain the upper-bound of $FRTT$, and then consider ways of determining $TBE$.

By assuming a zero processing delay of cells at the switch and the higher priority of backward RM cells over data cells, $FRTT$ is upper-bounded by the following equation.

$$FRTT = \tau + \frac{\min(Q_{max}, BL)}{BW}, \tag{4.30}$$

where $\tau$, $BW$ and $BL$ are the propagation delay, the bandwidth of the switch link, and the switch buffer size, respectively, and $Q_{max}$ is given by Eq. (4.20).

We first consider a case where only one connection is added to a network where $N_{VC}$ connections are actively transmitting cells and are already in a steady state. In this case, to avoid buffer overflow at the switch, $TBE$ for that connection should be set as

$$TBE = \max(BL - Q_{max}, 0). \tag{4.31}$$

As explained in Section 4.2, $Q(t)$ has a periodicity in a steady state. Hence, $TBE$ can take a larger value than Eq. (4.31) if the new connection starts cell transmission when $Q(t)$ is smaller than $Q_{max}$. However, we use Eq. (4.31) in the numerical examples since the periodicity of $Q(t)$ is difficult to anticipate, and our primary objective is to avoid cell loss at the switch.

Equation (4.31) is, however, insufficient to prevent buffer overflow when we consider the possibility that two or more connections are simultaneously established on a network and/or become active to start to transmit cells at the same time. That is, $TBE$ needs to be chosen more carefully than in the previous case. One may think that such an occasion rarely happens in a real situation. However, consider the case where the server in the network goes down due to its failure, and it is recovered afterward. Then, many client workstations access the server almost at the same time, which leads to the queue build-up at the switch [49]. In the remainder of this subsection, we will take account of such a case for safer operation of the rate-based congestion control.

In what follows, we consider four ways to allocate $TBE$ to multiple connections. Let $N_{VC}$ be the number of active connections in the steady state. We further introduce $N_{VC}'$ representing the possible number of connections that are added simultaneously. Noting that the inactive connections, which have been already established but are idle, start cell transmission in a same manner to the newly added connections [3], the inactive connections are not counted in $N_{VC}$, but in $N_{VC}'$. $\overline{N_{VC}}$ is also introduced as the maximum number of connections that the switch can accept. It may be limited by the physical configuration of the switch.

Note that the number of active connections, $N_{VC}$, can be known at the switch by introducing an additional function such as per-VC accounting [18]. In that case, the following Schemes 1 and 3 will work more correctly. Otherwise, $N_{VC}$ should be estimated according to the operating network systems at the expense of accuracy. Also note that estimation of $N_{VC}'$ becomes a key factor in Schemes 2 and 4.

**Scheme 1:** This scheme reserves the same amount of buffer capacity, $BL/\overline{N_{VC}}$, for every connection that can possibly be established on the switch. That is,

$$TBE = \frac{BL}{\overline{N_{VC}}}. \tag{4.32}$$

Therefore, buffer overflow is completely avoided regardless of the number of additional connections $N_{VC}'$. However, $TBE$ cannot take a larger value than other schemes even when $N_{VC}'$ is small.

**Scheme 2:** This scheme allocates the unused buffer capacity, $\max(BL - Q_{max}, 0)$, for additional connections, i.e.,

$$TBE = \frac{\max(BL - Q_{max}, 0)}{N_{VC}'}, \tag{4.33}$$

where $Q_{max}$ is obtained in Eq. (4.20) of our analysis. Newly established connections can gain a larger value of $TBE$ than other schemes. However, buffer overflow may occur if the estimation of $N_{VC}'$ is smaller than the actual value.

**Scheme 3:** This scheme allocates the unused buffer capacity, $\max(BL - Q_{max})$, for the connections that are currently not active but that might be established.

$$TBE = \frac{\max(BL - Q_{max}, 0)}{\overline{N_{VC}} - N_{VC}}, \tag{4.34}$$

This scheme can be considered as a more conservative one than Scheme 2.

**Scheme 4:** In this scheme, we set

$$TBE = \frac{\max(BL - Q'_{max}, 0)}{N_{VC}'}, \tag{4.35}$$

where $Q'_{max}$ is the maximum queue length obtained by replacing $N_{VC}$ in Eq. (4.20) with $\overline{N_{VC}}$. In other words, this scheme allocates the unused buffer capacity in the worst case (i.e., when all $\overline{N_{VC}}$ connections are active) for additional connections. Note that this scheme requires no information about $N_{VC}$. Therefore, the lower switch cost can be expected. Furthermore, the switch can determine $TBE$ a priori without considering the current switch status.

### 4.4.2 Numerical Examples

In what follows, we first show numerical examples for one additional connection (i.e., $N_{VC}' = 1$). We plotted $ICR$ and $C_{RM}$ as functions of the number of connections, $N_{VC}$, and the propagation delay, $\tau$, in Figs. 4.21 and 4.22. We used Eqs. (4.31) and (4.30) to determine $TBE$ and $FRTT$, respectively. The buffer size $BL$ and the threshold value $Q_T$ are fixed at 300 Kbyte and $BL/2$, respectively. In these figures, $RIF$ and $RDF$ are set to 1/64 and 1/16, which are commonly used values for the binary mode switch in the LAN environment [67, 55, 56]. For the other parameters, we use the same values as in Subsection 4.3.3. Figure 4.21 shows that $ICR$ should always be set to zero when the number of active connections, $N_{VC}$, is larger than 10.

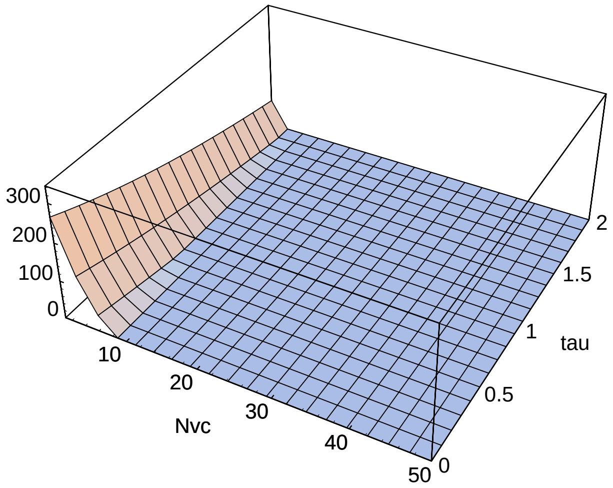

Figure 4.21: Initial Cell Rate for $BL = 300$ Kbyte, $Q_T = BL/2$, $RIF = 1/64$, and $RDF = 1/16$.

By setting both $RIF$ and $RDF$ appropriately, $ICR$ can take a larger value than in Figs. 4.21 and 4.22 as follows. We chosen $RIF = 1/128$ and $RDF = 1/2$ based on our the analytic results from Figs. 4.9, 4.11, 4.16, and 4.17. This parameter setting can fulfill both the conditions of

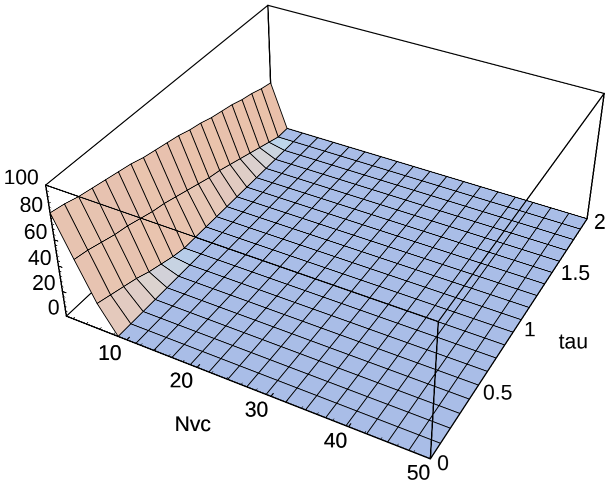

Figure 4.22: $C_{RM}$ for $BL = 300$ Kbyte, $Q_T = BL/2$, $RIF = 1/64$, and $RDF = 1/16$.

preventing buffer overflow and the condition of full link utilization. We plot $ICR$ and $C_{RM}$ for these parameters in Figs. 4.23 and 4.24. As shown in Fig. 4.23, $ICR$ can now take a larger value than in Fig. 4.21 for any $N_{VC}$ and $\tau$ because $Q_{max}$ is decreased by setting $RIF$ and $RDF$ appropriately. Remember that the smaller $Q_{max}$ becomes, the larger value $ICR$ can take (see Eqs. (4.28) and (4.31)). Therefore, a proper setting of $RIF$ and $RDF$ is also helpful to improve the performance in the initial transient state.

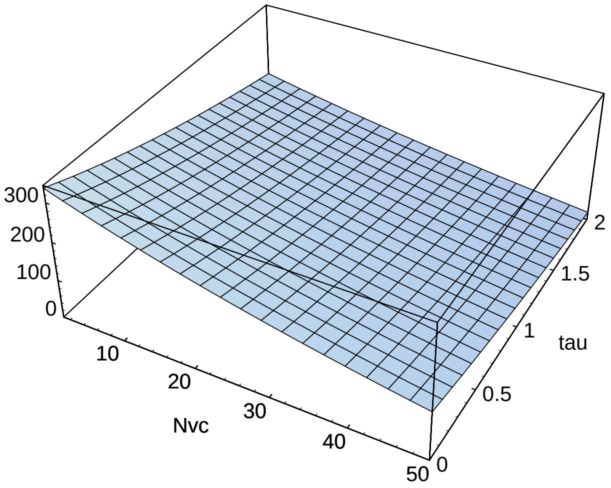

Figure 4.23: $ICR$ (Initial Cell Rate) for $BL = 300$ Kbyte, $Q_T = BL/2$, $RIF = 1/128$, and $RDF = 1/2$.

Next, we compare the four schemes for allocating $TBE$ to new connections. Figure 4.25 shows $ICR$ for each scheme obtained from Eq. (4.28) for $\tau = 0.02$ ms. To demonstrate the difference of these four schemes clearly, $N_{VC}'$ (the number of simultaneously added connections) is first set to 1. $\overline{N_{VC}}$ (the maximum number of connections that the switch can accept) is fixed at 50. For the other parameters, we use the same values as in Figs. 4.23 and 4.24. Schemes 2 and 4 give a larger value of $ICR$ than the others because of their compensation for the possibility of cell loss. We than change $N_{VC}'$ from 1 to 2 and 4, and plot $ICR$ for each scheme in Figs. 4.26

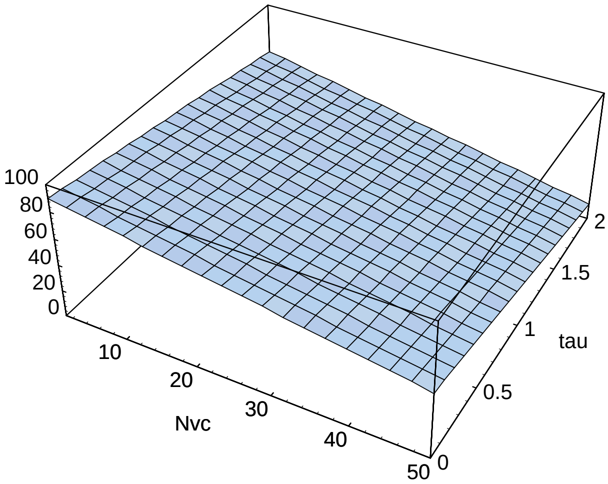

Figure 4.24: $C_{RM}$ for $BL = 300$ Kbyte, $Q_T = BL/2$, $RIF = 1/128$, and $RDF = 1/2$.

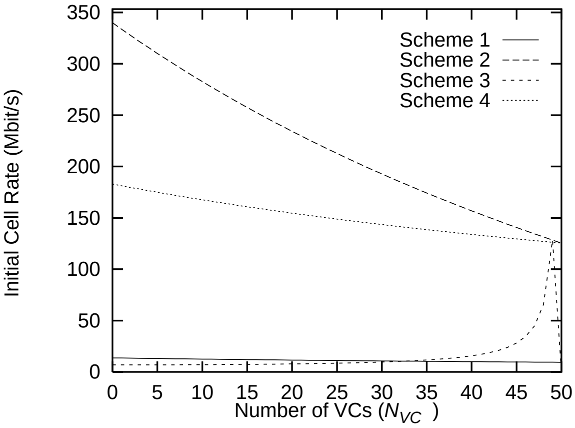

Figure 4.25: $ICR$ (Initial Cell Rate) for $N_{VC}{}' = 1$ and $\tau = 0.02$ ms.

and 4.27, respectively. Note that a larger value of $N_{VC}'$ makes Schemes 2 and 4 safer, but these schemes still offer a larger $ICR$ than Schemes 1 and 3. However, Schemes 1 and 3 is appropriate when $N_{VC}'$ is difficult to be estimated by the network or when even a small number of cell losses cannot be tolerated.

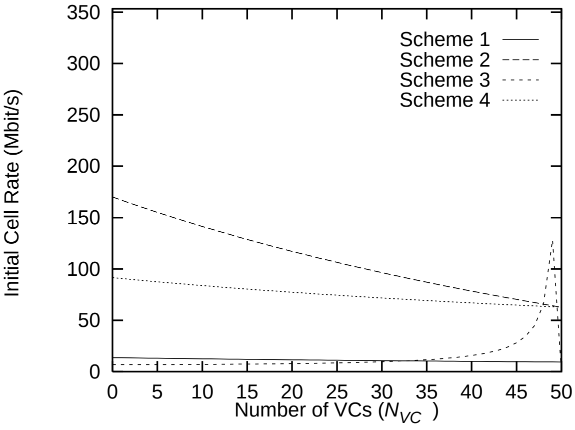

Figure 4.26: $ICR$ (Initial Cell Rate) for $N_{VC}' = 2$ and $\tau = 0.02$ ms.

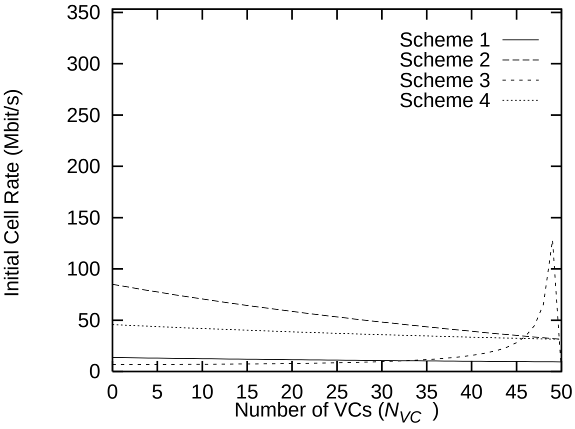

Figure 4.27: $ICR$ (Initial Cell Rate) for $N_{VC}' = 4$ and $\tau = 0.02$ ms.

Figure 4.28 shows the $ICR$ values of each scheme for $\tau = 2.00$ ms and $N_{VC}' = 2$, and Fig. 4.29 shows the same for $\tau = 2.00$ ms and $N_{VC}' = 4$. There is little difference between Schemes 1, 3 and 4 when the propagation delay is large. Although the switch buffer overflows with Scheme 2 for more than two additional connections, Scheme 2 can obtain a higher $ICR$ than the other schemes.

Hence, we conclude that Scheme 2 is appropriate for determining $ICR$ if the number of additional connections, $N_{VC}'$, can be estimated correctly by the network. Although Scheme 4 underestimates $ICR$ compared with Scheme 2, it is also suitable because of its implementation simplicity. However, if the estimation of $N_{VC}'$ is difficult and any cell loss cannot be tolerated, Scheme 3 should be used for safer operation.

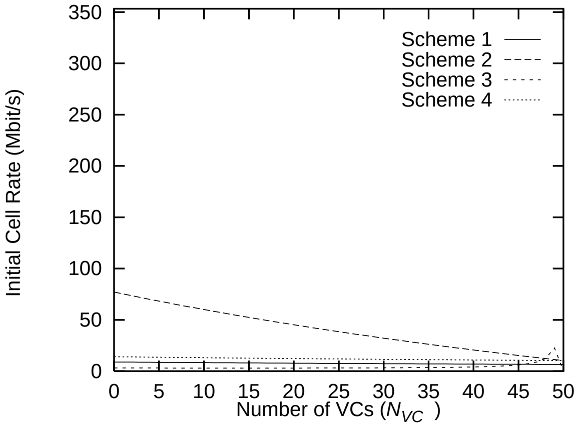

Figure 4.28: $ICR$ (Initial Cell Rate) for $N_{VC}' = 2$ and $\tau = 2.00$ ms.

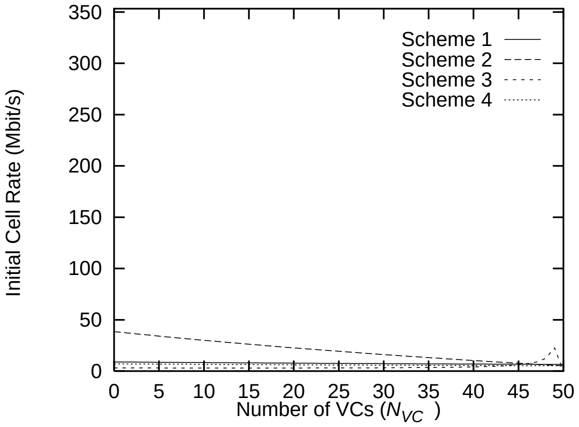

Figure 4.29: $ICR$ (Initial Cell Rate) for $N_{VC}' = 4$ and $\tau = 2.00$ ms.

## 4.5 Concluding Remarks

In this chapter, we have analyzed rate-based congestion control by using a first-order fluid approximation. We have also derived proper values of the control parameters — the source end system parameters and the switch parameters — to fulfill two objectives: avoidance of buffer overflow at the switch and full link utilization of the bottleneck link. Our investigation revealed that proper parameter settings also improve transient performance. In [64, 68, 69], we applied our parameter-tuning analysis to TCP over ABR as an example application of rate-based congestion control to an upper layer protocol. In [70], we extended our analysis to a more realistic model where each connection has a different propagation delay, and where CBR / VBR traffic coexists with ABR traffic. In that paper, we also presented simulation results for a multi-hop network configuration to exhibit the tradeoff relationship among cell loss probability, link utilization, and fairness.

# Chapter 5

# Robustness of Rate-Based Congestion Control Algorithm

In this chapter, by extending our analytic results presented in Chapter 4, proper settings of rate-control parameters in various circumstances are investigated. We first analyze the dynamical behavior of the rate-based congestion control algorithm for multiple groups of ABR connections with different propagation delays. Next, we evaluate the effect of CBR traffic on ABR connections. Simulation results for a multi-hop network configuration are also presented to exhibit tradeoffs among cell loss probability, link utilization and fairness. Finally, the selection method of control parameters in the multi-hop network is proposed based on our analytic methods and simulation results.

## 5.1 Parameter Tuning for Single Binary-Mode Switch

In what follows, we first summarize the analytic results of dynamical behavior of the rate-based congestion control obtained in [64, 71, 72] in Subsection 5.1.1. In Subsection 5.1.2, we then extend our analysis to the model in which multiple groups of connections with different propagation delays are allowed, and focus on a fairness problem by adding a newly established ABR connection. In Subsection 5.1.3, we analyze the effect of an additional CBR connection on the behavior of ABR connections in terms of the maximum queue length.

### 5.1.1 Analysis for The Homogeneous Connections: Summary

The analytic model consists of homogeneous traffic sources and a single bottleneck link as shown in Fig. 4.1. 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 defined by $\tau_{sx}$ and $\tau_{xd}$, respectively. The round-trip propagation delay at the switch 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 the source via the destination. Propagation delays $\tau_{sx}$ and $\tau_{xd}$ are assumed to be identical for all 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, all connections are assumed to start cell transmission simultaneously with the same rate-control parameters. We also assume that propagation delays $\tau_{sx}$ and $\tau_{xd}$ are the same for all connections. Thus, all connections behave identically. This assumption is, however, relaxed in Subsection 5.1.2 such that several groups of connections can have different propagation delays.

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 the source end system via the destination end system by marking an EFCI (Explicit Forward Congestion Indication) bit in the data cell, or by marking the CI (Congestion Indication) bit in the forward RM (Resource Management) cell. By receiving congestion information from the network, each source end system decreases its rate. After the queue length goes under the low threshold value $Q_L$, it is regarded as congestion termination. The EFCI bit in the data cell or the CI bit in the forward RM cell is then cleared to indicate congestion relief to source end systems via the destination. This sort of switch operation is often referred to as an "EFCI bit marking switch", or simply as a "binary switch" [16].

Behavior of source and destination end systems is standardized in the ATM forum, and is specified in [3]. Each source end system periodically generates a forward RM cell in proportion to its rate; that is, it sends one forward RM cell per $(N_{RM} - 1)$ data cells. On the receipt of the forward RM cell, the corresponding destination end system sends it back as the backward RM cell to the corresponding source end system. The allowed cell rate $ACR$ of the source is then changed in response to the backward RM cell. When it receives the CI bit cleared cell, rate increase is performed as

$$ACR \leftarrow \min(ACR + RIF \times PCR, PCR), \tag{5.1}$$

where $PCR$ (Peak Cell Rate) is a maximum rate for this connection negotiated at its connection setup, and $RIF$ (Rate Increase Factor) is a ratio for proportional rate increase. The CI bit marked cell decreases $ACR$ as

$$ACR \leftarrow \max(ACR - ACR \times RDF, MCR), \tag{5.2}$$

where $MCR$ (Minimum Cell Rate) is a 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, the forward RM cell is not explicitly considered; congestion indication is performed by the EFCI bit of the data cell. However, the forward RM cell can easily be taken into account by replacing $BW$ in our analysis with $BW'$, which is defined as

$$BW' = BW \, \frac{N_{RM} - 1}{N_{RM}}. \tag{5.3}$$

It is noted that we will use $BW'$ instead of $BW$ in numerical examples presented in Subsections 5.1.2 and 5.1.3.

Let us introduce $ACR(t)$ and $Q(t)$ as the allowed cell rate $ACR$ of each source and the queue length at the switch observed at time $t$, respectively. In what follows, we present analytic results for $ACR(t)$ and $Q(t)$ that are taken from our previous work [64, 71, 72].

As shown in Fig. 4.2, $ACR(t)$ and $Q(t)$ have periodicity in steady state. The behavior of $ACR(t)$ and $Q(t)$ are divided into four phases called Phase 1 through Phase 4. We introduce $ACR_i(t)$ and $Q_i(t)$ as $ACR(t)$ and $Q(t)$ in Phase $i$, which are defined as

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

where $t_i$ is the time when Phase $i$ terminates. For simplicity, we further introduce $t_{i-1,i}$ as the interval of Phase $i$, which is defined as $t_i - t_{i-1}$. $ACR_i(t)$ is given by the following equations.

$$\begin{aligned}
ACR_1(t) &= ACR_1(0)e^{-\frac{BW\,RDF}{N_{VC}\,N_{RM}}t} && (5.4)\\
ACR_2(t) &= ACR_2(0) + \frac{BW\,RIF\,PCR}{N_{VC}\,N_{RM}}t && (5.5)\\
ACR_3(t) &\cong ACR_3(0)e^{\frac{RIF\,PCR}{N_{RM}}t} && (5.6)\\
ACR_4(t) &= ACR_4(0) + \frac{BW\,RIF\,PCR}{N_{VC}\,N_{RM}}t && (5.7)\\
0 \leq & \quad t \quad < t_{i-1,i}
\end{aligned}$$

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

$$Q_i(t) = Q_i(\tau_{sx}) + \int_{\tau_{sx}}^{t} \max(N_{VC}\,ACR_i(x-\tau_{sx}) - BW, 0)dx, \quad \tau_{sx} \leq t < \tau_{sx} + t_{i-1,i}.$$

The duration of Phase $i$, $t_{i-1,i}$, is obtained as

$$t_{i-1,i} = \begin{cases} Q_1^{-1}(Q_L) + \tau_{xds} & i = 1 \\ \min(Q_2^{-1}(Q_H) + \tau_{xds}, Q_2^{-1}(0) + \tau_{xds}) & i = 2,4 \\ ACR_3^{-1}(BW/N_{VC}) + \tau & i = 3 \end{cases} \qquad (5.8)$$

where $ACR_i^{-1}(t)$ and $Q_i^{-1}(t)$ are defined as the inverse representations of $ACR_i(t)$ and $Q_i(t)$, respectively.

