UMI
300 North Zeeb Road
Ann Arbor, MI 48103

ABSTRACT

EXPLICIT CONGESTION CONTROL ALGORITHMS FOR AVAILABLE BIT RATE SERVICES IN ASYNCHRONOUS TRANSFER MODE NETWORKS

by
Ambalavanar Arulambalam

Congestion control of available bit rate (ABR) services in asynchronous transfer mode (ATM) networks has been the recent focus of the ATM Forum. The focus of this dissertation is to study the impact of queueing disciplines on ABR service congestion control, and to develop an explicit rate control algorithm.

Two queueing disciplines, namely, First-In-First-Out (FIFO) and per-VC (virtual connection) queueing, are examined. Performance in terms of fairness, throughput, cell loss rate, buffer size and network utilization are benchmarked via extensive simulations. Implementation complexity analysis and trade-offs associated with each queueing implementation are addressed. Contrary to the common belief, our investigation demonstrates that per-VC queueing, which is costlier and more complex, does not necessarily provide any significant improvement over simple FIFO queueing.

A new ATM switch algorithm is proposed to complement the ABR congestion control standard. The algorithm is designed to work with the rate-based congestion control framework recently recommended by the ATM Forum for ABR services. The algorithm's primary merits are fast convergence, high throughput, high link utilization, and small buffer requirements. Mathematical analysis is done to show that the algorithm converges to the max-min fair allocation rates in finite time, and the convergence time is proportional to the distinct number of fair allocations and the round-trip delays in the network. At the steady state, the algorithm operates without causing any oscillations in rates. The algorithm does not require any parameter tuning, and proves to be very robust in a large ATM network.

The impact of ATM switching and ATM layer congestion control on the performance of TCP/IP traffic is studied and the results are presented. The study shows that ATM layer congestion control improves the performance of TCP/IP traffic over ATM, and implementing the proposed switch algorithm drastically reduces the required switch buffer requirements.

In order to validate claims, many benchmark ATM networks are simulated, and the performance of the switch is evaluated in terms of fairness, link utilization, response time, and buffer size requirements. In terms of performance and complexity, the algorithm proposed here offers many advantages over other proposed algorithms in the literature.

EXPLICIT CONGESTION CONTROL ALGORITHMS FOR AVAILABLE BIT RATE SERVICES IN ASYNCHRONOUS TRANSFER MODE NETWORKS

by
Ambalavanar Arulambalam

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

May 1996

APPROVAL PAGE

EXPLICIT CONGESTION CONTROL ALGORITHMS FOR AVAILABLE BIT RATE SERVICES IN ASYNCHROUNOUS TRANSFER MODE NETWORKS

Ambalavanar Arulambalam

Dr. Nirwan Ansari, Dissertation Advisor — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Xiaoqiang Chen, Co-Advisor — Date
Member of Technical Staff, Bell Laboratories, Lucent Technologies,
Holmdel, NJ

Dr. Ali Akansu, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Zoran Siveski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Dennis Karvelas, Committee Member — Date
Assistant Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Ambalavanar Arulambalam

Degree: Doctor of Philosophy

Date: May, 1996

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1996
- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1993
- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1992

Major: Electrical Engineering

Presentations and Publications:

A. Arulambalam, X. Chen and N. Ansari, "A Fast Max-Min Fair Rate Allocation Algorithm for Available Bit Rate Services in ATM Networks," submitted to *IEEE/ACM Transactions on Networking.*

A. Arulambalam, X. Chen and N. Ansari, "A Fast Explicit Rate Congestion Control Algorithm for Available Bit Rate Services in ATM Networks," submitted to *GLOBECOM 96*, London, England.

N. Ansari, A. Arulambalam and S. Balasekar, "Traffic Management of a Satellite Communication Network Using Stochastic Optimization," *IEEE Transactions on Neural Networks*, May 1996.

A. Arulambalam, X. Chen and N. Ansari, "Impact of Queueing Disciplines on Available Bit Rate Congestion Control in ATM Networks," presented at the *30th Annual Conference on Information Sciences and Systems*, Princeton, NJ, March 20 - 22, 1996.

A. Arulambalam and N. Ansari, "Traffic Management of a Satellite Communication Network Using Mean Field Annealing," presented at the *International Conference on Neural Networks*, Orlando, Florida, June 28 - July 2, 1994.

This dissertation is dedicated
to my family

ACKNOWLEDGMENT

I wish to express my sincere gratitude to my supervisor, Dr. Nirwan Ansari, for his guidance, friendship, and moral support throughout this research. I also thank my advisor for giving me a free hand with the research and for providing continuous encouragement throughout my career at NJIT.

I acknowledge Dr. Xiaoqiang Chen of Lucent Technologies, for introducing me to the area of ABR congestion control and giving numerous technical suggestions for this research. I have been influenced by his style of research and emphasis on bridging the gap between theory and practice, and has benefited from his detailed comments on every aspect of this dissertation.

I extend my gratitude to Drs. Ali Akansu, Zoran Siveski and Dennis Karvelas for serving as members of the dissertation committee and for their comments.

Lisa Fitton provided timely help and suggestions during the editing of the dissertation.

I deeply appreciate the eternal love and continuous encouragement from my parents and my sister. I also thank my uncle T. Sivendran, M.D. and his family for helping me in countless ways throughout my college career.

Last but not least, I wish to acknowledge my present and former colleagues at the Center for Communications and Signal Processing Research at NJIT, for their help, advice, and friendship.

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

TABLE OF CONTENTS
(continued)

Chapter Page

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(continued)

LIST OF FIGURES
(continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Over the years, our society has become more and more dependent on fast and timely delivery of information. As a result of dramatic developments in high speed VLSI and information processing technologies, many services using multi-media – including voice, video and data – have rapidly started to emerge to satisfy the needs of our society. To support this wide variety of services, telecommunications carriers have traditionally operated a number of networks with specialized structures optimized toward the services they support. Traditionally, these services have been carried via separate networks – voice on the telephone network; data on computer networks or local area networks; video teleconferencing on private, corporate networks; and television on broadcast radio or cable networks [1]. These networks are largely engineered for a specific application and are not suitable for other applications. Customers with access to multiple services needed to be equipped with multiple access methods, multiple hardware requirements, and various access protocols. It is often desirable to have a single network to provide all these communication services in order to achieve the economy of sharing. Integration prevents the need for many overlaying networks, which complicates network management and reduces the flexibility in the introduction and evolution of services.

The recent growth of digital technology has meant that many components of non-data networks have been migrating towards digital technologies for economic and quality reasons. As this trend continues, more components of the separate logical networks can be physically shared within the carrier's infrastructure. With this in mind the International Telecommunications Union (ITU) has provided a new set of recommendations that overcome the limitations of conventional networks. This new, integrated digital network, which supports a wide range of bandwidth and delay requirements, is called the Broadband Integrated Services Digital Network (B-ISDN) [2].

The primary goal of the B-ISDN is to provide users with a universal platform to integrate all kinds of multi-media services. Various services required by business and the scientific community have become the driving force behind the realization of B-ISDN in the near future. Many users require sophisticated features and wider transmission bandwidth, a situation resulting from the following developments:

- The increasing number of LANs installed and the need to interconnect them;
- The requirement for high bandwidth links due to the increase in processor speed, software sophistication and file size, particularly the growth in graphic-based systems;
- The requirement to support systems that employ distributed computing and to support systems that are distributed in many locations;
- The increasing strategic value of information for the business community and the recognition of improved telecommunication services as a critical business advantage.

In order to realize B-ISDN, the ITU has recommended Asynchronous Transfer Mode (ATM) technology as the transport mechanism for B-ISDN services. ATM technology is based on the statistical multiplexing of short fixed-size (53 Bytes) cells. In recent years ATM technology has quickly become the most talked about subject in the communications and computer industry. ATM technology employs the concept of cell switching, which combines the benefits of traditional packet switching, used widely in data networks, and circuit switching, used widely in voice networks [3].

1.1 The ABR Service Category

As an established principle, ATM provides the flexibility of integrating the transport of different classes of traffic with a wide range of service requirements. ATM also provides the potential of obtaining resource efficiency through the statistical multiplexing of a diverse mix of traffic streams. Gains due to statistical multiplexing, however, come at the risk of potential congestion. Flow and congestion control in ATM networks is concerned with ensuring that users get their desired quality of service, and it can be broadly separated into two categories: open-loop control and closed-loop control. The objective of open-loop control is to ensure *a priori* that the network traffic intensity normally will never reach a level to cause unacceptable performance degradation. If a source is able to characterize in detail its traffic characteristics and performance requirements, the network can then take these values into account in the call acceptance decision (e.g., Call Admission Control). In addition, mechanisms such as policing units need to be provided at the entrance of the network to ensure that the source is consistent with the pre-specified traffic parameters. Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services fall into this category. On the other hand, most data applications cannot predict their own traffic parameters, and usually require a service that dynamically shares the available bandwidth. Such a service is referred to as Available Bit Rate (ABR).

The ABR service, aimed at the economical support of applications with vague requirements for throughputs and delays, will systematically and dynamically allocate available bandwidth to users by controlling the rate of offered traffic through feedback. A user might know, for example, that a particular application runs well across a lightly loaded 10-Mbps Ethernet and poorly through a 9.6 Kbps modem. That same user, however, could have difficulty selecting a single number to serve as both a guarantee and a bound on the bandwidth for the application, as would be required to set up a CBR connection across an ATM network. Trial and error would help to narrow the range of viable bandwidths for the application, but typically not to a single number. The ABR service is suited for such an application.

While the ABR service is potentially useful for a wide variety of applications, the main motivation for its development was the economical support of data traffic, where each packet of data is segmented into ATM cells, the loss of any one of which causes the re-transmission of the entire packet by a higher protocol layer. After studies [4] showed that the indiscriminate dropping of ATM cells under congestion could lead to the collapse of throughput for packet data applications, the goals of the ABR service were expanded to include the support of sharply defined objectives for cell loss. As a result, the class of control mechanisms considered for the ABR service was restricted to those that, based on feedback from the network to the traffic source, could tightly control cell loss within the network. The ABR service would guarantee a particular cell-loss ratio for all traffic offered in proper response to network feedback. To maximize the odds that the vague requirements of an ABR connection were met by a network's available bandwidth, the class of control mechanism considered for the ABR service was further restricted to those that can use the available bandwidth efficiently and allocate it systematically among the connections active at a given time.

Vague requirements for throughputs and delays are requirements nonetheless and are best expressed as ranges of acceptable values. An additional goal of the ABR service was to allow a user to specify, at the time of connection setup, a lower and upper bound on the bandwidth allotted to the connection over its life. The value of allowing users to do so will increase over time as network bandwidths grow and the throughputs required by different types of applications become farther and farther apart.

A final goal of the ATM protocol was to support connections across Local Area Networks (LANs), Metropolitan Area Networks (MANs), or Wide Area Networks

(WANs). The physical link that spans the ATM User Network Interface (UNI), e.g., between a terminal adapter and a switch, may itself extend to arbitrary distances. The ability to work in a variety of environments is particularly important when traffic is controlled using feedback, as with the ABR service, for then the sizes of queue fluctuations depend not so much on the absolute size of network distances as on the amount of data that a connection can transmit during the time it takes feedback to reach the traffic source. Hence, the performance issues experienced by an ABR service for the WAN of today will arise for the MAN of tomorrow that has one tenth the propagation delay of today's WAN but ten times the link bandwidth.

1.2 Traffic Management of ABR Services

The success of ABR services depend on how the traffic in the network is managed, and one of the challenges is how to react in the event of network congestion [5]. Because of its bursty nature, congestion control for ABR service poses more challenging problems than other services, and it has been the focus of recent standardization efforts at the ATM Forum. The ATM Forum has documented many proposals and studies regarding this issue [6]–[16]. After considerable debate, the ATM Forum has adopted a rate-based congestion control algorithm, which is based on the closed-loop feedback flow control principle, to control congestion for ABR traffic in ATM networks [17]. The approach chosen by the ATM Forum's Traffic Management Working Group as the best match for the goals of the ABR service is to control the bandwidth of connections directly. Since each ATM cell contains the same number of bits, control of a connection's bandwidth, measured as a bit rate, is achieved by directly controlling its cell rate, hence the term rate-based flow control. Control of the cell rate for a connection would occur at least at the source Terminal Adapter (TA), which would shape the connection's traffic as directed by feedback from the network. As an option it might occur at points within the network as well.

Under a rate-based framework, the share of bandwidth allocated to a connection does not depend on the delays between points where data is shaped on a per-connection basis, so that a rate-based framework is ideal for architectural flexibility. In addition, a rate-based framework can support fair (even) allocations of bandwidth, as well as bandwidth guarantees, even when the simple First-In-First-Out (FIFO) discipline is used at network queues [18]. The degree of architectural flexibility allowed by a rate-based framework for the ABR service distinguishes it from other approaches to flow control for high-speed networks [19], [20].

As an enhancement, the scheme allows the intermediate switches to specify an explicit rate of transmissions for VCs. The challenge is to calculate the explicit rate, which is a fair share of available bandwidth, in a distributed and asynchronous manner. Many such switch algorithms, with varying degrees of complexity and performance characteristics, have been proposed. The focus of this dissertation is to evaluate such fair allocation algorithms, and propose a new, fast, and adaptive-rate allocation algorithm suitable for a large ATM network, while keeping the complexity low and achieving a high level of performance.

1.3 Outline of the Dissertation

A complete description of the ABR congestion control framework is presented in Chapter 2, which describes the basic elements of congestion control. The problem of congestion in networks and the issues relating to congestion control are addressed in Chapter 2. The behavior of source, destination, and the possible switch mechanisms are summarized. It is necessary to understand the fairness criterion under which we develop the fair-rate allocation algorithm. The mathematical model for the fairness criterion is presented in Chapter 3, which also includes a small survey of existing fair-rate algorithms. Chapter 4 was written to accomplish three goals. The first was to discuss the different queueing strategies possible to complement ABR traffic control. The second was to provide a good understanding of queueing disciplines versus different ABR implementations. The third goal was to evaluate complexity and performance trade-offs among the possible queueing disciplines. The proposed new fair-rate allocation algorithm is presented in Chapter 5, which presents the basic elements, the key features, and the implementational details of the algorithm. The proof of convergence of the algorithm is also provided. Extensive simulation study is done in order to verify the correctness of the algorithm. Many benchmark simulation scenarios are simulated and the results are presented in Chapter 6. The simulation results show that our algorithm functions effectively, and offers many advantages over other proposed algorithms. In Chapter 7, the algorithm presented in Chapter 5 is enhanced to provide support for minimum cell-rate requirements of the ABR service category. The possible fairness criterions, which take into account minimum cell-rate requirements, are outlined. Simulation results are provided to illustrate the enhancement. In Chapter 8, the impact of ATM switching and the ATM layer congestion control on the performance of TCP/IP traffic is addressed. The goal is to address the issues of switch buffer requirements and their effects on packet re-transmissions. A simple network topology is simulated to illustrate the performance. The summary of this dissertation, conclusions drawn from this work, and the possible future directions are presented in Chapter 9.

CHAPTER 2

THE CONGESTION CONTROL FRAMEWORK FOR ABR SERVICES

ATM is a networking protocol with the potential of supporting applications with distinct tolerances for delay, jitter, and cell loss, and distinct requirements for bandwidth or throughput. To address this spectrum of needs, the ATM Forum has defined a family of service categories. Due to the wide range of traffic characteristics it is necessary to employ a traffic management scheme that satisfies the broad diversity of quality of service required by different applications, while making efficient use of network resources. While ATM promises many advantages over traditional networks, it suffers from the potential risk of congestion, and this is one of the major problems a traffic management scheme should deal with. Congestion is generally defined as the condition reached when the demand for resources exceeds the available resources, over a certain time interval. More specific to ATM, congestion is defined as the condition where the offered load (demand) from the user to the network approaches or exceeds the Quality of Services (QoS) specified in the traffic contract. In other words, congestion is a state where the network elements are not able to provide the negotiated network performance objectives for the already established connections [21].

In ATM networks, the resources that can become congested include switch ports, buffers, transmission links, ATM Adaptation Layer (AAL) processors, and Call Admission Control (CAC) processors [22]. The network elements such as link bandwidth and buffer space act as a bottleneck (the resource where the demand exceeds the capacity). For example, when large bursts of cells arrive at a switch port in an ATM network, the switch may not have enough resources (i.e., buffer space) to hold them in a queue and multiplex them. This condition will lead to cell dropping and the degradation of quality of service performance. Congestion control in ATM networks is difficult because of the large number of combinations of application characteristics, network characteristics, and levels of congestion detection and reaction. One congestion control scheme that works very well for certain applications and network characteristics at a certain level may work poorly for different characteristics, and/or different levels. For these reasons, congestion control in broadband networks has been the subject of intensive research.

This chapter provides insight into ABR congestion control in general. Section 2.1 provides the selection criteria for an effective congestion control scheme for ABR services. A brief history on the selection process of ABR control at the ATM Forum is provided in section 2.2. A detailed description of the rate-based congestion control approach, which was recently recommended by the ATM Forum as the ABR congestion control approach, is presented in section 2.3. We conclude this chapter by providing a brief summary in section 2.4.

2.1 Selection Criteria

ATM congestion control refers to the set of actions taken by the network to minimize the intensity, spread and duration of congestion. The primary role of traffic control and congestion control parameters and procedures is to protect the network and the user in order to achieve network performance objectives while optimizing the use of network resources [21]. In general a congestion control scheme must exhibit the following properties [23].

1. Efficiency: There are two aspects of efficiency we must consider. First of all, the amount of overhead that the control scheme imposes on the network should be very minimal. Sometimes the control cells used by the control scheme may congest the network further. Second, we do not want the network resources to be under-utilized because of the control scheme. Inefficient control schemes may throttle down sources even when there is no danger of congestion, leading to under-utilization of resources.

Therefore, it is necessary that our control scheme fully utilizes the network resources without causing additional overhead to the network.

2. Stability: There is a time delay between the detection of the congestion and the receipt of the congestion signal by the source. The control mechanism should receive this congestion information and try to bring the network to an uncongested state. It is important that the control mechanism converge to a steady state with a small amount of oscillations in few round trip times; otherwise, the network may operate under unstable conditions.

3. Robustness: We need a control scheme that will operate well with the changes in traffic patterns, and should be robust under minor parameter changes. It should be able to work under circumstances where control information is lost. A scheme may work very well under certain conditions and with certain traffic characteristics, but if there is a sudden change in traffic flow the scheme may not operate as desired. It is imperative that the control scheme be stable when the parameters are mistuned or presence of cell loss in the network.

4. Simplicity: In order for a scheme to be implementable with a low cost, the scheme must be simple. Simplicity is very important in high speed networks such as ATM, because – due to high switching and transfer rates – we do not want the control schemes processing many parameters and making a decision. If the control scheme is very simple then the processing of control information would be fast and will not delay any traffic flow. Moreover, the hardware requirement, such as the buffer space required to hold the incoming cells should be minimal. Larger buffers are not only expensive, but also increases delays in the network. Simple protocols are very easy to implement, and are easily accepted as international standards.

5. Scalability: It is natural for network systems to grow in time, and we have seen an explosive growth in network sizes in the last decade. Furthermore, the introduction of optical fibers has tremendously increased the bandwidth. These developments make it imperative that any proposed congestion control scheme scale well with the growth of network-size (i.e., distance, speed, number of users, number of nodes) and with the increase in available bandwidth.

6. Fairness: During congestion it is necessary for some sources to reduce their transmission rate, in order to control congestion. The congestion control scheme must make a choice on which source to throttle (either by requesting it to do so, or by dropping its cells), and should determine how to allocate network resources in a fair manner. One control should not discriminate against a set of connections, and no connections should be arbitrarily favored. For example, if an excessive demand by some source A causes the throttling of another source B, then clearly the network is treating B unfairly. We need a congestion control scheme which treats the sources fairly.

2.2 Development of ABR Control

The ATM Forum has received many congestion control schemes over the last two years. Many proposals, such as France Telecom's Fast Resource Management Scheme [24], Fujitsu's Delay-Based Rate Control [7], Backward Explicit Congestion Notification (BECN) Scheme [25],[26], and Sun Microsystem's Early Packet Discard method [27], have been rejected very early since they failed to meet the adequate requirements of congestion control selection criteria.

For the past few years, the congestion control research for ABR services has been centered around two basic schemes: credit-based and rate-based congestion control. Several approaches, which use a closed-loop feedback control mechanism that allows the network to control the cell transmission rate at each source, were proposed to the ATM Forum [17],[19],[20]. The benefits of a closed-loop control that dynamically regulates the flow of traffic entering a network based on congestion information have been demonstrated in many networks such as TCP/IP [28], DECnet [29], and

Frame Relay [30]. After a considerable debate, the ATM Forum has adopted a rate-based congestion control approach, without making commitments to any particular switch algorithm. The endorsement for the rate-based approach came about because of the architectural flexibility it provided, which includes a wide range of possible switch implementations with varying degrees of cost and complexity.

2.3 The Rate-Based Congestion Control Framework

In the ABR service, the source adapts its rate to changing network conditions. Information about the state of the network, such as bandwidth availability, state of congestion, and impending congestion, is conveyed to the source through special probe cells called Resource Management Cells (RM-cells). The scheme is based on a closed-loop, "positive feedback" rate control principle [31]. Here, the source only increases its sending rate for a connection when given an explicit positive indication to do so, and in absence of such a positive indication, continually decreases its sending rate. The decision to "remove an opportunity for a rate increase" is made independently by each intermediate network based on the state of the resources it is protecting. The decision may be based on a cell queue depth in a switch or a threshold on an aggregate rate of cells flowing on a link. This allows considerable freedom in network equipment design, and allows the network provider to trade off cell buffer memory and link bandwidth utilization.

2.3.1 Functional Elements of ABR Control

Figure 2.1 illustrates the elements of a typical communication network implementing the ABR closed-loop (or feedback) framework. These elements are defined below.

1. Source and Destination: The source and destination generate and receive the ATM cells transported through the network. They typically reside in the terminal adapters, or network interface cards, at the extreme points of an ATM virtual

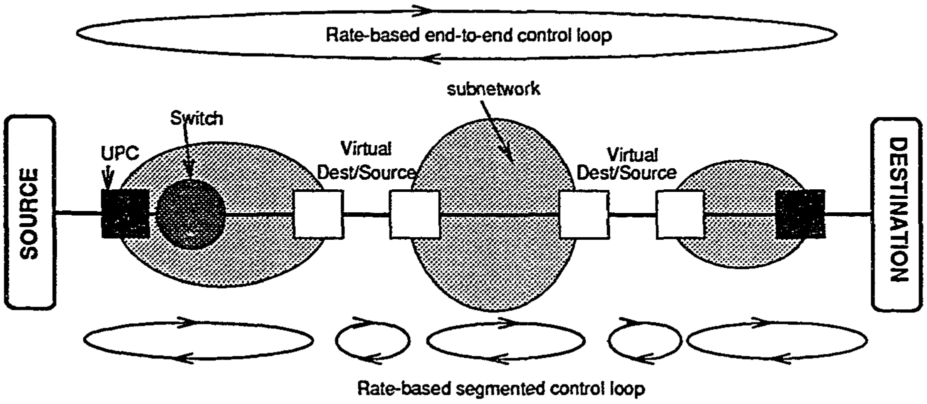

Figure 2.1 Rate-Based End-to-End Congestion Control Scheme

connection. The virtual connection is routed through the network and includes a forward (from source to destination) and a backward (from destination to source) path. For both bi-directional point-to-point and point-to-multipoint connections, the forward and backward components of a virtual connection use the same connection identifiers, and pass through identical transmission facilities. A distinguishing feature of a source for the ABR service is its ability to submit cells into the network at a variable but controlled or shaped rate. An ABR source and destination also must form the two ends of the ABR control loop: the ABR source transmits cells for conveying feedback information toward the destination and the destination returns them toward the source.

2. Network Switch: Switching elements provide the necessary resources for storing and forwarding ATM cells from sources to destinations, namely port bandwidth and buffers. These are limited resources, the contention for which may lead to congestion in the form of loss or the excessive delay of cells. An ABR switching element must monitor the use of its resources to provide proper feedback to the source.

3. Feedback Mechanisms: Feedback from network switches to end systems gives users the information necessary to respond, by appropriately modifying their submission rates, to changes in the available bandwidth, so that congestion is

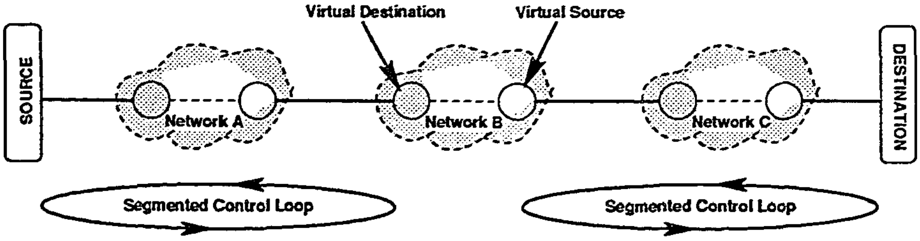

Figure 2.2 Rate-Based Control: Virtual Source/Destination

controlled or avoided and the available bandwidth is used. The use of feedback by the ABR service to control the source rate is one form of closed-loop flow control.

4. Virtual Source/Destination (Optional): While at least one control loop between source and destination end systems is required, segmentation of the control loop can be optionally implemented by way of virtual sources and destinations, as shown in Figure 2.2. An intermediate switching element can close the control loop and initiate a new control loop by functionally behaving as a destination and as a new source. The main motivations for a virtual source/destination are to reduce the length of individual control loops and to create separate control domains for administration.

5. Usage Parameter Control (UPC) (Optional): User parameter control, or policing (at the UNI), of traffic submitted by an end system is an essential requirement for public networks supporting multiple services. Service providers typically must support lower bounds on the bandwidth provided, as well as QoS objectives applying to delays and cell loss. The need for mechanisms that protect users of the ABR service from misbehaving users of the same service and limit how services affect each other is satisfied via the joint action of policing and of scheduling and buffer management at the switch ports. Policing may not be needed in some private-network environments, although efficient mechanisms to ensure fair access to resources are still important.

