MONASH UNIVERSITY

THESIS ACCEPTED IN SATISFACTION OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

ON............3 May 2002...............

...................[redacted].............

Sec. Research Graduate School Committee

Intelligent Quality of Service Algorithms for High Speed Networks

BY

Hiam Hiok Lim

BDigSys (Hons)

A THESIS SUBMITTED IN FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

School of Computer Science and Software Engineering

M O N A S H

U N I V E R S I T Y

January 2002

DECLARATION

I, Hiam Hiok Lim, hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university. Further to the best of my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

[redacted]

23/1/2002.

Hiam Hiok Lim   date

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Bin Qiu, senior lecturer of School of Computer Science and Software Engineering (CSSE), for his guidance, encouragement, patience and support throughout this work, as well as his constructive comments on the presentation of this thesis.

I would also like to thank Associate Professor Jim Breen for the concepts of TCP protocols, and Jojada Tirtowidjojo for the concepts of Bi-directional Explicit Rate Marking Optimizer (BEMO) algorithm.

Special thanks are due to Kadar Tanuwidjaja for selflessly giving me permission to use his computer for simulation purposes during my earlier days of study.

I would also like to thank Jane Moodie, lecturer of Language and Learning Services Unit from Faculty of Engineering, for her invaluable advise in paper and thesis writing, oral presentation, as well as correcting English Grammar and spelling. Many thanks also go to Stewart Hore for his careful proof-reading of this thesis. His input and comments have been very valuable.

I am greatly indebted to my parents and brothers for their love, understanding and encouragement throughout my study over these years. I am also grateful to my wonderful friends and roommates for the support through good and bad times.

Finally, scholarships granted by the Australian Government and CSSE are gratefully acknowledged.

Abstract

Long round-trip delays, heavy traffic fluctuation and uncertainty of traffic parameters pose a significant challenge for any traffic control scheme in high speed networks. An enormous amount of proposed solutions to these problems are based on conventional control algorithms which normally require exact mathematical equations and a fixed set of precise threshold values. However, when the complexity of systems increases, the ability to describe control mechanisms mathematically becomes difficult. This may lead to low network efficiency, quality of service (QoS) degradation and fairness problems.

This thesis attempts to address these problems by proposing several intelligent QoS algorithms based on fuzzy logic prediction and control. It has been observed that the approximate nature of human reasoning provided by the fuzzy sets theory is generally able to overcome the problems of imprecision and uncertainty.

This thesis first provides an examination of closed-loop reactive control for asynchronous transfer mode (ATM) available bit rate (ABR) service. A novel fuzzy logic congestion control scheme is proposed to alleviate the problem of long round-trip delays in the feedback control loop. The feedback control message incorporates a fuzzy logic link utilization factor which is targeted one round-trip delay in advance based on the predicted queue length in the switch. As a result, end systems are able to react to congestion earlier. Simulation results demonstrate that the proposed scheme exhibits a faster convergence time in rate and queue length, lower buffer requirements and cell loss, and is able to maintain high link utilization over the conventional control scheme. This design method is also applicable to other similar end-to-end feedback control schemes.

The open-loop preventive connection admission control (CAC) is considered next. To overcome the problem of uncertainty in describing traffic parameters during the connection setup phase, and the inaccuracies and errors involved in the conventional measurement-based CAC, a novel predictive fuzzy logic measurement-based CAC scheme is introduced. Traffic parameters are predicted by an on-line fuzzy logic predictor, and the QoS requirements are targeted indirectly by a fuzzy logic adaptive weight factor. Admission decisions are then based on real-time measurement of aggregate free bandwidth with the fuzzy logic adaptive weight factor as well as the predicted traffic parameters. Simulation results show that the proposed CAC achieves higher link efficiency with guaranteed QoS when compared to a conventional model-based CAC. The benefits of fuzzy logic prediction are also demonstrated by applying the prediction technique to the conventional model-based CAC.

Due to the increasing demand of Internet services, and the adoption of ATM as a backbone for Internet protocol (IP) networks, this thesis also provides an investigation of congestion control for transporting Internet applications over ATM networks. An intelligent fuzzy logic selective cell drop scheme for transmission control protocol (TCP) over ATM unspecified bit rate (UBR) service is proposed to tackle the problem of long round-trip delays. The scheme is able to accept or drop a new incoming packet dynamically based on the predicted future buffer condition in the switch. It can thereby compensate the effect of long round-trip delays and react to congestion earlier. In addition, fairness is maintained by a fuzzy logic inference system using soft-computing. Consequently, cell loss is kept to a minimum, and high TCP throughput with fairness is delivered. Simulation results show that the proposed scheme achieves significant improvement in TCP efficiency and fairness over the conventional schemes. Fuzzy logic prediction and control on TCP over ATM ABR congestion control is also shown to have a better performance.