### 5.1.2 Analysis for Multiple Groups of Connections

In this subsection, we derive the dynamical behavior of the rate-based congestion control for $N$ groups of connections with different propagation delays. Through numerical examples, we show the importance of parameter tuning for achieving good fairness and the short ramp-up time for an additional ABR connection.

#### Analysis

We divide ABR connections into $N$ groups with different propagation delays. Within a group, connections have identical propagation delays. Figure 5.1 depicts our analytic model in the case of $N = 2$. Propagation delays from each source to the switch, and from the switch to each destination of group $n$ ($1 \leq n \leq N$) are denoted by $\tau_{sxn}$ and $\tau_{xdn}$, respectively. For brevity, we introduce $\tau_n (= 2\tau_{sxn} + 2\tau_{xdn})$ and $\tau_{xdsn} (= \tau_{sxn} + 2\tau_{xdn})$. The number of connections in group $n$ is denoted by $N_{VCn}$. Thus, we have a relation:

$$N_{VC} = \sum_{n=1}^{N} N_{VCn}$$

We assume that all connections in each group behave identically. Namely, all connections in each group have the same control parameters. Let us introduce $RIF_n$, $RDF_n$ and $N_{RMn}$ as $RIF$, $RDF$ and $N_{RM}$ of group $n$, respectively. We also assume $\tau_{sxi} \leq \tau_{sxj}$ and $\tau_{xdi} \leq \tau_{xdj}$ for any $i$ and $j$ ($i < j$) without loss of generality.

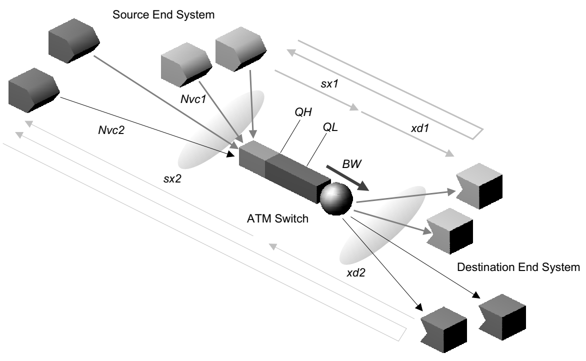

Figure 5.1: Analytic Model for Multiple Groups for $N = 2$.

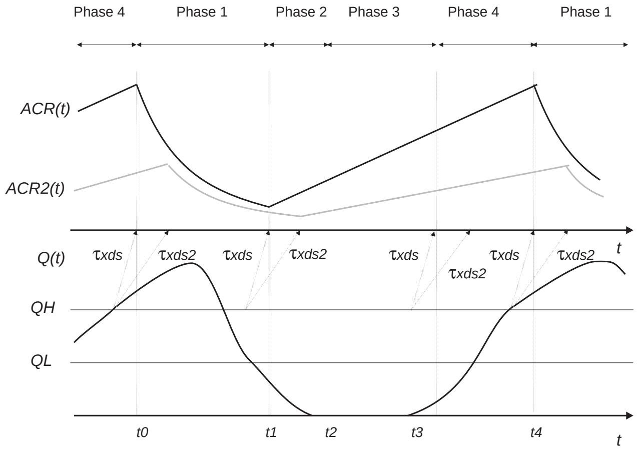

Figure 5.2: Pictorial View of $ACR^n(t)$ and $Q(t)$.

Let us introduce $ACR^n(t)$ and $Q(t)$ that represent $ACR$ of the source end system in group $n$ and the queue length at the switch observed at time $t$, respectively. As with the case in Subsection 5.1.1, $ACR^n(t)$ and $Q(t)$ have periodicity (see Fig. 5.2 for a pictorial view of $ACR^n(t)$ and $Q(t)$ for $N = 2$). We further introduce $ACR^n_i(t)$ and $Q_i(t)$ as the $ACR^n(t)$ and $Q(t)$ in Phase $i$, which are defined as

$$\begin{aligned} ACR^n_i(t) &= ACR^n(t - t_{i-1}), \\ Q_i(t) &= Q(t - t_{i-1}). \end{aligned}$$

Because of the difference in propagation delays between the switch and the source via the destination ($\tau_{xdsn}$), congestion information from the switch arrives at the sources at different time. Hence, $ACR^n_i(t)$ is obtained as follows (see Eqs.(5.4)–(5.7)).

$$\begin{aligned} ACR^n_1(t) &= ACR^n_1(\tau_{xdsn} - \tau_{xds1})e^{-\frac{BW\,RDF_n}{N_{VC}\,N_{RMn}}(t-(\tau_{xdsn}-\tau_{xds1}))} \\ ACR^n_2(t) &= ACR^n_2(\tau_{xdsn} - \tau_{xds1}) + \frac{BW\,RIF_n\,PCR}{N_{VC}\,N_{RMn}}(t - (\tau_{xdsn} - \tau_{xds1})) \\ ACR^n_3(t) &\cong ACR^n_3(\tau_{xdsn} - \tau_{xds1})e^{\frac{RIF_n\,PCR}{N_{RMn}}(t-(\tau_{xdsn}-\tau_{xds1}))} \\ ACR^n_4(t) &= ACR^n_4(\tau_{xdsn} - \tau_{xds1}) + \frac{BW\,RIF_n\,PCR}{N_{VC}\,N_{RMn}}(t - (\tau_{xdsn} - \tau_{xds1})) \end{aligned}$$

for

$$\tau_{xdsn} - \tau_{xds1} \le \quad t \quad \le \tau_{xdsn} - \tau_{xds1} + t_{i-1,i}.$$

At time $t$, the switch observes $ACR^n(t - \tau_{sxn})$ for group $n$ because of the propagation delay from the source to the switch, $\tau_{sxn}$. Therefore, $Q_i(t)$ in Phase $i$ is obtained as

$$Q_i(t) = \max(Q_i(\tau_{sx1}) + \int_{\tau_{sx1}}^{t} (\sum_{n=1}^{N} N_{VCn}\,ACR^n_i(x - \tau_{sxn}) - BW), 0), \quad \tau_{sx1} \le t < \tau_{sx1} + t_{i-1,i}.$$

The duration of Phase $i$, $t_{i-1,i}$, is simply obtained by replacing $\tau_{xds}$ in Eq. (5.8) with $\tau_{xds1}$.

### Numerical Examples

In this subsection, we provide several numerical examples. To exhibit the effect of the rate-control parameters on the ramp-up time of an additional ABR connection, we first add connections of group 1 in the network. After these connections are stabilized, another connection of group 2 with $ICR = PCR/20$ is established. The number of connections for each group is set to $N_{VC1} = 10$ for group 1 and $N_{VC2} = 1$ for group 2. We fixed the bandwidth of bottleneck link $BW$ at 353.7 cell/ms assuming 150 Mbit/s ATM link. At the switch, its buffer size $BL$ is assumed to be infinite for the purpose of obtaining the maximum queue length. Both high and low threshold values $Q_H$ and $Q_L$ are fixed at 150Kbyte. At each source end system, $N_{RM}$ is set to 32.

We first examine the effect of the propagation delay on the ramp-up time. In Figs. 5.3, we plot $ACR^n(t)$ and $Q(t)$ for $\tau_1 = \tau_2 = 0.02$ ms. In this figure, $RIF = 1/64$ and $RDF = 1/16$ (i.e., $RIF_n = 1/64$ and $RDF_n = 1/16$) are chosen to satisfy two objectives — preventing cell loss and achieving full link utilization — for connections of group 1 [72]. We add group 2 to the network when group 1 is at the beginning of Phase 1. In 5.4, we change only the round-trip delay of group 2, $\tau_2$, from 0.02 ms to 2.00 ms. In Table 5.1, we also show effective throughput normalized by the link capacity for connections in each group where $\tau_1$ is fixed at 0.02 ms but

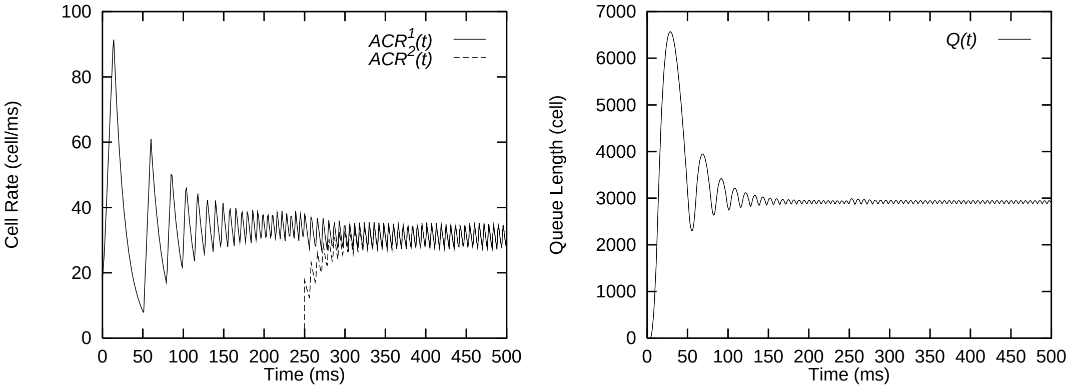

Figure 5.3: Effect of Propagation Delay for $\tau_1 = 0.02$ ms and $\tau_2 = 0.02$ ms.

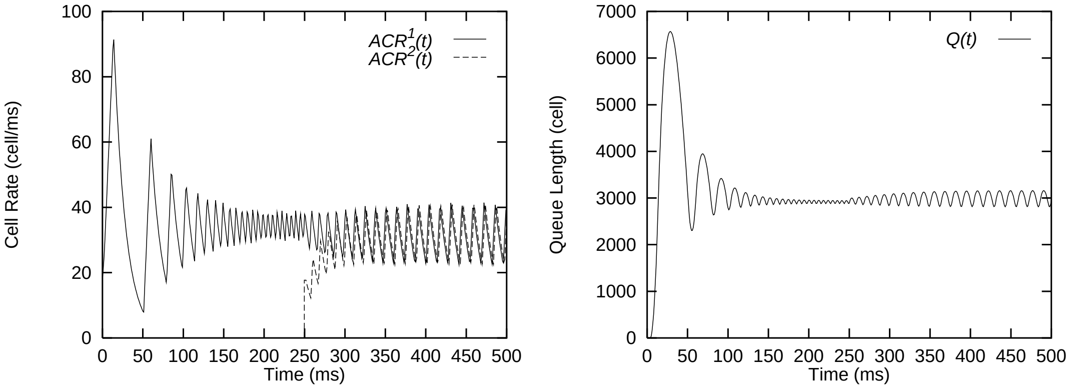

Figure 5.4: Effect of Propagation Delay for $\tau_1 = 0.02$ ms and $\tau_2 = 2.00$ ms.

Table 5.1: Effective Throughput for Each Group.

| Round-Trip Delay of Group 2 ($\tau_2$) | Group 1 | Group 2 |
|---|---|---|
| 0.02 ms | 0.0880 | 0.0880 |
| 0.20 ms | 0.0880 | 0.0880 |
| 2.00 ms | 0.0882 | 0.0875 |

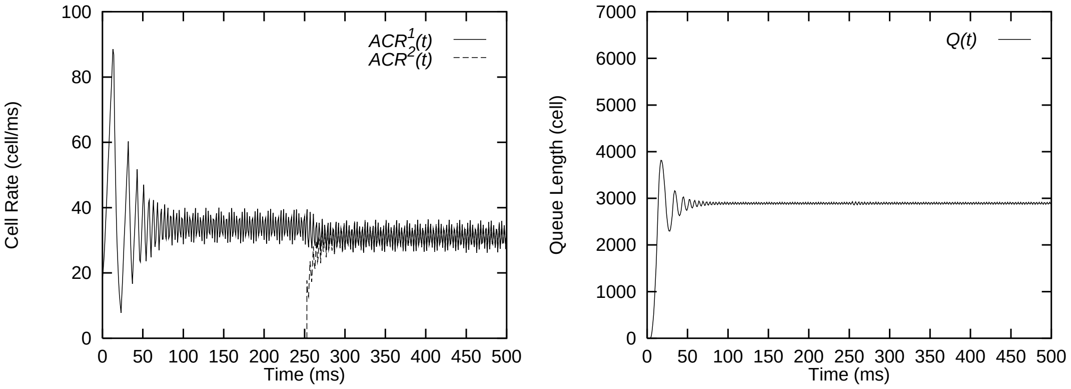

Figure 5.5: Effect of Control Parameters for $RIF = 1/64$ and $RDF = 1/4$.

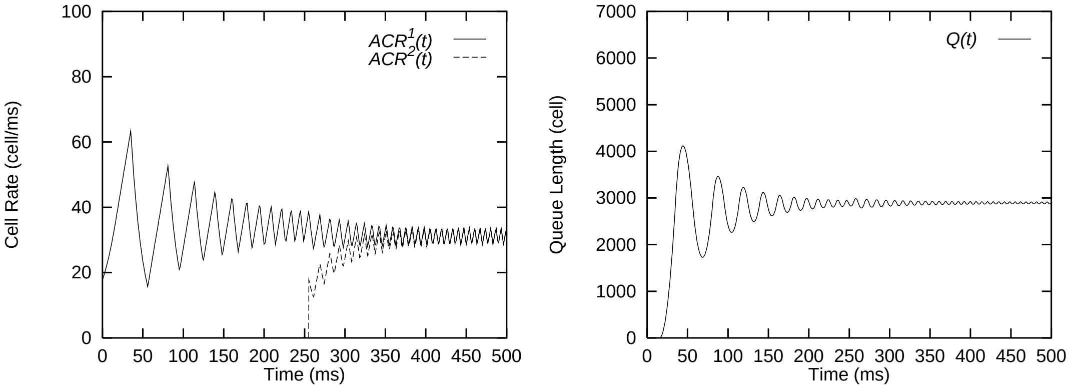

Figure 5.6: Effect of Control Parameters for $RIF = 1/256$ and $RDF = 1/16$.

$\tau_2$ is varied as 0.02 ms, 0.20 ms and 2.00 ms. From these results, one can find that the difference in round-trip delays of group 2 has little effect on fairness and the rump-up time. For example, the ramp-up time in Fig. 5.4 is almost equivalent to Fig. 5.3.

The effect of $RIF$ and $RDF$ on the additional ABR connection is next investigated. Figure 5.5 shows the case where a larger value of $RDF$ is used; that is, the rate decrease is faster than the case of Fig. 5.3. Here, $RDF = 1/4$ is used instead of 1/16 while $RIF = 1/64$ is unchanged. On the other hand, slower rate increase is considered in Fig. 5.6 where we use $RIF = 1/256$ and $RDF = 1/16$. These parameter sets also prevent cell loss and achieve full link utilization. It can be found that the ramp-up time of group 2 is considerably affected by the setting of $RIF$ and $RDF$. Namely, the ramp-up time becomes shorter by increasing $RDF$, and longer by decreasing $RIF$. Especially, the small value of $RIF$ leads to much larger ramp-up time as can be observed in Fig. 5.6. Therefore, for fulfilling good responsiveness, $RIF$ and $RDF$ should be set to large values as long as no cell loss and full link utilization can be satisfied.

### 5.1.3 Effect of CBR Traffic

In this subsection, by extending analytic results provided in Subsection 5.1.1, we derive the maximum queue length at the switch when a CBR connection is newly established.

### Analysis

We add a CBR connection to the model presented in Subsection 5.1.1 (see Fig. 4.1) at time $t'$ with a fixed bandwidth $p \times BW (0 \leq p \leq 1)$. The available bandwidth to ABR traffic is therefore suddenly changed from $BW$ to $(1-p)BW$ at the time $t'$. Let us introduce $Q_{max}$ as the maximum queue length after the establishment of the CBR connection at the time $t'$. First, $Q_{max}$ is given by

$$Q_{max} = Q(t' + \tau_{sx}) + \int_{t'+\tau_{sx}}^{t'_{max}} \max(N_{VC} ACR'(x - \tau_{sx}) - (1-p)BW, 0)\, dx, \tag{5.9}$$

where $ACR'(t)$ is defined as the allowed cell rate $ACR$ at time $t (\geq t')$, and $t'_{max}$ is the time when $Q(t)$ takes its maximum value (see Fig. 5.7). Since $Q(t)$ starts to decrease again after $\tau_{sx}$ from when the aggregate cell rate of ABR connections is decreased to $(1-p)BW$, $t'_{max}$ is obtained as

$$t'_{max} = ACR'^{-1}\left[\frac{(1-p)BW}{N_{VC}}\right] + \tau_{sx},$$

where $ACR'^{-1}(x)$ is the inverse representation of $ACR'(t)$.

After the time $t'$, each source receives backward RM cells with a fixed interval since the switch has always cells in the buffer. By letting $T_{RDF}$ be the interval of two successively received backward RM cells at the source end system, $T_{RDF}$ is given by

$$T_{RDF} = \frac{N_{RM}\, N_{VC}}{(1-p)BW}.$$

However, when the arrival rate of the backward RM cell is too slow, each source end system decreases its rate by $CDF$ (Cutoff Decrease Factor). In particular, when it receives no backward RM cell after transmitting the number $C_{RM}$ of forward RM cells, it begins to reduce its $ACR$ at each forward RM cell transmission as

$$ACR \leftarrow \max(ACR - ACR \times CDF, MCR). \tag{5.10}$$

The main purpose of the rate reduction mechanism introduced by $C_{RM}$ and $CDF$ is to allow the source end system to emit cells before receiving the first backward RM cell in its initial transient state [3]. Thus, $C_{RM}$ may be set to a rather large value. However, as will be shown in numerical examples, this mechanism is also helpful to avoid cell loss for ABR connections caused by background traffic such as CBR traffic.

By letting $T_{CDF}$ denote the duration of transmitting $C_{RM}$ forward RM cells without receipt of backward RM cells, $T_{CDF}$ is given by

$$T_{CDF} = \frac{N_{RM}\, C_{RM}}{ACR}.$$

According to the relation between $T_{RDF}$ and $T_{CDF}$, $ACR'(t)$ is obtained as follows.

1. $T_{RDF} \leq T_{CDF}$; the source end system receives one or more backward RM cells before transmitting $C_{RM}$ forward RM cells.

   In this case, $ACR'(t)$ is equivalent to $ACR_1(t)$ in Phase 1. Therefore, we have

   $$ACR'(t) = ACR(t')e^{-\frac{(1-p)BW\, RDF}{N_{VC}\, N_{RM}}(t-t')}.$$

2. $T_{RDF} > T_{CDF}$; no backward RM cell is received by the source end system before transmitting $C_{RM}$ forward RM cells.

   After the time $(t' + T_{CDF})$, the source end system decreases its rate according to Eq.(5.10) for each forward RM cell transmission. Thus, we have a differential equation as

$$\frac{dACR'(t)}{dt} = -\frac{(ACR'(t))^2\, CDF}{N_{RM}\, C_{RM}}.$$

   By solving this equation, we have

$$ACR'(t) = \begin{cases} ACR(t'), & t' \le t < t' + T_{CDF} \\ \left[\frac{CDF}{N_{RM}}(t - t') + \frac{1}{ACR(t')}\right]^{-1}, & t' + T_{CDF} \le t \end{cases}$$

   Actually, the backward RM cell arrives at the source end system at $t = t' + T_{RDF}$, and it decreases $ACR$ by $RDF$. In the above analysis, we ignored the rate reduction by receiving backward RM cells at the source end system since the arrival rate of backward RM cells is slow enough, and $RDF$ is usually smaller than $CDF$. Furthermore, even in the case where $RDF$ is not small compared with $CDF$, our analysis gives the upper-bound of the maximum queue length.

As can be found from Eq. (5.9), $Q_{max}$ depends on the initial values such as $Q(t' + \tau_{sx})$ and $ACR'(t')$ that further depends on time $t'$. In what follows, we derive the maximum of $Q_{max}$ for any $t'$, which is defined as

$$Q'_{max} = \max_{t'}(Q_{max}). \tag{5.11}$$

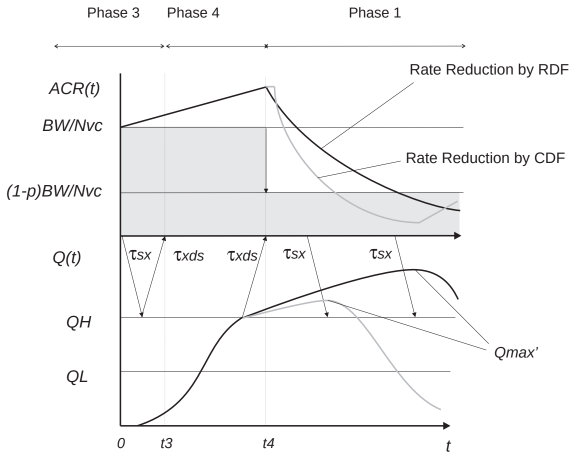

Figure 5.7: Pictorial View of $ACR(t)$ and $Q(t)$ with CBR Traffic.

As shown in Fig. 4.2, $ACR$ takes its maximum value at the end of Phase 4 (at the beginning of Phase 1). In addition, $ACR(t')$ is maximized when the switch is not fully utilized since the large amplitude of $Q(t)$ means the large amplitude of $ACR(t)$. Therefore, $Q'_{max}$ is obtained by setting $t' = t_4$, and by giving initial values of Phase 4 as

$$\begin{aligned} ACR(t_3) &= \frac{BW}{N_{VC}}, \\ Q(t_3 + \tau_{sx}) &= 0. \end{aligned}$$

At last, we note that the maximum queue length $Q'_{max}$ is given by a closed-form equation.

### Numerical Examples

In the following numerical examples, both $\tau_{sx}$ and $\tau_{xd}$ are fixed at 0.005 ms (about 1 km) as a typical value of the LAN environment. Furthermore, the number of ABR connections $N_{VC}$ is set to 10. For other control parameters except $RIF$ and $RDF$, we use the same values employed in Subsection 5.1.2.

We first show the maximum queue length $Q'_{max}$ obtained by Eq.(5.11) as a function of $p$ in Fig. 5.8. In this figure, $RIF$ is fixed at 1/64, and $C_{RM}$ and $CDF$ is at 32 and 1/2, respectively, while $RDF$ is varied as 1/4, 1/16 and 1/64. It can be found that $Q'_{max}$ increases as $p$ increases at first. For example, once a CBR connection that requires a half of the link bandwidth (75Mbit/s, in this case) is added, the switch should have 17,000 cells of buffer capacity to avoid cell loss of ABR connections with $RDF = 1/16$. Then, $Q'_{max}$ is suddenly reduced around $p = 0.9$. It is because the source end system decreases its rate by $CDF$ rather than $RDF$ when the available bandwidth for ABR connections becomes too small. Moreover, one can find that the maximum queue length can be reduced by setting $RDF$ to a large value (i.e., faster rate decrease). In Fig. 5.9, $RIF$ is changed from 1/64 to 1/1024, which means slower rate increase.