One additional element, which is not shown in Figure 2.1, is needed for ABR point-to-multipoint connections: branch point (for point-to-multipoint connections only). The role of a branch point for an ABR point-to-multipoint tree is to replicate cells traveling from the root to the leaves and to consolidate feedback traveling from the leaves to the root at points where branches of the tree intersect. The ABR branch point must assure that the flow of feedback transmitted onto each branch conforms to the expected behavior for a point-to-point connection. By doing so, a branch point makes it possible for sources destinations, virtual sources, virtual destinations, and switches to behave the same way for point-to-multipoint connections as for point-to-point connections.

2.3.2 Basic Mechanism

The precise definitions of the source, the destination, and the switch behavior are presented in the ATM Forum Traffic Management Specifications 4.0. In this section we give a brief overview of the framework and a detailed description of the source, the destination and the switch behavior.

Prior to transmission, the source creates a connection with a call setup request. During this call setup, the values for a set of ABR-specific parameters are identified. These parameters are shown in Table 2.1. Some values are requested by the source and possibly modified by the network (e.g., the lower and upper bounds on the source rate) while others are directly chosen by the network (e.g., the parameters characterizing the process for dynamically updating rates). A partial list of parameters negotiated or provided by the network is tabulated in Table 2.1. For a more complete listing, please refer to [32].

The typical operation of the rate-based control framework is illustrated in Figure 2.3. Once the source has received permission, it begins cell transmission. The rate at which an ABR source is allowed to schedule cells for transmission is

Table 2.1 ABR Control Framework Parameters

Parameter	Description	Comments
PCR	Peak Cell Rate	Maximum allowed rate
MCR	Minimum Cell Rate	Minimum rate guaranteed by the network
ICR	Initial Cell Rate	Start-up rate after source being idle
AIR	Additive Increase Rate	Amount of rate increase permitted
N_{RM}	Data cells per RM cell	Provided by the network
RDF	Rate Decrease Factor	Used when CI bit is set
ACR	Allowed Cell Rate	Used to control source's transmission rate

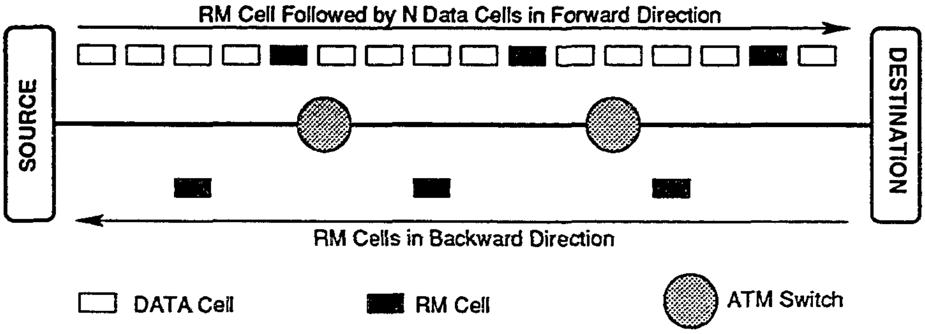
RM Cell Followed by N Data Cells in Forward Direction
SOURCE
DESTINATION
RM Cells in Backward Direction
DATA Cell
RM Cell
ATM Switch

Figure 2.3 Rate-Based End-to-End Congestion Control Scheme

Table 2.2 Fields in RM Cell and Their Sizes

Field	Description	Size in bits
Header	ATM Header	40
ID	Protocol ID	8
DIR	Direction of RM cell	1
CI	Congestion Indication	1
BN	BECN cell (switch generated)	1
CCR	Current Cell Rate	16
MCR	Minimum Cell Rate	16
ER	Explicit Cell Rate	16
$CRC - 10$	Cyclic Redundancy Check	10

denoted as Allowed Cell Rate (ACR). The ACR is initially set to the Initial Cell Rate (ICR) and is always bounded between the Minimum Cell Rate (MCR) and the Peak Cell Rate (PCR). Transmission of data cells is preceded by the sending of an ABR Resource Management (RM) cell. The basic structure of an RM cell is shown in Figure 2.4. Table 2.2 describes the RM-cell fields and their sizes. The source will continue to send RM cells, typically after every N_{RM} data cells. The source rate is controlled by the return of these RM cells, which are looped back by the destination or by a virtual destination.

The source places the rate at which it is allowed to transmit cells (its ACR) in the Current Cell Rate (CCR) field of the RM cell, and the rate at which it wishes to transmit cells (usually the PCR) in the Explicit Rate (ER) field. The RM cell travels forward through the network, thus providing the switches in its path with the information in its content for switches' use in determining the allocation of bandwidth among ABR connections. Switches may also decide at this time to reduce the value of the explicit rate field ER, or set the Congestion Indication (CI) bit to 1. Switches supporting only the Explicit Forward Congestion Indication ($EFCI$) mechanism (by which an indicator in the header of each data cell is set under congestion) will ignore the content of the RM cell. Switches optionally may generate a controlled number of ABR RM cells on the backward path, in addition to those originally supplied by

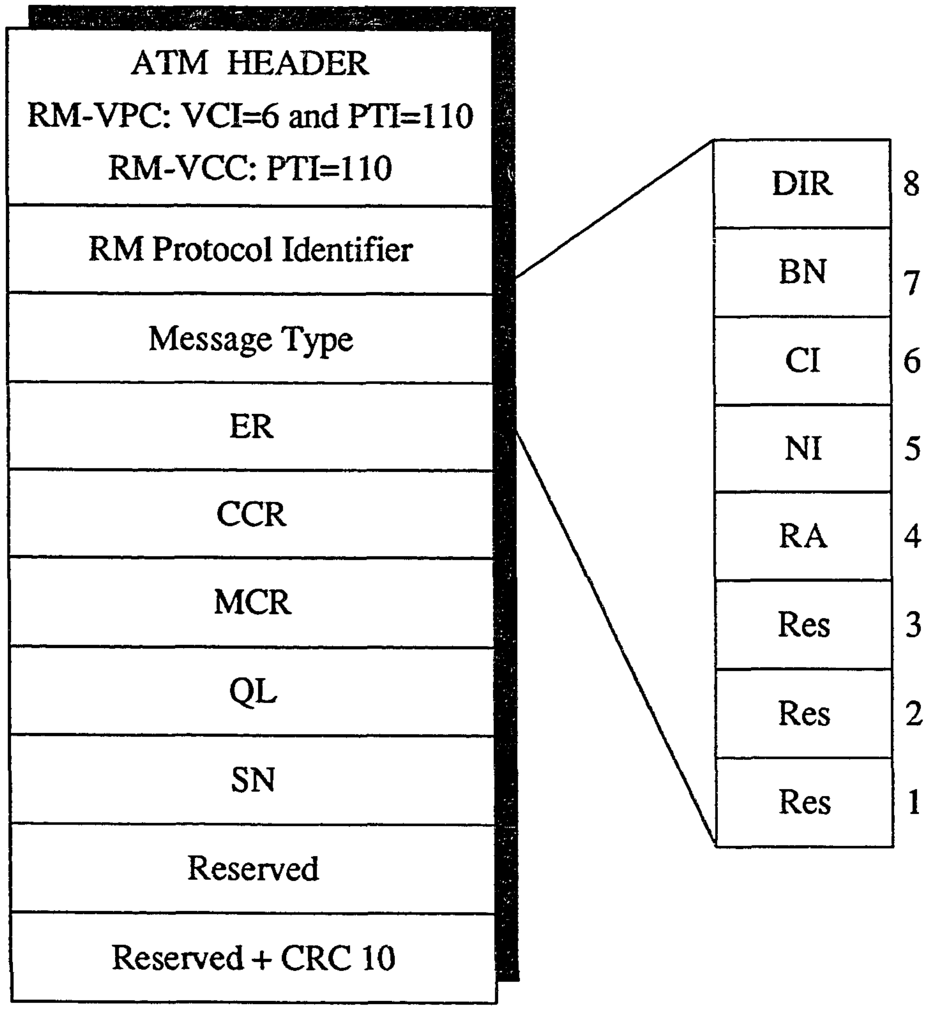

Figure 2.4 RM Cell Structure

the source. Switch-generated RM cells must have the Backward Notification (BN) bit set to 1 and either the CI bit or the No Increase (NI) bit set to 1.

When the cell arrives at the destination, the destination should change the direction bit in the RM cell and return the RM cell to the source. If the destination is congested and cannot support the rate in the ER field, the destination should then reduce ER to whatever rate it can support. If, when returning an RM cell, the destination observed a set $EFCI$ since the last RM cell was returned, then it should set the RM cell's CI bit to indicate congestion.

As the RM cell travels backward through the network, each switch may examine the cell and determine if it can support the ER for this connection. If the ER is too high, the switch should reduce it to the rate that it can support. No switch should increase the ER, since information from switches previously encountered by the RM cell would then be lost. The switches should try to modify the ER for only those connections for which there is a bottleneck, since this promotes a fair allocation of bandwidth.

When the RM cell arrives back at the source, the source should reset its ACR, based on the information carried by the RM cell. If the congestion indication bit is not set ($CI = 0$), then the source may increase its ACR by a fixed increment determined at call setup, toward (or up to) the ER value returned, but never exceeding the PCR. If the congestion indication bit is set ($CI = 1$), then the source must decrease its ACR by an amount greater than or equal to a proportion of its current ACR, the size of which is also determined at call setup. If the ACR is still greater than the returned ER, the source must further decrease its ACR to the returned ER, although never below the MCR. A set NI bit tells the source to observe the CI and ER fields in the RM cell, but not to increase the ACR above its current value. This can be expressed as follows:

$$ACR = \begin{cases} \max(\min(PCR, ER, ACR + N_{RM}AIR), MCR) & \text{if } CI = 0 \text{ and } NI = 0, \\ \max\left(\min\left(PCR, ER, ACR\left(1 - \frac{N_{RM}}{RDF}\right)\right), MCR\right) & \text{if } CI = 1, \end{cases} \tag{2.1}$$

where AIR is the Additive Increase Rate, and RDF is the Rate Decrease Factor. Note that the factors AIR and RDF control the rate at which the source increases and decreases its rate, respectively.

To make the ABR framework robust to synchronized surges in traffic from different users and to network failures, the source also must decrease its ACR if it is not taking full advantage of it or is not receiving the expected return flow of RM cells. Additional operational details of the ABR specifications can be found in [32].

2.4 Switch Mechanisms for ABR Congestion Control

The various switch mechanisms can be classified broadly depending on the congestion monitoring criteria used and the feedback mechanism employed. The three distinct switch mechanisms are 1) simple $EFCI$ marking, 2) selective marking and 3) explicit rate marking.

2.4.1 EFCI Marking

In an EFCI-based switch, if congestion is experienced in an intermediate switch during the connection, the $EFCI$ bit in the data cell will be set to 1 to indicate congestion. The CI field in the RM cell is set (i.e., $CI = 1$) by the destination if the last received data cell has the $EFCI$ field set (i.e., $EFCI = 1$) and is returned back to the source. The RM cells generated by the source and then returned by the destination represent opportunities for rate increases for the connection. If the source receives an RM cell with no congestion indication (i.e., $CI = 0$) the source is allowed to increase its rate. If the congestion indication bit is set (i.e., $CI = 1$) the source decreases its rate. The parameters AIR and RDF control the rate by which

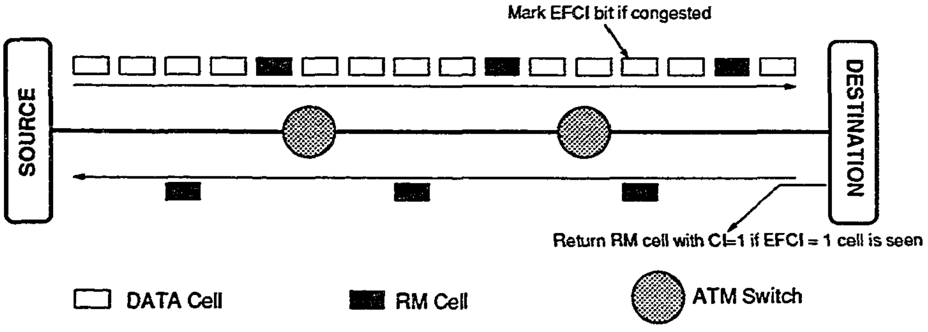
Mark EFCI bit if congested
SOURCE
DESTINATION
Return RM cell with CI=1 if EFCI = 1 cell is seen
DATA Cell
RM Cell
ATM Switch

Figure 2.5 Rate-Based Congestion Control with EFCI Marking

the source increases or decreases its rate. A basic operation of this type is shown in Figure 2.5.

The $EFCI$-based switches suffer from a phenomenon called the *beat-down problem.* In a network using only $EFCI$-based switches, where a congested switch marks the $EFCI$ bit of the data cell, sources traveling more hops have a higher probability of getting their cells marked than those traveling fewer hops. To illustrate this, consider a VC that traverses i hops. For the purpose of illustration, consider a network with the same level of congestion at all switches. Let the marking probability at each switch be P_m. Then, the end-to-end marking probability, P, of a data cell traversing i hops can be given by

$$P = iP_m. \tag{2.2}$$

This clearly shows that VCs traveling more hops have a higher probability of having their bit set than those traveling fewer hops. The long path VCs have very few opportunities to increase their rate and are beaten down more often than short path VCs. This is the beat-down problem.

2.4.2 Selective Marking

To overcome the beat-down problem one may use the current cell rate (CCR) value found in the RM cell and selectively indicate congestion on connections traversing a congested link to ensure fair-rate allocation among the competing connections. That is, during congestion periods, some connections with a CCR value higher than their fair share will be signaled to reduce their rates, while others whose CCR is lower than their fair share, may be allowed to increase their rates. This is sometimes referred to as "intelligent marking" or "selective marking."

2.4.3 Explicit Rate Marking

The basic option offered by the approach described above consists of single-bit congestion feedback, supported via $EFCI$ mechanisms in the switches and the CI bit in the RM cells. In addition to $EFCI$ marking and selective marking the switches may employ sophisticated switch mechanisms, which compute an explicit rate value that is fed back in the appropriate RM field [33],[34]. These more advanced approaches require switches that compute the fair share for each connection in a distributed fashion and explicitly set the source transmission rate. This specific rate is ER may be used by intermediate networks with small cell buffers that drive a connection rate lower to quickly respond to transient conditions where, say, a large number of idle connections sharing a link become active within a short time. A basic operation of this type is shown in Figure 2.6. During the initial transmission the ER field in the RM cell is set to PCR, and as the RM cell loops through the network, intermediate switches are allowed to reduce the ER value depending on their state of congestion. When the RM cell is received back at the source, the ER value placed in the RM cell is used explicitly to force the current ACR at the source to the smaller of the current ACR and the ER. Many algorithms, featuring sophisticated intelligent and explicit-rate mechanisms, have been proposed, demonstrating the enormous potential of the rate-based approach [11], [33]–[37].

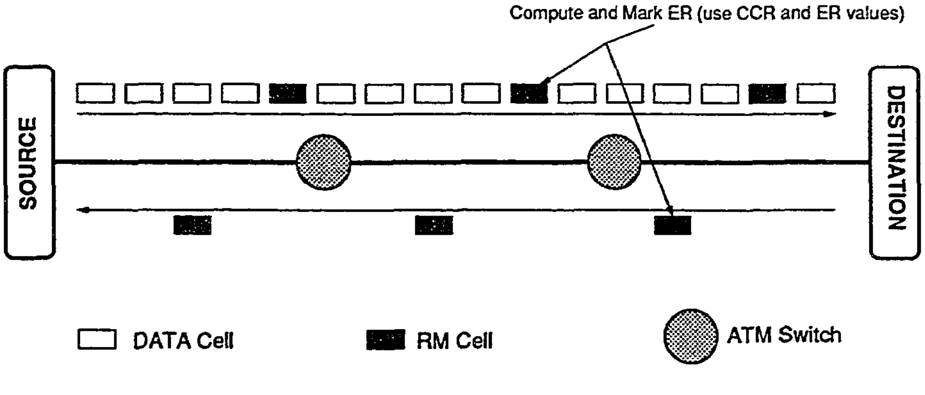

Figure 2.6 Rate-Based Congestion Control Scheme with ER Marking

2.5 Summary

In summary, the rate-based, closed-loop control mechanism for ABR services, defines the source-end and the destination-end system behavior, and it defines the means of forward and backward notification of congestion information. The ABR framework

1. supports end-to-end flow control, but also defines the option for intermediate switches or networks to segment the control loop;

2. allows switches to limit their participation in support of ABR connections using the simple 1-bit *EFCI* mechanism, but also to provide more detailed feedback that dynamically changes an explicit upper bound on the source rate;

3. defines mechanisms and control-information formats to allow switches implementing any of the above types of feedback to coexist within the same control loop and interoperate with the end systems.

The ABR framework is fundamentally a protocol for informing the source about the bandwidth made available to it by the network. The option for ABR switches to provide explicit-rate feedback does not specify how the network would derive the explicit rate, beyond the natural constraints described above. As a result, the ABR specification can support an ever widening family of rate-based, closed-loop implementations for congestion control. The flexibility inherent in the specification of the rate-based framework should offer service and equipment providers broad latitude in their creation of implementations appropriate to their various markets. At one extreme, the ABR framework allows for almost instantaneous access to available bandwidth (within one round-trip time for the control loop), as may be appropriate for LAN environments, and can be accomplished by using explicit-rate feedback (the *ER* field of RM cells). At the other extreme, the ABR framework allows for the more gradual change of rates, as may be appropriate for connections with long propagation delays, and can be accomplished with explicit-rate feedback, single-bit feedback (i.e., through *EFCI* or the *CI* field of RM cells), or with a coordinated use of both.

The flexibility of the ABR framework is grounded in a single set of behaviors implemented by the source and destination of the ABR connection. By fixing these behaviors, the ABR specification assures that users can take advantage of innovation and evolution in switch algorithms without changing their network interface cards and terminal adaptors. By following these behaviors, end systems fulfill their end of a contract for obtaining dynamic access to bandwidth and the reliable transport of data.

CHAPTER 3

FAIR RATE ALLOCATION FOR ABR SERVICE USERS

Providing ABR services to customers or users successfully depends on how the traffic in the network is managed, and one of the goals of traffic management is how to react in the event of network congestion. The ATM Forum has adopted a rate-based congestion control approach, which is based on a feedback flow control principle, to control congestion. In feedback flow control the network users adjust the load of traffic they send into the network, based on some information about the network status. Recall that in the rate-based approach, for each virtual connection the Resource Management (RM) cells are used to carry congestion information among the source, the switch and the destination. As noted earlier, the switches implementing the simple EFCI marking provides an unfair rate allocation to the sources traveling many hops. The explicit rate (ER) switches overcome the unfairness problem by computing a fair share of bandwidth and communicating this fair value to the sources. There are many such explicit rate calculation algorithms existing today. This chapter is intended to provide a survey of such algorithms and to evaluate them in terms of performance and complexity.

First it is necessary to provide the meaning of "fairness" and to provide a mathematical model of a fairness criterion. This issue is addressed in section 3.1. In general an ER algorithm must have many desired properties in terms of its performance and complexity. Section 3.2 presents such properties. Section 3.3 provides a brief survey of many proposed ER algorithms. This section also evaluates each algorithm in terms of its performance merits, shortcomings and complexity issues. We conclude this chapter by providing a brief summary and motivation for a new algorithm that improves the performance while keeping the complexity low.

3.1 Fairness Criterion

The issue of fairness in a multiuser environment, such as in ATM networks, has gotten a great deal of attention in the research community. The term "fairness" is conceived in a number of ways in the literature [38]–[40]. According to the fairness criterion that they employ, the various schemes fall into the following three categories [41]:

1. Fair utilization of network resources (e.g., link capacity, buffer);
2. Fairness in user performance (e.g., throughput, delay);
3. Balanced interference among users.

In the context of ABR service in ATM networks, since the ABR users are willing to accept delay variations, with the goal of using any unutilized link bandwidth, the notion of fairness which will involve fair distribution of available link capacity among many competing ABR users is suitable. Thus, the issue of fairness becomes the central element of congestion control. A commonly used fairness criterion is "max-min fairness," which was introduced in [42]. The ATM Forum has accepted the notion of max-min fairness as the criterion to decide fairness in an ATM network. This definition, however applies in an unambiguous way if no ABR connections receive bandwidth guarantees ($MCR = 0$), but various conflicting interpretations exist if connections use different, non-zero MCR values [43]. These issues will be addressed in Chapter 7.

3.1.1 Max-Min Fairness

The max-min criterion allows maximizing the link capacity allocated to users with the minimum rate allocation. Let us define $\mathcal{L}$ as the set of links and $\mathcal{S}$ as the set of virtual connections established in a given ATM network. Each session $j \in \mathcal{S}$ has a fixed path and traverses a subset of links $\mathcal{L}_j$. Denote $\mathcal{S}_l$ as the set of sessions that traverse link $l \in \mathcal{L}$. The capacity (or bandwidth) of each link $l \in \mathcal{L}$ is denoted by C_l.

Let λ_j^{CCR} denote the allocated rate at which the cells of session j enter the network (i.e., current cell rate), and let λ_j^{ER} denote the maximum permitted transmission rate of session j (i.e., explicit rate). In a network node, connections that are competing for bandwidth can be grouped into two categories:

1. Bottlenecked Connections – The connections that cannot achieve the their fair (equal) share of bandwidth at the particular node because of constraints imposed by their PCR requirements or by limited bandwidth available at other nodes along the route of the connection;

2. Unbottlenecked Connections – The connections that can achieve high bandwidth, and the bandwidth that they can achieve is limited by the bandwidth available at the considered node (usually referred to as a bottleneck node for that connection).

Definition 1 *A link $l \in \mathcal{L}$ is called the* bottleneck *link for a session j, such that $F_l = C_l$, and λ_j is at least as large as the rate of any other sessions using the bottleneck link. Furthermore, the rate, λ_j, is referred to as the* bottleneck bandwidth *for the session j.*

In the above definition F_l is the total flow on a link l given by

$$F_l = \sum_{j \in S_l} \lambda_j. \tag{3.1}$$

We require the fair allocations for each session to be non-negative while the total flow does not exceed the link capacity. These constraints can be formulated as follows.

$$\begin{aligned} \lambda_j &\geq 0, \quad \forall j \in \mathcal{S} \\ F_l &\leq C_l, \quad \forall l \in \mathcal{L}. \end{aligned} \tag{3.2}$$

Thus, the problem is to find a rate-allocation vector, $\Lambda = [\lambda_1, \lambda_2 \cdots \lambda_N]$, that is feasible (i.e., satisfies Equation (3.2)), and that is fair in the max-min sense. The key ideas behind max-min fairness are:

- Each VC has at least one bottleneck link along its path;
- Rates allocated to VCs bottlenecked at a link should be equal and can be given by

$$\text{Fair Share} = \frac{C_l - \sum \text{Rates of VCs bottlenecked elsewhere}}{N_l - \sum \text{VCs bottlenecked elsewhere}}, \qquad (3.3)$$

  where N_l is the number of connections using the link.

A simple procedure for finding the max-min fair rate allocations can be formulated iteratively as follows:

1. Find the equal share of each session on each link.
2. Find the VC with minimum allocation.
3. Subtract this rate and eliminate the VC with minimum allocation.
4. Recompute equal share of each link in the reduced network.
5. Repeat procedures 2 - 4 until all the VCs are eliminated.

The max-min principle is fair since all the users share a link, get equal share of bandwidth of the link provided that they can all use that fair share, and the only factor that prevents the user from obtaining higher allocation is the bottleneck link. Moreover, the max-min principle is efficient in the sense that it maximizes the throughput.

3.1.2 Example of Max-Min Fairness Criterion

As an example consider the network configuration shown in Figure 3.1. The network consists of four switches connected via three links. The bandwidths of links L1, L2 and L3 are 10Mbps, 50Mbps and 150Mbps respectively. Four VCs are setup such that the first link L1 is shared by two sources S1 and S2. The second link is shared by sources S2 and S3. The third link is shared by the sources S2, S3 and S4.

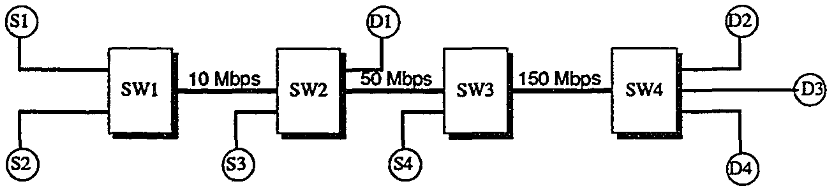

Figure 3.1 Network Configuration to Illustrate Max-Min Fairness

Table 3.1 Max-Min Fair Allocation Procedure

Iteration#	S1	S2	S3	S4
1	5	5	25	50
2	5	5	45	72.5
3	5	5	45	100

In order to calculate the fair share let us divide the link bandwidths fairly among contending sources. On link L1, we can give 5Mbps to each of the sources S1 and S2. On link L2, we could allocate 25Mbps for the two contending sources, S2 and S3. On link L4 we could give 50Mbps to each of the sources S2, S3 and S4. However, source S2 cannot use its 25Mbps share at link L2 since it is allowed to use only 5Mbps at link L1. Therefore, we give 5Mbps to source S2 and reallocate the bandwidth. Since source S2 only uses 5Mbps on link L2 we can allocate 45Mbps bandwidth to source S3. Now the new available bandwidth on link L3 is 145Mbps. We will divide this among the contending sources S3 and S4 equally at 72.5Mbps. Since the source S3 is bottlenecked at link L2 it can only use 45Mbps of its fair share of 72.5Mbps on link L3. Therefore, we will give the source S3 45Mbps on link L3 and reallocate the bandwidth for source S4. Now the available bandwidth on link L3 is $150 - (5 + 45)$ or 100Mbps and we will allocate this bandwidth to source S4. Thus, the fair allocation vector for this configuration is $\{5, 5, 45, 100\}$. The procedure is outlined in Table 3.1.

3.2 ATM Switch Mechanisms for ABR Service and Fairness

The various switch mechanisms can be classified broadly depending on the congestion monitoring criterion used and the feedback mechanism employed. Typical feedback mechanisms include binary feedback – setting of the $EFCI$ bit in the cell header, or explicit rate (ER) calculation – sending this information to the source through the RM cell. The inability of the $EFCI$-based switches to satisfy max-min fairness have lead to the development of many sophisticated ER-based switches. In this section, we will focus on the design of distributed explicit rate algorithms and provide a survey of some of the proposed algorithms.