This thesis demonstrates the advantages of fuzzy logic applications in three major aspects of traffic control in high speed networks. It appears that fuzzy logic systems also have a great potential to control traffic more effectively in other emerging technologies, such as satellite networks and wireless communications.

Publications

Following is the list of refereed publications which are related to the work presented in this thesis.

- Hiam Hiok Lim and Bin Qiu, "Performance Enhancement of ERICA+ Using Fuzzy Logic Prediction", in Proceedings, IEEE Asia-Pacific Conference on Communications (APCC'99), Beijing, China, 19-21 October 1999, vol. 1 of 2, pp.134-138.

- Hiam Hiok Lim and Bin Qiu, "Fuzzy Logic Buffer Control in ABR Service for Performance Refining", in CD Proceedings, International Conference on Information, Communications and Signal Processing (ICICS'99), Singapore, 7-10 December 1999, Session 3C2, Paper No. 176.

- Hiam Hiok Lim and Bin Qiu, "A Dynamic Fuzzy Logic QoS Improvement Scheme", in CD Proceedings, The 7th IEEE Singapore International Conference on Communication Systems (ICCS 2000), Singapore, 20-24 November 2000, 5pp.

- Hiam Hiok Lim and Bin Qiu, "Fuzzy Logic Target Utilization and Prediction for Traffic Control", in Proceedings, IEEE Global Telecommunications Conference (GLOBECOM 2000), San Francisco, USA, 27 November - 1 December 2000, vol. 3 of 3, pp.1644-1648.

- Hiam Hiok Lim and Bin Qiu, "A Predictive Measurement-based Fuzzy Logic Connection Admission Control", in Proceedings, IEEE International Conference on Communications (ICC 2001), Helsinki, Finland, 11-15 June 2001, vol. 3 of 10, pp. 920-924.

- Hiam Hiok Lim and Bin Qiu, "Predictive Fuzzy Logic Buffer Management for TCP/IP over ATM-UBR and ATM-ABR", in Proceedings, IEEE Global Telecommunications Conference (GLOBECOM 2001), San Antonio, Texas, USA, 25-29 November 2001, vol. 4 of 6, pp. 2326-2330.
- Hiam Hiok Lim and Bin Qiu, "Performance Improvement of TCP using Fuzzy Logic Prediction", in Proceedings, International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2001), Nashville, USA, 20-21 November 2001, pp. 152-156.
- Hiam Hiok Lim and Bin Qiu, "Fuzzy Logic Traffic Control in Broadband Communication Networks", in Proceedings, The 10th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2001), Melbourne University, Australia, 2-5 December 2001, vol. 1 of 3, Session T2a, Paper No. 148.

Contents

List of Figures

List of Tables

List of Procedures

Chapter 1

Introduction

Current and future networking technologies are expected to deliver multimedia applications such as voice, video and data on the same physical link. These applications generate traffic streams with different burstiness and magnitude. They also have different quality of service (QoS) requirements. QoS requirements are typically expressed in terms of delay and loss within a negotiated rate across the network. In order to be competitive, network operators have to achieve the highest possible efficiency by multiplexing different types of traffic as well as maintain QoS guarantees.

Asynchronous Transfer Mode (ATM) is a modern communication network technology. It has been widely deployed in local backbones and wide area networks. It is designed to support different types of service from a single network infrastructure, and has the ability to enforce QoS guarantees and regulate traffic flows. Services provided by the ATM can be categorized into real-time services such as constant bit rate (CBR) and variable bit rate (VBR) services, and non real-time services such as unspecified bit rate (UBR) and available bit rate (ABR) services. Real-time services have a higher priority over non real-time services as they cannot tolerate delays. Examples of real-time services are voice and video, whereas non real-time services are data applications.

Over the past decade, demand on Internet services such as *email* (electronic mail), *telnet* (terminal access), file transfer protocol (FTP) and world wide web (WWW) access has increased exponentially. These services are built based on the transmission control protocol (TCP) and the Internet protocol (IP). Therefore, it is very important that ATM as a backbone technology supports the seamless and efficient transport of TCP/IP data.

Traffic management is thus necessary for the efficient network resource utilization and delivery of the negotiated QoS to users. It is a set of mechanisms that manages and controls traffic and congestion in the ATM networks. These mechanisms include connection admission control (CAC), traffic policing and shaping (also known as usage parameter control, UPC), scheduling, selective cell discard and feedback control.