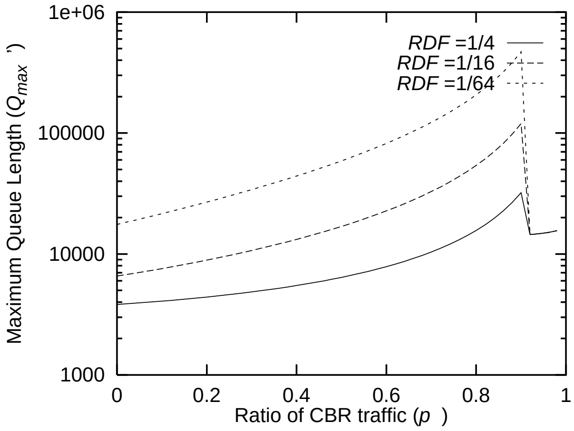

Figure 5.8: The Maximum Queue Length vs. Ratio of CBR Traffic for $RIF = 1/64$ and $C_{RM} = 32$.

In this figure, the maximum queue length is decreased to some extent when compared with Fig. 5.8. However, a large amount of buffer capacity is still required to prevent cell loss if $p$ is large.

By setting $C_{RM}$ properly, cell loss can be prevented even when the CBR connection reserves the bandwidth close to the link capacity as shown in Figs. 5.10 and 5.11. In these figures, as with the previous examples, $RIF$ is set to 1/64 and 1/1024, respectively. However, $C_{RM}$ that decides the duration to rate reduction by $CDF$ is changed from 32 to 4. These figures show that the maximum queue length can be limited even when $p$ becomes large. For example, 12,000 cells of the buffer capacity is sufficient for preventing cell loss with $RDF = 1/16$ even when the CBR connection requires the entire bandwidth.

We plot $Q'_{max}$ as the functions of $C_{RM}$ and $p$ in Figs. 5.12 and 5.13. In these figures, $RIF$ is set to 1/64 and 1/1024, respectively, while $RDF$ is fixed at 16 in both cases. The z-axis is ranged from 0 to 20,000 cells. As can be found from these figures, $C_{RM}$ should be set to be a smaller value to avoid cell losses completely for any traffic load of the CBR connection. However, by setting $RIF$ to a smaller value such as 1/1024, the queue buildup may be limited to some degree in the region where $p$ is not large. Therefore, we can conclude that to limit the

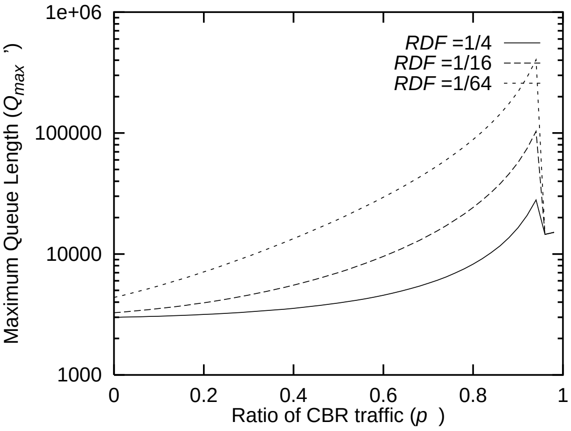

Figure 5.9: The Maximum Queue Length vs. Ratio of CBR Traffic for $RIF = 1/1024$ and $C_{RM} = 32$.

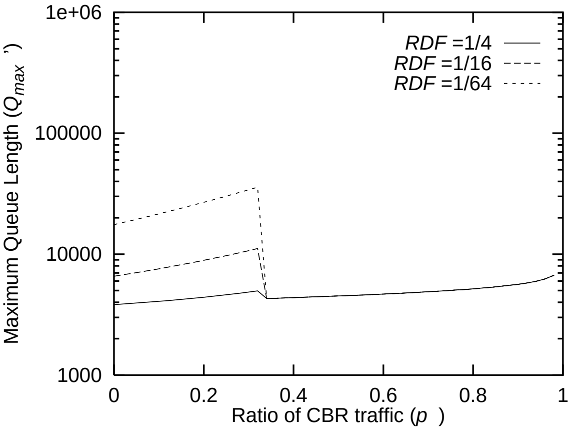

Figure 5.10: The Maximum Queue Length vs. Ratio of CBR Traffic for $RIF = 1/64$ and $C_{RM} = 4$.

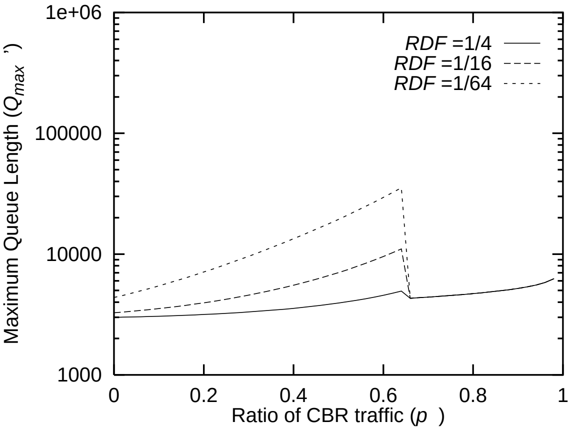

Figure 5.11: The Maximum Queue Length vs. Ratio of CBR Traffic for $RIF = 1/1024$ and $C_{RM} = 4$.

queue buildup by a new CBR connection, each of $RIF$ and $RDF$ should be small and large, respectively. Moreover, a smaller value of $C_{RM}$ is helpful to prevent cell loss.

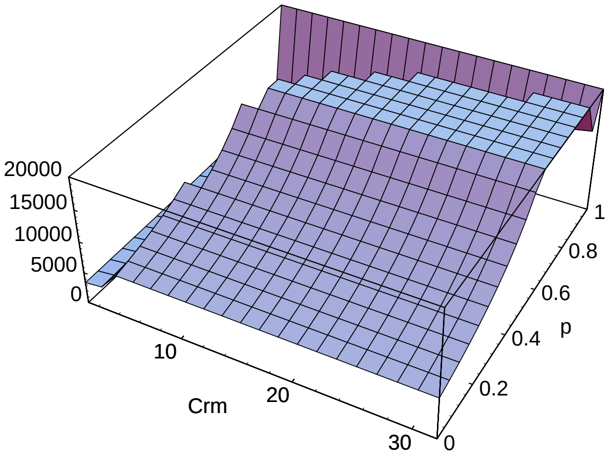

Figure 5.12: The Maximum Queue Length for $RIF = 1/64$ and $RDF = 1/16$.

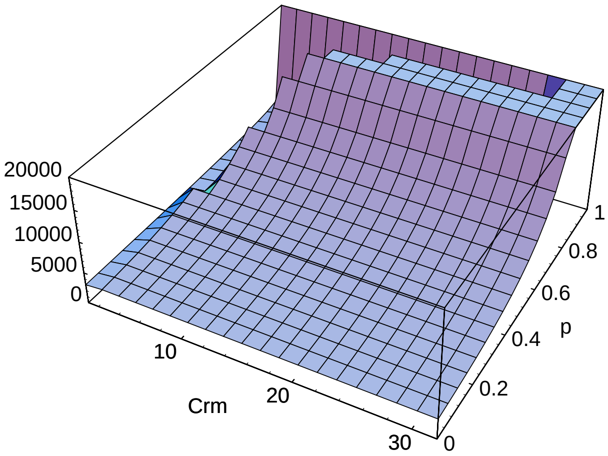

Figure 5.13: The Maximum Queue Length for $RIF = 1/1024$ and $RDF = 1/16$.

## 5.2 Simulation Results of Multi-Hop Network Configuration

In this section, we investigate a proper setting of rate-control parameters for a generic network configuration by simulation. In Subsection 5.2.1, we first introduce our simulation model. In Subsection 5.2.2, we then show simulation results to discuss the robustness of the rate-based congestion control in terms of cell loss, link utilization and fairness.

### 5.2.1 Simulation Model

Figure 5.14 illustrates our simulation model that is commonly referred to as the "parking lot" configuration [73, 18]. This model consists of five interconnected switches and four connections with different numbers of hops. The connection VC$n$ is established from SES$n$ to DES$n$. Each

VC$n$ enters the network at SW$n$, and all exit from SW5. Since each connection has the different number of hops, unfairness among these connections may be caused, which is our main concern in this section. Note that the link between SW4 and SW5 possibly becomes bottleneck in this model. The operation algorithm of source and destination end systems follows the standard draft [3]. Each source end system is assumed to always have cells to transmit, by which we evaluate robustness of the rate-based control in the worst case condition.

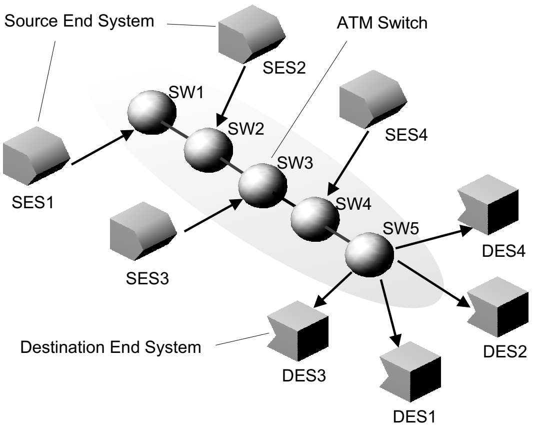

Figure 5.14: Parking Lot Configuration.

Bandwidth of all links is fixed at 150Mbit/s (353.7 cell/ms), and propagation delays between the source and the switch, $\tau_{sx}$, and between the destination and the switch, $\tau_{xd}$, are also fixed at 0.00 1ms (about 0.2 km). On the other hand, propagation delays between two interconnected switches, $\tau_{xx}$, are 0.01 ms or 1.00 ms (about 2 km and 200 km, respectively) as values for LAN and WAN environments. For intermediate switches, we model the binary mode switch with the FIFO scheduling, and provide 300 Kbyte (5,796 cells) of the buffer. Upper and lower threshold values in the buffer, $Q_H$ and $Q_L$, are fixed at the half of the buffer size. Other control parameters used in our simulation are $PCR = 150$ Mbit/s, $MCR = PCR/1000$, $ICR = PCR/20$, $TCR = 0.01$ cell/ms, $Mrm = 2$, $Trm = 100$, $C_{RM} = 32$, $CDF = 1/2$ and $TOF = 2$. Refer to [3] for the description of these control parameters.

As we have shown in [71, 72], key parameters that determine the efficiency and stability of the rate-based congestion control are $RIF$ and $RDF$. In these papers, we have analytically derived two boundary conditions for $RIF$ and $RDF$ to prevent cell loss and achieve full link utilization for the same model presented in Subsection 5.1.1. In [74], we have proposed a guideline for parameter tuning based on our analytic results and simulation experiments. Here, we summarize this guideline.

1. Estimate the round-trip delay, $\tau$, and the number of active connections, $N_{VC}$, in the worst case condition.

2. Obtain two boundary conditions for preventing cell loss and achieving full link utilization for these parameters from our analysis [71, 72].

3. Set $RDF$ to be a smaller value than 1/8, and determine $RIF$ that satisfies the condition of preventing cell loss.

In our simulation, the number of active connections is set to be constant but the round-trip delay and the number of hops for each connection is varied. Thus, it is impossible to directly apply our analysis to the simulation model. In what follows, we investigate how our analytic methods, which is for a single-hop model and homogeneous sources, should be applied to a more generic model.

One problem is in determination of the round-trip delay, $\tau$, that is used for obtaining two boundary conditions in our analysis. As we have shown in Subsection 5.1.2, the difference in propagation delays of connections has little effect on fairness. However, cell loss probability and link utilization are affected by the propagation delay since the larger round-trip delay implies the larger feedback delay [71, 72]. By letting $\tau_n$ be the round-trip delay for the $n$th connection, VC$n$, we consider three schemes for determining $\tau$ being applied to our analysis as follows.

**Scheme 1:** Adjust to the shortest connection

This scheme tunes parameters for the connection with the shortest round-trip delays. Thus, by assuming that VC1 has the shortest round-trip delay, we simply have

$$\tau = \tau_1$$

**Scheme 2:** Adjust equally to all connections

This scheme determines $\tau$ as an average of propagation delays of all connections. Thus, we have

$$\tau = \frac{1}{N_{VC}} \sum_{n=1}^{N_{VC}} \tau_n$$

**Scheme 3:** Adjust to the longest connection

This scheme is the opposite of Scheme 1; that is, parameters are tuned for the longest connection. Thus, by assuming that VC$N$ has the longest round-trip delay, we have

$$\tau = \tau_N$$

To compare these schemes, we plot two boundary lines for preventing cell loss and achieving full link utilization for $\tau_{xx} = 0.01$ ms, $\tau_{xx} = 0.10$ ms and $\tau_{xx} = 1.00$ ms in Figs. 5.15 through 5.17, respectively. $N_{VC}$ is fixed at 4, and $BL$ is 300 Kbyte to conform to the simulation parameters. The line labeled by "$Q_{max} = BL$" is the upper-bound of control parameters for preventing cell loss; that is, by selecting $RIF$ and $RDF$ from the lower region of this line, cell loss can be avoided. On the contrary, full link utilization can be fulfilled by selecting $RIF$ and $RDF$ from the upper region of the line labeled by "$Q_{min} = 0$". Hence, for preventing cell loss and achieving full link utilization, $RIF$ and $RDF$ should be chosen from the region between these two curves. One can find from these figures that when the round-trip delay is small, the boundary line for preventing cell loss (the "$Q_{max} = BL$" line) is nearly independent of schemes. However, the boundary line for full link utilization (the "$Q_{min} = 0$" line) is affected by schemes especially when the round-trip delay is large. Thus, for simulation of a WAN environment, we compare these three schemes although only Scheme 3 is used for simulation of a LAN environment.

### 5.2.2 Simulation Results

#### Case of LAN Environment

In this subsection, we show simulation results for a small propagation delay, $\tau_{xx} = 0.01$ ms, as a LAN environment. As described in the previous subsection, Scheme 3 is used for determining

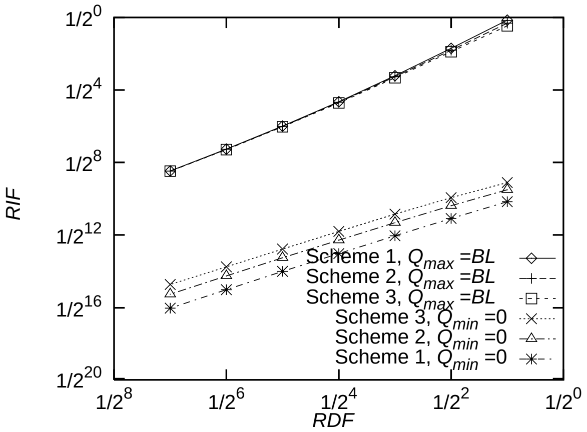

Figure 5.15: Analytic Results for Appropriate Parameters for $\tau_{xx} = 0.01$.

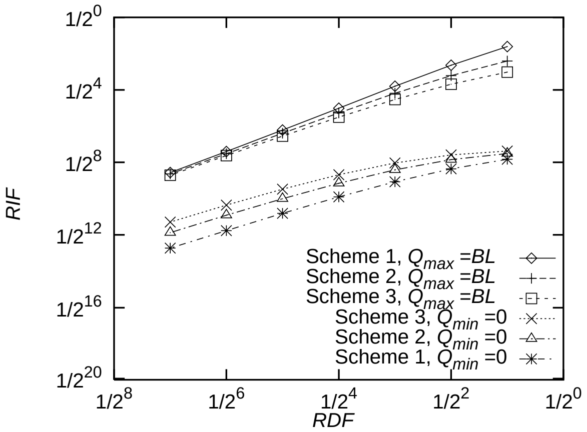

Figure 5.16: Analytic Results for Appropriate Parameters for $\tau_{xx} = 0.10$.

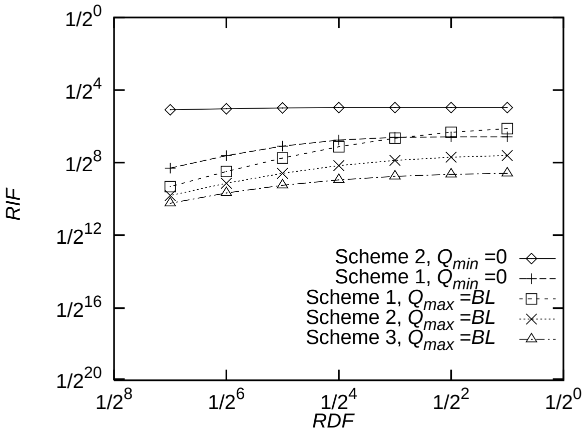

Figure 5.17: Analytic Results for Appropriate Parameters for $\tau_{xx} = 1.00$.

Table 5.2: Values of $(RIF, RDF)$ for LAN Environment.

| | fast down | moderate down | slow down |
|---|---|---|---|
| $Q_{max} = BL$ | (1/4, 1/4) | (1/32, 1/16) | (1/256, 1/64) |
| $\vdots$ | (1/32, 1/4) | (1/256, 1/16) | (1/1024, 1/64) |
| $Q_{min} = 0$ | (1/512, 1/4) | (1/2048, 1/16) | — |

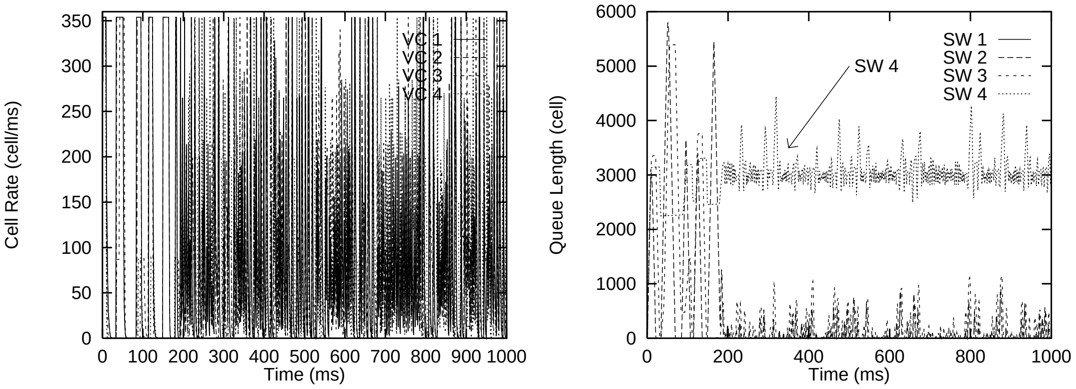

Figure 5.18: LAN Case for $RIF = 1/4$ and $RDF = 1/4$.

$\tau$. However, in this subsection, we use three values of $RIF$ for a given $RDF$ to investigate how the values of $RIF$ and $RDF$ should be chosen form the region between two boundary lines. We first fix $RDF$ to be 1/4 as a fast rate-down case. Then, three values of $RIF$ (1/4, 1/32 and 1/512) are chosen from Fig. 5.15. That is, $RIF = 1/4$ is chosen from the "$Q_{max} = BL$" line for preventing cell loss, and $RIF = 1/512$ is from the "$Q_{min} = 0$" line for achieving full link utilization. $RIF = 1/32$ is chosen as the midst of these values. Note that $RIF = 1/4$ is slightly smaller than the "$Q_{max} = BL$" line, and $RIF = 1/512$ is larger than the "$Q_{min} = 0$" line because $RIF$ and $RDF$ is represented in a form of $1/2^n$ [3]. We also use $RDF = 1/16$ and 1/64 as moderate and slow rate-down cases, respectively. We summarize values of $RIF$ and $RDF$ used in this subsection in Table 5.2.

Figures 5.18 through 5.20 show the cell transmission rate of each connection and the queue length at the switch for $RIF = 1/4$, $1/32$ and $1/512$, respectively. It can be found from these figures that SW4 is fully utilized when $RIF = 1/4$ and $1/32$, and that cell loss is prevented in all cases. However, fairness among connections is not fulfilled; longer connections (VC1 and VC2) transmit more cells than shorter connections (VC3 and VC4) (for example, VC1 reaches $PCR$ but VC4 does not in Fig. 5.18). it can be explained as follows. If SW4 becomes congested, each source decreases its rate by receiving backward RM cells of $CI = 1$. Because of different propagation delays, longer connections require more time to respond to congestion, and their ACR's remain high compared with shorter connections. Noting that an arrival rate of backward RM cells is proportional to its ACR, longer connections can receive much backward RM cells of $CI = 0$ after the congestion relief. Thus, longer connections can increase their ACR faster than the others, and it results in unfairness among connections.

We then change $RDF$ to 1/16 for slower rate decrease (the third column of Table 5.2). Simulation results for $RIF = 1/32$, 1/256 and 1/2048 are plotted in Figs. 5.21 through 5.23, respectively. We choose the values of $RIF$ similarly to the previous case. From the figures, we can

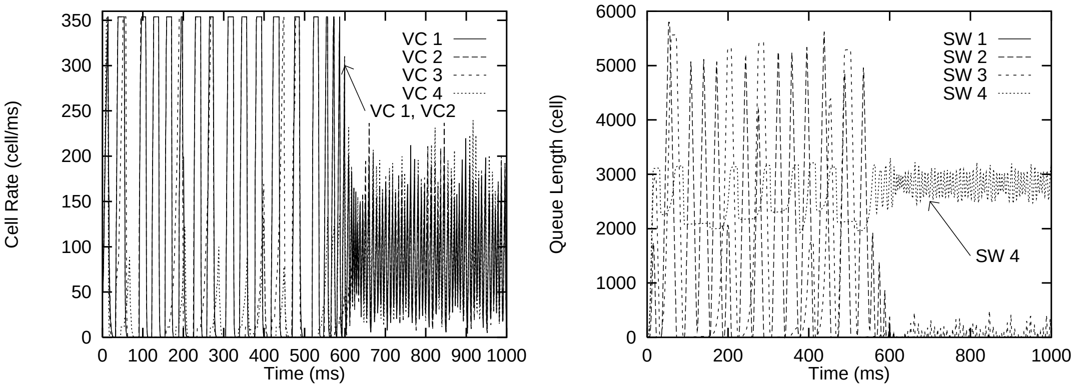

Figure 5.19: LAN Case for $RIF = 1/32$ and $RDF = 1/4$.

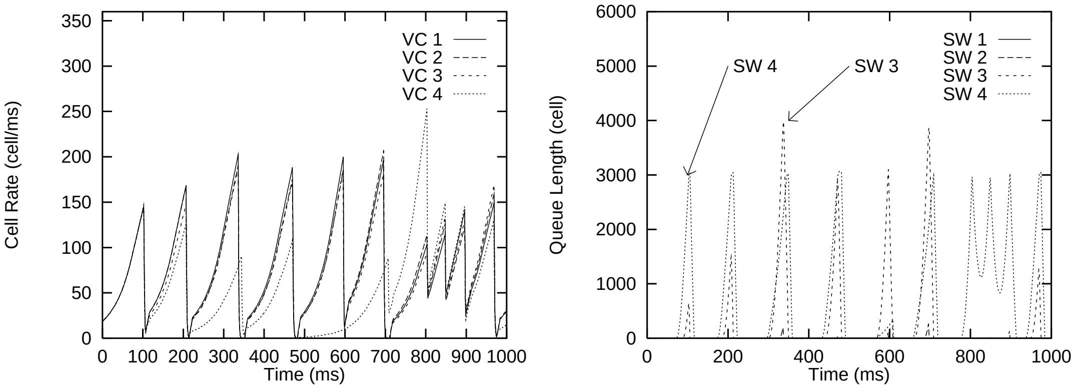

Figure 5.20: LAN Case for $RIF = 1/512$ and $RDF = 1/4$.