3.2.1 Requirements for Explicit Rate Algorithms

Correct and efficient operation of ABR control vastly depends on the quality of the ER algorithm implemented at the switches. The objective is to determine the fair share rate in a distributed network under dynamic changes in the absence of centralized knowledge about the network and without the synchronization of different network components. An effective distributed algorithm must exhibit the following properties.

1. *Convergence:* Max-Min fairness criterion should be guaranteed for all the connections especially for a start-up connection, which has difficulty obtaining bandwidth from existing connections. In steady state the allowed rates for each connection should converge to their fair share without causing large oscillations. The large oscillations generally result in poor link utilization, low throughput and buffer overflow problems.

2. *Responsiveness:* The available bandwidth for ABR service connections vary rapidly since they operate with CBR, VBR and other ABR connections that generate competing traffic. Therefore, it is important to address the transient performance of the explicit rate algorithm, which must respond well in a dynamic environment. The

ability to provide fast access to the available bandwidth and rapid rate reductions under congestion are necessary requirements for an explicit rate algorithm. Moreover, the rates should converge to the new max-min fair allocations very quickly when new connections are formed and old connections exit.

3. *Implementation Complexity:* The algorithm should be simple enough to be implementable without adding much cost to the switch design. The required computations should be kept to a minimum and have to be done in a very short time. As an example, consider a switch-output port operating at 155 Mbps with a corresponding cell time of $2.75\mu s$. In order to achieve full efficiency of the line and cell-backlog, the switch must process and schedule each cell within $2.75\mu s$. In the future, the link speeds may go up to 622Mbps or even up to 2.4Gbps with corresponding cell-times of $0.68\mu s$ and $0.18\mu s$.

4. *Scalability:* The design of fair-rate allocation algorithms should scale well to support a large number of virtual connections in the ATM network. The number of calculations performed at each arrival of an RM cell should be kept to minimum to allow the switch processors to complete the required calculations within a short interval (i.e., a few cell-times).

5. *Robustness:* The algorithm must work correctly even in the presence of heavy cell loss, dynamic load changes, network failures and parameter mis-tunings. When many parameters are used in explicit rate calculations, any parameter mis-tuning may lead to performance degradation. It is desirable to have a small number of parameters where they can be set easily.

6. *Inter-operability:* The ER switches should be able to operate effectively with existing simpler switches that only perform $EFCI$ marking. The algorithms that use only the ER field in the RM cell suffer from inter-operability problems. Since the simpler switches do not modify the ER field, a switch that uses only the ER field loses any congestion information given by the simpler switch. This may lead

to poor link utilization, low throughput or serious buffer overflow problems. The algorithms that take advantage of both ER and CCR fields in the RM cell not only perform well, but also have the capability to inter-operate with other switches. The CCR field reflects the actual transmission rate of the source and thus any congestion information obtained from the simpler switches would have been taken into account of current allowed cell rate computations.

3.2.2 Design Issues

The choice among different explicit rate algorithms influences implementation in a significant way. Since ATM technology is meant to be scalable to very high speeds, a significant portion of the algorithms needs to be performed in hardware. The implementational issues concerning a switch must include effective congestion detection, buffer management, and explicit rate calculations. These issues are addressed below.

1. **Congestion Detection:** Effective congestion detection techniques that have broad implications on complexity and performance must be implemented. Congestion detection may be based on the use of the following methods: 1) a single threshold, 2) multiple thresholds, 3) differential of queue length, 4) output port link utilization or 5) variation of successive cell delay for the same connection.

2. **Per-VC Queueing vs. Per-VC Counting:** It has been recognized that per-VC queueing in ATM networks is prohibitive because its implementation does not scale with the number of connections to be supported, and it requires complex buffer management and scheduling algorithms and a high implementation cost. This very reason has led to the defeat of several congestion control proposals for ATM networks, for instance [19], [20]. At intermediate switches, queuing disciplines (buffer management and scheduling) can affect the behavior of traffic flows, and if properly implemented, lessen congestion. Queueing disciplines, however, do not affect congestion and flow control directly in that they do not change the total traffic

admitted into the network. Any flow controls, either open-loop or closed-loop, should be designed in such a way that they function well even in the presence of simple queueing disciplines in the network. We believe that implementing some class-based queueing strategies is necessary to handle complex dynamic traffic fluctuations in high-speed links, to isolate traffic flows and to ensure that quality of service for each class will be maintained. We also believe, however, that per-VC queuing may not necessarily provide any significant performance improvement over simple queuing disciplines, and its implementation complexity may not be justified.

Per-VC counting means that a switch stores some information related to each VC in its VC-table. Many of the algorithms take into consideration the expense of per-VC counting, and stay away from implementing per-VC counting-based algorithms. The algorithms which attempt to eliminate the need for per-VC counting use various approximation techniques requiring many parameters to force convergence. Many such algorithms that use various degrees of approximation suffer from degradation of convergence properties and poor transient response. Use of per-VC counting makes it possible to calculate exact fair-rates and thus it provides improved convergence and fast transient response.

3. **Explicit Rate Computation** This is the primary routine of any explicit rate algorithm. The explicit rate calculations can be made as a function of queue length, output link utilization, CCR of each connection, or ER value in the RM cell. Different algorithms use one or many of the above in their calculation routine. The key is to use the information available in an effective manner to achieve a high level of performance with few calculations.

3.3 Proposed Distributed Explicit Rate Algorithms

The ABR flow control specifications put forth by the ATM Forum present considerable freedom to switch vendors in designing and developing switch mechanisms that handle ER calculations. Many such algorithms that calculate ER have been proposed and can be classified into two broad categories based on the information used for ER computations. In the first category, the queue length is used as the congestion indicator, and rates are incrementally updated based on the level of congestion. The algorithms that fall in the second category require the switches to explicitly compute the available bandwidth and feedback of ER, which is calculated directly based on the available bandwidth. In this section a brief survey of some of the major proposed switch algorithms in terms of their performance and complexity is presented.

3.3.1 MIT Algorithm

The scheme is based on the Master's Thesis work of Anna Charny at MIT, and was originally formulated in the context of packet switching [44]. It can, however, be easily adapted for use in ATM networks.

Each switch monitors its traffic and calculates its available capacity per flow. This quantity is called the "advertised rate." The switches keep track of the bottlenecked connections and the last seen CCR values. When an RM cell arrives at the switch, if its ER is less than the advertised rate then the connection is assumed to be bottlenecked elsewhere, and the status bit and the current rate of the connection are stored in the VC-table. Equation (3.3) is used directly to compute the advertised rate. If at any time a connection previously marked transmits at a rate larger than the advertised rate, it must be unmarked and the advertised rate must be recalculated.

The algorithm converges to the optimal, max-min rates from any initial conditions, and the convergence time is upper-bounded by $4M$ round-trip times, where M is the number of bottleneck links. The steady state bandwidth utilization of the connections do not oscillate and have a fast transient response. Since the advertised rate calculations require the switch to refer all the connections' statuses, the algorithm has a computational complexity of $O(N)$, where N is the number of connections. These computations require significant processing time. The algorithm requires that all the switches along its path execute the same explicit rate algorithm, which prevents this algorithm to inter-operate with any other switch mechanisms. This is due to the fact that the ER calculations are based primarily on the ER field on the RM cell; consequently, if a simple switch that does not mark the ER field exists in the network, the congestion information of the simple switch will be ignored.

In order to overcome computational complexity, a quantized (discrete) rate allocation process is presented in [45]. This scheme requires that sources be allowed to send cells at a rate chosen from a discrete set of possible transmission rates. Thus the switches are only needed to store these discrete values, which reduces the complexity significantly. Unfortunately, this new approach requires that all the sources be aware of this discrete set of rates, and that all the switches in the network are implemented with the same algorithm.

3.3.2 Enhanced Proportional Rate Control Algorithm (EPRCA)

The EPRCA algorithm was originally proposed in [36]. The basic approach of this algorithm is based on the algorithm in [33] and the work in [34]. This algorithm is a heuristic approach, designed in such a way such that it does not require any VC-table reference. The EPRCA algorithm uses the queue length as the congestion indicator and computes an approximate fair rate, called the *Mean Allowed Cell Rate* (MACR). The fair share, based on the level of congestion, CCR, and the computed fair rate, is conveyed to the sources, and the sources adjust their rates accordingly.

In this approach, during uncongested periods the the switches estimate the fair-rate by computing the $MACR$ for all VCs, as shown below.

$$MACR = (1 - AV)(MACR) + (AV)(CCR), \tag{3.4}$$

where AV is the averaging factor. By this approach the average of the CCR's of the VCs that are not bottlenecked elsewhere can be estimated. The fair share (γ) is set as a fraction of this average.

$$\gamma = (DPF)(MACR), \tag{3.5}$$

where DPF is a multiplier called the *switch down pressure factor*. The ER field in the returning RM-cells are reduced to fair share if the switch is in a congested state, as follows.

$$\lambda^{ER} = \min(\lambda^{ER}, \gamma). \tag{3.6}$$

The switch may also set the CI bit in the cells passing when it is congested, which is sensed by monitoring its queue length. When the switch is congested, $MACR$ is then used to selectively mark the reverse direction RM cells so that the ones with a CCR greater than the $MACR$ are marked with $CI = 1$ (congested), and those below the $MACR$ are left to increase. This is the basic solution to beat-down, the case where a VC through many nodes is marked congested too often and gets an unfair, low capacity.

To make this scheme work in practice, however, the closed loop of the switches and the sources with their CCR's and MACR's must converge under all conditions. To insure this, the algorithm has several multiplier factors used to force convergence. Although the EPRCA scheme is simple and does not require any per-VC accounting, it uses many parameters to force convergence, and unless these parameters are tuned properly, the algorithm does not perform well under a wide range of network scenarios. Improper selection of these parameters results in large oscillations in rates, poor link utilization, large buffer requirements, and poor response time.

3.3.3 Explicit Rate Indication for Congestion Avoidance (ERICA)

The ERICA (Explicit Rate Indication for Congestion Avoidance) approach proposed in [35] computes a fair rate based on the number of active connections, the input rate of queue and the available bandwidth. Each ABR queue measures its input rate, $F(t)$, over a fixed "averaging interval" and divides it by the target rate, F^{tgt} (which is set slightly below the link bandwidth), as shown in Equation (3.7). A target utilization value is used to control the queue growth rate. Based on the known available capacity of the link, the switch computes a load factor, ρ, which is a ratio of actual input flow and target operating speed of the link.

$$\rho = \frac{F(t)}{F^{tgt}}. \tag{3.7}$$

Measurement of load consists of simply counting the number of cells received during a fixed averaging interval. A load factor of less than 1 indicates that the queue is underloaded and a load factor of greater than 1 indicates that the queue is overloaded. Utilizing the load factor the queue, the VC share bandwidth, γ^{vc}, is calculated as

$$\gamma^{vc} = \frac{CCR}{\rho}. \tag{3.8}$$

The CCR value used to compute γ^{vc} comes from the forward RM-cells and the feedback is given in the backward RM-cells. This ensures that the most current information is used to provide the fastest feedback.

In order to achieve fairness, this scheme allows underloaded VCs to increase their rate to fair share. The switch calculates fair share, γ, as:

$$\gamma = \frac{F^{tgt}}{N}, \tag{3.9}$$

where N is the number of active VCs in the queue. Upon reception of an RM cell in the backward direction the ER field in the backward RM cell is marked as follows.

$$\lambda^{ER} = \min\left\{\lambda^{ER}, \max(\gamma, \gamma^{vc})\right\}. \tag{3.10}$$

The ERICA algorithm operates in a congestion-avoidance basis, insensitive to parameter variations, and proves to be very robust. The rates converges very quickly and operates without any oscillation. This algorithm, however, has some fundamental limitations in terms of achieving fairness for all the connections and buffer requirements. In some cases a connection that starts late – although it gets its equal link share – it does not get the max-min fair rate. Furthermore, during transient periods, and if the desired target utilization is set close to the full link rate, the queue grows rapidly and results in heavy cell loss.

3.3.4 Congestion Avoidance Using Proportional Control (CAPC)

The CAPC algorithm proposed in [37], the switches also set a target utilization and measure the input rate to compute load factor ρ. During underload ($\rho < 1$), fair share is increased as:

$$\gamma = \gamma \ \min(ERU, 1 + (1 - \rho)\, Rup), \tag{3.11}$$

where Rup is a slope parameter between 0.025 and 0.1, and ERU is the maximum increase allowed. During overload ($\rho_l \geq 1$), fair share is decreased as:

$$\gamma = \gamma \ \max(ERF, 1 - (\rho_l - 1)\, Rdn), \tag{3.12}$$

where Rdn is a slope parameter between 0.2 and 0.8, and ERF is the minimum decrease required. The ER field in the RM cell is updated as follows.

$$\lambda^{ER} = \min(\lambda^{ER}, \gamma). \tag{3.13}$$

The CAPC algorithm does not require any per-VC counting approach; however, the parameters that are required to force convergence must be set carefully. Incorrect settings of the parameters exhibits large oscillations in rates, and does not converge to the correct rates.

3.4 Summary

The ABR flow control specifications put forth by the ATM Forum present considerable freedom to switch vendors in designing and developing switch mechanisms that handle *ER* calculations. In this chapter, a summary of many algorithms that calculate *ER* is presented. These algorithms have varying degrees of complexity and performance characteristics.

The issues relating to convergence, responsiveness, robustness, scalability, and implementational complexity dominate the design of an effective distributed algorithm. The performance characteristics of non per-VC counting approaches suffer significantly in terms of convergence, robustness, and responsiveness. The per-VC counting approaches, although more complex, provide significant performance merits. The algorithm developed in this dissertation attempts to use per-VC counting in an effective manner to achieve a high level of performance while keeping the computational complexity low.

CHAPTER 4

IMPACT OF QUEUEING DISCIPLINES ON ABR CONGESTION CONTROL

It has long been recognized that per-VC queueing in ATM networks is prohibitive because its implementation does not scale with the number of connections to be supported, and it requires complex buffer management and scheduling algorithms and a high implementation cost. This very reason has led to the defeat of several congestion control proposals for ATM networks, for instance [19], [20]. The rapid growth in both use and size of ATM networks, however, has sparked a renewed interest in incorporating per-VC implementation in commercial ATM switches [46]. Several ATM switch vendors have recently announced their per-VC implementation strategies. There is no clear technical evidence, however, to show what benefits per-VC queueing can provide to justify its implementation cost. The objective of this chapter is to address this issue and provide insights and understanding of the impact of queueing disciplines on ATM networks.

At intermediate switches, queueing disciplines (buffer management and scheduling), which control the usage of buffer space and the order in which cells are sent, can affect the behavior of traffic flows, and if properly implemented, lessen congestion. However, queueing disciplines, do not affect congestion and flow control directly in that they do not change the total traffic admitted into the network. Any flow controls, either open-loop or closed-loop, should be designed in such a way that they function well even in the presence of simple queueing disciplines in the network. We believe that implementing some class-based queueing strategies is necessary to handle complex dynamic traffic fluctuations in high-speed links, to isolate traffic flows and to ensure that quality of service for each class will be maintained. We shall argue, however, that per-VC queueing may not necessarily provide any significant performance improvement over simple queueing disciplines, and its implementation complexity may not be justified. To evaluate this, we examine two queueing disciplines and their impact on ABR traffic.

This chapter is written in order to accomplish three goals. The first was to discuss the different queueing strategies possible to complement ABR traffic control. The second was to provide a good understanding of queueing disciplines versus different ABR implementations. This is done in section 4.1, where we present simulation results for a specific benchmark network. The third goal was to evaluate complexity and performance trade-offs between two queueing disciplines. This point is discussed in section 4.2. The major conclusion obtained in this study is that simple FIFO queueing is adequate for ABR control, where connection-based closed-loop control can handle congestion effectively and fairly. We observe that the explicit connection-based control can implicitly achieve the same effect as per-VC implementation. Following a similar line of reasoning, we believe that per-VC queueing may not even be required for open-loop controlled traffic such as CBR and VBR, and effective policing and traffic shaping on a per-connection basis may well protect well-behaved users, and ensure that quality of service can be achieved.

4.1 Buffer Management and Cell Scheduling

A network switch provides the necessary resources; namely, port bandwidth and buffers, for routing ATM cells. The port bandwidth and the buffer size are limited resources and are heavily contended. The heavy contention over a period of time may lead to congestion and, as a consequence, to possible loss and excessive delay of ATM cells. Therefore, in order to provide high quality-of-service a switch must employ efficient buffer management techniques and cell scheduling policies. As we have already mentioned, the rate-based framework provides many degrees of freedom in the behavior and implementation of network switches. One such freedom is the capability of employing various buffer management strategies and cell scheduling policies.

Buffer management strategies control the usage of buffer space, monitor the level of congestion in its resources, and employ cell discarding strategies. Cell scheduling policies decide how to schedule each cell and transmit over the link (i.e., the order in which the cells are transmitted). There are several options for such a buffer management scheme. For example, single FIFO queueing or per-VC queueing can be used as two distinct techniques. In the single FIFO queue approach the queue is served in a FIFO manner, thus there is no need for a complex scheduling mechanism. The per-VC queueing approach may employ various cell scheduling mechanisms, such as weighted round-robin or weighted fair queueing [47]. In addition to queueing disciplines, effective congestion detection techniques that have broad implications on implementation complexity and performance must be implemented. The congestion detection may be based on the use of the following methods: 1) a single threshold, 2) multiple thresholds, 3) differential of queue length, 4) output port link utilization or 5) variation of successive cell delay for same connection. The single FIFO queueing and per-VC queueing techniques are discussed below. For both approaches a two threshold-based congestion detection mechanism is used.

4.1.1 Single FIFO Queueing

In this approach, a centralized output-port memory is completely shared by a single queue, where all the cells from different sources form a single queue and the cells are scheduled in a FIFO manner. This type of discipline is the simplest, most economical and commonly implemented queueing discipline. If the queue length exceeds the available buffer space then the incoming cells are discarded. To minimize cell-loss, congestion must be detected effectively. In this work, congestion detection is done by using two queue thresholds. When the queue length is above the high-threshold

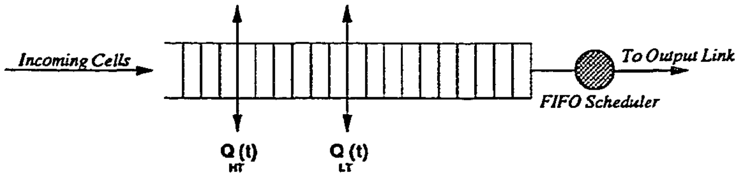

Figure 4.1 The Single FIFO Queueing and Two Threshold Congestion Detection Approach

(Q_{HT}) level the congestion indication flag is set to 1 (i.e. $\sigma(t) = 1$) and remains set until the queue length drops below the low-threshold (Q_{LT}) level. This can be mathematically expressed as follows:

$$\sigma(t) = \begin{cases} 1 & \text{if } Q(t) > Q_{HT} \\ 1 & \text{if } Q(t) > Q_{LT} \text{ and } \sigma(t-) = 1 \\ 0 & \text{if } Q(t) < Q_{LT} \end{cases} \qquad (4.1)$$

This two threshold detection method ensures that the oscillations are minimized. This detection mechanism is illustrated in Figure 4.1.

4.1.2 Per-VC Queueing

Unlike the first approach where all the VCs are queued in one single queue, in the per-VC approach, cells from different VCs are queued in separate queues and the buffer space is allocated on a per-VC basis. Multiple classes of traffic with varying degrees of priority and delay requirements can be fairly served with the per-VC implementation in conjunction with a fair scheduling policy such as the weighted fair queueing or weighted round-robin scheduling policy. The output-port buffer space could be divided among all the VCs in a fixed manner or dynamically shared among VCs. In the fixed buffer size allocation method (static) each VC is only allowed to occupy its own VC buffer share, but in adaptive buffer management VCs can take up more than their share. When the buffer becomes full, however, a cell from the queue with the largest queue length is dropped to allow room for the newly arriving cells. The adaptive buffer management scheme utilizes the buffer space more

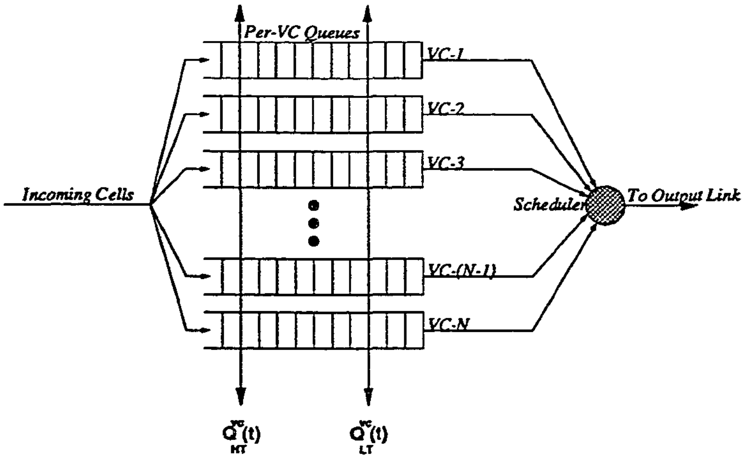

Figure 4.2 The Per-VC Queueing and Two Threshold Congestion Detection Approach

efficiently than a fixed buffer management scheme without wasting any buffers. In order to compare the performance of per-VC queueing with single FIFO queueing the state of congestion is determined similarly using the two-threshold approach described earlier. When the aggregate sum of all individual queue lengths is above the high-threshold (Q_{HT}) level, the individual queue lengths are compared with the per-VC-threshold, Q^{vc}_{HT}, and the congestion indication flag for the VC, $\sigma^{vc}(t)$ which exceeds the threshold, is set. This congestion flag remains set until the individual queue length drops below its low-threshold, Q^{vc}_{LT}.

$$\sigma^{vc}(t) = \begin{cases} 1 & \text{if } Q^{vc}(t) > Q^{vc}_{HT} \\ 1 & \text{if } Q^{vc}(t) > Q^{vc}_{LT} \text{ and } \sigma^{vc}(t-) = 1 \\ 0 & \text{if } Q^{vc}(t) < Q^{vc}_{LT} \end{cases} \tag{4.2}$$

The per-VC queue lengths are calculated as follows.

$$Q^{vc}_{HT} = \frac{Q_{HT}(t)}{N_{vc}} \tag{4.3}$$

$$Q^{vc}_{LT} = \frac{Q_{LT}(t)}{N_{vc}} \tag{4.4}$$

It is essential to note that, in the simulations below we are not concerned with any priority or delay requirements for ABR traffic, and thus the scheduling

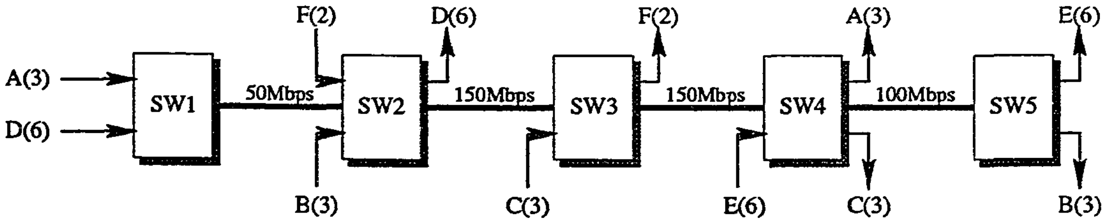

Figure 4.3 The GFCl Network

mechanism we incorporate (i.e., weighted round-robin) translates into a simple round-robin scheduling policy.

4.2 Simulation Results and Discussions

In order to study the impact of queueing policies on the ABR congestion control scheme in ATM networks, a simple network, shown in Figure 4.3, is simulated. This network configuration is referred to as the Generic Fairness Configuration 1 (GFC1), and it is one of the benchmark configurations recommended by the ATM Forum [17]. The GFCl network consists of five switches and 23 connections grouped into six classes (A-F). In Figure 4.3 the number inside the parentheses next to the group label represents the number of VCs for that group. VCs in groups C, D, E, and F are single-hop traffic and VCs in groups A and B are three-hop cross-traffic. The links connecting hosts to switches have a capacity of 150Mbps. All the links are 400 meters in length and have a propagation delay of $4\mu s$ per Km. For performance measurement purposes, the switches are assumed to be non-blocking and output buffered. The sources are assumed to be well behaved, persistently greedy, and can transmit at the peak link rate when the bandwidth is available. The use of persistent sources presents a tough challenge for multiple congested links. The parameters used throughout this study are tabulated in Table 5.1.

Table 4.1 GFC1 Simulation Parameters

Parameter	$EFCI$ Mark	Intelligent Mark	ER Stamping
N_{RM}	32	32	32
PCR (Mbps)	150	150	150
MCR (Mbps)	0.150	0.150	0.150
ICR (Mbps)	7.5	7.5	7.5
AIR (Mbps)	0.0157	0.0157	0.0157
RDF	256	256	256
Buffer Size	750	750	750
Q_{LT}	50	50	50
Q_{HT}	100	100	100
VCS	NA	0.875	0.875
DPF	NA	0.875	0.875
DQT	NA	500	500
AV	NA	0.0625	0.0625
ERF	NA	NA	0.94
MRF	NA	NA	0.25

In this study, the following scenarios, which represent possible combinations of implementations, are simulated and the results are compared in terms of fairness, throughput, link-utilization, cell-loss rate and switch output-port memory utilization.

- **S1** : Simple $EFCI$ marking with single-FIFO queueing
- **S2** : Simple $EFCI$ marking with per-VC queueing
- **S3** : Selective marking with single-FIFO queueing
- **S4** : Explicit Rate marking with single-FIFO queueing
- **S5** : Explicit Rate marking with per-VC queueing

4.2.1 Throughput and Fairness

One of the major goals of ABR service is to achieve fairness while maximizing the throughput. The ATM Forum has adopted the notion of the "*max-min*" fair allocation principle, which has been studied extensively in the literature as the basis for the evaluation of fairness [33], [39]. The max-min criterion provides the maximum possible bandwidth to the source receiving the least among all contending sources. This is done by first maximizing the link capacity allocated to the users with the minimum allocation and then using the remaining link capacity for other users in a way that it maximizes the allocation of the most poorly treated users. The max-min principle is fair since all VCs sharing a link get an equal share of bandwidth provided they can all use the fair share. Moreover, the max-min principle is efficient in the sense that it maximizes the throughput. The second column in Table 4.2 shows the max-min fair allocation of each group. Table 4.2 also shows the average throughput (in Mbps) achieved with different simulation scenarios.

Figure 4.4 illustrates the impact of queueing policies on fairness using the $EFCI$ and the ER marking schemes. The fairness percentage is calculated by computing average throughput over a fixed time and dividing it by the fair value of the throughput. From the figure it can be seen that marking only the $EFCI$ bit and using the single FIFO queueing policy (case **S1**) leads to unfairness among classes. In particular, connections in Groups A and B, which compete for bandwidth in multiple links, receive a lower than the max-min fair share, while connections in the other classes take advantage of this situation and receive more than their fair share. Using a single FIFO queue in conjunction with selective marking results in a fair bandwidth allocation (case **S3**). Since only the VCs that exceeding their fair share are marked, the beat-down effect disappears. Similarly, the $EFCI$ marking with per-VC queueing (case **S2**) leads to global fairness among all the groups. A simple $EFCI$ marking scheme with per-VC queueing solves the beat-down problem even though per-VC queueing is a local policy; namely, per-VC queueing isolates individual connections and has the selective effect under congestion, where only the cells from the VCs using the buffer extensively are marked. Implementing ER schemes using either queueing discipline (cases **S4** and **S5**) produces almost identical fairness. Since the

Table 4.2 Throughput (in *Mbps*) Comparison for the GFC1 Network

Group	Expected	*EFCI* FIFO	*EFCI* per-VC	Select FIFO	ER FIFO	ER per-VC
A	5.56	3.13	5.18	5.33	5.31	5.37
B	11.11	5.67	10.18	10.50	10.68	10.80
C	33.33	36.81	30.40	32.06	32.4	32.30
D	5.56	6.48	5.19	5.39	5.41	5.37
E	11.11	12.76	10.18	10.96	10.76	10.78
F	50.00	53.27	46.68	50.15	48.26	48.52

ER mechanism implemented in the switches computes the fair share and informs the ER via RM cells, queueing policies do not provide any additional advantages in terms of fairness using the ER method. The ER marking schemes allow the sources to reduce the sending rate very quickly under severe congestion situations and help the switches recover from congestion.

4.2.2 Link and Buffer Utilization

High link or network utilization is critical to ensure efficient and profitable operation of the network. The average link utilization at the steady state is shown in Figure 4.5. From the figure it can be seen that the ER mechanism using either queueing discipline (cases S4 and S5) and the selective marking scheme (case S3) results in high utilization of the links. However, the simple *EFCI* mechanism under-utilizes the network regardless of the queueing policies implemented. Network utilization is highly dependent on buffer occupancy levels. The utilization can be made very high by simply adjusting the parameters (HT, LT) to force the queue length to increase. With a larger queue, slight variations originating from on-off sources causing oscillations in buffer usage are absorbed by the queue without the queue ever becoming empty. However, the larger the queue length, the higher the delay. Therefore, there exists a clear trade-off among buffer size, delay and network utilization. Figures 4.6 and 4.7 illustrate peak queue length and the steady-state average queue length of

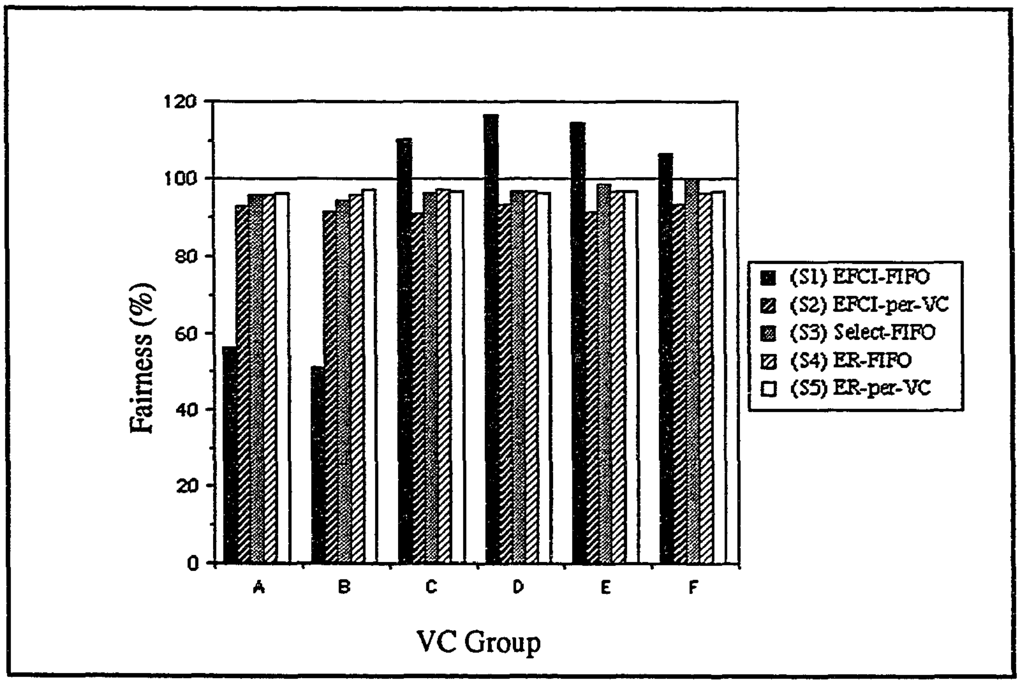

Figure 4.4 Impact of Queueing Policies on Fairness

switch ports. The queue peaks shown are for the worst case, including both the initial transient period and the steady-state region. Using single-FIFO or per-VC queueing disciplines in conjunction with the ER mechanism does not offer any further advantages. In both cases the resulting average queue length and peak queue length are almost identical. From these results it is clear that the per-VC approach does not save any buffer space or produce higher utilization than the single FIFO queue.

4.2.3 Cell-loss Rate

For all the scenarios simulated, no cell loss was observed. This shows that for the same amount of memory, the queueing policies do not have an impact on cell-loss. Again, since the rate-based approach implicitly controls the rate at which each VC transmits cells, the overall system works very effectively without providing any

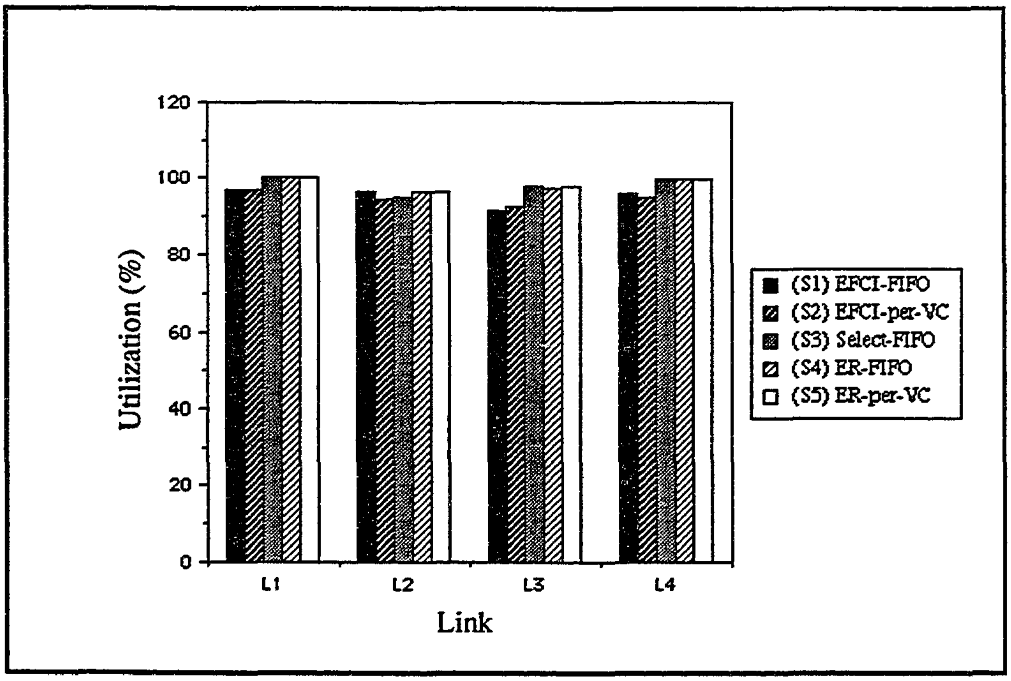

Figure 4.5 Impact of Queueing Policies on Link Utilization

complex local queueing and scheduling policy. A single queue with enough intelligence may be able to provide the minimum cell-loss requirement desired by the ABR traffic sources.

4.3 Performance - Complexity Tradeoffs

The choice among different queueing disciplines influences the implementations in a significant way. The implementation of these two queueing disciplines comes widely varying complexity. In the past, with lower speed networks, the issue of implementations of buffer management and scheduling policies was somewhat less important, since much of it is implemented in software. ATM, however, is meant to be scalable to much higher link or port speeds (i.e., 2.4 Gbps), thus the implementation of control schemes needs to be performed in hardware. In this section we will compare

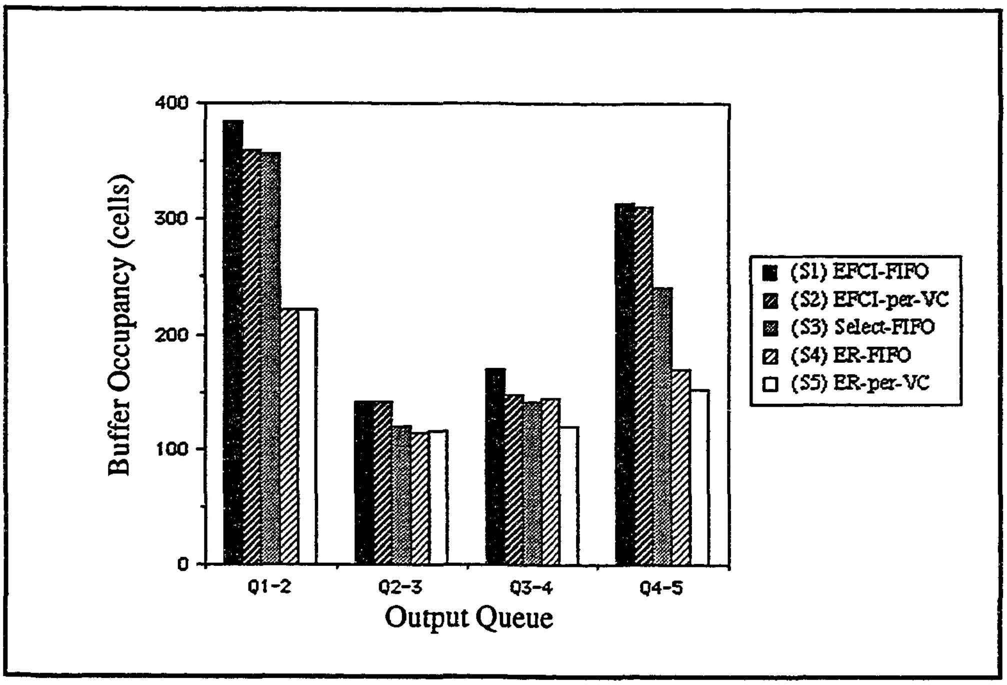

Figure 4.6 Impact of Queueing Policies on Peak Queue Length

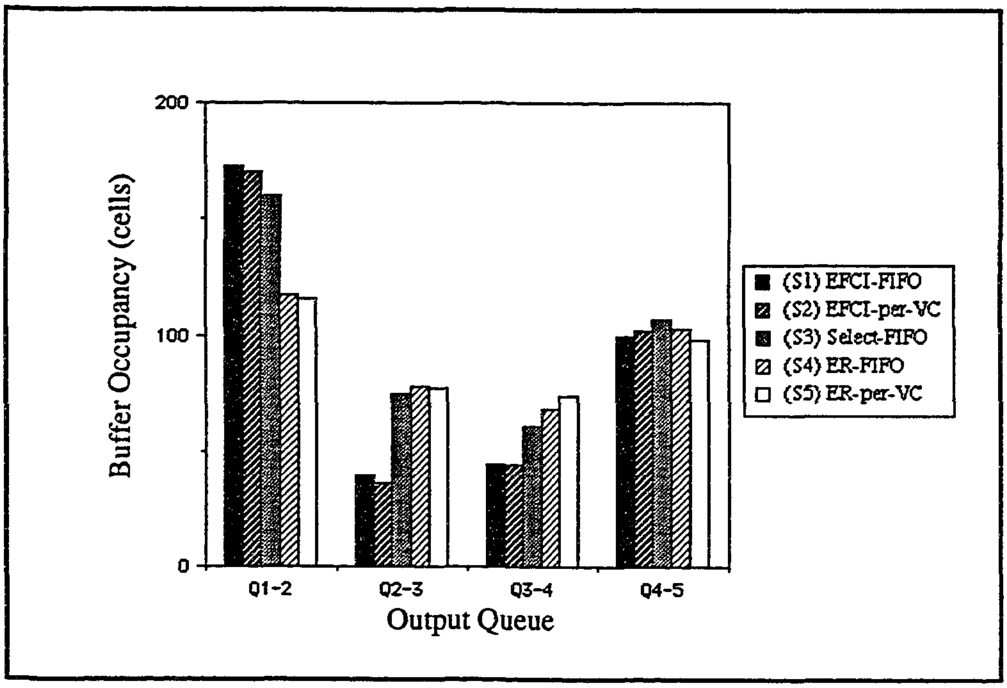

Figure 4.7 Impact of Queueing Policies on Average Queue Length

the advantages and disadvantages of the FIFO and per-VC queueing policies and address the implementation issues.

A shared port buffer served in a FIFO fashion is the simplest, most economical and commonly implemented queueing discipline. Because of this simple nature it is easily implementable particularly at very high-speed ports. It does not, however, provide any local mechanisms to enforce a fair access to buffers and bandwidth, and it leaves such resources open to abuse by malicious users. Due to this unfair access to buffers and bandwidth the simple $EFCI$-marking control scheme suffers considerably in terms of fairness. This fairness problem can be overcome by using ER-based switches, which calculate the fair share for each VC and inform the source of this fair rate. In addition to intelligent techniques, the single-FIFO queueing discipline will require an external mechanism that serves a policing function to alleviate the problems caused by malicious sources.

The per-VC approach requires switches to keep a separate queue for each VC. The accounting of occupancy of the buffer is performed on an individual VC basis. The isolation provided by the separate queues ensures fair access to buffer space and bandwidth. This also allows the delay and loss behavior of individual VCs to be isolated from each other. Furthermore, per-VC information is readily available to help congestion control, such as early packet discard mechanisms. This per-VC information can also be used to help police misbehaving users effectively. It is important to note that in a static buffer management scheme, where the VCs are given a fixed buffer share, the policing can be completely eliminated. This is because in the static buffer scheme if a VC is misbehaving only its queue will grow and overflow. On the other hand the adaptive buffer scheme must utilize some intelligent mechanism in order to achieve efficient policing.

Although per-VC queueing offers many advantages over single FIFO queueing, per-VC implementation suffers considerably in terms of implementation and scheduling complexity. Since per-VC queueing causes switch complexity to be proportional to the number of VCs, the approach will not scale well given that some large switches will support millions of VCs causing considerable complexity. In addition, complex scheduling policies must be implemented on a per-VC basis, which adds extra complexity and cost.

As an example, consider a switch-output port operating at 155 Mbps with corresponding cell time of $2.75\mu s$. In order to achieve full efficiency of the line and cell-backlog, the switch must process and schedule each cell within $2.75\mu s$. In the future, the link speeds may go up to 622 Mbps or even up to 2.4 Gbps with corresponding cell-times of $0.68\mu s$ and $0.18\mu s$. In addition to link speeds the size of the network is expected to grow, which makes it necessary for the switches to handle millions of VCs. Implementing per-VC queueing means that a switch must be able to handle complex cell scheduling techniques. The scheduling techniques may be implemented without any added cost for low-speed switch ports with a small number of VCs. On the other hand, at higher link speeds and with the increasing number of VCs the complex scheduling mechanisms have to process each cell within the time specified above, which will consequently will result in very complex and expensive hardware.

4.4 Summary and Conclusions

We have examined two queueing disciplines in terms of their performance and complexity in the presence of ABR traffic and rate-based control. In studying the impact of queueing policies, it is clear that there is no significant increase in performance using per-VC queueing over single FIFO queueing. Per-VC queueing is good if selective marking has not been implemented for *EFCI* switches, but it is unnecessary for ER switches. In the context of ABR traffic, since the rate-based scheme implicitly controls congestion on a per-VC level, a single-FIFO queue with some intelligent marking or explicit rate setting scheme can achieve the same performance as the per-VC approach. Furthermore, selective *EFCI* marking is considerably simple in the implementation against per-VC queueing, and can sustain as good performance as per-VC queueing in terms of fairness, throughput, and link-utilization. Using a similar line of reasoning we can argue that the effective and properly implemented open-loop control for CBR and VBR traffic may eliminate the necessity to implement per-VC queueing discipline.

This conclusion, however, does not exclude the merits that per-VC queueing can provide; namely, isolation, fairness and elimination of policing. On the other hand, the hardware complexities and non-scalable nature of the per-VC queueing approach make it very costly to implement in real networks. It's no doubt that in some places in the network the per-VC queueing is required, such as virtual source (VS) and virtual destination (VD) terminating points. At these points control loop needs to be segmented to apply proprietary control schemes or to add extra protection. Moreover, the per-VC-queueing approach may cost-effectively used at the network entry points where traffic shaping and policing are necessary. This, however, in no way justifies per-VC queueing at every switching points.

It is certain that at the connection level, the single-FIFO approach will not be sufficient in guaranteeing the quality of service requirements for CBR and VBR traffic, and handling the complex dynamic traffic fluctuations. This, however, does not mean that we must resort to the per-VC approach. Rather a per-class queueing approach may be implemented and justified in terms of performance, complexity and cost.

CHAPTER 5
THE FAST MAX-MIN RATE ALLOCATION ALGORITHM

One of the flexibility of the rate-based control technique is the calculation of explicit cell transmission rates by the switches for ABR service users. Since many ABR service users will be competing for the bandwidth, the available bandwidth must be shared fairly among all the ABR service users. To achieve "fairness," each switch in an ATM network should execute a rate allocation algorithm. The challenge is to calculate the explicit rate, which is a fair share of available bandwidth, in a distributed and asynchronous way. Many such algorithms have been proposed and can be broadly classified into two categories: queue length based algorithms (e.g., EPRCA algorithm [36]) and link utilization based congestion avoidance algorithms (e.g., ERICA algorithm [35]).

The queue length-based approach, such as the EPRCA (Enhanced Proportional Rate Control Algorithm), monitors the queue length and computes an approximate fair rate, called *Mean Allowed Cell Rate* (MACR). The congestion information, based on the level of congestion and the computed fair rate, is conveyed to the sources, and the sources adjust their rates accordingly. Although the EPRCA scheme is simple and does not require any per-VC accounting, it uses many parameters to force convergence, and unless these parameters are tuned properly, the algorithm does not perform well under wide range of network scenarios. Improper selection of these parameters results in large oscillations in rates, poor link utilization, large buffer requirements, and poor response time.

The ERICA (Explicit Rate Indication Congestion Avoidance) approach proposed in [35] computes a fair rate based on the number of active connections, load factor and available bandwidth. Unlike the EPRCA scheme, the ERICA algorithm requires the switches to keep track of each VC's status. ERICA, however, does not compute accurate fair rates, and in some instances fails to converge to the correct fair rates, especially the sessions that start late sometimes fail to obtain their fair share.

It has been demonstrated that the credit-based approach has significant merits in the performance that it provides, but with added cost and complexity [19]. In this chapter, an algorithm which can achieve the high level of performance that the credit-based approach provides but without the necessity of complex and expensive switch design. It has been argued in [19] that the rate based approach under utilizes the network resources when the network consists of bursty sources. The credit-based approach achieves full utilization by directly controlling the buffer usage and over-allocating the resources, where if a connection goes idle for some time the other VCs cells can be used to fill in the extra bandwidth. In the rate-based approach it is true that the simpler switches without any added intelligence will suffer significantly in the bursty environment, however, with proper design and buffer usage a switch algorithm designed to complement rate-based approach can achieve the same result.

It is required that an algorithm should work well with varying propagation delays in the network ranging from few microseconds (in a LAN) to few milliseconds (in a WAN) and the propagation delays in the network significantly influences the responsiveness of a control scheme. The performance requirements in LAN, however, vary widely with those of a WAN. In a LAN the available bandwidth is usually in abundance, and in a WAN the bandwidth is more scarce and highly shared. Moreover the users in a LAN requires faster response than the response time required by a WAN user. Therefore, it is necessary that the rate allocation algorithm in a LAN should be very fast in allocating bandwidth, thus requires a quick ramp-up time, and in a WAN the links should be fully utilized to maximize the number of connections.

With these in mind we propose a new, fast, fair-rate allocation algorithm primarily intended to achieve a high level of performance while keeping the switch complexity low and overcoming some of the problems found in other rate allocation algorithms. This algorithm is called *Fast Max-Min Rate Allocation* (FMMRA) algorithm. FMMRA algorithm functions in a congestion avoidance basis, requires per-VC accounting, and calculation of load level. The use of per-VC accounting in an effective manner is the main reason which enables the FMMRA algorithm to compute exact fair shares and to obtain high level of performance. The key features of our algorithm are very fast convergence, oscillation free steady state, low buffer size requirements, high link utilization and high-start feasibility.

The remainder of this chapter is organized as follows. A detailed description of the proposed rate allocation algorithm is presented in section 5.1. Section 5.2 presents the convergence analysis to validate the claim that the algorithm converges to the correct fair allocations. Some benchmark network simulation scenarios are simulated and the results are presented in section 5.3. We conclude the chapter by providing a summary and some remarks in section 5.4.

5.1 The FMMRA Algorithm

This section describes our switch algorithm for ABR congestion control to achieve max-min fairness. The algorithm can be executed by any switch component experiencing congestion (i.e., input port, output port, etc.). The key idea behind the FMMRA algorithm is that each ABR queue in the switch computes a rate that it can support. We refer to this rate as the *advertised rate*, γ_l, where subscript l refers to the link that the queue serves. The *ER* field in the RM cell is read and marked in both directions to speed up the rate allocation process. If a session cannot use the advertised rate, the session is marked as a bottlenecked elsewhere, and its bottleneck bandwidth is recorded. The advertised rate is recomputed incorporating the bottleneck status of the sessions. The advertised rate computation requires an updating rule, and the development of this rule is given in section 4.1. To facilitate the fair rate calculations, we define the following functional modules:

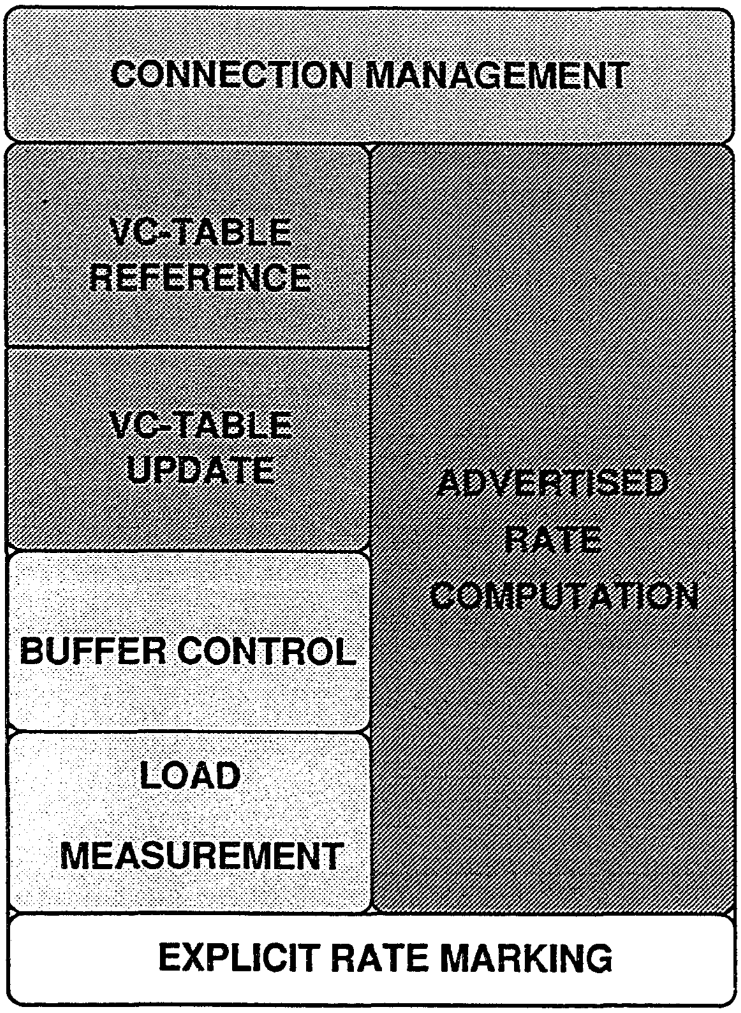

Figure 5.1 A Layered View of the FMMRA Algorithm

1) the ABR connection management module, 2) the bandwidth management module, 3) the explicit rate calculation module, and 4) the congestion detection and buffer management module. A layered view of the FMMRA algorithm is shown in Figure 5.1, and a detailed description of these modules are given in the subsequent subsections.

5.1.1 Derivation of Updating Rule

The objective of any max-min fair allocation algorithm is to identify the bottleneck bandwidth of each connection in an iterative manner. This is done by first maximizing the link capacity that is allocated to the sessions with the minimum allocation, and then using the remaining link capacity for other sessions, in a way that it maximizes the allocation of the most poorly treated among those sessions. For the purpose of determining the fair allocation rates for other sessions, the rate of the bottleneck connections will be fixed, and a reduced network is considered. In the reduced network, the bottlenecked connections are eliminated and the capacity of all the links (excluding bottleneck links) is reduced by the total bottleneck bandwidth elsewhere. The procedure is repeated in the reduced network until all the sessions have received the fair allocation rates. We could accomplish the above task by keeping track of each session's bottleneck status and computing the advertised rate, γ_l, as follows:

$$\gamma_l = \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}, \tag{5.1}$$

where C_l^A is the available bandwidth for ABR traffic, $\bar{C}_l$ is the sum of the bandwidth of connections bottlenecked elsewhere, N_l is the total number of sessions traverse link l, and $\bar{N}_l$ is the total number of bottlenecked connections elsewhere.