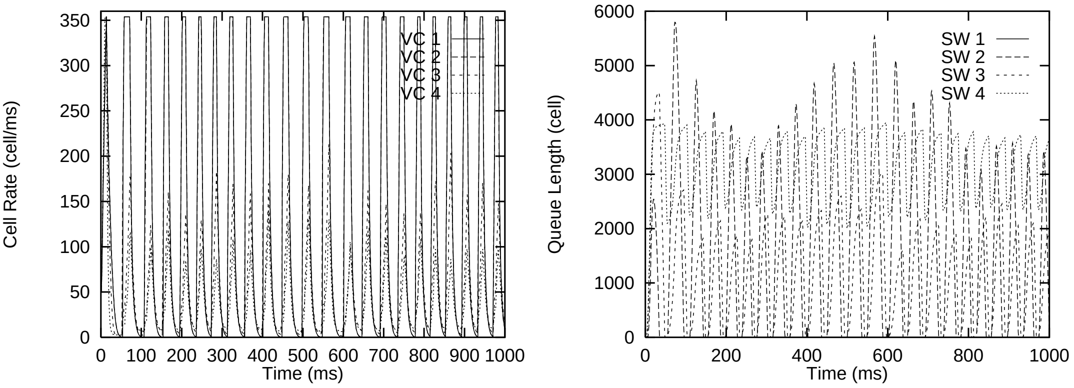

Figure 5.21: LAN Case for $RIF = 1/32$ and $RDF = 1/16$.

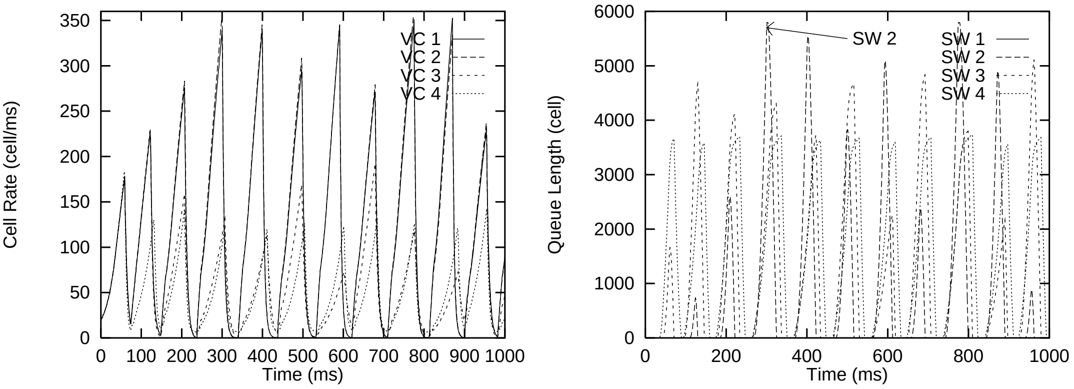

Figure 5.22: LAN Case for $RIF = 1/256$ and $RDF = 1/16$.

observe that $RIF = 1/2048$ achieves good fairness as well as full utilization of the bottleneck link while $RIF = 1/256$ can do neither. Furthermore, some cells are lost when $RIF = 1/256$. Therefore, we conclude that too fast rate increase/decrease degrades fairness among connections and utilization of the bottleneck link.

Figures 5.24 and 5.25 show simulation results for $RIF = 1/256$ and 1/1024 when $RDF$ is 1/64, which means much slower rate decrease. Both these figures indicate good fairness compared with the cases of $RDF = 1/4$ and 1/16. However, it should be noted that it takes longer time for each connection to be settled (around 500 ms in both cases). We conclude that a smaller value of $RDF$ (i.e., slower rate decrease) is proper for achieving good fairness and stable operation, and that $RIF$ should be chosen from the "$Q_{max} = BL$" line.

### Case of WAN Environment

The objective in this subsection is to compare three schemes explained in Subsection 5.2.1, and to investigate a proper setting of $RIF$ and $RDF$ in the WAN environment. We set the propagation delay between switches, $\tau_{xx}$, to 1.00 ms (about 200 km). As with the cases in Subsection 5.2.2, we use $RDF = 1/4$, 1/16 and 1/64. Then, for each of three schemes, we choose $RIF$ from the "$Q_{max} = BL$" line in Fig. 5.17. The values of $RIF$ and $RDF$ are summarized in

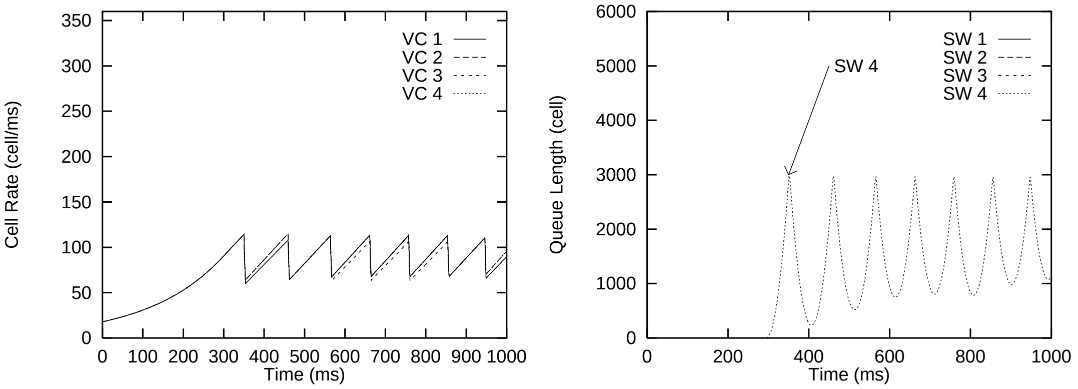

Figure 5.23: LAN Case for $RIF = 1/2048$ and $RDF = 1/16$.

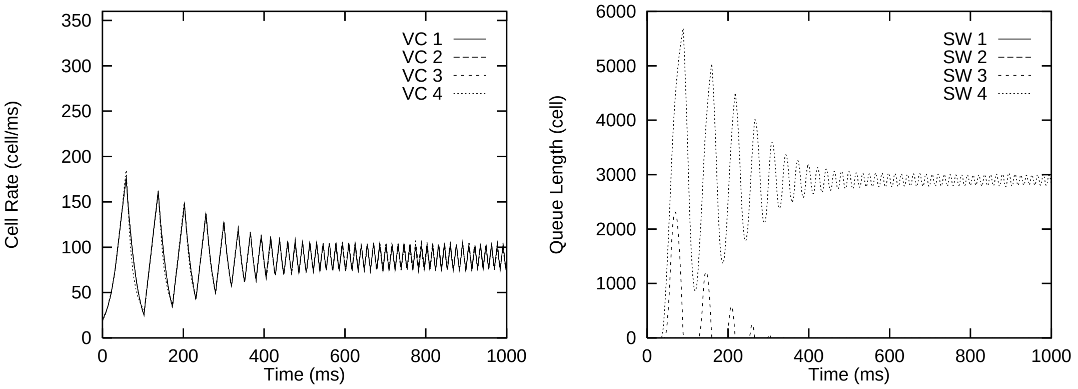

Figure 5.24: LAN Case for $RIF = 1/256$ and $RDF = 1/64$.

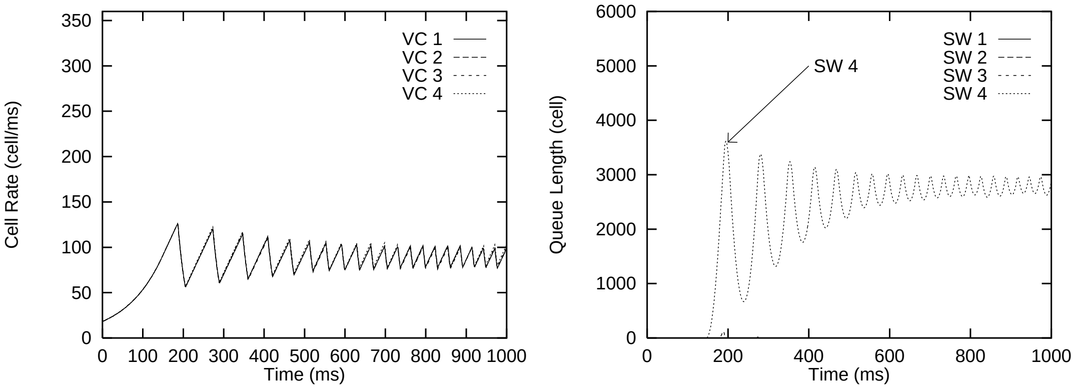

Figure 5.25: LAN Case for $RIF = 1/1024$ and $RDF = 1/64$.

Table 5.3: Values of $(RIF, RDF)$ for WAN Environment.

| | fast down | medium down | slow down |
|---|---|---|---|
| Scheme 1 | (1/128, 1/4) | (1/256, 1/16) | (1/512, 1/64) |
| Scheme 2 | (1/256, 1/4) | (1/512, 1/16) | (1/1024, 1/64) |
| Scheme 3 | (1/512, 1/4) | (1/512, 1/16) | (1/1024, 1/64) |

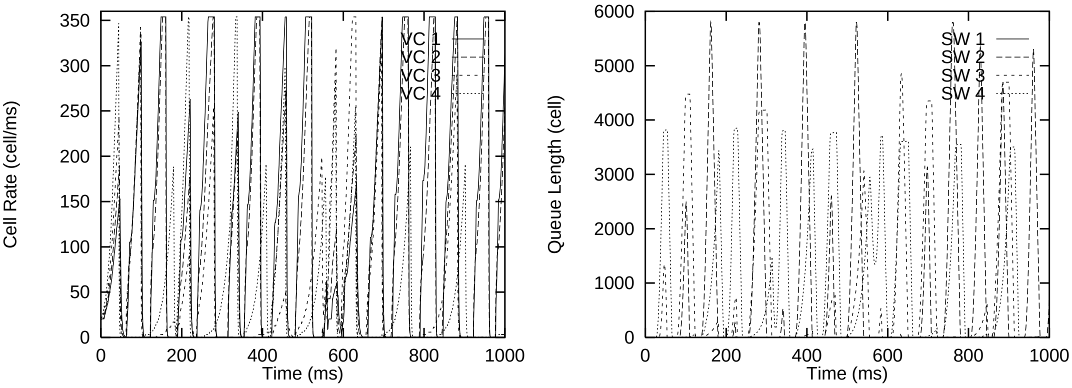

Figure 5.26: WAN Case for $RIF = 1/128$ and $RDF = 1/4$ (Scheme 1).

Table 5.3.

We first show simulation results for each scheme with $RDF = 1/4$ in Figs. 5.26 through 5.28. It is noted that in every scheme fairness among connections cannot be fulfilled, and that cell loss occurs regardless of $RIF$. In these figures, the case of $RIF = 1/128$ (Scheme 1) is at least better than others, but the use of these parameters should be avoided as explained later.

We next change $RDF$ from 1/4 to 1/16, which means slower rate decrease, and plot simulation results for each scheme in Figs. 5.29 and 5.30. As can be found from these figures, there is little improvement over the cases of $RDF = 1/4$. Although cell loss can be avoided by setting $RIF = 1/512$ (Schemes 2 and 3), fairness among connections is not still accomplished.

Finally, we change the rate decrease to be much slower ($RDF = 1/64$). Results are shown in Figs. 5.31 (Scheme 1) and 5.32 (Schemes 2 and 3). It can be easily found that a fairness problem is dramatically improved compared with previous two cases with $RDF = 1/4$ and 1/16. Namely, when the rate decrease is slow as $RDF = 1/64$, every scheme shows good performance in terms of good fairness, no cell loss and full link utilization. Finally, we conclude that rate-control parameters should be chosen from the "$Q_{max} = BL$" line with $\tau$ given by Scheme 2 and a smaller value of $RDF$ (slow rate decrease) in multi-hop network configurations.

## 5.3 Conclusion

In this chapter, we have investigated a proper setting of control parameters for the rate-based congestion control with binary-mode switch. For this purpose, we have mainly focused on two rate-control parameters, $RIF$ and $RDF$, which decides the envelope of rate increase / decrease.

First, we have presented two sorts of analyses. One was the analysis for the model with several groups of connections with different propagation delays in order to reveal the fairness

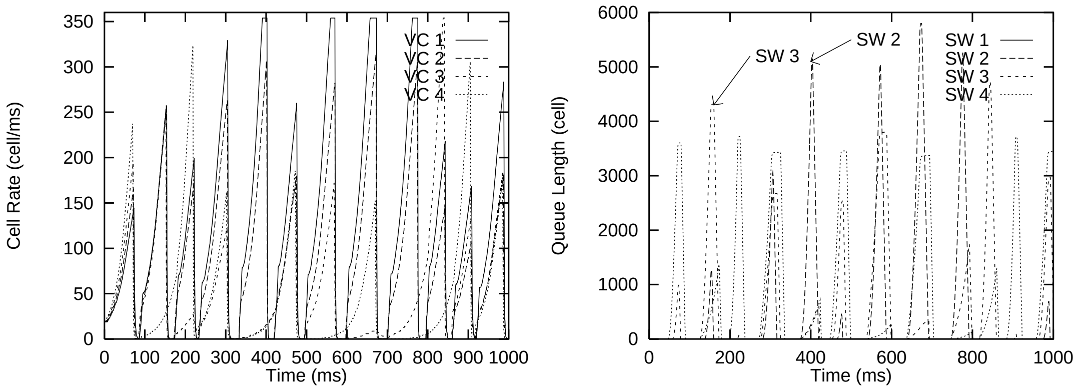

Figure 5.27: WAN Case for $RIF = 1/256$ and $RDF = 1/4$ (Scheme 2).

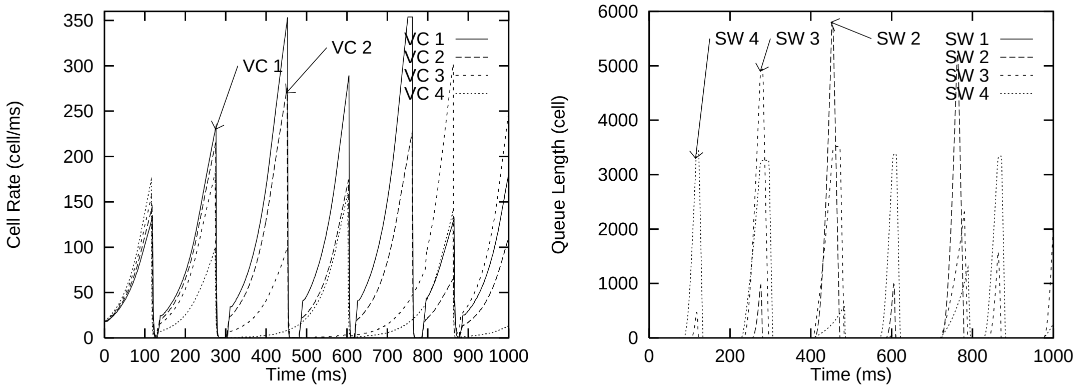

Figure 5.28: WAN Case for $RIF = 1/512$ and $RDF = 1/4$ (Scheme 3).

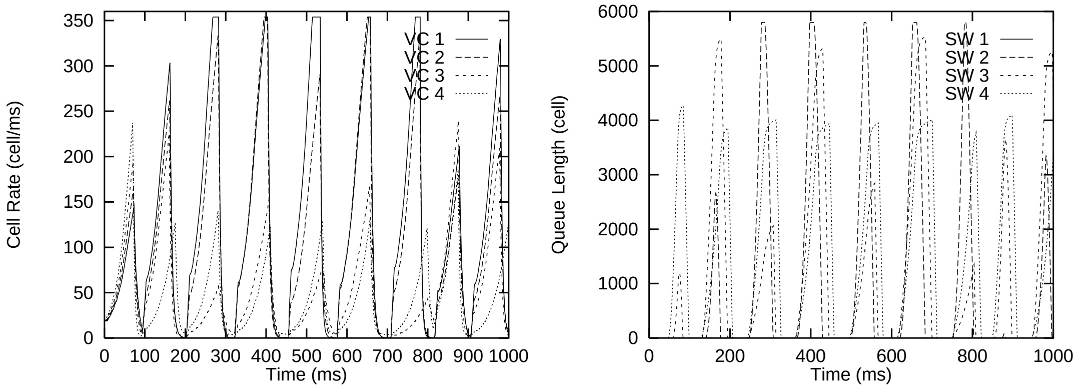

Figure 5.29: WAN Case for $RIF = 1/256$ and $RDF = 1/16$ (Scheme 1).

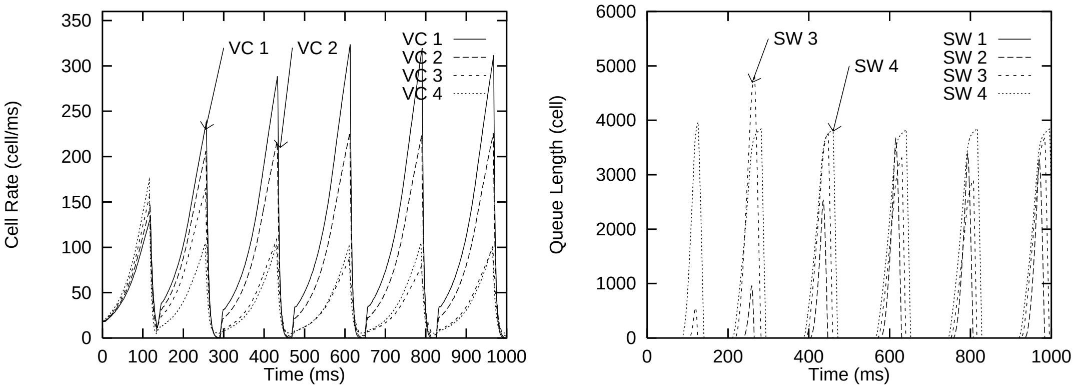

Figure 5.30: WAN Case for $RIF = 1/512$ and $RDF = 1/16$ (Schemes 2 and 3).

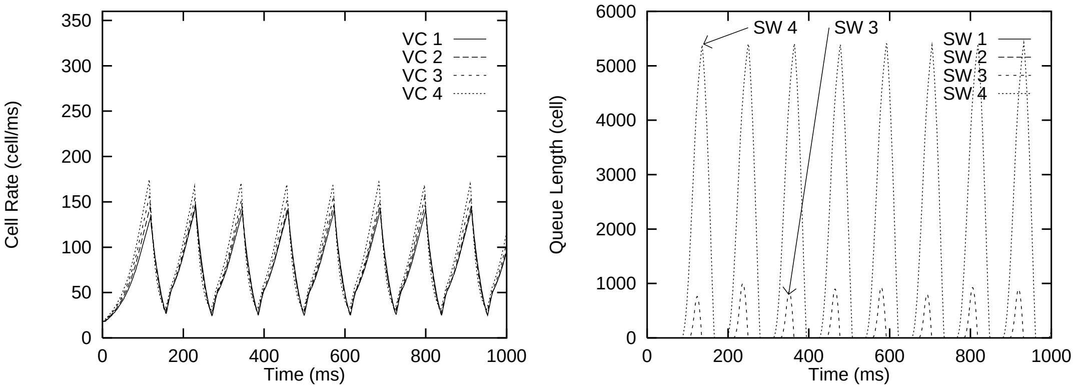

Figure 5.31: WAN Case for $RIF = 1/512$ and $RDF = 1/64$ (Schemes 1 and 2).

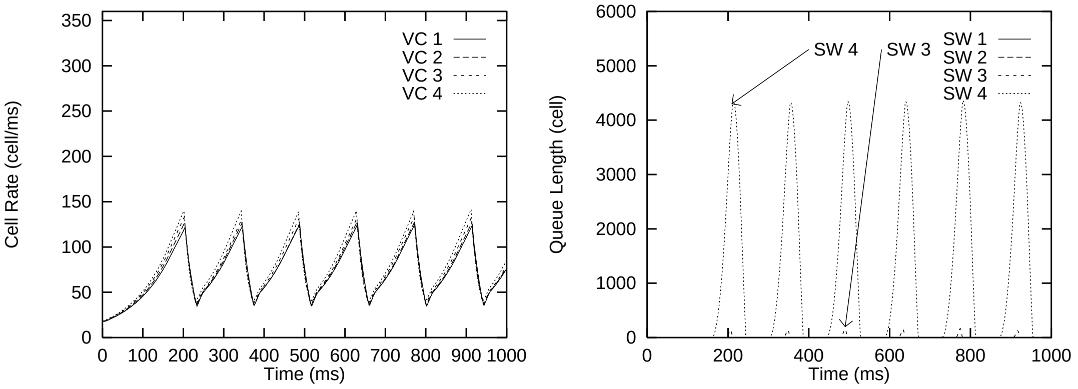

Figure 5.32: WAN Case for $RIF = 1/1024$ and $RDF = 1/64$ (Scheme 3).

problem among connections and the ramp-up time of an additional ABR connection. The other was the derivation of the maximum queue length at the switch buffer affected by an addition of background traffic such as CBR traffic. Through numerical examples, we have shown that a large value of $RIF$ (i.e., fast rate increase) is helpful to shorten the ramp-up time, and that a small value of $C_{RM}$ dramatically reduces the maximum queue length caused by CBR traffic.

Next, we have examined proper setting of $RIF$ and $RDF$ by simulation experiments. As a simulation model, we have used the parking lot configuration having five interconnected switches and four connections with different numbers of hops. We have compared three schemes for applying our analysis to more generic network configurations. It has been shown that $RDF$ should be set to a small value around 1/64 (i.e., slow rate decrease), and that $RIF$ should be set to a large value as long as cell loss can be prevented — the maximum value of $RIF$ is given by our analysis using Scheme 2 for parameter determination.

At the end of this chapter, we summarize the guideline for determining control parameters of the rate-based congestion control algorithm.

1. Estimate the number of active connections, $N_{VC}$, and their round-trip delays, $\tau_n$.

2. Choose $RDF$ around 1/64.

3. Calculate the average round-trip delay, $\tau$, as

$$\tau = \sum_{n=1}^{N_{VC}} \frac{\tau_n}{N_{VC}}.$$

4. For these $N_{VC}$, $\tau$,$RDF$ and other given parameters, solve the equation $Q_{max} = BL$ in [64, 72] for $RIF$ to obtain the maximum of $RIF$ that can prevent cell loss.

5. Choose $RIF$ smaller but closest to this solution.

6. $C_{RM}$ can be set to a small value (for example, 2) for preventing buffer overflow caused by background traffic.

# Chapter 6

# Designing Efficient Explicit-Rate Switch Algorithm for Rate-Based Congestion Control Algorithm

In the standard of the rate-based congestion control algorithm, two types of congestion notification methods of the switch are specified: EFCI marking and explicit-rate marking. In this chapter, we focus on more complicated explicit-rate marking switch. We first discuss design criteria of an explicit-rate switch to achieve high performance in terms of throughput, cell loss probability, fairness and so on. We next propose our explicit-rate switch algorithm that meets these design criteria and evaluate its performance through simulation experiments.