The advertised rate, γ_l, is updated every time a backward RM cell is received at the link. Let t be the time when a backward RM cell is received, and let t^+ be the time of the new update of the advertised rate based on information from the recently received RM cell. Denote $\gamma_l(t)$ as the advertised rate when a backward RM cell is received and $\gamma_l(t^+)$ as the new advertised rate resulted after the update. By incorporating time indices in Equation (5.1) we get

$$\gamma_l(t^+) = \frac{C_l^A - \bar{C}_l(t^+)}{N_l - \bar{N}_l(t^+)}. \tag{5.2}$$

Note that in Equation (5.2), $\bar{C}_l(t^+)$ is the sum of the total bottleneck bandwidth prior to the update, $\bar{C}_l(t)$, and change in the bottleneck bandwidth of the session, $\Delta\lambda$, at the time of update. Similarly, $\bar{N}_l(t^+) = \bar{N}_l(t) + \Delta\beta$, where $\Delta\beta$ represents the change in bottleneck status of the session. The bottleneck status and bottleneck bandwidth of a session is determined by comparing the ER field in the RM cell and the advertised rate of the link. Let β_l^i be an indicator to decide if the session, i, is bottlenecked elsewhere, and λ_l^i be the corresponding bottleneck bandwidth. Based on the explicit rate value, λ_i^{ER}, found on the received RM cell, the variables $\Delta\lambda$ and $\Delta\beta$ are computed as follows:

$$\Delta\lambda = \begin{cases} \lambda_i^{ER} - \lambda_l^i\beta_l^i & \text{if } \lambda_i^{ER} < \gamma_l, \\ 0 - \lambda_l^i\beta_l^i & \text{if } \lambda_i^{ER} \geq \gamma_l, \end{cases} \tag{5.3}$$

$$\Delta\beta = \begin{cases} 1 - \beta_l^i & \text{if } \lambda_i^{ER} < \gamma_l, \\ 0 - \beta_l^i & \text{if } \lambda_i^{ER} \geq \gamma_l. \end{cases} \tag{5.4}$$

Now Equation (5.2) becomes

$$\gamma_l(t^+) = \frac{C_l^A - \bar{C}_l(t) - \Delta\lambda}{N_l - \bar{N}_l(t) - \Delta\beta}. \tag{5.5}$$

Equation (5.5) can be written as

$$\gamma_l(t^+) = \frac{C_l^A - \bar{C}_l(t)}{N_l - \bar{N}_l(t)} + \frac{\frac{C_l^A - \bar{C}_l(t)}{N_l - \bar{N}_l(t)}\Delta\beta - \Delta\lambda}{N_l - \bar{N}_l(t) - \Delta\beta}. \tag{5.6}$$

Note that in Equation (5.6)

$$\frac{C_l^A - \bar{C}_l(t)}{N_l - \bar{N}_l(t)} = \gamma_l(t), \tag{5.7}$$

where $\gamma_l(t)$ denotes the advertised rate just before the update. The new advertised rate, $\gamma_l(t^+)$, can be expressed in terms of the old advertised rate, $\gamma_l(t)$:

$$\gamma_l(t^+) = \gamma_l(t) + \frac{\gamma_l(t)\Delta\beta - \Delta\lambda}{N_l - [\bar{N}_l(t) + \Delta\beta]}. \tag{5.8}$$

5.1.2 ABR Connection Management

The FMMRA algorithm described here requires per-VC accounting, and two per-VC variables are used to keep track of status of all the sessions. A one bit variable, β_l^i, will be used to decide if the session, i, is bottlenecked elsewhere, and the corresponding bottleneck bandwidth is recorded in variable λ_l^i. These variables are referred and updated upon arrival of RM cells.

In addition to the VC table, each queue maintains variables such as the total number of ABR connections using the queue (N_l), the advertised rate (γ_l), the total number of ABR connections bottlenecked elsewhere ($\bar{N}_l$), and bandwidth information obtained via the bandwidth management module. When a connection is opened or closed, the number of ABR connections, N_l, is updated as follows:

$$N_l = \begin{cases} N_l + 1 & \text{if a new VC opens at link } l, \\ N_l - 1 & \text{if an existing VC closes at link } l. \end{cases} \tag{5.9}$$

At the time when the connection opens, the per-VC variables are initialized to zero, and the advertised rate is reduced as follows:

$$\gamma_l = \gamma_l - \frac{\gamma_l}{N_l - \bar{N}_l}. \tag{5.10}$$

When a connection closes, the information regarding this VC should be erased, and the status of the link should be adjusted accordingly. Let j be the connection that was closed. The updating is as follows:

$$\begin{aligned} \bar{N}_l &= \bar{N}_l - \beta_l^j, & (5.11) \\ \gamma_l &= \gamma_l + \frac{\lambda_l^j + \gamma_l \beta_l^j}{N_l - \bar{N}_l}. & (5.12) \end{aligned}$$

5.1.3 Available Bandwidth Management

The bandwidth available for ABR traffic, $C_l^A(t)$, is the difference of link capacity, C_l, and the bandwidth used for guaranteed traffic, $C_l^{GUR}(t)$. In our algorithm we compute the ABR traffic bandwidth as follows:

$$C_l^A(t) = \mu_l \left(C_l - C_l^{GUR}(t) \right), \tag{5.13}$$

where μ_l represents the desired bandwidth utilization factor on link l for ABR traffic. The network manager could set μ_l to 1 to ensure a high link utilization, or set μ_l

slightly below 1 to ensure that the links operate on a congestion avoidance basis, sacrificing link utilization. In a Local Are Network (LAN), it is possible to set this value close to 1 because of the small propagation delays, and consequently the control action can take place fast in case of link overflows.

At each bandwidth update, the advertised rate is adjusted to reflect the change in available bandwidth as follows:

$$\gamma_l(t^+) = \gamma_l(t) + \frac{C_l^A(t^+) - C_l^A(t)}{N_l - \bar{N}_l}. \tag{5.14}$$

In addition to the ABR traffic bandwidth estimation, this routine also computes a load factor, $\rho_l(t)$, which is the ratio of actual ABR traffic bandwidth, $F_l(t)$, and the available bandwidth for ABR traffic. $F_l(t)$ is equivalent to the sum of current cell rates of all the ABR VCs or total flow on the link. The load factor can be computed as

$$\rho_l(t) = \frac{F_l(t)}{C_l^A(t)}. \tag{5.15}$$

The load factor is used to distribute the bandwidth not used by idle sessions to the active sessions. The bandwidth can be estimated by making each queue to measure the incoming cell rate over a fixed "averaging interval."

5.1.4 Explicit Rate Calculation Module

The rate-based approach specifies that a switch should not increase the ER field but could reduce the field to a lower value. The algorithm achieves this by comparing the ER field in the RM cell with the advertised rate, and rewriting the ER field as follows:

$$\lambda_i^{ER} \leftarrow \min(\gamma_l, \lambda_i^{ER}). \tag{5.16}$$

The above assignment is done whenever an RM cell is received at the switch, regardless of the direction of the RM cell. The downstream switches learn the bottleneck status of each connection whenever a forward RM cell is marked by an upstream switch, and the upstream switches learn the bottleneck status of a connection whenever a backward RM cell is marked by a downstream switch. This bi-directional updating of ER in the RM cell is a key feature of FMMRA algorithm, which is not found in any other ER algorithm, and plays a significant role in drastically reducing the convergence time of max-min fair rate allocation process. The advertised rate is updated only when a backward RM cell is received, since this RM cell has been seen by all the switches along its path, and the fields contain the most complete information about the status of the network.

At the reception of a backward RM cell, the change of status of the connection is determined by calculating the change in per-VC variables, $\Delta\lambda$ and $\Delta\beta$, using Equation (5.4), and the new advertised rate is calculated as follows:

$$\gamma_l = \begin{cases} C_l^A & \text{if } N_l = 0, \\ \gamma_l + \frac{\gamma_l \Delta\beta - \Delta\lambda}{N_l - [N_l + \Delta\beta]} & \text{if } N_l > \bar{N}_l, \\ \gamma_l & \text{if } N_l = \bar{N}_l. \end{cases} \tag{5.17}$$

In contrary to the algorithm presented in [33], which requires the switch inspecting all the sessions and calculate fair rate, the FMMRA algorithm only requires the knowledge of the session which is seen by the switch at the time of update. This feature makes the computational complexity of FMMRA to be of $O(1)$, whereas the algorithm in [33] has a computational complexity of $O(N_l)$.

Next, the new number of bottlenecked connections is updated as follows:

$$\bar{N}_l = \begin{cases} \bar{N}_l + 1 - \beta_l^i & \text{if } \lambda_i^{ER} < \gamma_l, \\ \bar{N}_l - \beta_l^i & \text{if } \lambda_i^{ER} \geq \gamma_l. \end{cases} \tag{5.18}$$

Finally, once the explicit rates are marked on any backward RM cell and the port variables are updated, the switch updates the corresponding per-VC variables in the VC table as follows:

$$\beta_l^i = \begin{cases} 1 & \text{if } \lambda_i^{ER} < \gamma_l, \\ 0 & \text{if } \lambda_i^{ER} \geq \gamma_l, \end{cases} \tag{5.19}$$

$$\lambda_l^i = \begin{cases} \lambda_i^{ER} & \text{if } \lambda_i^{ER} < \gamma_l, \\ 0 & \text{if } \lambda_i^{ER} \geq \gamma_l. \end{cases} \tag{5.20}$$

5.1.5 Congestion Detection and Buffer Management

It is essential to note that the fair rate assignment as in Equation (5.16) is conservative, since if any session is idle, the link will not be fully utilized. Moreover, there are no mechanisms specified to control the queue growth. We provide some enhancements in this section to provide control over buffer utilization.

The FMMRA algorithm takes advantages of the CCR field found in the RM cell, bottleneck status of the session, the utilization factor, ρ_l, and the queue length. As we have described before, ρ_l is computed every time the ABR traffic bandwidth is estimated. The load factor reflects how well the ABR bandwidth is utilized, for example, $\rho_l < 1$ reflects that some of the sessions are sending cells at a rate less than their allowed rate. This presents an opportunity for the non-bottlenecked sessions in link l to increase their rate. The new explicit rate assignment operation is formulated by modifying Equation (5.16) as follows:

$$\lambda_i^{ER} \leftarrow \min\left\{\max\left(\frac{\hat{\lambda}_l^i(1-\beta)}{\rho_l}, \gamma_l\right), \lambda_i^{ER}\right\}, \tag{5.21}$$

where $\hat{\lambda}_l^i$ denotes the last seen CCR value for connection i at link l at the time of estimation of ρ_l, and β is the status of the connection.

It is also necessary to control the queue growth to prevent potential cell loss. If the queue length reaches a *Low-Threshold* (Q_{LT}) and the load factor, $\rho_l > 1$ (i.e, input rate is larger than the available capacity), the operation as in Equation (5.21) is turned off, and the assignment as in Equation (5.16) is turned on. This ensures that whenever a potential for congestion is detected, even if some sessions are idle, the non-idle sessions are not given any extra bandwidth, which allows the queues to drain. Furthermore, if the queue length is above a *High-Threshold*, Q_{HT}, (indicating heavy congestion) the target utilization factor is reduced by a *Target Rate Reduction Factor* (*TRRF*), until the queue length drops below the low threshold, Q_{LT}. This technique allows the queue to drain, and operate at a desired queue level, whenever the switch is heavily congested, which may happen during opening of new connections.

The use of CCR also ensures that the algorithm is interoperable in an environment consisting of simple EFCI switches or switches employing a different ER algorithm. A similar approach is taken in [35], however the ERICA algorithm does not use the bottleneck status of the session, and the load factor is used regardless of the queue length. The Q_{LT}, Q_{HT}, and $TRRF$ can be set as a function of round trip delays in the network. These issues will be addressed in a future contribution.

5.2 Convergence Analysis

In this section we provide mathematical analysis to show convergence, and develop an expression to calculate an estimate of convergence time for the FMMRA algorithm. Let us denote λ_j^* as the optimal max-min fair allocation rate for session j. Denote $\Gamma^* = [\gamma^1, \gamma^2, \cdots, \gamma^M]$ to be a vector where the elements of vector Γ^* are arranged in an increasing order and represent the M distinct values of max-min fair rates of a given ATM network and for the given set of sessions. Let $\mathcal{S}^m$ be the set of VCs with a max-min fair allocation of γ^m. Let us denote $\mathcal{L}_j^*$ as the set of links which are bottleneck for session j, and let $\mathcal{L}^m$ be the set of links which are bottleneck for sessions receiving γ^m as their fair allocation rate. We assume that all the sessions are greedy, and they send the cells at the maximum allowed rate.

Definition 2 *A saturated session, j, is a session that has stabilized to a rate allocation equal to its fair rate, λ_j^*, and the VC j saturates link l if*

$$\lambda_j^* = \gamma_l = \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}. \tag{5.22}$$

Based on the above definition, let us denote $\mathcal{S}_l^*$ as the set of sessions saturated on link l.

We should note that whenever a session is bottlenecked on a link, it does not mean that the session has saturated since the session may be bottlenecked on a link temporarily. If the session has reached saturation, however, the session is said to be bottlenecked. For the purpose of analysis, denote the total saturation bandwidth as C_l^*, which is the total bandwidth occupied by the saturated sessions on link l, as expressed below.

$$C_l^* = \sum_{j \in S_l^*} \lambda_j^*. \tag{5.23}$$

Although the quantity C_l^* is not used explicitly in the rate allocation algorithm, it helps us to study the behavior of γ_l.

5.2.1 Proof of Convergence

The FMMRA algorithm is shown to converge by showing that the sessions with the minimum fair rate allocation saturate first, and then subsequently all the sessions saturate. Furthermore, the session that saturate will remain saturated. In order to prove these arguments we formulate the following three lemmas.

Lemma 1 *The session with the minimum fair allocation rate, γ^1, and the smallest round trip time among the sessions in set S^1 will saturate when its first RM cell is returned back to the source.*

Lemma 2 *If a session $j \in S^m$ saturates at time t_j^m with a fair allocation rate of γ^m, then*

1.) The advertised rate, $\gamma_l(t)$, for $t \geq t_j^m$ satisfies the following

$$\gamma_l(t) \geq \frac{C_l^A - C_l^*(t_j^m)}{N_l - N_l^*(t_j^m)}, \tag{5.24}$$

2.) all the unsaturated sessions with γ^m as their fair rate allocation and sharing one of j's bottleneck link (i.e., $l \in \mathcal{L}_j^*$) , become saturated, within the time when the RM cells of each session completes a round trip, and

3.) the session j remains saturated for any time $t > t_j^m$, and session j remains marked as bottlenecked elsewhere for any time $t > t_j^m$ on all links $l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}$ with the bottleneck bandwidth of γ^m.

Lemma 3 *Given that there are a set of unsaturated sessions in the network, a session $j \in \mathcal{S}^m$, $2 \leq m \leq M$ will saturate once the sessions $i \in \mathcal{S}^{m-1}$ saturates.*

The proofs of Lemmas 1, 2 and 3 are given in the appendix. Based on these three lemmas we formulate the following theorem.

Theorem 1 *Given arbitrary initial conditions, the FMMRA algorithm converges in finite time, and subsequent iterations of the algorithm do not modify the optimal, max-min fair rates.*

Proof: The proof of Theorem 1 directly follows from Lemmas 1-3. By Lemma 1, the session with the minimum fair allocation rate, γ^1, and the smallest round trip time among the sessions in set $\mathcal{S}^1$ will saturate at the end of its first round trip time. By Lemma 2 all the other sessions in the set $\mathcal{S}^1$ will saturate, and all the saturated sessions remain saturated. From Lemma 3, the sessions in $\mathcal{S}^2$ will saturate once the sessions in the set $\mathcal{S}^1$ are saturated. Repeating Lemmas 2 and 3 for all the other unsaturated sessions in the set, $\mathcal{S}^m$, $2 < m \leq M$ will saturate. When all the sessions are saturated, the bottleneck status and the bottleneck bandwidth do not change, and thus the advertised rates on all the links will remain unmodified. Therefore, the optimal rates will not change. This concludes the proof of Theorem 1.

5.2.2 Rate of Convergence

It is very difficult to predict an exact convergence time, since the algorithm operates in distributed fashion, and difficult to predict the traffic characteristics. It is useful, however, to find an upperbound on covergence time.

From Lemmas 1 and 2, all the sessions in the set $\mathcal{S}^1$ will saturate at the completion of their first round trip time (i.e., T^1 time units). Similarly from Lemma 3, the session in $\mathcal{S}^2$ will saturate T^2 time units after sessions $\mathcal{S}^1$ saturate and so forth. Then the total convergence time is $\sum_{m=1}^{M} T^m$. This implies that the convergence time is proportional to the number of distinct fair allocation rates and the round trip time delays. In order to give an approximate value of an upperbound let us assume that the worst case round trip time delay is D, and let M be the distinct fair allocation rates. We can conclude that the algorithm will converge to the fair allocation rates approximately within MD time units.

The delays associated with the quantity D includes propagation delay, transmission delay, queuing delays and RM cell generation interval. The RM cell generation interval is critical, since the opportunity to increase rates are controlled at the time of an RM cell's return. An RM cell is generated at the source every N_{RM} cells and the time duration between successive RM cells depends on ACR. The ACR computation depends on the specified source behavior. The algorithm specified here, allows the sources to set their ACR values to the ER value specified in the RM cell, which allows the sources to respond quickly to the feedback. In the worst case, RM cell will be generated within $\frac{N_{RM}}{\lambda_{ER}}$. The RM cell generation interval is significant when the propagation delays are small and negligible when there is large propagation delays. It can be seen from the simulation results that the algorithm converges to the fair allocation rates within a time which is much lower than MD time units.

5.3 Simulations and Results

5.3.1 Performance Comparison in a Single-Hop Network

A network topology, shown in Figure 5.2, is simulated using EPRCA, ERICA, and FMMRA switch algorithms. The network consists of five sessions and two ATM switches. For performance measurement purposes, the switches are assumed to be non-blocking and output buffered. The sources are assumed to be well behaved, persistently greedy, and always transmit at the maximum allowed cell rate. The use of persistent sources presents a tough challenge for congested links. For all the simulations the buffer size is characterized in order to achieve zero cell loss. We use this network model to compare the performance of various algorithms in Local Area Network (LAN) and Wide Area Network (WAN) configurations. In a LAN configuration, all the links are 1Km in length. In a WAN configuration, the distances of links L1 and L12 are 1000Km and 100Km, respectively. In both cases, all the links have 100Mbps capacity and a propagation delay of $5\mu s$ per Km.

For both cases we consider the situation where some connections are bottlenecked. A bottlenecked connection is a connection which cannot increase its transmission rate due to certain network conditions or the lack of network resources. In our simulations, we make sessions 1 and 3 bottlenecked by setting the PCR value of the sources S1 and S3 to $5Mbps$. This implies that the session 2 can use 90Mbps on link L12 until $t = 100ms$. At any time $t > 100ms$, sessions 2, 4 and 5 should receive equal share of bandwidth on link L12. Thus, the steady state bandwidth share for sessions 1 and 3 is $5Mbps$, and for sessions 2,4 and 5 is $30Mbps$.

From Figures 5.3 and 5.11, it can be seen that, although the EPRCA converges to fair rates, significant level of oscillations are found at steady state, especially in the WAN configuration. It was observed that the rate of sessions receiving higher bandwidth have very large oscillations than the rates of sessions receiving lower bandwidth. Note that in these simulations an $AIR = 0.1Mbps$ value is used, implying that the source may increase the allowed cell rate by $3.2Mbps$ every time an RM cell arrives at the source. In a LAN it is desired, however, to allow the sources to increase their rates to their fair share almost instantly by using a higher value of AIR, that can increase ACR up to PCR or ER instantly. It was observed that using a higher value of AIR with the EPRCA, results in poor convergence, very large oscillations and poor link utilization. Moreover, the selection of threshold values is also very important to achieve good performance. Improper selection of threshold values results in poor link utilization, especially in a WAN configuration.

Thus, the EPRCA significantly limits the network to operate aggressively (i.e., use of high AIR) and to be robust (i.e., selection of parameters).

From Figures 5.5 and 5.13, it can be seen that the ERICA algorithm fails to converge to the fair rates, for both LAN and WAN environments. The sessions which start late do not get their fair share of 30Mbps, instead they get a share of 20Mbps. This is due to the fact that the ERICA algorithm does not calculate the exact fair rate. Furthermore, the queue levels grow without a bound, and it is necessary to set the desired target utilization below 100% to avoid such a large growth. Even using a lower target utilization factor in the WAN configuration (95%), results in very high peak queue length, as illustrated in Figure 5.14.

Both the FMMRA algorithm with and without buffer control results in oscillation free steady state, as shown in Figures 5.7, and 5.15. In the WAN model it can be seen that the FMMRA algorithm provides significant improvement of convergence time over the EPRCA. The advantage of using buffer control can be seen in the WAN configuration model where the FMMRA algorithm without buffer control results in a steady state queue length of about 240 cells. On the other hand the FMMRA algorithm with buffer control has a steady state queue length of zero. It is observed that the FMMRA algorithm converges to the correct fair rates without any oscillations regardless of the AIR used. In fact, using a very high AIR value results in much faster convergence.

5.3.2 Measurement of Worst Case Convergence Time

In the analysis above an upperbound for convergence time is derived. To find out the worst case convergence time the network shown in Figure 5.19 is simulated. The network consists of 3 sources, a switch and 3 receivers. The link connecting the two switches is $1000Km$ in length, and has a bandwidth of 150Mbps. All the other links have a distance of 10 meters. The links connecting the sources to the switch SW1

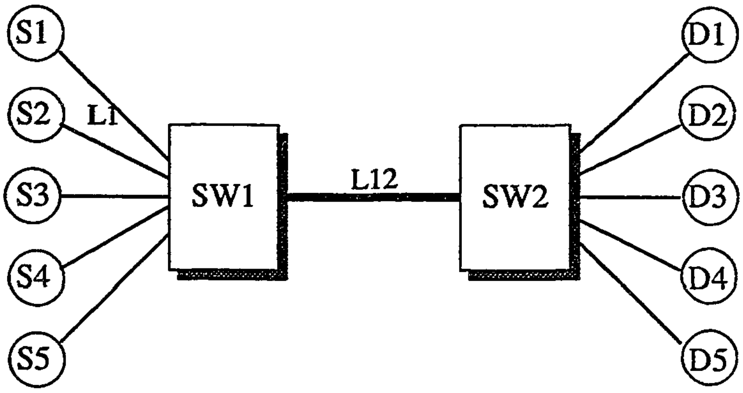

Figure 5.2 The Single-Hop Network

Table 5.1 Single-Hop Network Simulation Parameters

Parameter	EPRCA LAN	EPRCA WAN	ERICA LAN	ERICA WAN	FMMRA LAN	FMMRA WAN
N_{RM}	32	32	32	32	32	32
$MCR(Mbps)$	2	2	2	2	2	2
$ICR(Mbps)$	5	5	5	5	5	5
$AIR(Mbps)$	0.1	0.1	0.1	0.1	0.1	0.1
RDF	256	256	256	256	256	256
Q_T	30	100	N/A	N/A	30	50
DQT	60	150	N/A	N/A	50	100
μ	N/A	N/A	100%	95%	100%	100%
$TRRF$	N/A	N/A	N/A	N/A	5%	5%
VCS	0.875	0.875	N/A	N/A	N/A	N/A
DPF	0.875	0.875	N/A	N/A	N/A	N/A
AV	0.0625	0.0625	N/A	N/A	N/A	N/A
ERF	0.94	0.94	N/A	N/A	N/A	N/A
MRF	0.25	0.25	N/A	N/A	N/A	N/A

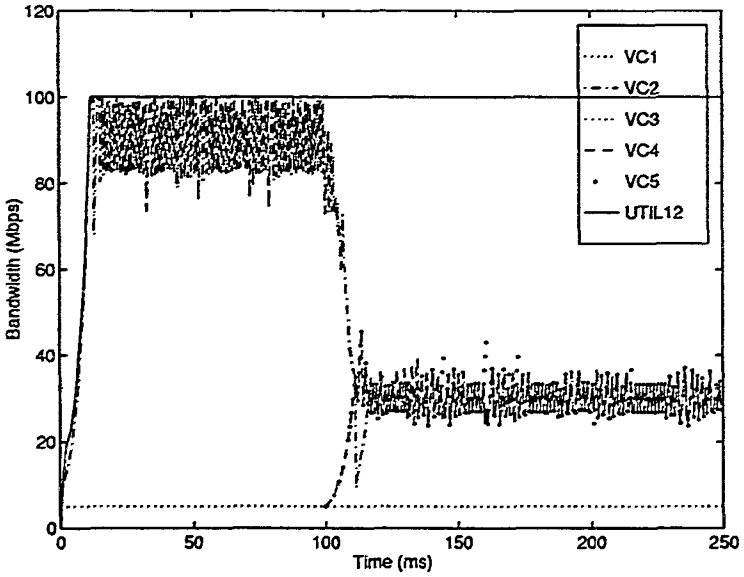

Figure 5.3 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck LAN Configuration - EPRCA Algorithm

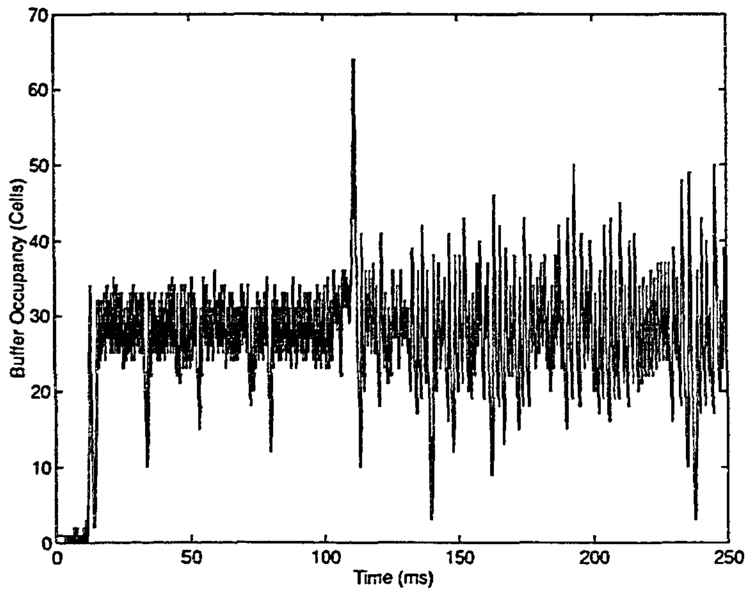

Figure 5.4 Instantaneous Queue Length in Single-Hop Bottleneck LAN Configuration - EPRCA Algorithm

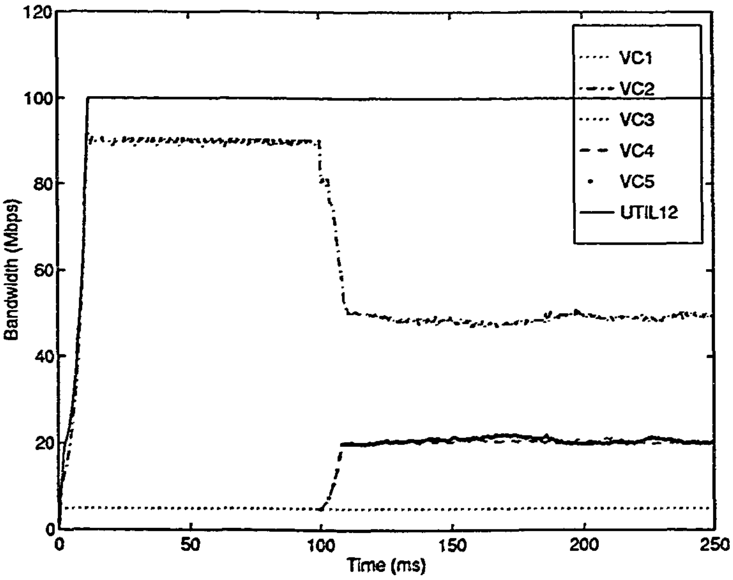

Figure 5.5 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck LAN Configuration - EPRCA Algorithm

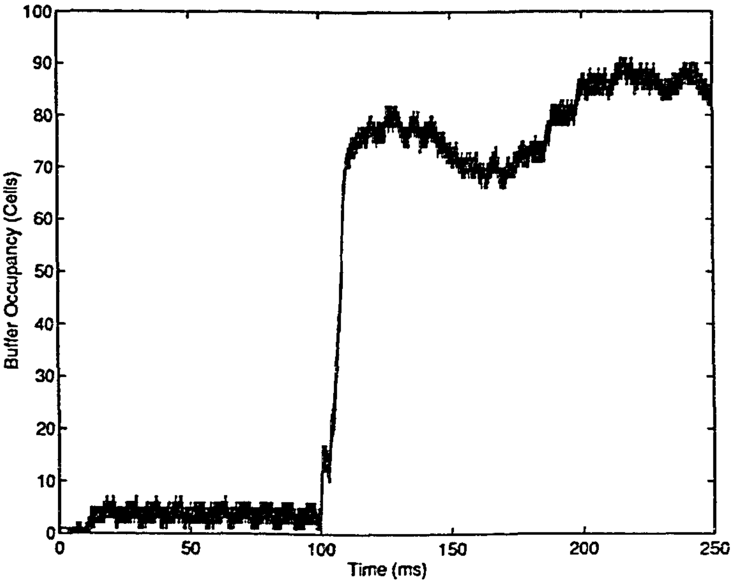

Figure 5.6 Instantaneous Queue Length in Single-Hop Bottleneck LAN Configuration - ERICA Algorithm

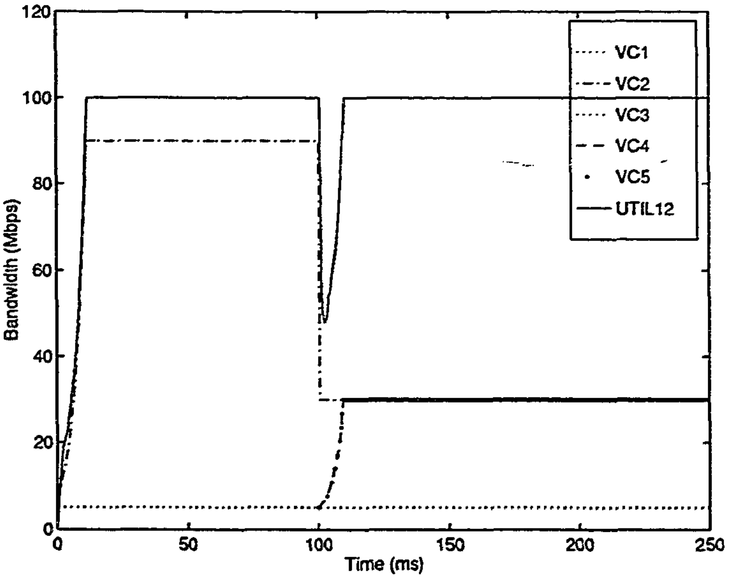

Figure 5.7 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck LAN Configuration - FMMRA Algorithm without Buffer Control

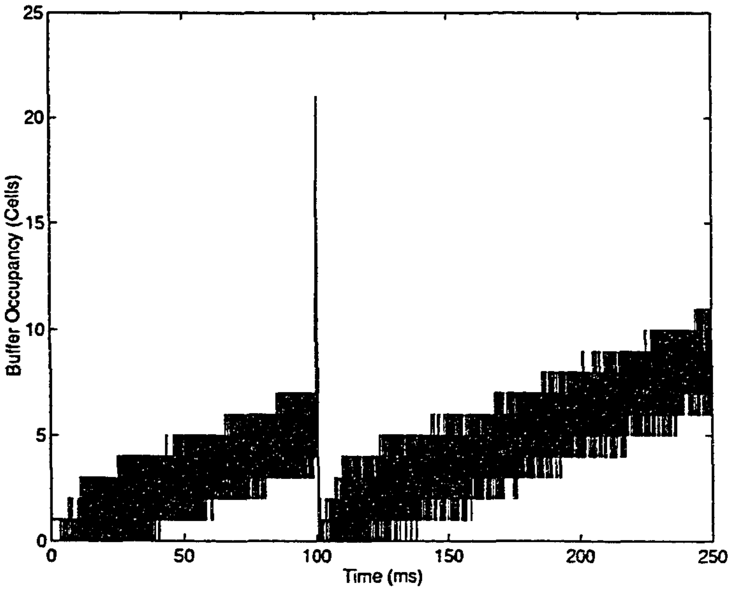

Figure 5.8 Instantaneous Queue Length in Single-Hop Bottleneck LAN Configuration - FMMRA Algorithm without Buffer Control

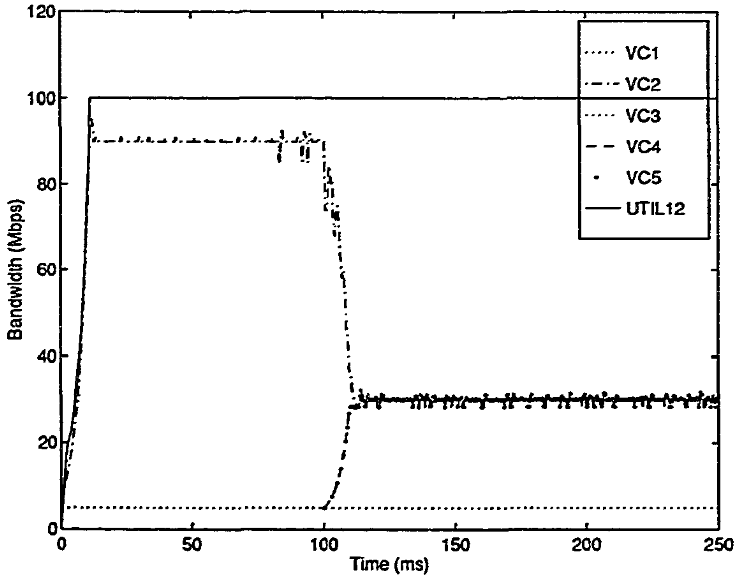

Figure 5.9 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck LAN Configuration - FMMRA Algorithm with Buffer Control

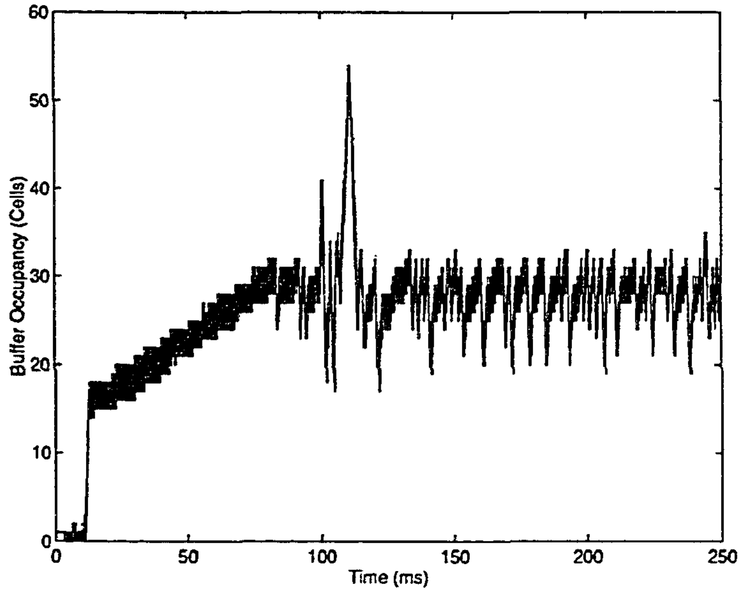

Figure 5.10 Instantaneous Queue Length in Single-Hop Bottleneck LAN Configuration - FMMRA Algorithm with Buffer Control

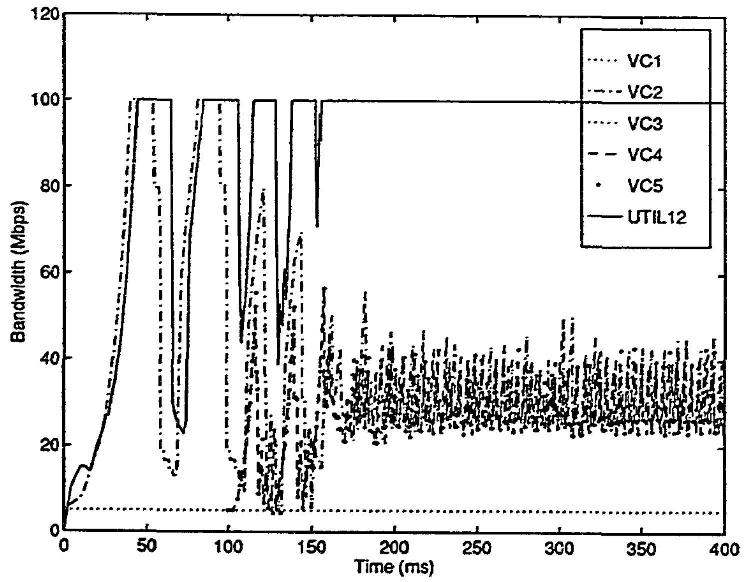

Figure 5.11 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck WAN Configuration - EPRCA Algorithm

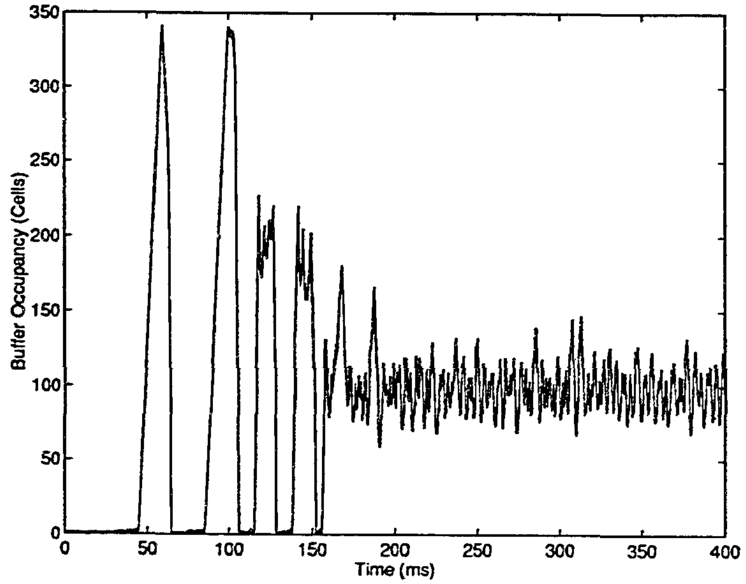

Figure 5.12 Instantaneous Queue Length in Single-Hop Bottleneck WAN Configuration - EPRCA Algorithm

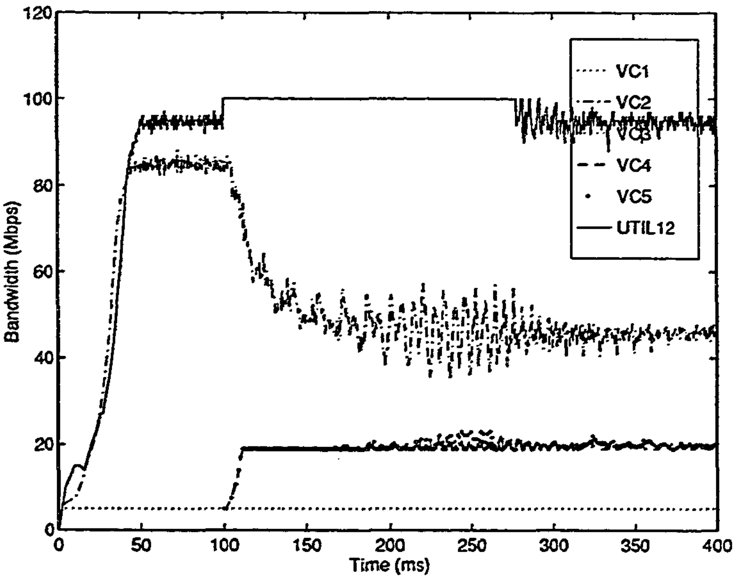

Figure 5.13 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck WAN Configuration - ERICA Algorithm

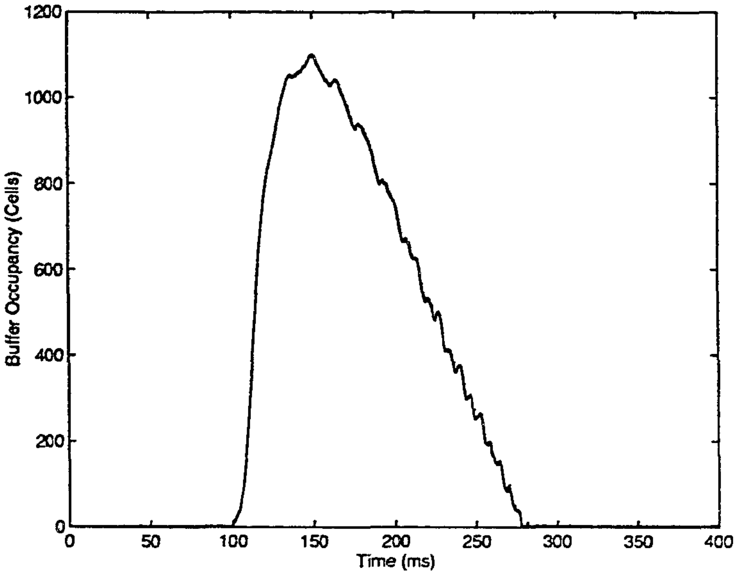

Figure 5.14 Instantaneous Queue Length in Single-Hop Bottleneck WAN Configuration - ERICA Algorithm

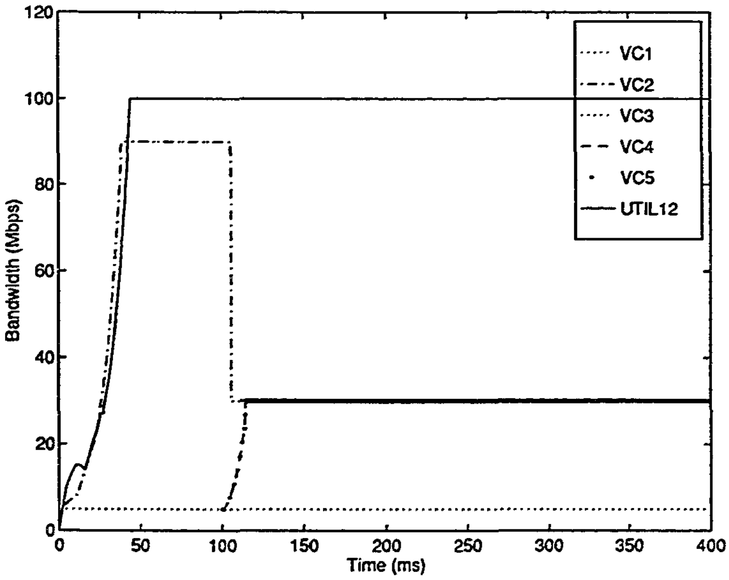

Figure 5.15 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck WAN Configuration - FMMRA Algorithm without Buffer Control

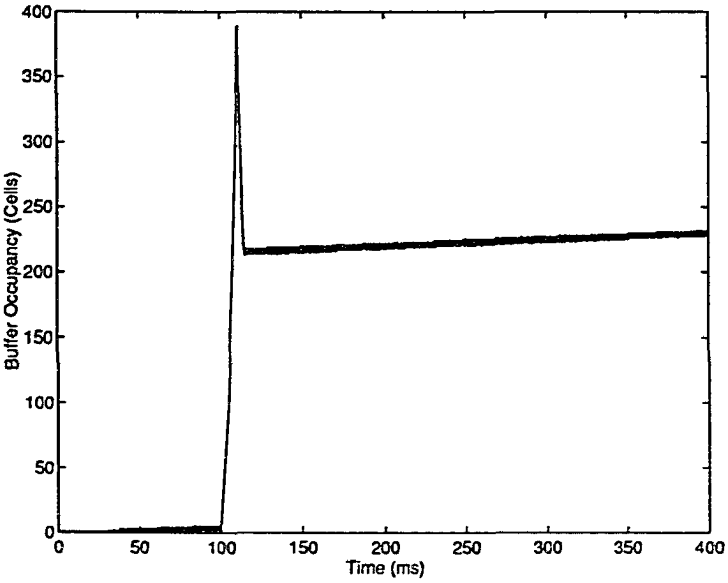

Figure 5.16 Instantaneous Queue Length in Single-Hop Bottleneck WAN Configuration - FMMRA Algorithm without Buffer Control

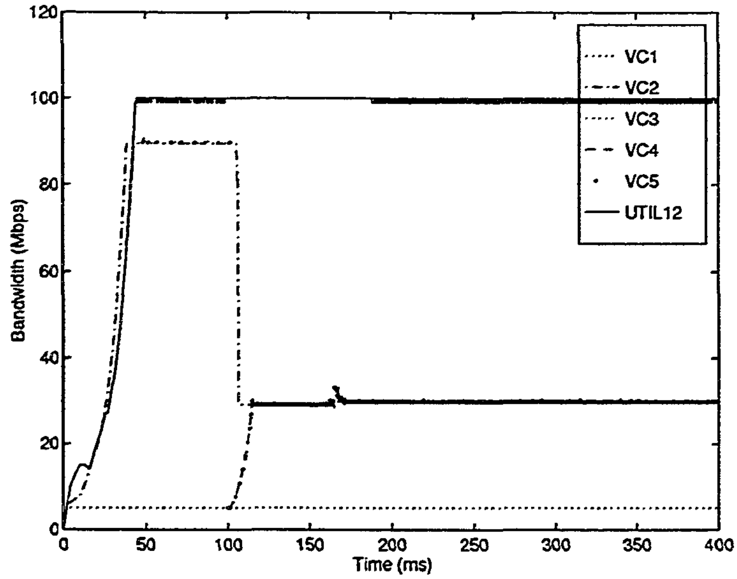

Figure 5.17 Instantaneous Bandwidth Utilization in Single-Hop Bottleneck WAN Configuration - FMMRA Algorithm with Buffer Control

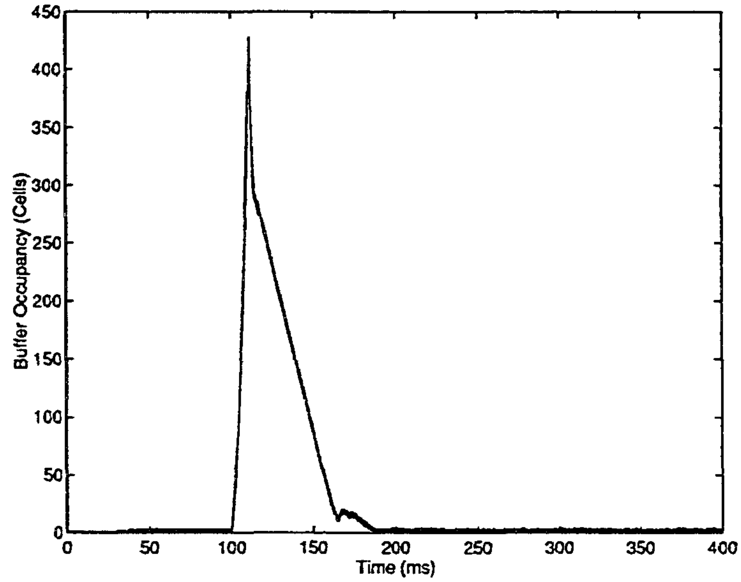

Figure 5.18 Instantaneous Queue Length in Single-Hop Bottleneck WAN Configuration - FMMRA Algorithm with Buffer Control

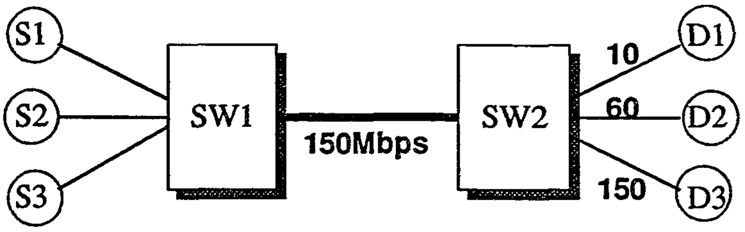

Figure 5.19 Configuration to Illustrate Worst Case Convergence Time

have 150Mbps link capacity. The capacities of the links are fixed such that there are three distinct fair allocation rates (i.e., VC1 = $10Mbps$, VC2 = $60Mbps$, and VC3 = $80Mbps$). The worst case round-trip time in this network is approximately $10ms$. Thus, the upperbound of convergence time is $30ms$. The instantaneous bandwidth allocation for each session against time is presented in Figure 5.20. From the simulation results it can be seen that the convergence time is about $23ms$, which is significantly lower than the predicted convergence time.

5.4 Summary and Conclusions

In this chapter we have presented and evaluated a fair, fast adaptive rate allocation algorithm designed for ATM switches implementing explicit rate congestion control supporting ABR services. The most important features of this algorithm are $O(1)$ computational simplicity, fast convergence, oscillation free steady state, high link utilization and low buffer requirements. The algorithm requires per-VC accounting, that enables the switches to calculate an exact fair rate and provides the means for quick convergence.

The algorithm is robust in the sense that any loss of RM cells does not affect the functionality, or stability of the algorithm. However, it is obvious that such losses can affect the response time of the sources. Furthermore, the algorithm does not require any parameter tuning, which is a desirable feature because, as the size

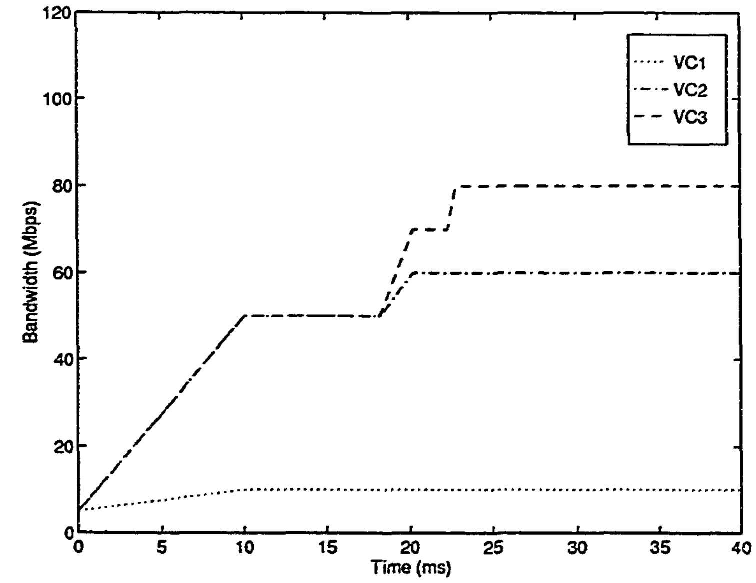

Figure 5.20 Allowed Cell Rate vs. Time: Illustration of Worst Case Convergence Time

of the network grows, there is no need to reconfigure the parameter values or worry about any incorrect settings of various parameters.

The performance of the algorithm is evaluated in terms of fairness, throughput, link utilization and buffer utilization by simulating a benchmark network topology. Simulation results validate that the algorithm operates effectively under wide range of traffic scenarios, and offers significant advantages over the EPRCA and ERICA algorithms.

CHAPTER 6

PERFORMANCE OF FMMRA ALGORITHM IN CHALLENGE CONFIGURATIONS

At the July, 1994 meeting of the Traffic Management Group, some simulation configurations were specified as suitable fairness tests for ABR flow control methods [48]. In this chapter some of these models are simulated and the results are presented. Some of the simulation configurations used are taken from [49]. The simulations are intended to show the effectiveness of the FMMRA algorithm under a wide range of network conditions and topology configurations.

6.1 Performance in a Multi-Hop Network

A multi-hop network, shown in Figure 4.3, is simulated. This network configuration is referred to as the Generic Fairness Configuration 1 (GFC1). The GFC1 network consists of five switches and 23 connections grouped into six classes (A-F). In Figure 6.1 the number inside the parentheses next to the group label represents the number of VCs for that group. VCs in groups C, D, E, and F are single-hop traffic and VCs in groups A and B are three-hop cross-traffic. The links connecting hosts to switches have a capacity of 150Mbps. The expected max-min fair allocation rates (in *Mbps*) for the VCs from groups A - F and their bottleneck links are shown in Table 6.1.