## 6.1 Design Goals of Explicit-Rate Switch Algorithm

In this section, we discuss design goals of the explicit-rate switch algorithm: high performance, transient performance, fairness, configuration simplicity, applicability to various environments and interoperability.

### 6.1.1 High Performance

When a cell is lost in the network due to, for example, buffer overflow at the switch, the entire packet (upper-layer protocol data unit) containing the lost cell should be retransmitted by the upper-layer protocol for reliable data communication. Thus, it is natural that the first design goal of the explicit-rate switch algorithm is to prevent cell loss. Since the buffer capacity is finite because of cost and / or technology limitation, the queue length should be controlled to prevent buffer overflow and resultant cell losses. An intuitive way to minimize the queue length is to set the sum of bandwidth allocated for all sources to be less than the actual bandwidth available to ABR connections. This approach is taken by ERICA as the *target utilization* (See Subsection 6.3.2). It is simple but it cannot fully utilize the available bandwidth.

Since the rate-based congestion control is inherently closed-loop, it takes a while for the source end system to respond to the feedback information; at least one round-trip time is required for all connections to adapt its $ACR$ to the ER value in the backward RM cell. Until all source end systems change their $ACR$ to new allocations, the queue length increases (or decreases) if the switch is overloaded (or under-loaded) because of divergence between the aggregation of the allocated bandwidth and the available bandwidth. Therefore, to minimize the queue growth, the ER value of the RM cell should be recomputed as soon as possible.

Since the network resources are limited, the rate-based congestion control algorithm should utilize the network resources effectively. Thus, the second design goal is to utilize the available bandwidth for ABR connections effectively. While minimizing the queue length is helpful for preventing cell loss, it sometime lowers the link utilization; that is, maintaining a small queue easily leads to buffer underflow. For these antithetical objectives — preventing cell loss and achieving full link utilization, the queue length should be kept at a certain level reasonably smaller than the buffer capacity but larger than zero. The third design goal is to shorten the cell delay experienced at the switch buffer. The small queue length is desirable for this purpose. In addition, cell delay variation should be minimized by limiting the queue length fluctuation at a certain level.

### 6.1.2 Transient Performance

The fourth design goal of the explicit-rate switch is to shorten a convergence time, defined as a time spent until the network reaches its steady state after the network status is changed (e.g., addition / disconnection of connections and increase / decrease of background traffic). For example, when a new connection is established on the network, the amount of incoming traffic is temporarily increased (i.e., the switch is overloaded), leading to the queue growth. On the other hand, when one connection is terminated and removed from the network, the amount of incoming traffic is decreased so that the switch may become idle. Therefore, it is important for the switch to reallocate the bandwidth for each connection immediately after the network status is changed.

It is desirable for the new connection to be able to gain an enough bandwidth quickly. Namely, a ramp-up time of the connection, defined as a time spent until obtaining enough bandwidth, should be minimized. This is the fifth design goal of the explicit-rate switch. The ramp-up time of the additional connection can be shortened by small control-loop of RM cells; One solution is to give a higher priority to RM cells than data cells. It is also helpful for the switch to generate an RM cell in backward direction. In the standard, the new connection is allowed to emit cells of $TBE$ (Transient Buffer Exposure) without receipt of backward RM cells [3]. $TBE$ is negotiated with the network at connection setup time, and is determined by the switch. The ramp-up time of the connection is minimized by assigning a large value of $TBE$, but too large $TBE$ may cause buffer overflow. Therefore, the switch should carefully determine a large $TBE$ for the new connection while preventing buffer overflow.

Moreover, the switch algorithm should operate effectively with the background traffic such as CBR and VBR traffic. Since the CBR / VBR traffic requires QoS (Quality of Services) guarantee, the cells of CBR / VBR service class must be given a higher priority than the cells of the ABR service class. Thus, the available bandwidth of the ABR service class is limited by existence of CBR and VBR traffic, and is dynamically changed. Henceforth, the sixth design goal of the explicit-rate switch is an adaptability to the background traffic. To recompute the bandwidth allocation according to the change of the background traffic, the switch must observe the bandwidth available to the ABR service class accurately. A possible scheme is to calculate the usable bandwidth in a fixed time interval by counting the number of arriving CBR / VBR cells. However, the monitoring interval should be set carefully since too small or too large interval sometimes results in inaccurate bandwidth estimation.

### 6.1.3 Fairness

Providing fair bandwidth allocation to all source end systems is also an important design goal of the explicit-rate switch. This is the seventh of the design goals. It is difficult for the binary-mode switch to allocate bandwidth for each connection fairly [75], but the explicit-rate switch has such an ability.

In conventional communication networks, an ideal bandwidth allocation scheme that maximizes the total throughput while preserving fairness among connections is called *max-min fairness* [59], which is adopted in [58]. In max-min fairness, the bandwidth is equally shared by connections if all connections are not constrained at other switches. That is, the fair share of the bandwidth, $FS$, is given by

$$FS = \frac{ABW}{N_{VC}}, \tag{6.1}$$

where $ABW$ is the available bandwidth for these connections, and $N_{VC}$ is the number of connections being established through the switch. However, the max-min fairness cannot be directly applied to the rate-based congestion control algorithm because it employs $PCR$ (Peak Cell Rate) and $MCR$ (Minimum Cell Rate) to guarantee the minimum and maximum transmission rate of the source end system. Therefore, in the rate-based congestion control algorithm, the fairness definition should take account of $PCR$ and $MCR$.

In what follows, we present several fairness definitions that are extensions of the max-min fairness to support $PCR$ and $MCR$. We consider the case when the number $N_{VC}$ of connections established at the switch. Let $FS_n$ be the fair share for the $n$th connection at the switch. We further introduce $MCR_n$ and $PCR_n$ as $MCR$ and $PCR$ of the $n$th connection, respectively. By considering $MCR_n$ and $PCR_n$, the fair share for the $n$th connection is generally represented by

$$FS_n \quad = \quad \alpha \times MCR_n + \beta \times (ABW - \alpha \times \sum_i MCR_i). \tag{6.2}$$

In the above equation, $\alpha$ and $\beta$ are weighting factors for bandwidth allocation determined as follows. Note that $MCR_n$ and $PCR_n$ are the lower and upper bounds of $FS_n$ so that the actual value of $FS_n$ is limited by $MCR_n$ and $PCR_n$. Namely,

$$FS_n \quad \leftarrow \quad \max(FS_n, MCR_n) \tag{6.3}$$

and

$$FS_n \quad \leftarrow \quad \min(FS_n, PCR_n). \tag{6.4}$$

**Scheme 1:** Max-Min Share

$$\alpha \quad = \quad 0 \text{ (or } 1) \tag{6.5a}$$

$$\beta \quad = \quad \frac{1}{N_{VC}} \tag{6.5b}$$

This scheme is similar to the max-min fairness criterion (equivalent if $\alpha = 0$). Namely, the bandwidth is allocated equally to all connections regardless of their $PCR$s and $MCR$s.

**Scheme 2:** Weighted Share with $MCR$

$$\alpha \quad = \quad 0 \text{ (or } 1) \tag{6.6a}$$

$$\beta \quad = \quad \frac{MCR_n}{\sum_i MCR_i} \tag{6.6b}$$

This scheme allocates the bandwidth proportional to the connection's $MCR$; that is, the connection with larger $MCR$ can obtain more bandwidth than other connections. Note that this scheme cannot be applied when there is a connection with $MCR_n = 0$.

**Scheme 3:** Weighted Share with $PCR$

$$\begin{aligned} \alpha &= 0 \text{ (or 1)} && \text{(6.7a)} \\ \beta &= \frac{PCR_n}{\sum_i PCR_i} && \text{(6.7b)} \end{aligned}$$

This scheme allocates bandwidth proportional to the connection's $PCR$; that is, the connection with larger $PCR$ can obtain more bandwidth than other connections.

**Scheme 4:** Weighted Share with $MCR$ and $PCR$

$$\begin{aligned} \alpha &= 0 \text{ (or 1)} && \text{(6.8a)} \\ \beta &= \left(\frac{MCR_n}{\sum_i MCR_i}\right)^{\gamma} \times \left(\frac{PCR_n}{\sum_i PCR_i}\right)^{\delta} && \text{(6.8b)} \end{aligned}$$

This scheme is a combination of Schemes 2 and 3; it allocates bandwidth according to both $MCR$ and $PCR$. In the above equation, $\gamma$ and $\delta$ are weight ratios ($0 \leq \gamma \leq 1$ and $0 \leq \delta \leq 1$).

The most suitable fairness definition is dependent on the policy of the network [76]; it is determined by the network designer or administrator. The switch algorithm, therefore, should be designed to work with all fairness definitions.

### 6.1.4 Configuration Simplicity

The eighth design goal of the explicit-rate switch is how easily control parameters of the source end system and the switch can be configured. The explicit-rate switch may have several internal control parameters depending on its algorithm. The optimal values of control parameters generally depend on the algorithm as well as the network configuration. The performance should be insensitive to the choice of control parameters. Otherwise, it should be easy for users or network administrators to configure control parameters intuitively or by the aid of a proper mechanism.

Moreover, the switch should know the number of active connections for exactly computing the bandwidth allocation for every connection. One possible solution would be to count the number of RM cells from different connections arriving within a fixed time interval. In this case, the interval must be chosen carefully. If it is too large, the switch cannot react rapidly to change of the number of connections. On the other hand, if it is too small, the switch would fail to count the actual number of active connections, leading miscomputation of the bandwidth allocation.

### 6.1.5 Applicability to Various Environments

An applicability to various environments is the ninth design goal of the explicit-rate switch algorithm. The algorithm of the explicit-rate switch should be designed by taking account of various network configurations, for example, LAN and WAN environments. Especially, the algorithm should work effectively in WAN environments as well as in LAN environments. The rate-based congestion control algorithm is inherently closed-loop so that difference in propagation delays of connections would result in dispersion of sources' responses to congestion. Namely, nearer connections to the congestion point may emit more or less cells than further ones, which causes unfairness among connections at different locations.

Table 6.1: Information table at the switch.

| Name | $VCI$ | $ER_F$ | $ER_B$ | $CA$ | constrained |
|---|---|---|---|---|---|
| Type | integer | float | float | float | boolean |

It is also required that the switch co-operates with other types of switches: binary-mode switches or other types of explicit-rate switches. Because of implementation complexity, the explicit-rate switch may be more expensive than the binary-mode switch. Therefore, it is likely that most of switches in the network are first binary-mode switches, and then some or all of them will be replaced by explicit-rate switches. In this scenario, the explicit-rate switch should work effectively when some of other switches are still binary-mode switches. One problem in such a mixed environment is how $RIF$ should be chosen. The source end system increases its $ACR$ by $PCR \times RIF$ at the receipt of the RM cell as in Eq. (1.4). To adjust $ACR$ to $ER$ immediately, $RIF = 1$ is an ideal value for the explicit-rate switch, but the binary-mode switch requires a smaller value [72]. For a compatibility, the explicit-rate switch should work effectively even with a smaller value of $RIF$.

### 6.1.6 Interoperability

As have been explained above, several types of switch algorithms may coexist in the same network. Thus, the explicit-rate switch algorithm should have interoperability with other types of switches, which is the last design goal. For this objective, it is essential for the switch to conform to the ATM Forum standard [3]. Namely, the switch should not place any assumption on the end systems and other switches except that these follow the standard specifications. It should also be avoided to create a new field in the RM cell unless its necessity is fully justified.

## 6.2 Designing Explicit Rate Switch Algorithm

In this section, we design an explicit-rate switch algorithm satisfying the design goals discussed in Section 6.1. Since our switch algorithm is based on the max-min scheme proposed by Tsang *et. al* in [58], we start this section with an introduction of the max-min scheme with reviewing its advantages and disadvantages. We next propose our enhancements to the max-min scheme, and explain how the defects of the original max-min scheme are resolved.

### 6.2.1 Max-Min Scheme

The max-min scheme always maintains an information table at the switch. Two entries are maintained for each connection. An entry of the table is listed in Table 6.1. In this table, $VCI$ corresponds to the VC identifier of the connection. $ER_F$ and $ER_B$ remember ER values written in the latest forward and backward RM cells, respectively. $CA$ is the current bandwidth allocation to this connection, and a *constrained* flag indicates whether this connection is constrained or not by other switches; if this flag is true, it means that this connection cannot achieve its fair share of the bandwidth at the switch. The constrained flag is used to allocate bandwidth according to the max-min fairness. At every receipt of forward and backward RM cells, the switch updates the associated entries and recomputes the bandwidth allocation for the connection as follows.

Suppose that the switch receives a forward RM cell. The switch first checks whether the ER value in the RM cell is different from $ER_F$. If different, it implies that the bandwidth allocation for this connection has been changed at other switches, and that the bandwidth allocation should be recomputed. Hence, the switch replaces $ER_F$ with the ER value in the RM cell, and updates the constrained flag by comparing $ER_F$ with the allocated bandwidth $CA$. Then, the following calculation of the bandwidth allocation is performed.

Let $FS$ be the fair share of the bandwidth for unconstrained connections (i.e., the constrained flag is false). $FS$ is computed as

$$FS = \frac{ABW - \sum_{n \in VC_C} CA_n}{|VC_U|}, \tag{6.9}$$

where $ABW$ is the available bandwidth to the ABR service class, and $CA_n$ is $CA$ of the $n$th connection. $VC_C$ and $VC_U$ are sets of constrained and unconstrained connection, respectively. $|VC_U|$ represents the number of unconstrained connections. The switch updates the constrained flag of each connection for $FS$, and assigns $FS$ to $CA$ of unconstrained connections. Namely, the constrained flag and $CA$ are determined as

$$\text{constrained} = \left\{ \begin{array}{ll} \text{true}, & FS \geq \min(ER_F, ER_B) \\ \text{false}, & FS < \min(ER_F, ER_B) \end{array} \right., \tag{6.10}$$

and

$$CA = \left\{ \begin{array}{ll} \min(ER_F, ER_B), & \text{if constrained} \\ FS, & \text{otherwise} \end{array} \right.. \tag{6.11}$$

The above process is repeated until there is no change in constrained flags. Finally, the ER value of the RM cell is updated as

$$ER \leftarrow CA. \tag{6.12}$$

Refer to [58] for more detail.

In what follows, we consider whether the max-min scheme satisfies the design goals described in Section 6.1 to clarify its advantages and disadvantages. In regard to *high performance* (Subsection 6.1.1), one defect of the max-min scheme is in lack of controllability of the queue length. The queue length should be controlled to prevent buffer overflow and to achieve full link-utilization. However, the queue length increases indefinitely in the max-min scheme as will be demonstrated in Section 6.3. Since the bandwidth allocation is finished in one round-trip time, the convergence time of the max-min scheme is shortest so that the criterion of *transient performance* (Subsection 6.1.2) is satisfied. However, a method to allocate $TBE$ for a new connection is not specified, and the effect of background traffic is not considered. The notable feature of the max-min scheme is that it can achieve the max-min fairness (Subsection 6.1.3). However, the max-min fairness cannot be fulfill in some condition due to deadlock of switches as will be exhibited in Subsection 6.3.2. Furthermore, $PCR$ and $MCR$ are not taken into account. The max-min scheme has no control parameters for switch operation. Thus, it is superior in terms of *configuration simplicity* (Subsection 6.1.4). In the point of *applicability* (Subsection 6.1.5), the improvement on the max-min scheme called MMDA (Max-Min Scheme with Delayed Adjustment), which achieves better fairness among connections with different propagation delays, is recently proposed in [77]. However, the max-min scheme has poor *interoperability* because the destination end system must reset the ER value of the RM cell to $PCR$, and an additional filed in the RM cell is required in MMDA.

### 6.2.2 Our Enhancements to Max-Min Scheme

In this subsection, we propose enhancements to the max-min scheme based on discussions in Subsection 6.2.1. The objective of our enhancements is to eliminate defects of the max-min scheme without losing its advantages. Advantages of our enhanced max-min scheme over the original max-min scheme are: (1) controllability of the queue length, (2) an effective $TBE$ allocation mechanism, (3) robustness against background traffic, (4) fairness achievement incorporating $PCR$ and $MCR$, and (5) interoperability. Details of our enhancements are described below.

The first enhancement is to control the queue length to a desired level. This mechanism is intended to prevent cell loss and to achieve full link-utilization as well as small cell delay. In our enhanced max-min scheme, the switch allocates the bandwidth to connections according to the current queue length. More strictly, the allocation of the ER value in Eq. (6.12) is changed as

$$ER \leftarrow CA \times z(Q(t)), \tag{6.13}$$

where $z(x)$ is a bandwidth adjustment function, and $Q(t)$ is a current queue length. The bandwidth adjustment function, $z(x)$, is a monotonically decreasing function having the following characteristics.

$$z(x) \quad = \quad \begin{cases} 1+\Delta_1, & x=0 \\ 1, & x=Q_T \end{cases} \tag{6.14}$$

and

$$1-\Delta_2 \le z(x) \le 1+\Delta_1 \tag{6.15}$$

$Q_T$ is a threshold value at the switch used to control the queue length. $\Delta_1$ and $\Delta_2$ are upper and lower bandwidth adjustment factors. For example, when the queue length is zero, the switch allocates $(1+\Delta_1)$ times larger bandwidth than the available bandwidth of the ABR service class. On the other hand, when the queue length is greater than $Q_T$, the switch reduces the bandwidth allocation. By introducing this mechanism, the queue length is managed to be kept at $Q_T$. Namely, if the queue length is below $Q_T$, the switch tries to increase its queue length by allocating more bandwidth. If the queue length is over $Q_T$, the switch tries to decrease its queue length. Hence, the queue length is restored at $Q_T$ even when the switch gets overloaded or underloaded.

The second enhancement for the max-min scheme is to support various fairness definitions with $PCR$ and $MCR$ defined in Subsection 6.1.3. To take account of $PCR$ and $MCR$, the equation for computing the fair share, Eq. (6.9), is further extended as

$$FS_n = \alpha \times MCR_n + \beta \times \left( \left[ ABW - \sum_{n \in VC_C} CA_n \right] - \alpha \times \sum_{n \in VC_U} MCR_n \right), \tag{6.16}$$

where $\alpha$ and $\beta$ are given by one of Eqs. (6.5)–(6.8). Note that Schemes 2 and 3 require an additional capability at the switch for maintaining $PCR$ values of all connections although $MCR$ values of all connections are stored in the RM cell. In our enhanced max-min scheme, the available bandwidth to the ABR service class is computed at the switch by monitoring the number of arriving CBR and VBR cells within a fixed interval. More specifically, by letting $I$ be the bandwidth monitoring interval and $N$ be the number of CBR and VBR cells received during $I$, the available bandwidth $ABW$ is computed as

$$ABW = BW - \frac{N}{I}. \tag{6.17}$$

We next explain our mechanism to allocate $TBE$ for a new connection. Let us assume that there are $N_{VC}$ active connections on the link, and $(N_{VC}+1)$th connection starts cell emission at $t=t_0$. At the connection setup time, the switch determines $TBE$ for this connection as

$$TBE = \min(RTT \times PCR,\, BL - \max(Q(t), Q_T) - \sum_{n=1}^{N_{VC}} R_n), \tag{6.18}$$

where $R_n$ is a reserved buffer capacity for $n$th connection, and $BL$ is the buffer size at the switch. $RTT$ is an estimated round-trip delay of the RM cell including processing delays, which is signaled at connection setup [3]. The buffer reservation, $R_n$, is valid until the source end system receives the first backward RM cell from the network; that is, $R_n$ is canceled at $t = t_0 + RTT$. Thus, the buffer reservation for $(N_{VC}+1)$th connection is given by

$$R_{N_{VC}+1} = \begin{cases} TBE, & t_0 \le t \le t_0 + RTT \\ 0, & t_0 + RTT < t \end{cases} \tag{6.19}$$

Given $TBE$ from the network, the source end system computes $ICR$ (Initial Cell Rate) as (see [3])

$$ICR \leftarrow \min(ICR, \frac{TBE}{RTT}). \tag{6.20}$$

By employing the $ICR$ negotiation mechanism, buffer overflow caused by activation of a new ABR connection can be completely avoided. Another possibility of buffer overflow is when background traffic suddenly increases its bandwidth requirements. In what follows, we investigate a proper setting of control parameters satisfying two objectives: preventing cell loss and achieving full link utilization.

From now on, we analyze the maximum and minimum of the queue length by assuming infinite buffer capacity. To analyze the worst case, we assume that all connections are not constrained at other switches, and that all source end systems always have cells to transmit. We further assume that the network is in steady-state; the queue length is equal to $Q_T$ because of the queue control mechanism of our enhanced max-min scheme. Let $N_{VC}$ denote the number of active connections. We introduce $\tau_{sxn}$ and $\tau_{xdn}$ $(1 \le n \le N_{VC})$ as the propagation delays between the $n$th source end system and the switch, and between the switch and the destination end system. The bandwidth of the link is denoted by $BW$.

When the amount of the background traffic is increased from $C$ to $C'$ $(C' > C)$ at $t = t_0$, the switch immediately recomputes new bandwidth allocations and notifies them to source end systems via the ER values of RM cells. In this case, the bandwidth allocation for each connection is changed from $(BW-C)/N_{VC}$ to $(BW-C')/N_{VC}$. Since the RM cell containing a new explicit-rate arrives at the $n$th source end system $\tau_{sxn}$ after the arrival rate of the background traffic is changed, cells are excessively injected into the network. Thus, the envelope of the queue length is given by

$$Q(t) = Q_T + \int_{t_0}^{t} \left( \sum_{n=1}^{N_{VC}} ACR_n(t - \tau_{sxn}) - (BW - C') \right) dx, \tag{6.21}$$

where $ACR_n(t)$ is the bandwidth allocated for the $n$th connection. The backward RM cell with the new bandwidth allocation of $(BW - C')/N_{RM}$ are received by the $n$th source at $t = t0 + \tau_{sxn} + t_{RM}$, where $\tau_{sxn}$ is the propagation delay from the switch to the source end system and $t_{RM}$ is a delay for the next RM cell at the switch. Thus, $ACR_n(t)$ is given by

$$ACR_n(t) = \begin{cases} \frac{BW-C}{N_{VC}}, & t \le t_0 + \tau_{sxn} + t_{RM} \\ \frac{BW-C'}{N_{VC}}, & t > t_0 + \tau_{sxn} + t_{RM} \end{cases}, \tag{6.22}$$

and

$$
\begin{aligned}
t_{RM} &\leq \frac{N_{RM}}{ACR_n(t - \tau_{sxn} - 2\tau_{xdn}))} && (6.23)\\
&= \frac{N_{RM} \times N_{VC}}{BW - C}. && (6.24)
\end{aligned}
$$

The maximum queue length, $Q_{max}$, is obtained as

$$
\begin{aligned}
Q_{max} &= \lim_{t \to \infty} Q(t) && (6.25)\\
&= Q_T + \sum_{n=1}^{N_{VC}} \left( \frac{C' - C}{N_{VC}} \times (2\tau_{sxn} + t_{RM}) \right) && (6.26)\\
&\leq Q_T + (C' - C) \times \left( 2 \times \max_n(\tau_{sxn}) + \frac{N_{RM} \times N_{VC}}{BW - C} \right) && (6.27)
\end{aligned}
$$

Hence, to prevent buffer overflow, $Q_T$ should be chosen to satisfy the following relation.

$$Q_{max} \leq BL \qquad (6.28)$$

The queue decreases when the amount of background traffic is decreased. When the amount of background traffic is changed from $C$ to $C''$ ($C'' < C$) at $t = t_0$, the envelope of the queue length is simply given by replacing $C'$ in Eq. (6.21) with $C''$. As with the previous case, the minimum queue length is given by

$$
\begin{aligned}
Q_{min} &= Q_T + \sum_{n=1}^{N_{VC}} \left( \frac{C'' - C}{N_{VC}} \times (2\tau_{sxn} + t_{RM}) \right) && (6.29)\\
&\leq Q_T + (C'' - C) \times \left( 2 \times \max_n(\tau_{sxn}) + \frac{N_{RM} \times N_{VC}}{BW - C} \right) && (6.30)
\end{aligned}
$$

Thus, full link utilization can be achieved by setting $Q_T$ to satisfy the following relation.

$$Q_{min} \geq 0 \qquad (6.31)$$

In our enhanced max-min scheme, three control parameters — $Q_T$, $\Delta_1$, $\Delta_2$ and $I$ — are adopted for fulfilling high performance in exchange for configuration simplicity. However, the threshold value, $Q_T$, can be configured according to the above analysis. An appropriate setting of the monitoring interval, $I$, is inevitable in our method so that it will be investigated through simulation experiments in Subsection 6.3.4.

In the original max-min scheme, the destination end system must reset the ER value in the RM cell to $PCR$. It requires an additional hardware to maintain $PCR$ values of active connections at the destination end system, and does not follow the ATM Forum standard. In our enhanced max-min scheme, such a mechanism is not required; the destination end system simply sends back the RM cell.

## 6.3 Performance Evaluation

### 6.3.1 Simulation Model

Figure 6.1 shows our simulation model, which consists of two inter-connected explicit-rate switches and four ABR connections with identical propagation delays. In the following simulation, the link bandwidth, $BW$, is fixed at 353.7 cell/ms assuming a 150 Mbit/s link. The

Table 6.2: Control parameters at the source end system.

| Parameter Name | Assigned Value |
|---|---|
| $PCR$ (Peak Cell Rate) | $BW$ |
| $MCR$ (Minimum Cell Rate) | $PCR/1000$ |
| $ICR$ (Initial Cell Rate) | $PCR$ |
| $TCR$ (minimum rate for data cells) | 0.01 |
| $RIF$ (Rate Increase Factor) | 1 |
| $RDF$ (Rate Decrease Factor) | 1 |
| $N_{RM}$ (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 |
| $TOF$ (Time Out Factor) | 2 |
| $TDF$ (Time out Decrease Factor) | $ICR$ / $2^{14}$ |

propagation delay of each link (source–switch, switch–switch and switch–destination) is fixed at an identical value denoted by $\tau$. A round-trip delay between source and destination end systems is, therefore, $6 \times \tau$. We use two values of $\tau$: 0.01 ms (about 2 km) as LAN environments and 1.00 ms (about 200 km) as WAN environments. Thus, the round-trip delay is 0.06 ms for LAN environments or 6.00 ms for WAN environments.

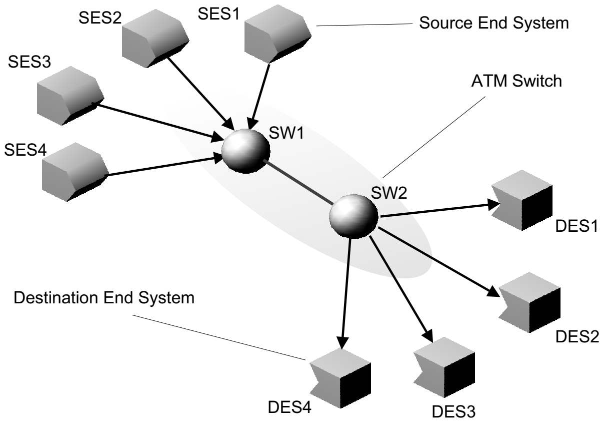

Figure 6.1: Our Simulation Model.

At each switch, its buffer size, $BL$, is set to 300 Kbyte (5,796 cells). We assume persistent sources; all source end systems always have cells to transmit. In other words, we assume that $CCR$ of the source end system is always equivalent to $ACR$. We summarize values of control parameters at the source end system used in our simulation in Table 6.2. See [3] for complete description of control parameters.

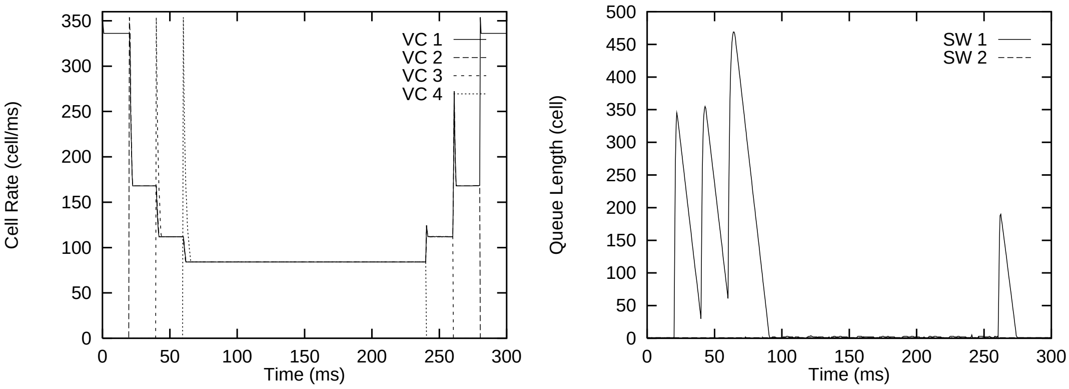

Figure 6.2: Effect of connection addition / disconnection in ERICA for $\tau = 0.01$ ms and target utilization of 0.95.

### 6.3.2 Addition and Departure of ABR Connections

In this subsection, we compare three explicit-rate switch algorithms: ERICA, the max-min scheme and our enhanced max-min scheme. The main objective of this section is to evaluate the influence of connection addition and departure. So we add four connections to the network at different starting points, $t$ = 0, 20, 40 and 60 ms, and remove them from the network at $t$ = 300, 280, 260 and 240 ms, respectively. For comparison purposes, the $TBE$ determination algorithm in Subsection 6.2.2 is not used. Instead, we set the initial cell rate, $ICR$, to be $PCR$.

We first show simulation results for ERICA in Figs. 6.2 and 6.3 for different propagation delays, $\tau$ = 0.01 and 1.00 ms, respectively. A target utilization and a load averaging interval are set to be 0.95 and 100 cell time. In ERICA, the target utilization is used to limit the bandwidth allocation for ABR connections; that is, (target utilization $\times BW$) of the bandwidth is shared by ABR connections, and the rest of the bandwidth is not allocated to absorb the rate fluctuation. The load averaging interval is an interval for monitoring the current traffic load at the switch. Readers should refer to [47] for details of ERICA.

Each graph shows $ACR$s of source end systems and queue lengths of switches. As can be found from these figures, the queue length grows when the new connection is activated (around $t$ = 20, 40 and 60 ms), and the maximum queue length is about 470 cells in the LAN environment. Since the target utilization is less than 1.0, the buffered cells are gradually processed and the queue length diminishes. In simulation, the queue length is decreased in 30 ms, and the maximum queue length is limited even with several new connections. In the WAN environment, however, many cells are lost due to buffer overflow as can be found from Fig. 6.3. The number of lost cells was 59,927 cells during the simulation run. It can also be found that fairness among connections is not fulfilled. This problem also occurs in EPRCA++, which is the previous version of ERICA [53]. Buffer overflow can be avoided by setting the target utilization to be a much smaller value [53]. In Fig. 6.4, we change the target utilization from 0.95 to 0.70. In this figure, cell loss can be prevented although the maximum queue length is still large. It should be noted that setting a small value of the target utilization causes lower utilization of the bandwidth.

In Figs. 6.5 and 6.6, we next show simulation results of the original max-min scheme for $\tau$ = 0.01 and 1.00 ms. From the figures, it can be found that cell loss can be prevented even in the WAN environment, and that the maximum queue length is much smaller than the one obtained by ERICA. It is because the max-min scheme can adjust $ACR$ of the new connection

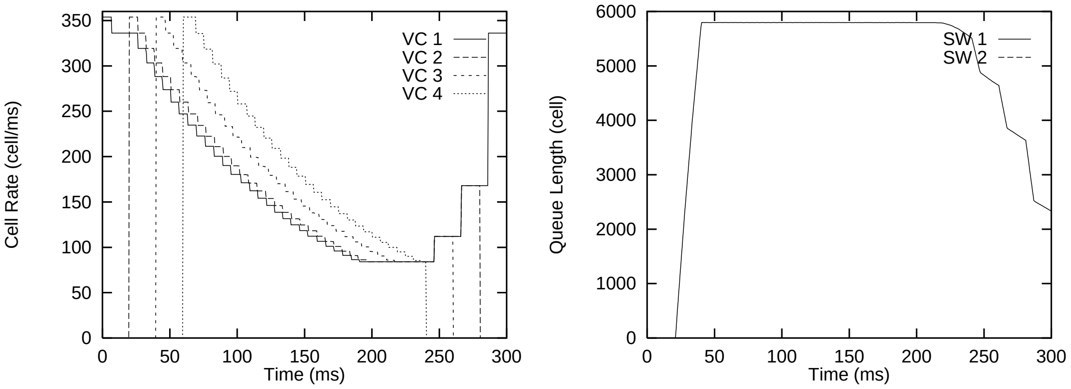

Figure 6.3: Effect of connection addition / disconnection in ERICA for $\tau = 1.00$ ms and target utilization of 0.95.

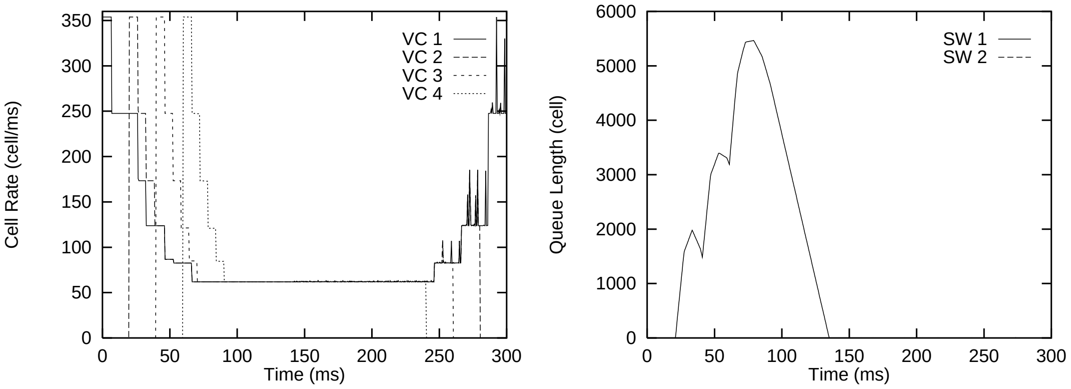

Figure 6.4: Effect of connection addition / disconnection in ERICA for $\tau = 1.00$ ms and target utilization of 0.70.

| $VCI$ | $ER_F$ | $ER_B$ | $CA$ | constrained |
|---|---|---|---|---|
| $1 \sim 4$ | 353.7 | 88.4 | 88.4 | true |

Table 6.3: Information table at SW1 before VC4 terminates.

| $VCI$ | $ER_F$ | $ER_B$ | $CA$ | constrained |
|---|---|---|---|---|
| $1 \sim 4$ | 88.4 | 353.7 | 88.4 | true |

Table 6.4: Information table at SW2 before VC4 terminates.

to the correct value in one round-trip time. However, the serious problem of the max-min scheme is that each connection cannot increase its $ACR$ even when one or more connections are terminated. Namely, max-min fairness is not satisfied after $t$ = 240 ms. This is due to a *deadlock problem* of the max-min scheme explained as follows. Tables 6.3 and 6.4 show information tables maintained at SW1 and SW2 before VC4 terminates at $t$ = 240 ms. Note that all connection have the same entry. When VC4 terminates, the switch tries to reallocate the available bandwidth. Since there are three active connections, the switch computes the fair share, $FS$, as $BW/3$ (= 117.9 cell/ms) according to Eq. (6.9). However, the minimum of $ER_F$ and $ER_B$ is 88.4 cell/ms at both SW1 and SW2, all connections are regarded as constrained connection. Consequently, the bandwidth allocation for each connection is still limited to 88.4 cell/ms (see Eqs. (6.10) and (6.11)).

Another problem of the max-min scheme is that the queue length is settled at a high level. It becomes more apparent in the WAN environment as shown in Fig. 6.6. In the figure, the maximum queue length is about 4,700 cells, and cells would be lost if one more connection is added to the network. In other words, it takes long time for the queue length to be decreased because the max-min scheme tries to fully utilize the available bandwidth even though the queue length is almost full. This problem can be avoided by restricting the bandwidth allocation to slightly smaller than the available bandwidth. We replace Eq. (6.9) of the max-min scheme with

$$FS = \frac{0.99 \times ABW - \sum_{n \subseteq VC_C} CA_n}{|VC_U|}, \tag{6.32}$$

although the authors do not notice this problem since they only consider the LAN environment [58]. Figures 6.7 and 6.8 are also of the max-min scheme but the above equation is adopted to calculate the fair share. The queue length is considerably decreased especially in the LAN environment. However, the queue length is still large and would not decrease in a short time in the WAN environment. It is due to lack of the queue length controllability as pointed out in Subsection 6.2.1.

We next show simulation results of our enhanced max-min scheme in Figs 6.9 and 6.10 for $\tau$ = 0.01 and 1.00 ms, respectively. In these figures, $Q_T$ is chosen according to our analysis presented in Subsection 6.2.2: in these cases, $Q_T$ = 138 in the LAN environment and $Q_T$ = 1,189 in the WAN environment. Bandwidth adjustment factors, $\Delta_1$ and $\Delta_2$, are set to be 0.2 and 0.5, respectively. It can be found from these figures that the maximum queue length is small, and that the queue length is stabilized at $Q_T$. It can also be found that the queue length is decreased quickly once the queue length exceeds $Q_T$. It is owing to the mechanism of our enhanced max-min scheme to control the queue length. Our enhanced max-min scheme frequently updates the bandwidth allocation when compared with the original one. However, frequent computation of the bandwidth allocation would be indispensable when the background traffic coexists in the network.

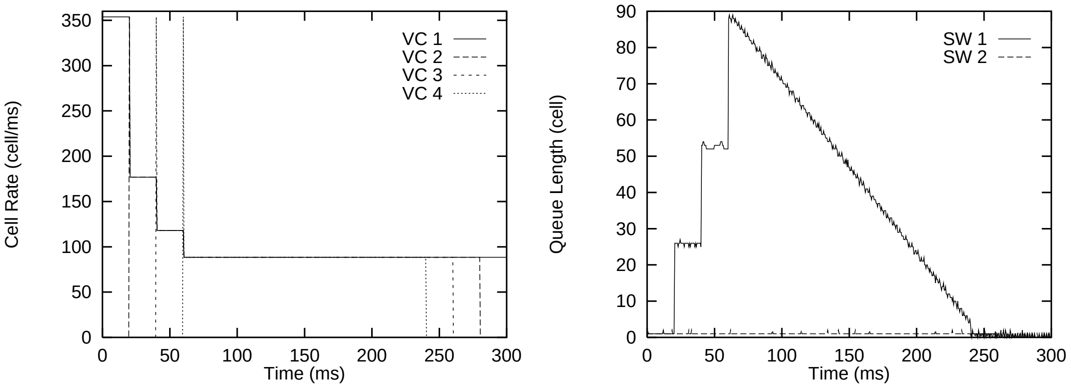

Figure 6.5: Effect of ABR connection arrival / departure in max-min scheme for $\tau = 0.01$ ms.

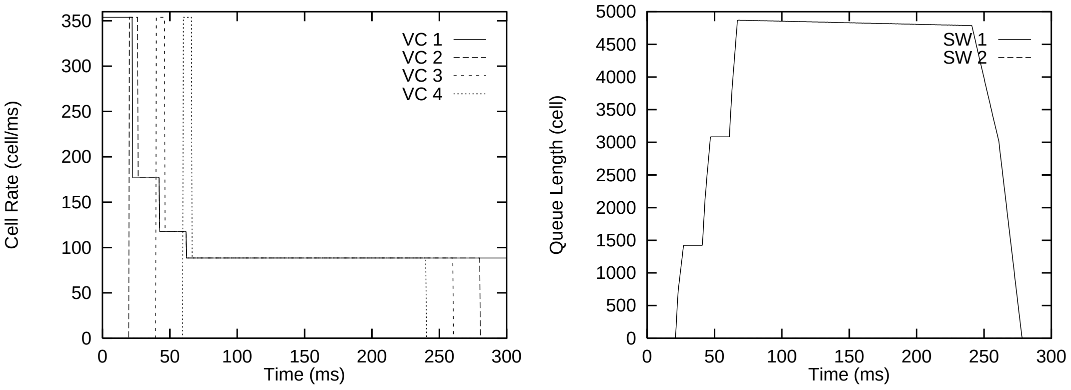

Figure 6.6: Effect of ABR connection arrival / departure in max-min scheme for $\tau = 1.00$ ms.

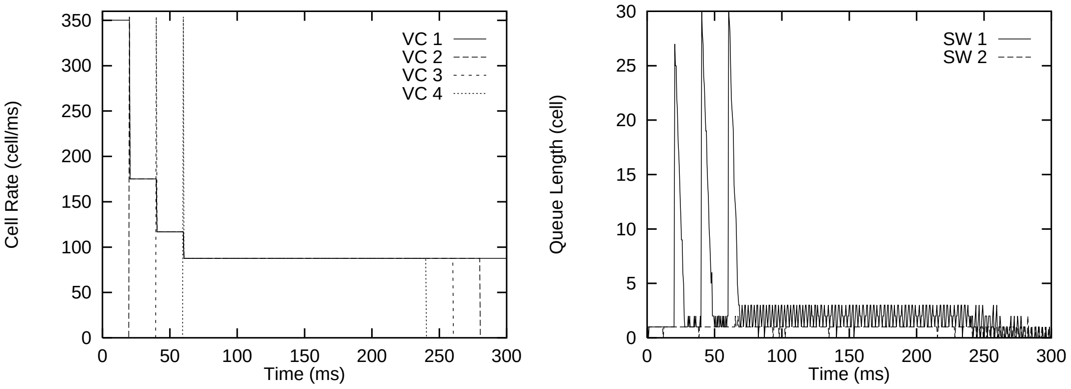

Figure 6.7: Effect of ABR connection arrival / departure in max-min scheme for $\tau = 0.01$ ms.

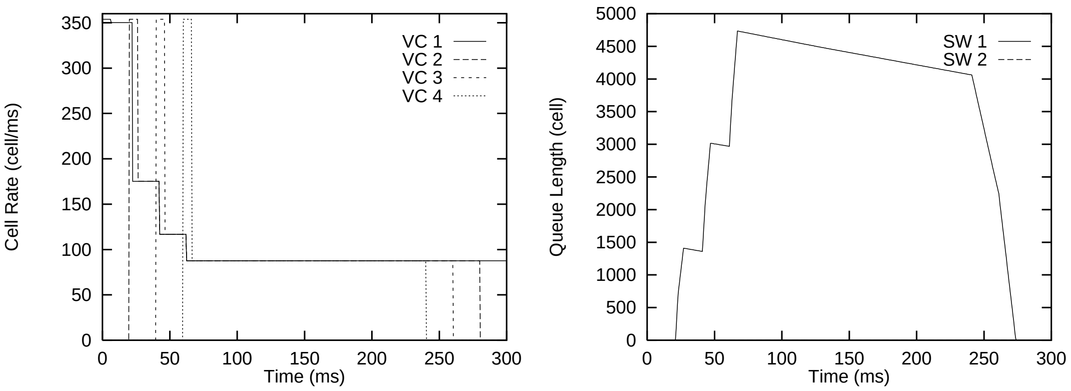

Figure 6.8: Effect of ABR connection arrival/ departure in max-min scheme for $\tau = 1.00$ ms.

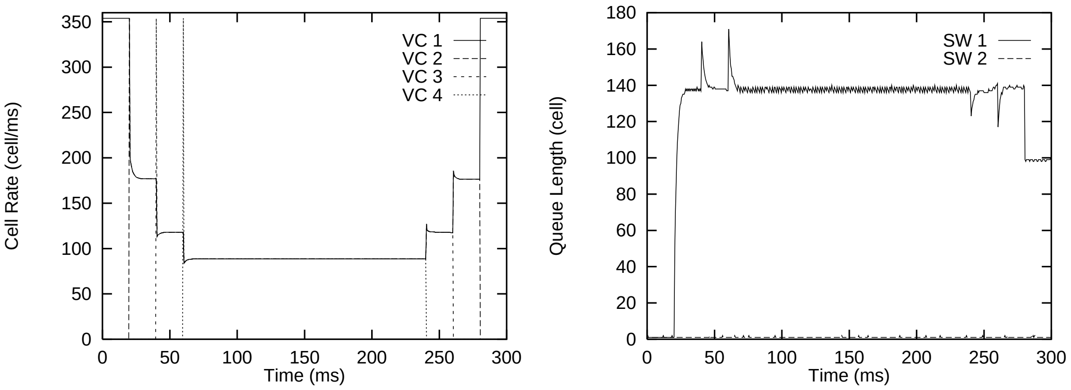

Figure 6.9: Effect of ABR connection arrival/ departure in enhanced max-min scheme for $\tau = 0.01$ ms.

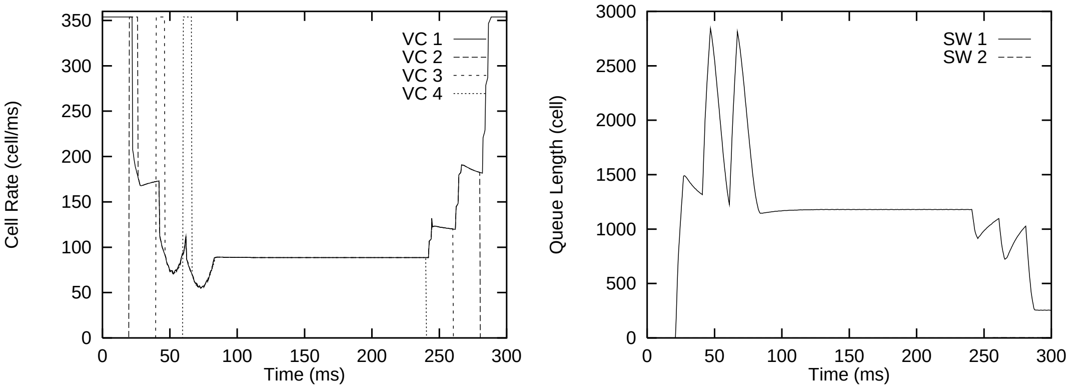

Figure 6.10: Effect of ABR connection arrival/ departure in enhanced max-min scheme for $\tau = 1.00$ ms.