For performance measurement purposes, the switches are assumed to be non-blocking and output-buffered. The sources are assumed to be well behaved, persistently greedy, and they always transmit at the maximum allowed cell rate. The use of persistent sources presents a tough challenge for multiple congested links.

The first simulation in this experiment is intended to illustrate the "beat-down" problem experienced by the EFCI switches. The GFC1 network, with all EFCI switches, is simulated and the results are shown in Figure 6.2. In Figure 6.2

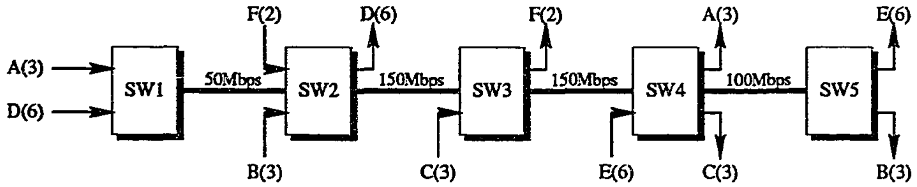

Figure 6.1 The GFC1 Network

the received number of cells for an arbitrarily chosen VC in each group is plotted against time. The slopes of the line segments represent the average throughput for each VC. From the figure it can be seen that marking only the EFCI bit leads to unfairness among classes. In particular, connections in Groups A and B, which compete for bandwidth in multiple links, receive a lower than the max-min fair share, while connections in the other classes take advantage of this situation and receive more than their fair share.

The same GFC1 network, with all ER switches employing the FMMRA algorithm, is simulated next. In order to show the algorithm's quick convergence time, we set the AIR values at the source very high (e.g., $AIR = 5$). This ensures that the source can transmit at the rate equal to the ER value in the received RM cell. The resulting ACR value versus time is plotted in Figure 6.6. From this figure it can be seen that the convergence is very fast. Also note that the ACR values do not oscillate at steady state. This feature allows the network to operate under stable conditions. Using a higher AIR value for an algorithm which is based on exponential averaging results in poor convergence, and very large rate-oscillations. In addition, allowed cell rates, the plot showing received number of cells versus time is also shown in Figure 6.7. From these figures it can be seen that, regardless of their geographic location and number of hops traveled, all VCs obtain their fair share very quickly while maximizing the throughput.

Table 6.1 GFC1 - Max-Min Fair Rates

VC Group	Max-Min Rate (Mbps)	Bottleneck Link
A	5.56	SW1-SW2
B	11.11	SW4-SW5
C	33.33	SW3-SW4
D	5.56	SW1-SW2
E	11.11	SW4-SW5
F	50.00	SW2-SW3

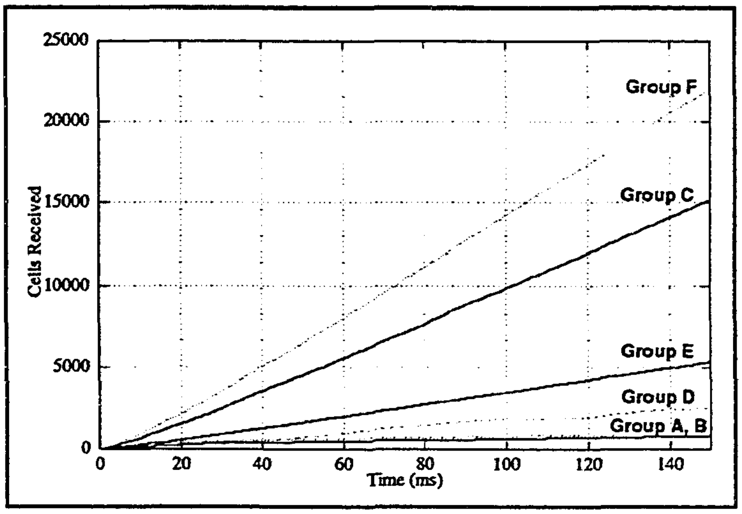

Figure 6.2 Fairness Comparison in GFC1 Network with All EFCI Switches

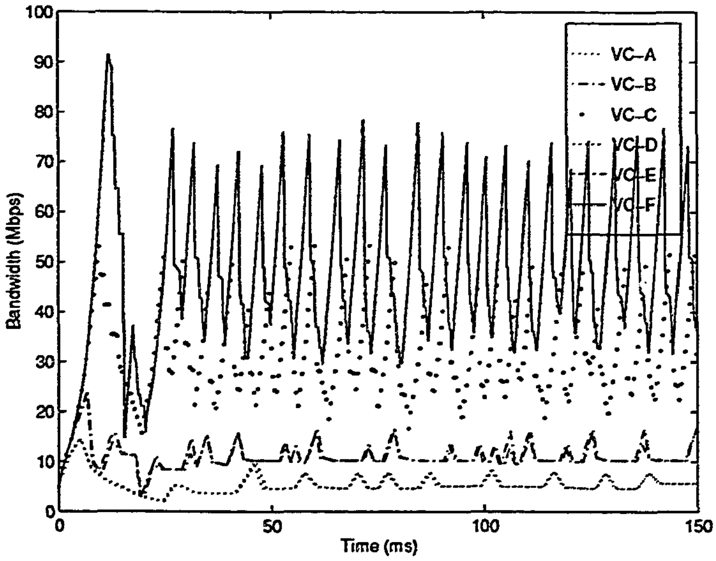

Figure 6.3 Instantaneous Bandwidth Utilization in GFC1 Network: EPRCA Algorithm

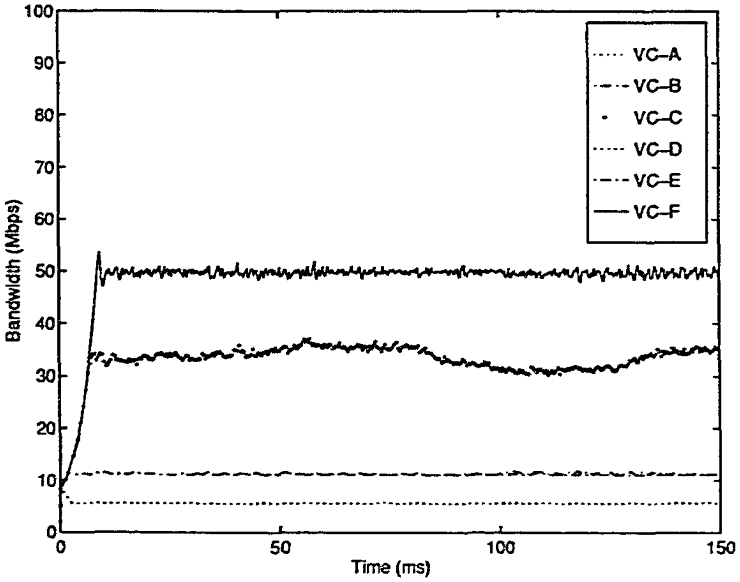

Figure 6.4 Instantaneous Bandwidth Utilization in GFC1 Network: ERICA Algorithm

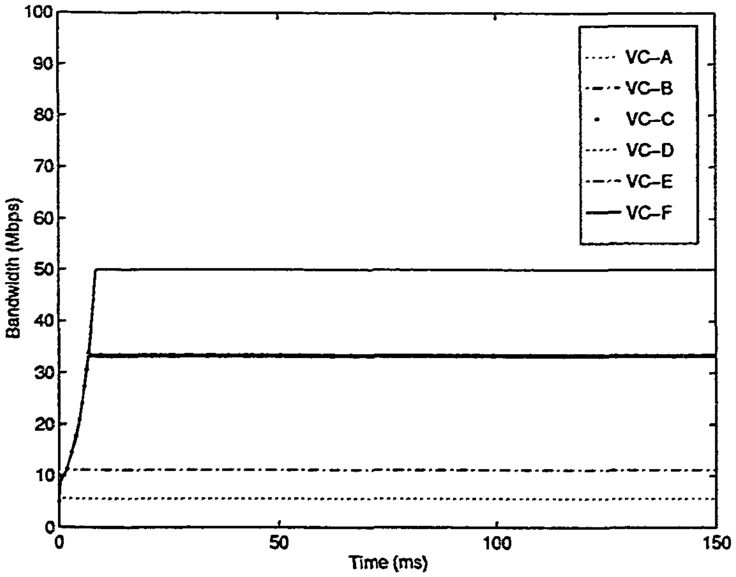

Figure 6.5 Instantaneous Bandwidth Utilization in GFC1 Network: FMMRA Algorithm

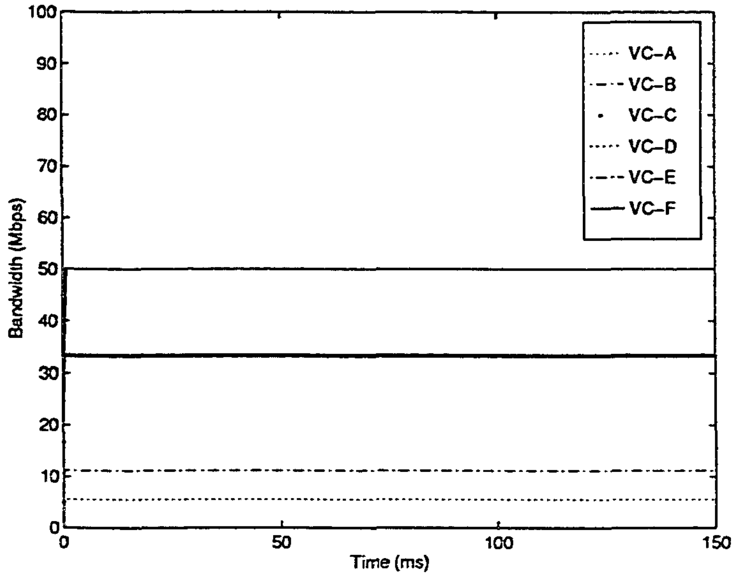

Figure 6.6 Instantaneous Bandwidth Utilization in GFC1 Network: FMMRA Algorithm with $AIR = 5Mbps$

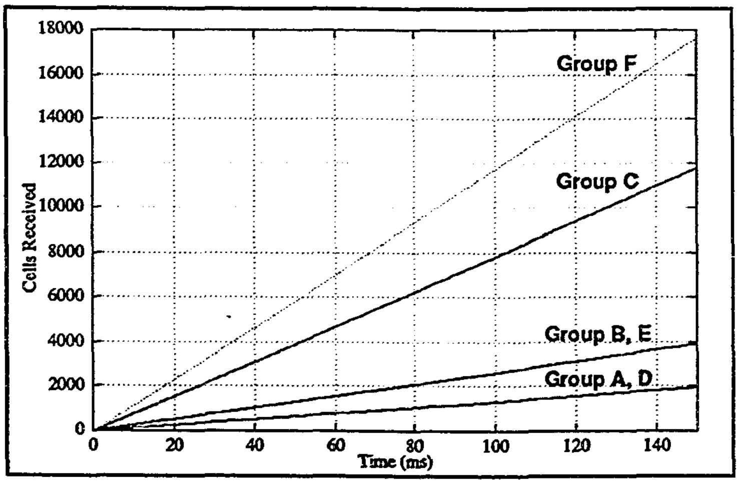

Figure 6.7 Received Number of Cells vs. Time in GFC1 Network: FMMRA Algorithm

Table 6.2 GFC3: The Network Parameters

Link	LA	LB	LC	LD	LE	LF	LG	LH	LI	LJ	L-SD
Bandwidth ($Mbps$)	150	100	50	1.5	150	100	1.5	150	50	100	150
Distance (Km)	10	10	10	10	0.01	5	5	2	2	0.01	1

6.2 Varying Propagation Delays and Link Speeds

The network topology as in Figure 6.8, which varies widely in terms of link bandwidth and propagation delays, is simulated. This network consists of one switch and 11 hosts. The bandwidth of the input links varies from $1.5Mbps$ to $150Mbps$, and the link distances vary from $0.01Km$ to $10Km$. The network characteristics are tabulated in Table 6.2. The results in Figures 6.9 and 6.10 show that the FMMRA algorithm works correctly in achieving the fair rates.

Table 6.3 Fair Rates for GFC3

VC	A	B	C	D	E	F	G	H	I	J
Fair Rate ($Mbps$)	18.4	18.4	18.4	1.5	18.4	18.4	1.5	18.4	18.4	18.4

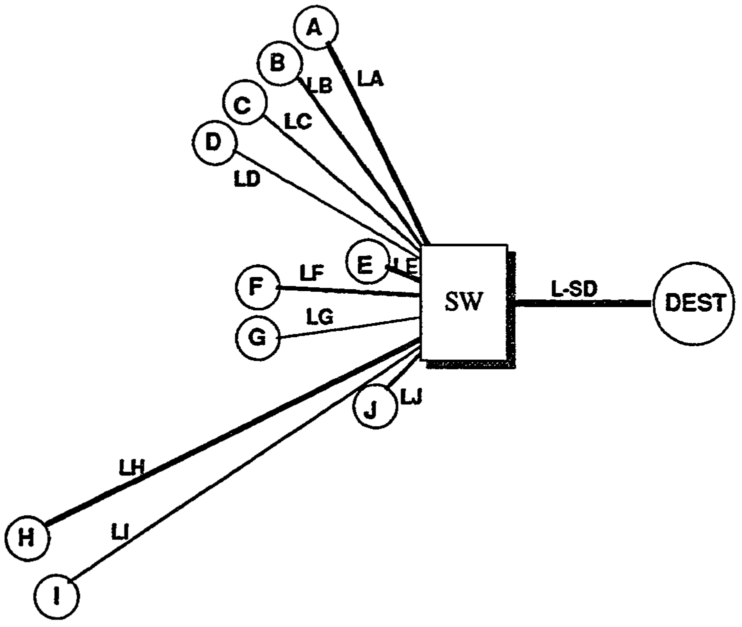

Figure 6.8 GFC3: The Network Configuration with Wide Disparity in Link Speeds and Distances

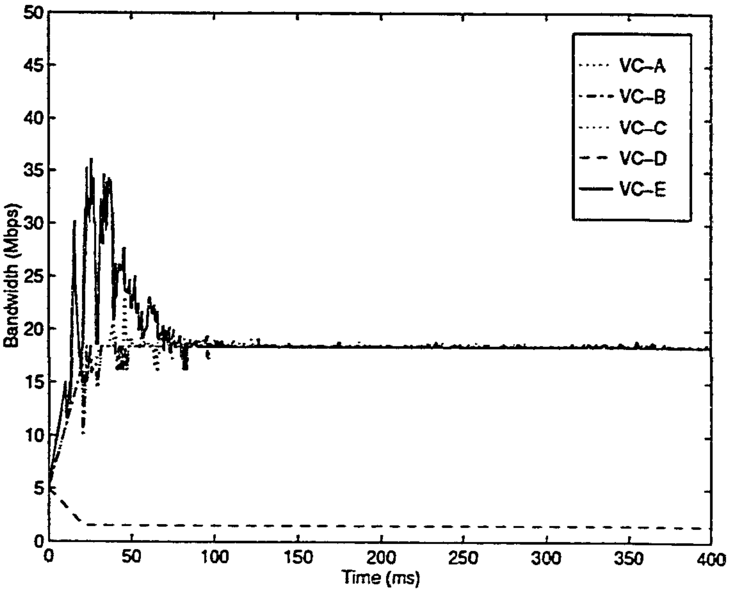

Figure 6.9 Instantaneous Bandwidth Utilization in GFC3 Network (Part 1): FMMRA Algorithm

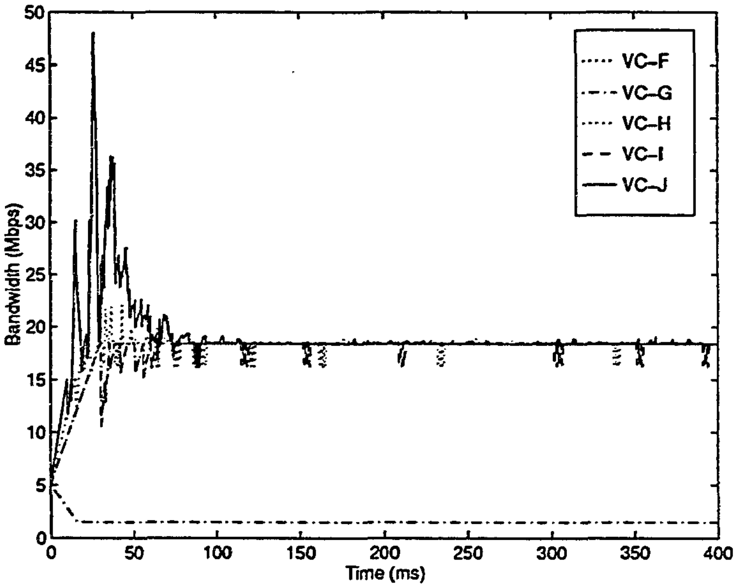

Figure 6.10 Instantaneous Bandwidth Utilization in GFC3 Network (Part 2): FMMRA Algorithm

CHAPTER 7

MINIMUM CELL RATE REQUIREMENTS AND FAIRNESS

In the ABR service definition, fairness is defined by the "max-min" criterion. This criterion may not fit well with the "minimum cell rate" (MCR) requirements defined in the ABR standard. The algorithm proposed previously does not take into account the MCR requirements. A new fairness definition must be adopted in order to support non-zero MCR requirements. In [43], many possible modifications to the fairness definition in the ABR service model are proposed. In this chapter, the issue of MCR requirements, a new possible fairness criterion, and a modification of the FMMRA are presented.

7.1 Modification of Max-Min Criterion

In the new ABR service definition, the fairness is defined according to max-min criterion. Basically, the max-min criterion states that fairness is achieved if each connection gains an equal share of its bottleneck bandwidth, which can be expressed as

$$\text{Fair Share} = \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}, \qquad (7.1)$$

where C_l^A is the ABR traffic bandwidth, $\bar{C}_l$ is the total bottlenecked bandwidth, N_l is the total number of connections, and $\bar{N}_l$ is the total number of bottlenecked connections elsewhere. The above fairness definition was defined without considering the MCR, as in the ABR service definition. If we simply apply max-min criterion to the ABR service, we may have undesired results. The problem becomes obvious when considering the explicit rate control mechanisms, where the above expression is used to determine the rate of each connection.

Let us consider the following simple example. A bottleneck link is shared by 4 connections. The first connection has an MCR equal to 1/2 of the link bandwidth and the other three connections have zero MCR. By the max-min criterion, each connection should have a share of 1/4 of the bandwidth. By the ABR service definition, however, the first connection can send traffic at a rate at least equal to 1/2 of the link bandwidth. The network control information informs each connection of how fast each one can send traffic according to the max-min criterion. Thus, the offered load can be 1.25 (i.e., first connection = 1/2 of the bandwidth; the other three = 1/4 of the bandwidth each), and this overload leads to congestion.

It is apparent that a more precise fairness definition is needed. A clear definition of fairness also helps the development of the control mechanism to support the ABR service.

In considering fairness for ABR service, the guaranteed bandwidth, MCR, should be taken into account. The available bandwidth to be shared should be the total bandwidth (for ABR) minus the bandwidth used for MCRs. Each connection should gain an equal share of this available bandwidth in addition to its minimum guaranteed bandwidth, MCR. In other words, the max-min criterion should apply to the available bandwidth (excluding the bandwidth used for MCRs) and the share of each connection in addition to its MCR.

Let's re-consider the previous example. The available bandwidth is 1/2 of the link bandwidth (i.e., excluding the bandwidth used for the *MCR* of the first connection). The share of the available bandwidth according to the max-min criterion is 1/8 of the link bandwidth (i.e., 1/4 of the available bandwidth). Thus, the bandwidths for the four connections are:

BW for the first one = 1/2 + 1/8 = 5/8 link bandwidth;

BW for the other three = 0 + 1/8 = 1/8 link bandwidth.

The maximum offered load here is 1, instead of 1.25. To be more general, the bandwidth for each ABR connection should be:

$$\text{Fair Share} = \lambda_i^{MCR} + \frac{C_l^A - \sum_{i \in S} \lambda_i^{MCR} - \bar{C}_l}{N_l - \bar{N}_l}, \qquad (7.2)$$

where λ_i^{MCR} is the minimum cell rate of session i. This form of max-min fairness is referred to as "MCR plus equal share [43]."

7.2 FMMRA Algorithm with *MCR* Guarantee

At the connection establishment time, the connections with non-zero *MCR* requirements are treated as CBR connections. The bandwidth for these connections must be reserved in advance. In addition to the reserved bandwidth, these connections will be given any additional available bandwidth while satisfying the "MCR plus equal share" fairness criterion.

The amount of bandwidth that must be reserved is given by the sum of all the connections' *MCRs* is as follows:

$$C_l^{MCR} = \sum_{i \in S_l} \lambda_i^{MCR}. \tag{7.3}$$

The quantity C_l^{MCR} can be calculated easily at the connection set up time. Whenever a connection opens or closes, C_l^{MCR} should be modified accordingly.

The bandwidth management module of the FMMRA algorithm will be modified to reflect the changes in the available bandwidth as follows:

$$C_l^A(t) = \mu_l \left(C_l - C_l^{GUR}(t) - C_l^{MCR}(t) \right). \tag{7.4}$$

We will apply this new fairness rule on the advertised rate calculation routine.

$$\gamma_l = \begin{cases} C_l^A & \text{if } N_l = 0, \\ \gamma_l + \lambda_i^{MCR} + \frac{\gamma_l \Delta\beta - \Delta\lambda}{N_l - [\bar{N}_l + \Delta\beta]} & \text{if } N_l > \bar{N}_l, \\ \gamma_l + \lambda_i^{MCR} & \text{if } N_l = \bar{N}_l. \end{cases} \tag{7.5}$$

7.3 Simulation Results

In this section, the FMMRA algorithm with an *MCR* guarantee is verified by simulating a simple network topology, as shown in Figure 7.1. The network consists of 3 sources, a switch and 3 receivers. The 3 sources have different *MCR* requirements.

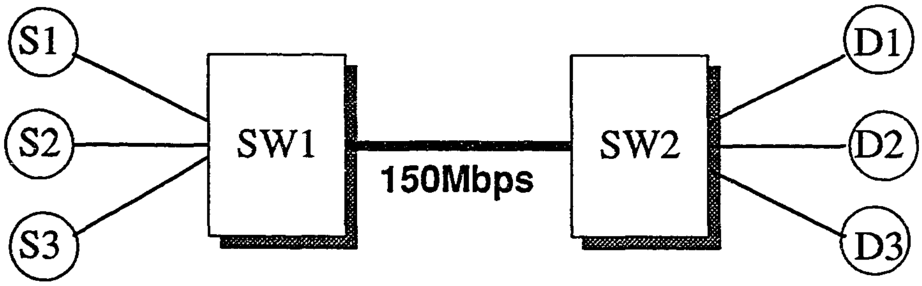

Figure 7.1 Network Configuration to Illustrate *MCR* Requirements

Table 7.1 Fair Rates with *MCR* Requirements

Connection	*MCR*	Fair-Rate
VC1	10	30
VC2	30	50
VC3	50	70

All the links have a distance of 1Km, and a capacity of 150Mbps. The modified max-min rates for the connections are tabulated in Table 7.3.

The results show that the FMMRA algorithm can easily use any variation of max-min fairness criterion, and achieve desired performance. From the results it can be seen that the sources achieve the correct fair share that satisfies the "MCR plus equal share" fairness criterion.

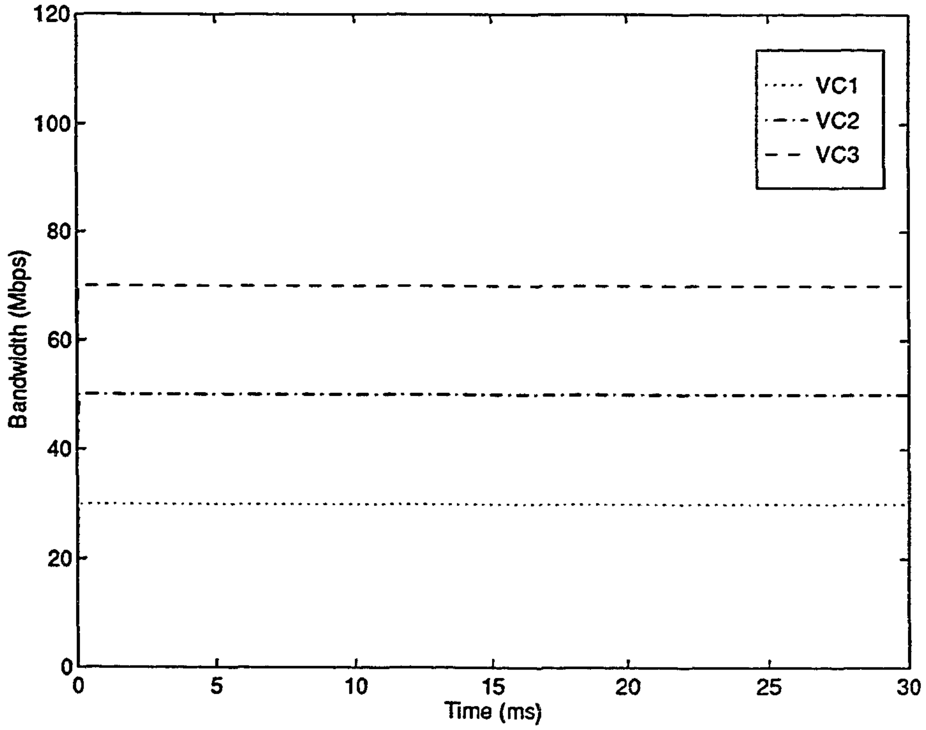

Figure 7.2 Instantaneous Bandwidth Utilization: FMMRA Algorithm with MCR Guarantee

CHAPTER 8

IMPACT OF ATM SWITCHING AND ABR CONGESTION CONTROL ON TCP PERFORMANCE

Since its birth in the early 1970's, the Transmission Control Protocol and Internet Protocol (TCP/IP) family has been widely used in today's local and wide area networks. The TCP provides reliable data communications to various applications, and contains the mechanisms used to guarantee that the data is error free, complete, and in sequence. Because of its current popularity and extensive deployment, TCP over ATM will be used heavily to transport data traffic. TCP implementations have been shown to perform well on data networks over a wide range of speeds. The recent emergence of ATM technology into today's communication networks has created many interests in studying the performance of TCP over ATM. Because of its solid installed base to support data traffic, the TCP will be used as the transport protocol to support data applications that will run on ATM networks.

One of the key components of the TCP is the collection of algorithms used to perform congestion control and recovery. The congestion control schemes applied in TCP implementations consists of many ideas proposed by Jacobson [28], some of which were later fine-tuned to improve performance. TCP uses an end-to-end flow control framework, which performs congestion control and recovery. A slow-start algorithm, a congestion avoidance mechanism, and a round-trip time estimation algorithm play the principle roles in the TCP flow control process [50]. The tuning and refinement of the TCP flow control mechanisms has been the subject of a great deal of research, and further improvements of the algorithms continue to emerge.

The study of interaction of TCP flow control and ATM layer congestion control has been an active research area. Simulation studies in [4] show that the performance of TCP over ATM without any ATM layer congestion control suffers considerably in terms of effective throughput. This happens because each TCP

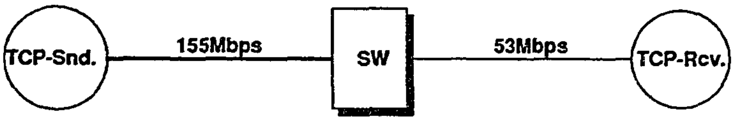

Figure 8.1 TCP over ATM: Network Model

segment is fragmented into a large number of small cells in the ATM network; the loss of single cell triggers the retransmission of an entire TCP segment. Even worse, when a switch discards a cell belonging to a TCP segment, the remaining cells continue to travel towards their destination, wasting network resources such as buffer space and link bandwidth.

In this chapter, we present some simulation results of the influence of FMMRA on the performance of TCP over ATM. More specifically, we study the TCP performance over an ATM network without any flow control enabled switches and an ATM network consisting of ER-based switches.

8.1 Network Model

This section describes the simulation environment set up to study the performance of TCP over ATM. A simple topology consisting of a source, a receiver, and a switch is considered. The congestion at the switch easily happens, because of large link capacity disparity. The topology is shown in Figure 8.1. A size of 64kB is used as the maximum TCP window size. The TCP packet size of 9180 bytes is the default for IP over ATM. For the simulations the ATM Adaptation Layer, Type 5 (AAL5), described below, is used.

8.1.1 ATM Adaptation Layer Type 5 (AAL5)

AAL5 is a mechanism for segmentation and reassembly of datagrams. That is, it is a rulebook which sender and receiver agree upon for taking a long datagram and dividing it up into cells. The sender's job is to segment the datagram and build the set of cells to be sent. The receiver's job is to verify that the datagram has been received intact, without errors, and then to put it back together again.

When a network node has a user datagram to transmit, it first converts the datagram into a CS-PDU (Convergence Sub-layer - Payload Data Unit) by adding a pad that consists of binary zeroes and an 8-byte trailer. A pad between 0 and 47 bytes in length, is added to the CS-PDU to ensure that the CS-PDU is divisible by 48 such that the chunks of CS-PDU can fit into the payload field of ATM cells. The trailer consists of some control information, the length of user information (excluding the pad) in the CS-PDU, and a 32 bit CRC to cover the data and the pad. Once the CS-PDU is made up it breaks it into many 48-byte SAR-PDUs (Segmentation and Reassembly - Payload Data Unit) and passes the 48-byte units to the ATM layer that will process the ATM header information. The last SAR-PDU is marked so that the receiver can recognize it. The payload type in the last cell (i.e., wherever the AAL5 trailer is) is marked to indicate that this is the last cell in a datagram. (The receiver may assume that the next cell received on that VCI is the beginning of a new packet.)

On the receiver side, the receiver simply concatenates cells as they are received, watching for the end-of-frame indication. When it is seen the receiver checks the length and the CRC, and then passes the PDU up to the next higher layer for further processing.

There are two problems that can happen during transit. First, a cell could be lost. In that case, the receiver can detect the problem either because the length does not correspond with the number of cells received, or because the CRC does not match what is calculated. Second, a bit error can occur within the payload. Since cells do not have any explicit error correction/detection mechanism, this cannot be detected except through the CRC mismatch.

8.1.2 Application Level Traffic Characterization

The two types of applications typically included in simulations of TCP/IP performance are the infinite source model and the bursty source model. Our goal is to study the effect of ATM and ATM layer congestion control on TCP performance; thus we use the infinite source model such that there will be a continuous flow of data from the application.

The infinite source model essentially emulates the file transfer protocol (FTP) application, whose traffic accounts for the majority of all traffic on the Internet. The emulation is imprecise only in that the "file" is infinitely long. This model is popular because it tends to make the application transparent, the better to reveal TCP behavior. Many simulations, typically those that concentrate solely or primarily on throughput, use this model more or less exclusively.

Implementation of such a model is generally done via some sort of "polling" scheme between the TCP object and the application object, wherein TCP requests data any time its control algorithm indicates that further transmission is allowed, and the application immediately obliges. The packet size is typically fixed in this model.

8.2 Performance of TCP over ATM

A network as shown in Figure 8.1 is simulated with and without the ATM layer flow control. Figure 8.2 shows the effective throughput achieved, and Figure 8.3 shows the re-transmission percentage as a function of switch buffer size. The effective throughput is defined as:

$$\begin{aligned}&\text{Effective Throughput (bits/sec)} \qquad (8.1)\\ &= \frac{\text{Number of Successfully Transmitted Packets} \times \text{Packet Size (bytes)} \times 8 \times \frac{48}{53}}{\text{Measurement Time Duration (seconds)}}.\end{aligned}$$

In the effective throughput calculations only the successfully transmitted packets are included. A factor of $\frac{48}{53}$ is used, since 5 bytes of the ATM cell is used for the header, and only 48 bytes are used for payload.

When there is no ATM layer congestion control, and if the switch buffer is not large enough to accept a full window size of cells, there is a significant amount of cell loss. This cell loss leads to packet re-transmissions, which results in poor throughput. As seen from the figure, in order to maximize the throughput the switch buffer size must be at least equal to the TCP receiver window size. In the simulation considered here, for a window size of 64Kb, a switch buffer size of 1500 cells is required.

When the ATM layer congestion control is turned on, and when the switch is implemented with the FMMRA algorithm, the required buffer size to achieve full utilization is reduced significantly. With the ATM layer control the cells can be injected into the network in a controlled manner rather than sending them at once. The number of re-transmissions are reduced because of the minimized cell loss from congestion avoidance. The RM cells used in ATM layer congestion control reduces the effective throughput. In the simulations done here, an RM cell is generated every 32 data cells. Therefore, with a very large buffer size, the effective throughput achieved with flow control is less (about 2Mbps) than the effective throughput achieved when there is no ATM layer control.

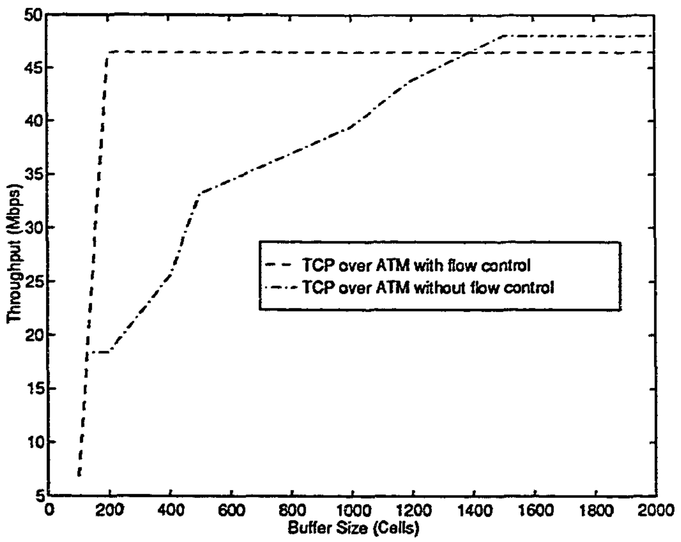

Figure 8.2 TCP over ATM: Effective Throughput vs. Buffer Size

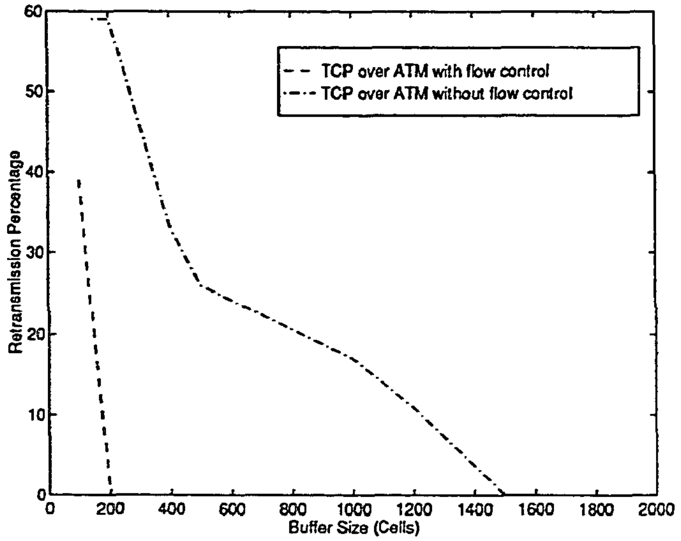

Figure 8.3 TCP over ATM: Re-transmission Percentage vs. Buffer Size

CHAPTER 9
SUMMARY AND CONCLUSIONS

Traffic management of ABR services in ATM networks poses many challenging problems, such as the congestion control problem. ABR services require that the network rely on feedback, closed-loop control techniques to achieve the desired QoS guarantees. The rate-based congestion control approach, which was accepted as the control technique for ABR services by the ATM Forum, requires that the switches monitor their congestion levels and inform the sources of their status of congestion. The focus of this dissertation is to provide an introduction to ABR congestion control techniques, to evaluate some proposed algorithms, and to provide a new, fast adaptive algorithm with many desirable features for ATM switches.

The present generation of ATM switches to be deployed in the next couple of years has the capability of providing EFCI-based control. The new generation of switches may consist of various implementation styles with different levels of performance and complexity. It is vital that the implementation of new-generation switches not dictate the overall functionality of the control mechanism; that is, they must co-exist with the simpler switches in the ATM network.

General guidelines for fair rate allocation is presented in terms of performance and complexity requirements. Many of the proposed algorithms are summarized, and their advantages and disadvantages are presented. One of these algorithms is used to show that the implementation of per-VC queueing is not only very expensive but also unnecessary when intelligent control techniques are used in conjunction with the FIFO queueing discipline.

Drawbacks of the other explicit rate algorithms are presented, and the new explicit rate algorithm is presented. The algorithm should be extended to support multicast requirements put forth by the ATM Forum. The basic issue is to how to consolidate congestion information from multiple RM cells. Further simulations should be done to study the buffer size requirements in the switches, especially to support TCP/IP traffic over ATM. Fairness, link utilization, and effective throughput performance can be studied utilizing many proposed algorithms.

In summary, in this dissertation, the problem of ABR traffic control is formulated, and a new switch algorithm is provided. The main contributions of this dissertation are listed below:

- General guidelines of switch algorithms with a high level of performance and low complexity as primary goals;
- Survey and evaluation of existing switch algorithms documented by the ATM Forum;
- A switch model and a new algorithm that overcome many drawbacks of the proposed algorithms;
- A study and analysis of the impact of two major queueing policies on ABR service;
- Mathematical analysis of convergence and transient behavior for the algorithm;
- Demonstration of the effectiveness of the algorithm through extensive simulations;
- Modification of the algorithm to support minimum cell requirements;
- A study of the impact of ATM switching and the impact of ATM layer congestion control on TCP/IP performance.

APPENDIX A
PROOF OF CONVERGENCE

A.1 Proof of Lemma 1

Let us say that there are N_l sessions traverse link l, and thus the initial advertised rated on link l is $\frac{C_l^A}{N_l}$. Let j be the session which receives the smallest fair allocation (i.e., $j \in \mathcal{S}^1$) and with the smallest round trip time. This connection will receive its RM cell back before any other session. Upon reception, the ER field in the RM cell will be set to γ^1, which is expressed as

$$\gamma^1 = \min\left\{\gamma \mid \gamma = \frac{C_l^A}{N_l},\ \forall l \in \mathcal{L}_j,\ j \in \mathcal{S}^1\right\}. \tag{A.1}$$

The set of bottleneck links where j is saturated can be given by,

$$\mathcal{L}_j^* = \left\{l \in \mathcal{L} \mid \gamma^1 = \frac{C_l}{N_l}\right\}. \tag{A.2}$$

From the above argument we can conclude the following.

$$\gamma_l = \gamma^1 \qquad \text{if } l \in \mathcal{L}_j^*, \tag{A.3}$$

$$\gamma_l > \gamma^1 \qquad \text{if } l \in \{\mathcal{L}_j - \mathcal{L}_j^*\} \tag{A.4}$$

This implies that at the completion of first round trip time of j's RM cell,

$$\beta_l^j = \begin{cases} 0 & \text{if } l \in \mathcal{L}^1, \\ 1 & \text{if } l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}, \end{cases} \tag{A.5}$$

and

$$\lambda_l^j = \begin{cases} 0 & \text{if } l \in \mathcal{L}_j^*, \\ \gamma^1 & \text{if } l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}. \end{cases} \tag{A.6}$$

Once RM cell of session j is seen by the link, the new advertised rates are computed via Equation (5.17). Note that, on link $l \in \mathcal{L}_j^*$, $\Delta\lambda = 0$, and $\Delta\beta = 0$. Thus we have,

$$\gamma_l(t^+) = \gamma_l(t),\ l \in \mathcal{L}_j^*. \tag{A.7}$$

On link $l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}$, $\Delta\lambda = \gamma^1$, and $\Delta\beta = 1$. Thus, the new advertised rate will be

$$\gamma_l(t^+) = \gamma_l(t) + \frac{\gamma_l(t) - \gamma^1}{N_l - 1},\ l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}. \tag{A.8}$$

Since the session is marked as bottlenecked on links $l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}$, $\gamma_l(t) > \gamma^1$, and therefore, $\gamma_l(t^+) > \gamma_l(t)$. On the other hand, note that γ_l for the links $l \in \mathcal{L}_j^*$ is not increased. This implies that the links $l \in \mathcal{L}_j^*$ are the bottleneck links for j. Thus, the session j is saturated.

A.2 Proof of Lemma 2

1) In Lemma 1 it was shown that the set $\mathcal{L}_1^*$ forms the bottleneck links for session, j, which receives the smallest fair rate allocation and the advertised rates on links $\mathcal{L}_1^*$ are not modified. It was shown that the advertised rate on links $\mathcal{L}_j - \mathcal{L}_j^*$ is increased (i.e.,$\gamma_l(t^+) > \gamma_l(t)$). The new advertised rate on links $\mathcal{L}_j - \mathcal{L}_j^*$ can also be given as

$$\gamma_l(t^+) = \frac{C_l^A - C_l^*(t_j^1)}{N_l - N_l^*(t_j^1)}, \tag{A.9}$$

where $C_l^*(t_j^1) = \lambda_j^* = \gamma^1$. The $\gamma_l(t^+)$ never falls below the quantity $\frac{C_l^A - C_l^*(t_j^1)}{N_l - N_l^*(t_j^1)}$. Thus, the statement 1 of lemma 2 follows.

2) If any other session, $i \in \mathcal{S}^1, i \neq j$, is seen by any $l \in \mathcal{L}_j^*$, then i will be allocated the rate, γ^1. Thus, when an RM cell from this group of VCs with the largest round trip time, completes their round trip, all the VCs in the set $\mathcal{S}^1$ get saturated with the fair rate of γ^1.

3) Now, consider the arrival of another RM cell at a link, which was sent after an RM cell was received by the saturated source j. Again, it should be noted that the γ_l for link $l \in \mathcal{L}^1$ is γ^1. Also note that the advertised rates on links $l \in \{\mathcal{L}_j - \mathcal{L}_j^*\}$, seen by the RM cells from the saturated connections are larger than γ^1. Thus, after the first round trip time, the sessions with the minimum max-min fair rate (i.e., $j \in \mathcal{S}^1$) will achieve saturation and will stay saturated. Now, the total saturation bandwidth on the non-bottleneck links is given by

$$C_l^* = \sum_{l \in \mathcal{L}_j - \mathcal{L}_j^*} \lambda_j^* \tag{A.10}$$

Following a similar line of argument, consider any session k, saturated with γ^m as its allocation (i.e. $k \in \mathcal{S}^m$) at time t.

$$\gamma^m = \min\{\gamma \mid \gamma = \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}, \forall l \in \mathcal{L}_k \text{ for any } k \in \mathcal{S}^m\}. \tag{A.11}$$

The advertised rates are given by

$$\gamma_l = \gamma^m \text{ if } l \in \mathcal{L}_k^*, \forall k \in \mathcal{S}^m, \tag{A.12}$$

$$\gamma_l > \gamma^m \text{ if } l \in \{\mathcal{L}_k - \mathcal{L}_k^*\}, \forall k \in \mathcal{S}^m, \tag{A.13}$$

and the new set of bottleneck links where k saturated can be given as,

$$\mathcal{L}_k^* = \{l \in \mathcal{L} \mid \gamma^m = \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}\} \tag{A.14}$$

Similar to the proof of Lemma 1, after the computations of γ_l we will have,

$$\gamma_l(t^+) = \gamma_l(t), \ l \in \mathcal{L}_k^*, \tag{A.15}$$

and

$$\gamma_l(t^+) \geq \gamma_l(t) + \frac{\gamma^m}{N_l - \bar{N}_l - 1}, \ l \in \{\mathcal{L}_i - \mathcal{L}_k^*\}. \tag{A.16}$$

Since on links $\mathcal{L}_i^*$ the advertised rates are not modified, the session k and any other session sharing any $l \in \mathcal{L}_k^*$, will be allocated the same rate, γ^m.

A.3 Proof of Lemma 3

Let us start with some unsaturated sessions in the network and assume that the sessions receiving fair allocation of γ^{m-1} have saturated already. After an RM cell of the set of unsaturated sessions completes its round trip, the session will be allocated a rate equal to the minimum advertised rate among all the links. At least one link must have the minimum γ_l among all links, which is larger than γ^{m-1}. Thus, any unsaturated session, j, that traverse this link will choose this γ_l as its allocation. The session which receives this allocation must be from the set $\mathcal{S}^m$, and once its RM cell completes its round trip it will get saturated. This will be true for any session from the set $\mathcal{S}^m$. This concludes proof of lemma 3.

REFERENCES

1. A. Miller, "From here to ATM," *IEEE Spectrum*, pp. 20–24, June 1994.

2. J. Lane, "ATM Knits Voice, Data on Any Net," *IEEE Spectrum*, pp. 42–45, February 1994.

3. M. D. Prycker, *Asynchronous Transfer Mode: Solutions for B-ISDN*, Elis Horwood, New York, NY, 1993.

4. S. Floyd and A. Romanov, "Dynamics of TCP Traffic over ATM Networks," *Proc. ACM SIGCOMM'94*, pp. 79–88, September 1994.

5. R. Jain, "Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey," *Computer Systems and ISDN Systems*, February 1995.

6. M. Hluchyj, et. al, "Closed-Loop Rate-Based Traffic Management," *The ATM Forum Contribution 94-0438R2*, September 1994.

7. D. Kataria, "Comments on Rate-Based Proposal," *The ATM Forum Contribution 94-0384*, May 1994.

8. J. Bennett and G. T. D. Jardins, "Comments on the July PRCA Rate Control Baseline," *The ATM Forum Contribution 94-0682*, July 1994.

9. G. C. Fedorkow, "Observations on Complexity of ABR Mechanisms," *The ATM Forum Contribution 94-0593*, July 1994.

10. P. Newman and G. Marshall, "BECN Congestion Control," *The ATM Forum Contribution 94-789R1*, July 1993.

11. A. W. Barnhart, "Use of the Extended PRCA with Various Switch Mechanisms," *The ATM Forum Contribution 94-0898*, September 1994.

12. L. Roberts, "New Pseudo Code Explicit Rate Plus EFCI Support," *The ATM Forum Contribution 94-0974*, October 1994.

13. L. Roberts, "The Benefits of Rate-Based Flow Control for ABR Service," *The ATM Forum Contribution 94-0796*, September 1994.

14. L. Roberts, "Rate-Based Algorithm for Point to Multipoint ABR Service," *The ATM Forum Contribution 94-0772R1*, November 1994.

15. H. T. Kung, et. al, "Adaptive Credit Allocation for Flow Controlled VCs," *The ATM Forum Contribution 94-0282R2*, March 1994.

16. H. T. Kung, et. al, "Flow Controlled Virtual Connection Proposal for ATM Traffic Management," *The ATM Forum Contribution 94-0632R2*, September 1994.

17. F. Bonomi and K. W. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," *IEEE Networks Magazine*, pp. 25–39, March-April 1995.

18. A. Arulambalam, X. Chen, and N. Ansari, "Impact of Queueing Disciplines on Available Bit Rate Congestion Control in ATM Networks," *Proc. The 30th Annual Conference on Information Sciences and Systems*, March 1996.

19. H. T. Kung and R. Morris, "Adaptive Credit Allocation for Flow Controlled VCs," *IEEE Network Magazine*, pp. 40–48, March-April 1995.

20. K. K. Ramakrishnan and P. Newman, "Integration of Rate and Credit Schemes for ATM Flow Control," *IEEE Network Magazine*, pp. 49–56, March-April 1995.

21. The ATM Forum, *ATM User-Network Interface Specifications, Version 3.0*, Prentice Hall, Englewood Cliffs, NJ, 1993.

22. D. McDysan and D. Spohn, *ATM - Theory and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1994.

23. R. Jain, "Congestion Control in Computer Networks: Trends and Issues," *IEEE Network Magazine*, pp. 24–30, May 1990.

24. P. E. Boyer and D. P. Tranchier, "A Reservation Principle with Applications to the ATM Traffic Control," *Computer Networks and ISDN Systems*, vol. 24, pp. 321–334, 1992.

25. P. Newman, "Traffic Management for ATM Local Area Networks," *IEEE Communications Magazine*, pp. 44–50, August 1994.

26. P. Newman and G. Marshall, "Update on BECN Congestion Control," *The ATM Forum Contribution 94-855R1*, September 1993.

27. A. Romanov, "A Performance Enhancement for Packetized ABR and VBR+ Data," *The ATM Forum Contribution 94-0295*, March 1994.

28. V. Jacobson, "Congestion Avoidance and Control," *Proc. ACM SIGCOMM'88*, August 1988.

29. K. K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," *ACM Transactions on Computer Systems*, vol. 8, no. 2, pp. 158–181, May 1990.

30. ANSI, "ISDN – Core Aspects of Frame Protocol for use with Frame Relay Bearer Service," *ANSI T1.618*, 1991.

31. M. Hluchyj and N. Yin, "On Closed-Loop Rate Control for ATM Networks," *Proc. INFOCOM'94*, pp. 99–108, 1994.

32. The ATM Forum, "The ATM Forum Traffic Management Specification,Version 4.0," *ATM Forum Contribution*, 1995.

33. A. Charny, D. D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," *Proc. ICC 95*, June 1995.

34. K. Siu and H. Tzeng, "Adaptive Proportional Rate Control for ABR Service in ATM Networks," Tech. Rep. 94-07-01, Electrical and Computer Engineering, University of California, Irvine, CA, July 1994.

35. R. Jain, S. Kalyanaraman, and R. Viswanthan, "The OSU Scheme for Congestion Avoidance using Explicit Rate Indication," Tech. Rep. OSU-CISRC-1/96-TR02, Ohio State University, Columbus, OH, September 1996.

36. L. Roberts, "Enhanced PRCA (Proportional Rate-Control Algorithm)," *The ATM Forum Contribution No. 94-0735R1*, August 1994.

37. A. W. Barnhart, "Explicit Rate Performance Evaluation," *The ATM Forum Contribution 94-0983R1*, September 1994.

38. K. B. Kumar and J. M. Jaffe, "A New Approach to Performance Oriented Flow Control," *IEEE Transactions on Communications*, vol. 29, pp. 427–435, April 1981.

39. D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, Englewood Cliffs, NJ, 2nd ed., 1987.

40. E. M. Gafni and D. P. Bertsekas, "Dynamic Control of Session Input Rates in Communication Networks," *IEEE Transactions on Automatic Control*, vol. 29, no. 11, pp. 1009–1016, November 1984.

41. M. Gerla, H. W. Chen, and J. R. de Marca, "Fairness in Communication Networks," *Proc. ICC 85*, pp. 1384–1389, 1985.

42. J. M. Jaffe, "Bottleneck Flow Control," *IEEE Transactions on Communications*, vol. 29, no. 7, pp. 954–962, July 1981.

43. N. Yin, "Fairness Definition in ABR Service Model," *The ATM Forum Contribution 94-0928R2*, September 1994.

44. A. Charny, "An Algorithm for Rate Allocation in a Packet-Switching Network with Feedback," Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1994.

45. A. Charny, K. K. Ramakrishnan, and A. G. Lauck, "Scalability Issues for Distributed Explicit Rate Allocation in ATM Networks," *Proc. INFOCOM 96*, pp. 1198–1205, March 1996.

46. FORE Systems, "Forethought: Bandwidth Management," *FORE Systems - White Papers*, April 1995.

47. A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," *Proc. of ACM SIGCOMM'89*, pp. 3–12, 1989.

48. R. J. Simcoe, "Configurations for Fairness and Other Test," *The ATM Forum Contribution 94-0557*, July 1994.

49. J. C. R. Bennett, K. Chang, H. T. Kung, and D. Lin, "A Comparison of EPRCA September 94 Version and FCVC Control Scheme: Simulation Results," *The ATM Forum Contribution 94-0929*, September 1994.

50. W. R. Stevens, *TCP/IP, Illustrated, Vol. I*, Addison Wesley, New York, NY, 1994.

51. S. Feit, *TCP/IP, Architecture, Protocols, and Implementation*, McGraw-Hill, New York, NY, 1993.

52. A. Arulambalam, X. Chen, and N. Ansari, "A New Fair-Rate Allocation Algorithm for Available Bit Rate Services in ATM Networks," *submitted to IEEE/ACM Transactions on Networking*, 1996.

53. H. T. Kung and K. Chang, "Receiver-Oriented Adaptive Buffer in Credit-Based Flow Control," *Proc. INFOCOM'95*, pp. 239–252, 1995.