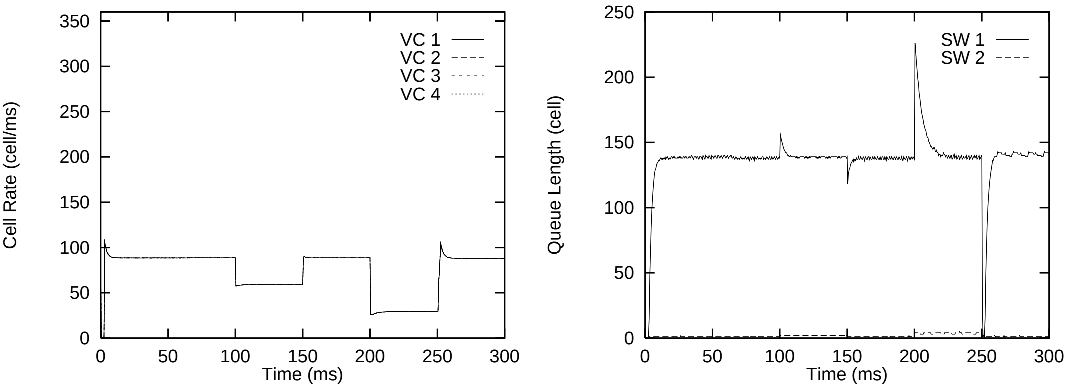

Figure 6.11: Effect of CBR traffic in enhanced max-min scheme for $\tau = 0.01$ ms.

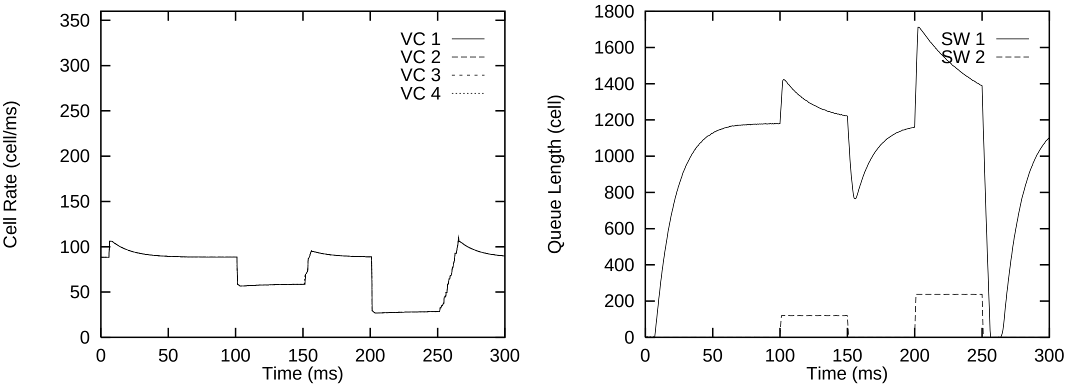

Figure 6.12: Effect of CBR traffic in enhanced max-min scheme for $\tau = 1.00$ ms.

### 6.3.3 Effect of CBR Traffic

We next focus on the effect of the CBR traffic on our enhanced max-min scheme. As explained in Subsection 6.1.2, the bandwidth available to the ABR service class is sustained by the CBR traffic, and suddenly changed by the CBR traffic. The main objective of this subsection is to evaluate the stability of our method against addition and disconnection of the CBR connection. For this purpose, we add two CBR connections to the model shown in Fig. 6.1: one of 50 Mbit/s from 100 ms to 150 ms, and the other of 100 Mbit/s from 200 ms to 250 ms. We assume that cells of the CBR connection are processed prior to cells of ABR connections at the switch.

In the following simulation, all source end systems start cell transmission simultaneously at $t$ = 0 ms. The initial cell rate, $ICR$, is set to $BW/4$ to minimize the effect of ABR connection establishment since our objective in this subsection is to investigate the effect of the CBR traffic.

Simulation results of our enhanced max-min scheme are shown in Figs. 6.11 and 6.12 for LAN and WAN environments. Our enhanced max-min scheme controls the queue length by adjusting the bandwidth allocation to ABR connections. As a result, the queue length is rapidly converged to the desired queue length, $Q_T$ (see around $t$ = 100 ms or 200 ms in Fig. 6.11).

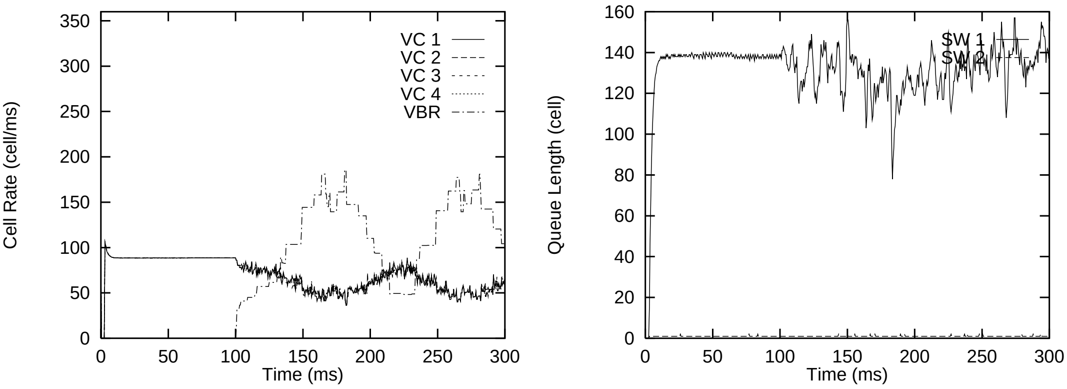

Figure 6.13: Effect of VBR traffic in enhanced max-min scheme for $\tau = 0.01$ ms and $I$ = 0.1 ms.

### 6.3.4 Effect of VBR Traffic

We next evaluate the effect of the VBR traffic on our enhanced max-min scheme. As a typical example of VBR traffic, we use an MPEG-1 encoded video stream of 30 frame/s, 352 × 240 pixels with average rate 4.5 Mbit/s and peak rate 14.84 Mbit/s. It means that up to ten video streams can be multiplexed since we assume each video frame is carried with the CBR service class. In our simulation, ten identical VBR sources but with different starting points are multiplexed and added into the network (see Fig. 6.1) at $t$ = 100 ms. As shown in Subsection 6.3.3, our enhanced max-min scheme works well even when the CBR traffic exists in the network. However, since the amount of the VBR traffic changes frequently, the switch must recompute the bandwidth allocation to ABR connections according to activity of the VBR traffic. As explained in Subsection 6.2.2, our enhanced max-min scheme estimates the available bandwidth by counting the number of CBR/VBR cells in the bandwidth monitoring interval, $I$. The correctness of the bandwidth allocation depends on how accurately the available bandwidth to the ABR service class is estimated. Hence, in this subsection, we investigate an appropriate setting of $I$.

Figures 6.13 and 6.14 show simulation results of our enhanced max-min scheme for $\tau$ = 0.01 and 1.00 ms. In these figures, the bandwidth monitoring interval, $I$, is set to 0.1 ms. It can be found that the bottleneck link (in this case, the outgoing link of SW1) is fully utilized because $ACR$ of the source end system is adaptively changed according to the amount of the VBR traffic. In the WAN environment, the queue length is also kept around $Q_T$. However, it can be noticed that $ACR$s of the source end systems fluctuates in a range of about 20 cell/ms. This is due to a small $I$; the available bandwidth estimation is inaccurate. This oscillation of $ACR$ can be eliminated by increasing $I$ as shown in the next.

In Figs. 6.15 and 6.16, the bandwidth monitoring interval, $I$, is changed from 0.1 ms to 1.0 ms. By comparing these figures with the previous case of $I$ = 0.1 ms (see Figs. 6.13 and 6.14), it can be found that the oscillation of $ACR$ becomes quite small, and that the fluctuation of the queue is almost identical.

The amplitude of the queue oscillation becomes noticeable when $I$ becomes 20 ms as shown in Figs. 6.17 and 6.18. As can be seen from figures, the queue length oscillation is much larger than the cases of $I$ = 0.1 and 1.0 ms. The queue length undesirably reaches zero at $t$ = 200 ms in the LAN environment. This is because the switch fails to estimate the available bandwidth correctly since the long monitoring interval cannot follow sudden decrease of VBR traffic.

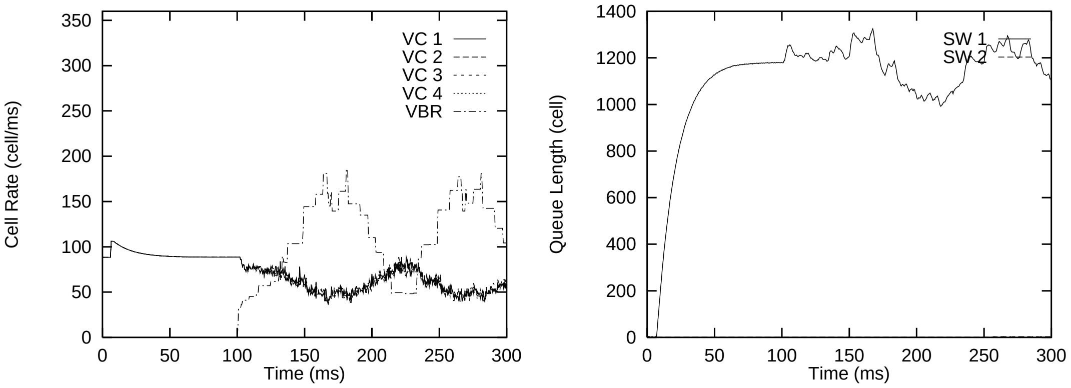

Figure 6.14: Effect of VBR traffic in enhanced max-min scheme for $\tau = 1.00$ ms and $I$ = 0.1 ms.

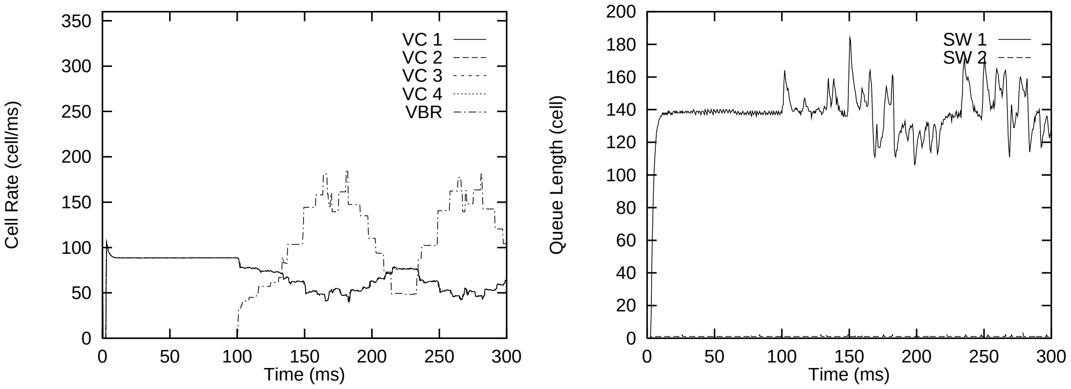

Figure 6.15: Effect of VBR traffic in enhanced max-min scheme for $\tau = 0.01$ ms and $I$ = 1.0 ms.

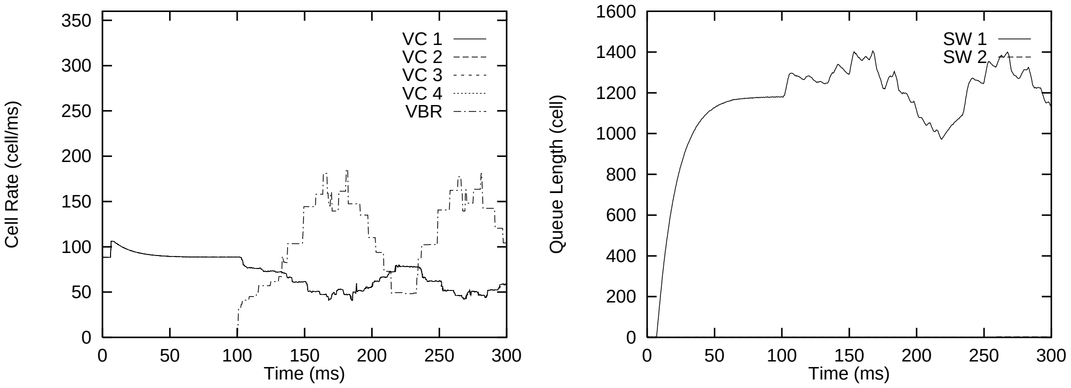

Figure 6.16: Effect of VBR traffic in enhanced max-min scheme for $\tau = 1.00$ ms and $I$ = 1.0 ms.

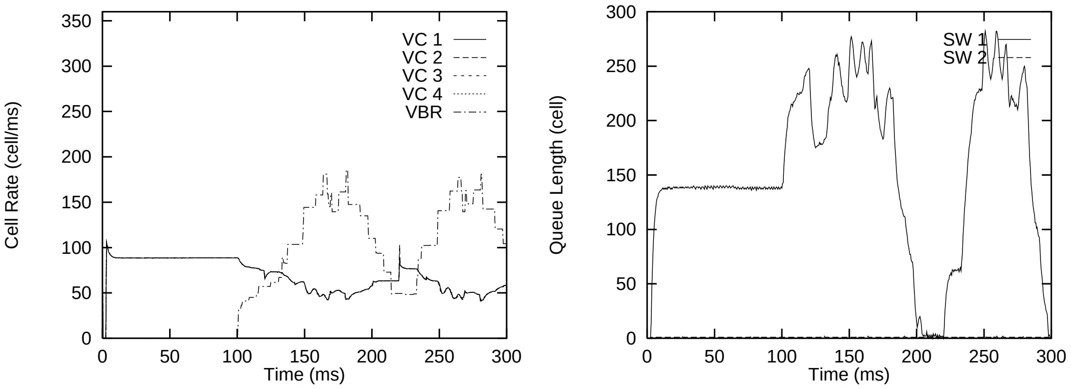

Figure 6.17: Effect of VBR traffic in enhanced max-min scheme for $\tau = 0.01$ ms and $I$ = 20 ms.

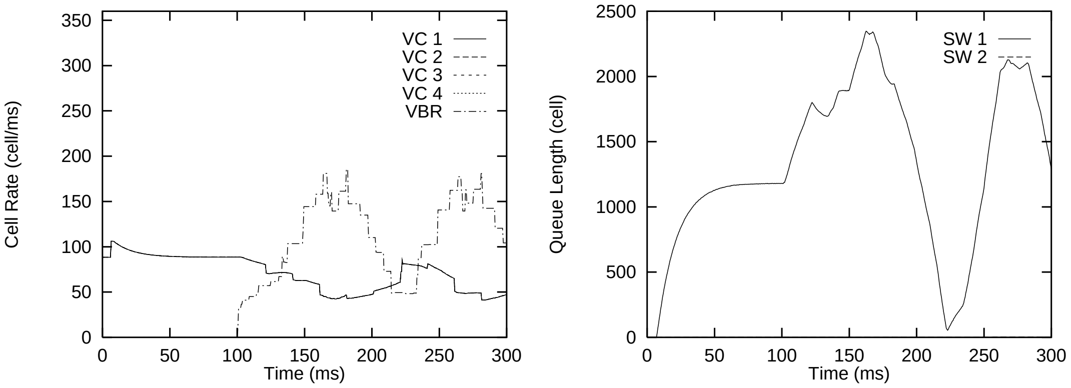

Figure 6.18: Effect of VBR traffic in enhanced max-min scheme for $\tau = 1.00$ ms and $I$ = 20 ms.

From these observations, the bandwidth monitoring interval should be set to around 1.0 ms for achieving stable bandwidth allocation and avoiding unnecessary queue oscillation in the current case. Of course, the adequate monitoring interval must depend on the traffic characteristics, and further study is required.

### 6.3.5 Coexistence with Binary-Mode Switches

In this section, we investigate how the explicit-rate switch is affected by the binary-switches when both switches coexist in the network. A simulation model used in this subsection is shown in Fig. 6.19. The model consists of two switches and four connections with different routes. We first show simulation results when both SW1 and SW2 employ our enhanced max-min scheme. We then change SW1 and SW2 in turn to the binary-mode switch. To evaluate the effect of the binary-mode switch. There are four connections in the network denoted by VC$n$ ($1 \leq n \leq n$). Each connection, VC$n$, starts cell transmission at $t$ = 0, 50, 100 and 150 ms, respectively. We use the values listed in Table 6.2 for control parameters of source end systems. Note that both $RIF$ and $RDF$ are set to 1 in this case. Propagation delays of all links are 0.01 ms (about 2 km). The bandwidth allocation in each period satisfying the max-min fairness criterion is listed in Table 6.5.

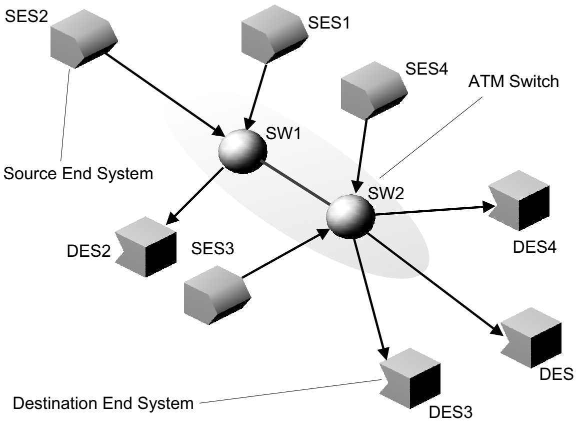

Figure 6.19: Our Simulation Model with Binary-Mode Switch.

Table 6.5: Bandwidth allocation with max-min fairness (in cell / ms).

| | $0 \sim 50$ ms | $50 \sim 100$ ms | $100 \sim 150$ ms | $150 \sim 200$ ms |
|---|---|---|---|---|
| VC1 | 353.7 | 176.8 | 176.8 | 117.9 |
| VC2 | – | 176.8 | 176.8 | 235.8 |
| VC3 | – | – | 176.8 | 117.9 |
| VC4 | – | – | – | 117.9 |

We first show a simulation result when both SW1 and SW2 employ our enhanced max-min scheme in Fig. 6.20. The threshold value, $Q_T$, are chosen as 30 and 5766 cells, and the bandwidth adjustment factors, $\Delta_1$ and $\Delta_2$ are set to 0.2 and 0.5, respectively. By comparing this figure with Table 6.5, it can be found that the max-min fairness is satisfied. It can be also found that the queue length is settled at $Q_T$, and that the queue length never becomes empty.

We next replaces one of SW1 and SW2 with the binary-mode switch. Figures 6.21 and 6.22 show simulation results when the binary-mode switch is placed at SW1 or SW2, respectively. A threshold value of the binary-mode switch, which is used to detect congestion, is set to the half of the buffer capacity (150 Kbyte) [72]. In the figures, neither the bandwidth allocation with max-min fairness nor full link-utilization can be satisfied. It is due to inappropriate settings of $RIF$ and $RDF$ for the binary-mode switch although these settings (in this case, $RIF = 1$ and $RDF = 1$) are ideal for our enhanced max-min scheme.

By setting $RIF$ and $RDF$ appropriately, performance is greatly improved as shown in Figs. 6.23 and 6.23. In these figures, $RIF = 1/256$ and $RDF = 1/64$ are chosen based on our previous work [72]; that is, $RIF$ and $RDF$ are tuned parameters to obtain high performance for binary-mode switches. One can easily find that the queue length of SW1 is stabilized at 150 Kbyte (2,898 cells). In Fig. 6.23, even though $ACR$ of the source end system fluctuates, average of $ACR$ in each period is close to that of the max-min fairness Note that the maximum of $ACR$s of VC1, VC3 and VC4 are restricted to about 130 cell / ms. It is because SW2 of our enhanced max-min scheme allocates 130 cell / ms for these connections. Moreover, it can be found that Fig. 6.23 shows better performance than Fig. 6.24 in terms of the maximum queue length and fairness. The reason can be explained as follows. In our simulation model (Fig. 6.19), the bottleneck switch is SW2. Namely, SW2 is likely to be congested compared with SW1. Since the binary-mode switch needs more time to respond to congestion than our enhanced

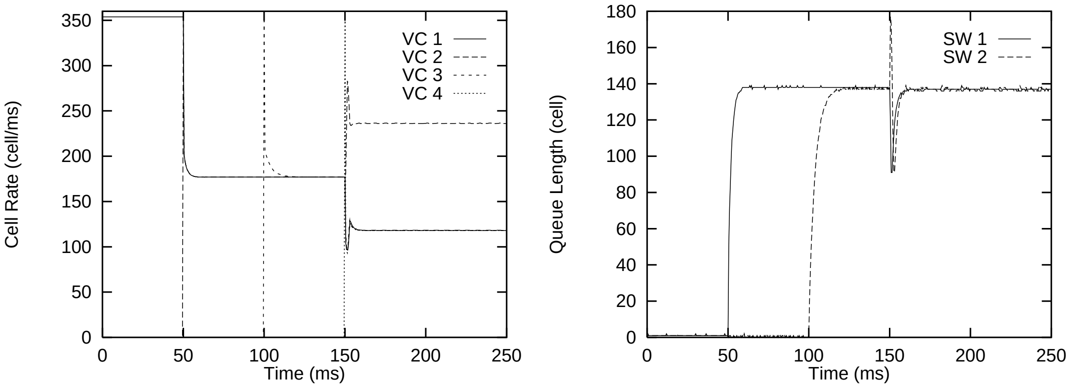

Figure 6.20: Case of enhanced max-min scheme at SW1 and SW2.

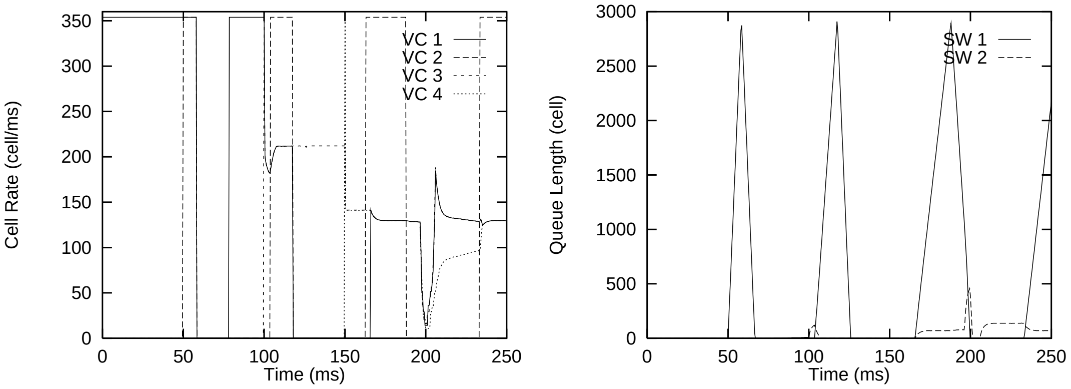

Figure 6.21: Case of binary-mode switch at SW1 and enhanced max-min scheme at SW2.

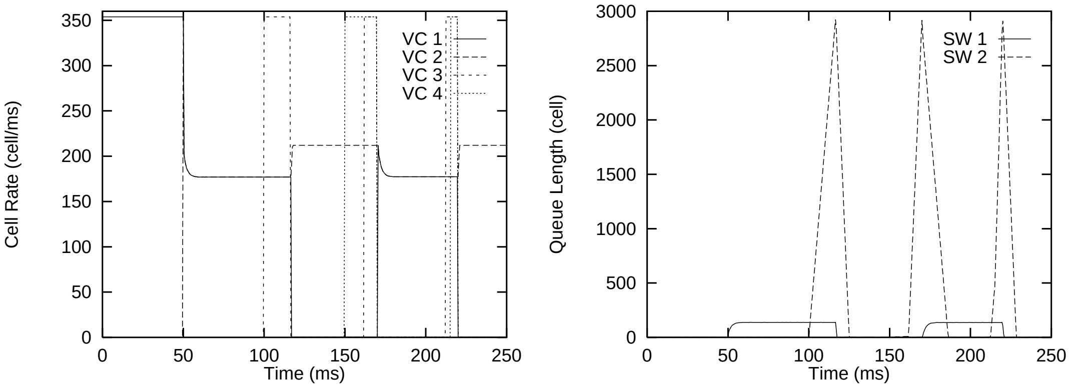

Figure 6.22: Case of enhanced max-min scheme at SW1 and binary-mode switch at SW2.

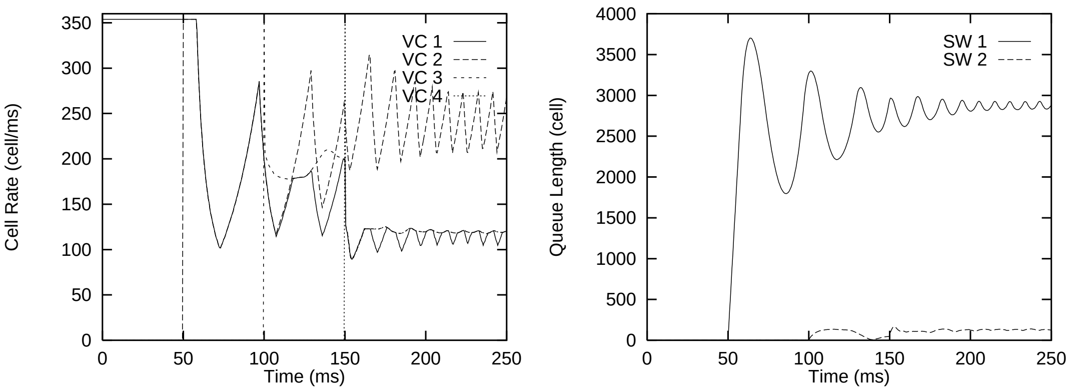

Figure 6.23: Case of binary-mode switch at SW1 and enhanced max-min scheme at SW2 for $RIF = 1/256$ and $RDF = 1/64$.

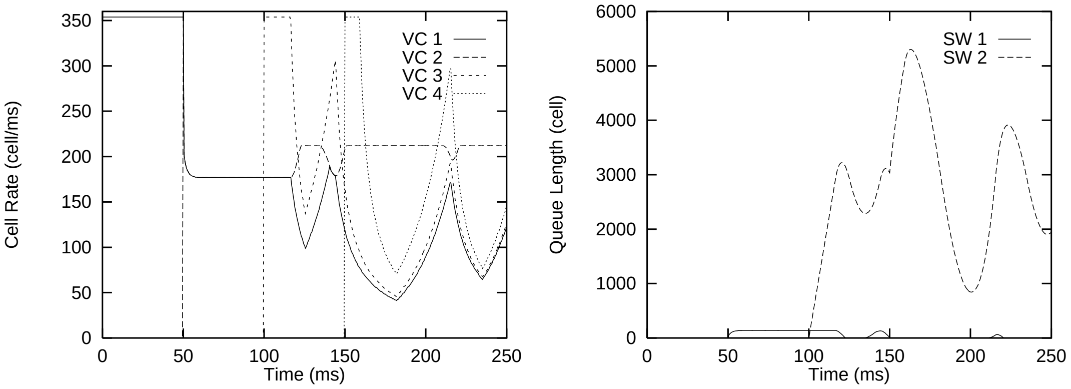

Figure 6.24: Case of enhanced max-min scheme at SW1 and binary-mode switch at SW2 for $RIF = 1/256$ and $RDF = 1/64$.

max-min scheme, replacing SW2 with the binary-mode switch results in a larger queue length. From these results, we find that the control parameters of source end systems should be tuned to the binary-mode switch if it exists, and that our scheme can still work effectively. Or we may say that the network performance is limited by the binary-mode switch, not by our enhanced max-min scheme when binary-mode and explicit-rate switches coexist in the network.

## 6.4 Conclusion

The rate-based congestion control algorithm has been standardized in the ATM Forum as the congestion control mechanism for the ABR service class. Two types of congestion notification mechanisms are specified in the standard: EFCI marking and explicit-rate marking. In this chapter, we have focused on an explicit-rate marking switch, which utilizes the ER value in the RM cell for allocating bandwidth to each connection. We have discussed design goals of an explicit-rate switch in terms of high performance, transient performance, fairness, configuration simplicity, applicability and interoperability. Based on these design goals, we have then proposed our explicit-rate switch algorithm, which is an enhanced version of the max-min scheme. Through simulation experiments, we have evaluated the performance of our switch algorithm in various environments, and have shown that our switch algorithm can achieve better efficiency and stability compared width other switch algorithms.

For future work, we should consider the effect of bursty sources. In this chapter, we have assumed persistent source traffic. However, cell generation from the upper-layer protocol may have bursty nature. We should take account of an explicit-rate switch algorithm, which works efficiently with bursty sources as well as persistent ones.

# Conclusion

The ATM technology has been regarded as the promising technology for realizing a high-speed multimedia network. A lot of researches have been devoted by many researchers and organizations in development and standardization of ATM networks. However, there still remains a number of issues to be solved for success of the ATM technology as the fundamentals for realizing high-speed multimedia networks.

Congestion control as traffic management is one of the most important topics in ATM networks. Without adequate congestion control mechanism, the ATM network cannot satisfy the demands by customers to provide efficient and stable multimedia networks. Moreover, traditional congestion control schemes are usually difficult to apply to ATM networks because the bandwidth–delay product in such a high-speed network tends to be quite large, and because QoS requirements of different applications cannot be satisfied with these schemes.

Recently, there is rapid increase in the number of multimedia applications such as audio and video transmission. These applications can be successfully accommodated into the ATM networks by using service classes such as CBR and VBR service classes. These service classes require the application to declare its traffic parameters and QoS requirements in prior to its connection setup. However, in reality, almost all existing applications cannot predict characteristics of its own traffic patterns precisely. Therefore, many applications will still use best-effort service classes such as ABR and UBR service classes.

In this thesis, we have investigated two promising congestion control schemes for best-effort traffic: an input/output buffered type ATM switch with the back-pressure function and the rate-based congestion control algorithm for the ABR service class. The back-pressure function is a mechanism to prohibit cell transmission from input ports to the corresponding output port to avoid buffer overflow at the output buffer. The rate-based congestion control algorithm regulates cell emission process of source end systems based on feedback information from the network.

In Chapter 2, we have treated the ATM switch with input and output buffers equipped with the back-pressure function. We have analyzed its performance under bursty traffic condition, and have derived the maximum throughput, the packet delay distribution and the approximate packet loss probability under the assumption of infinite switch size. Through numerical examples, we have shown that larger packet length drastically degrades the performance of the switch. However, it is possible to lessen such a performance degradation to some extent by providing a large amount of output buffers. At least, the output buffer size comparable to the average packet length is necessary to gain a sufficient performance.

In Chapter 3, we have evaluated the performance of two switch algorithms of the rate-based congestion control algorithm — EPRCA, which is a basis of the standardization process, and EPRCA++, a more intelligent scheme — by simulation technique. We first compared these schemes for only ABR traffic, and have pointed out a problem that EPRCA++ causes serious queue explosion in WAN environment unless careful parameter setting is applied. As a typical application of VBR traffic, multiplexed MPEG streams were added on the switch to exhibit how VBR traffic influences the performance of these schemes. We have shown the effect of VBR traffic on cell emission rates of ABR connections, the maximum queue length, and the throughput at the switch. It should be emphasized that control parameters of complicated schemes should be set carefully in order to achieve effective and stable operation.

Next, in Chapter 4, we have analyzed the dynamical behavior of the rate-based congestion control algorithm by using a first-order fluid approximation. We have also derived proper values of control parameters — the source end system parameters and the switch parameters — to fulfill two objectives: prevention of buffer overflow at the switch and full link utilization of the bottlenecked link. Our investigation have revealed that proper parameter setting also improves transient performance.

In Chapter 5, we have presented two sorts of analyses. One was the analysis for the model with several groups of connections with different propagation delays in order to exhibit the fairness problem among connections, and the ramp-up time of an additional ABR connection. The other was the derivation of the maximum queue length at the switch buffer affected by an addition of background traffic such as CBR traffic. Through numerical examples, we have shown that a large value of $RIF$ (i.e., fast rate increase) is helpful to shorten the ramp-up time, and that a small value of $C_{RM}$ dramatically reduces the maximum queue length caused by CBR traffic. We have also examined the proper setting of $RIF$ and $RDF$ by simulation experiments. As a simulation model, we have used the parking lot configuration having five interconnected switches and four connections with different numbers of hops. We have compared three schemes for applying our analysis to more generic network configurations. It has been shown that $RDF$ should be set to a small value around 1/64 (i.e., slow rate decrease), and that $RIF$ should be set to a large value as long as cell loss can be prevented.

Finally, in Chapter 6, we have focused on an explicit-rate marking switch, which utilizes the ER value in the RM cell for allocating bandwidth to each connection. We have discussed design goals of an explicit-rate switch in terms of high performance, transient performance, fairness, configuration simplicity, applicability and interoperability. Based on these design goals, we have then proposed our explicit-rate switch algorithm, which is an enhanced version of the max-min scheme. Through simulation experiments, we have evaluated the performance of our switch algorithm in various environments, and have shown that our switch algorithm can achieve better efficiency and stability compared with other switch algorithms.

# Abbreviation List

| | |
|---|---|
| **ABR** | Available Bit Rate |
| **ACR** | Allowed Cell Rate |
| **AIR** | Additive Increase Rate |
| **APRC** | Adaptive Proportional Rate Control |
| **ATM** | Asynchronous Transfer Mode |
| **BES** | Binary Enhanced Switch |
| **CAC** | Call Admission Control |
| **CAPC** | Congestion Avoidance and Proportional Control |
| **CBR** | Constant Bit Rate |
| **CCR** | Current Cell Rate |
| **CDV** | Cell Delay Variation |
| **CDVT** | Cell Delay Variation Tolerance |
| **CI** | Congestion Indication |
| **CLR** | Cell Loss Ratio |
| **CTD** | Cell Transfer Delay |
| **EDS** | Explicit Down Switch |
| **EFCI** | Explicit Forward Congestion Indication |
| **EPD** | Early Packet Discard |
| **EPRCA** | Enhanced Proportional Rate Control |
| **ER** | Explicit Rate |
| **ERICA** | Explicit Rate Indication for Congestion Avoidance |
| **FECN** | Forward Explicit Congestion Notification |
| **FIFO** | First-In-First-Out |
| **HOL** | Head of Line |
| **ICR** | Initial Cell Rate |
| **IP** | Internet Protocol |
| **LAN** | Local Area Network |
| **MBS** | Maximum Burst Size |
| **MCR** | Minimum Cell Rate |
| **MMDA** | Max-Min Scheme with Delayed Adjustment |
| **MPEG** | Motion Picture Expert Group |
| **PCR** | Peak Cell Rate |
| **PDU** | Protocol Data Unit |
| **PGF** | Probability Generation Function |
| **PRCA** | Proportional Rate Control Algorithm |
| **RDF** | Rate Decrease Factor |
| **RIF** | Rate Increase Factor |

| | |
|---|---|
| **RIRO** | Random-In-Random-Out |
| **RM** | Resource Management |
| **SCR** | Sustainable Cell Rate |
| **TBE** | Transient Buffer Exposure |
| **TCP** | Transmission Control Protocol |
| **TUB** | Target Utilization Band |
| **UBR** | Unspecified Bit Rate |
| **UI** | Update Interval |
| **VBR** | Variable Bit Rate |
| **VC** | Virtual Connection |
| **VCI** | Virtual Connection Identifier |
| **VP** | Virtual Path |
| **VPI** | Virtual Path Identifier |
| **WAN** | Wide Area Network |

# Bibliography

[1] David E. McDysan and Darren L. Spohn, *ATM theory and application*. McGraw-Hill, 1994.

[2] T. A. Forum, *ATM User-Network Interface Specification Version 3.0*. PTR Prentice Hall, October 1993.

[3] The ATM Forum Technical Committee, "Traffic management specification version 4.0 (draft version)," *ATM Forum Contribution 95-0013R10*, February 1996.

[4] ITU-T, "B-ISDN recommendation I.363," tech. rep., International Telecommunication Union, October 1993.

[5] ITU-T, "ITU-T recommendation in I.371, traffic control and congestion control in B-ISDN," tech. rep., International Telecommunication Union, 1995.

[6] A. Puri, "Vido coding using the MPEG-2 compression standard," *SPIE*, vol. 2094, pp. 1701–1713, 1994.

[7] ISO, "Generic coding of moving pictures and associated audio," *ISO/IEC/JTC1/SC29/WG11*, March 1993.

[8] "Special issue on `Congestion control in high speed networks'," *IEEE Communications Magazine*, vol. 29, October 1991.

[9] G. Woodruff and R. Ksitpaiboon, "Multimedia traffic management principles for guaranteed ATM network performance," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 437–446, April 1990.

[10] P. Newman, "Traffic management for ATM local area networks," *IEEE Communications Magazine*, vol. 32, no. 8, pp. 44–51, 1994.

[11] A. Gersht and K. J. Lee, "A congestion control framework for ATM networks," *IEEE J. Select. Areas in Commun.*, vol. 9, pp. 1119–1130, September 1991.

[12] J. S. Turner, "Design of a broadcast packet switching network," *IEEE Transactions on Communications*, vol. 36, pp. 734–743, June 1988.

[13] R. Rooholamini and V. Cherkassky, "Finding the right ATM switch for the market," *IEEE Computer*, pp. 17–28, April 1994.

[14] R. Fan, H. Suzuki, K. Yamada, and N. Matsuura, "Expandable ATOM switch architecture (XATOM) for ATM lans," *ICC '94*, 5 1994.

[15] H. T. Kung and A. Chapman, "Credit-based flow control for ATM networks: Credit update protocol, adaptive credit allocation, and statistical multiplexing," *ACM SIGCOMM '94*, vol. 24, pp. 101–114, October 1994.

[16] 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.

[17] 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.

[18] R. Jain, “Congestion control and traffic management in ATM networks: recent advances and a survey,” *ATM Forum Contribution 95-0177*, 1995.

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

[20] P. Newman, “Backward explicit congestion notification for ATM local area netwoks,” *IEEE GLOBECOM '93*, pp. 719–723, December 1993.

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

[22] C. Ikeda and H. Suzuki, “Adaptive congestion control schemes for ATM LANs,” *IEEE INFOCOM '94*, pp. 829–838, June 1994.

[23] A. Berger, F. Bonomi, K. Fendick, and J. Swenson, “Control of multiplexed connections using backward congestion control,” *ATM Forum Contribution 94-064*, January 1994.

[24] B. Makrucki et al., “Closed-loop rate-based traffic management,” *ATM Forum Contribution 94-438*, May 1994.

[25] A. W. Barnhart, “Baseline model for rate control simulations,” *ATM Forum Contribution 94-0399*, May 1994.

[26] J. C. R. Bennett and G. T. D. Jardins, “Failure modes of the baseline rate based congestion control plan,” *ATM Forum Contribution 94-0682*, July 1994.

[27] L. Roberts et al., “Closed-loop rate-based traffic management,” *ATM Forum Contribution 94-0438R1*, June 1994.

[28] J. C. R. Bennett and G. T. D. Jardins, “Comments on the July PRCA rate control baseline,” *ATM Forum Contribution 94-0682*, July 1994.

[29] H. Hsiaw et al., “Closed-loop rate-based traffic management,” *ATM Forum Contribution 94-0438R2*, September 1994.

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

[31] K.-Y. Siu and H.-Y. Tzeng, “Adaptive proportional rate control for ABR service in ATM networks,” Tech. Rep. 94-07-01, UC Irvine, July 1994.

[32] L. Roberts and A. W. Barnhart, “New pseudocode for explicit rate plus EFCI support,” *ATM Forum Contribution 93-0974*, October 1994.

[33] Y. Chang, N. Golmie, L. Benmohamed, and D. Su, “Simulation study of the new rate-based EPRCA traffic management mechanism,” *ATM Forum Contribution 94-0809*, September 1994.

[34] K.-Y. Siu and H.-T. Tzeng, “Adaptive proportional rate control (APRC) with intelligent congestion indication,” *ATM Forum Contribution 94-0888*, September 1994.

[35] K.-Y. Siu and H.-Y. Tzeng, “Limits of performance in rate-based control schemes,” *ATM Forum Contribution 94-1077*, November 1994.

[36] H.-Y. Tzeng and K.-Y. Siu, “Comparison of performance among existing rate control schemes,” *ATM Forum Contribution 94-1078*, November 1994.

[37] R. Jain, S. Kalyanaraman, and R. Viswanathan, “Simulation results: The EPRCA+ scheme,” *ATM Forum Contribution 94-0988*, October 1994.

[38] R. Jain, S. Kalyanaraman, and R. Viswanathan, “The OSU scheme for congestion avoidance using explicit rate indication,” *ATM Forum Contribution 94-0883*, September 1994.

[39] R. Jain, S. Kalyanaraman, and R. Viswanathan, “Current default proposal: Unresolved issues,” *ATM Forum Contribution 94-1175R1*, November 1994.

[40] R. Jain, S. Kalyanaraman, and R. Viswanathan, “Transient performance of EPRCA and EPRCA++,” *ATM Forum Contribution 94-1173*, November–December 1994.

[41] I. Iliadis, “Performance of a packet switch with input and output queueing under unbalanced traffic,” in *Proceedings of IEEE INFOCOM '92*, vol. 2, (Florence, Italy), pp. 743–752 (5D.4), 5 1992.

[42] I. Iliadis, “Head of the line arbitration of packet switches with input and output queueing,” in *Fourth International Conference on Data Communication Systems and their Performance*, (Barcelona, Spain), pp. 85–98, 6 1990.

[43] I. Iliadis, “Synchronous versus asynchronous operation of a packet switch with combined input and output queueing,” *Performance Evaluation*, no. 16, pp. 241–250, 1992.

[44] A. I. Elwalid and I. Widjaja, “Efficient analysis of buffered multistage switching networks under bursty traffic,” *IEEE GLOBECOM '93*, vol. 2, pp. 1072–1078, November–December 1993.

[45] S. Gianatti and A. Pattavina, “Performance analysis of shared-buffered banyan networks under arbitrary traffic patterns,” in *Proceedings of IEEE INFOCOM '93*, vol. 3, pp. 943–952, IEEE Computer Society Press, 3 1993.

[46] R. Fan, H. Suzuki, K. Yamada, and N. Matsuura, “Expandable ATOM switch architecture (XATOM) for ATM LANs,” in *IEEE International Conference on Communications*, vol. 1 of 3, pp. 402–409, May 1994.

[47] R. Jain, S. Kalyanaraman, R. Viswanathan, and R. Goyal, “A sample switch algorithm,” *ATM Forum Contribution 95-0178*, February 1995.

[48] 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,” *IEEE GLOBECOM '95*, pp. 296–303, November 1995.

[49] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, “Analysis of rate-based congestion control methods in ATM networks, Part 2: initial transient state analysis —,” *IEEE GLOBECOM '95*, pp. 1095–1101, November 1995.

[50] M. Ritter, "Network buffer requirement of the rate-based control for ABR services," *IEEE INFOCOM '96*, pp. 1190–1197, March 1996.

[51] H. Ohsaki, N. Wakamiya, M. Murata, and H. Miyahara, "Performance of an ATM LAN switch with back-pressure function," in *Data Communications and their Performance: Proceedings of the 6th IFIP WG 6.3 Conference on Performance of Computer Networks* (S. Fdida and R. O. Onvural, eds.), pp. 99–113, Chapman & Hall, October 1995.

[52] J.-C. Bolot and A. U. Shankar, "Dynamical behavior of rate-based flow control mechanisms," *Computer Communication Review*, vol. 20, pp. 35–49, 4 1990.

[53] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Performance evaluation of rate-based congestion control algorithms in multimedia ATM networks," *IEEE GLOBECOM '95*, pp. 1243–1248, November 1995.

[54] N. Yin, "Analysis of a rate-based traffic management for ABR service," *IEEE GLOBECOM '95*, pp. 1076–1082, November 1995.

[55] A. Lin and C. Fang, "Simulation study of ABR robustness under binary switch modes," *ATM Forum Contribution 95-1019*, August 1995.

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

[57] A. W. Barnhart, "Explicit rate performance evaluations," *ATM Forum Contribution 94-0983R1*, October 1994.

[58] D. H. K. Tsang, W. K. F. Wong, S. M. Jiang, and E. Y. S. Liu, "A fast switch algorithm for ABR traffic to achieve max-min fairness," in *1996 International Zurich Seminar on Digital Communications* (B. Plattner, ed.), pp. 161–172, Springer, February 1996.

[59] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[60] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input vs. output queueing on a space-division packet switch," in *Proceedings of IEEE GLOBECOM '86*, (Houston, Texas), pp. 659–665, 12 1986.

[61] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge University Press, 1988.

[62] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[63] H. Takagi, "Queueing analysis volume 3: Discrete-time systems," *North-Holland*, 1993.

[64] H. Ohsaki, G. Hasegawa, M. Murata, and H. Miyahara, "Parameter tuning of rate-based congestion control algorithms and its application to TCP over ABR," *First Workshop on ATM Traffic Management IFIP WG 6.2*, pp. 383–390, December 1995.

[65] Y. Oie, M. Murata, K. Kubota, and H. Miyahara, "Performance analysis of nonblocking packet switches with input / output buffers," *IEEE Transactions on Communications*, vol. 40, pp. 1294–1297, 8 1992.

[66] L. G. Roberts, “Computation of Xrm and ICR in ABR signaling,” *ATM Forum Contribution 95-0817*, August 1995.

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

[68] G. Hasegawa, H. Ohsaki, M. Murata, and H. Miyahara, “Performance evaluation and parameter tuning of TCP over ABR service in ATM networks,” *IEICE Transactions on Communications*, vol. E79-B, pp. 668–683, May 1996.

[69] G. Hasegawa, H. Ohsaki, M. Murata, and H. Miyahara, “Performance improvement of TCP over ABR service class based on parameter tuning of rate-based congestion control,” *in preparation*, 1996.

[70] H. Ohsaki, M. Murata, and H. Miyahara, “Robustness of rate-based congestion control algorithm for ABR service class in ATM networks,” submitted to *International Journal of Communication Systems*, Jun 1996.

[71] H. Ohsaki, M. Murata, H. Miyahara, C. Ikeda, and H. Suzuki, “Parameter tuning for binary mode switch — analysis,” *ATM Forum Contribution 95-1483*, 1995.

[72] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, “Parameter tuning of rate-based congestion control algorithms for ATM networks,” submitted to *IEEE/ACM Transactions on Networking*, November 1996.

[73] L. Wojnaroski, “Baseline text for traffic management sub-working group,” *ATM Forum Contriubtion 94-0394r5*, October 1994.

[74] C. Ikeda, H. Suzuki, H. Ohsaki, and M. Murata, “Recommendation parameter set for binary switch,” *ATM Forum Contribution 95-1482*, December 1995.

[75] S. Liu, T. Chen, and V. K. Samalam, “Fairness in closed-loop rate-based traffic control schemes,” *ATM Forum Contribution 94-0387*, May 1994.

[76] N. Yin, “Fairness definition in ABR service model,” *ATM Forum Contribution 94-0928R2*, November 1994.

[77] 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,” *IEEE INFOCOM '96*, pp. 1174–1181, March 1996.