There has been an enormous amount of research and publications since the introduction of ATM technology, especially in the area of traffic management. Despite all these efforts, there are still many substantial unsolved problems. Particularly, the problems of uncertainty in traffic parameters and large bandwidth-delay product. This thesis aims to address these problems using an unconventional approach.

1.1 Motivations and Direction

Fuzzy logic control has become one of the most attractive areas of fuzzy set theory applications following the pioneering works of Zadeh [Zad73, Zad65]. It has been successfully deployed in many real-world automatic control systems such as washing machines, air-conditioners, automobile transmissions, and industrial robots. However, the applications of fuzzy logic control in telecommunication systems and networks are recent and considerably less compared to the automatic control. Some research efforts in the areas of traffic management using fuzzy logic are CAC [RR00, CC96], UPC [CFP96, CD95] and congestion control [LH00, PcR97]. (A comprehensive survey could be found in [CD97]). As it can be observed, these works are quite recent, hence there is an enormous room to realize the full potential of fuzzy logic control in telecommunication networks, in particularly, the traffic management area.

Another motivation for using fuzzy logic in ATM traffic management has come to light from the analysis of the limitations in conventional control algorithms. These algorithms normally require exact mathematical equations and a fixed set of precise threshold values for control mechanisms. When the complexity of systems increases, especially due to traffic fluctuation and uncertainty, the ability to describe control mechanisms mathematically becomes difficult. This may lead to poor dynamic action in the control mechanisms, which in turn can cause low network efficiency, QoS degradation and fairness problems.

In contrast, fuzzy sets theory provides a robust mathematical framework for dealing with imprecision, non-linear and uncertain processes. This is because it allows the approximate nature of human decision making and the soft-computing capability. It has been observed that the approximate nature of human reasoning is generally able to overcome the problems of imprecision and uncertainty. Fuzzy logic control thus appears to have greater potential to adapt to fluctuations of network traffic.

The ability to forecast workload is an important factor for efficient network performance. The main idea is to predict traffic or workload far enough in advance so that one can control traffic flows or resource usage in time to avoid congestion and achieve high network utilization. Traffic prediction has been successfully applied to various traffic management areas, such as dynamic resource allocation [MPS96, WCS95], UPC [dlCAM97, JM95] and congestion control [JT00, HLP95]. These applications use traditional prediction models, such as autoregressive (AR) or autoregressive moving-average (ARMA) or autoregressive, integrative, moving average processes (ARIMA), to exploit the high correlation inherent in traffic data. These models express stochastic processes in different orders. The higher the order of the model, the more tuning parameters are needed. In addition, higher order models require more computation and do not necessarily always provide better results. Intelligent technique such as artificial neural networks (ANN) can also be used to predict traffic [LD97, LO97, MLNH96]. However, neural network approach tends to have long training periods which often have to be performed off-line. Recent research shows that fuzzy logic prediction outperforms the traditional autoregressive predictions for multimedia network traffic [Qiu01, Qiu98]. This is because fuzzy logic control allows the modeling of complex systems based on intuition and expert knowledge. Hence it appears that fuzzy logic systems have a potential to control traffic more effectively. The objective of this research is thus to apply the fuzzy logic prediction and control techniques in three major aspects of traffic management: (1) flow and congestion control for the ABR service, (2) CAC for the VBR service and (3) congestion control for TCP/IP over ATM. Their performances are examined by extensive simulation and analysis.

1.1.1 Flow and Congestion Control

The first aspect of traffic management under investigation in this thesis is the flow and congestion control for the ATM ABR service. It is a *reactive control* with closed-loop feedback that aims to detect and control congestion within the network by limiting flow rates of appropriate sources. As a result, network efficiency can be increased and cell loss ratio can be kept to minimum. This control action takes place during the life time of a connection. It is for applications which can tolerate delay and accept a low or zero cell loss. These types of application usually cannot predict their own bandwidth requirements, and are expected to share fairly the left-over bandwidth by the higher priority services with other ABR sources.

The ATM Forum has adopted a rate-based flow and congestion control scheme as the standard for the ABR service [ATM96]. Hence the fair allocated bandwidth is conveyed to sources by control messages using the explicit rate (ER) field of resource management (RM) cells. However, these control messages may be delayed due to propagation delays in the round-trip feedback control loop. Consequently, the control messages are usually inaccurate for most current network conditions. This can cause buffer overflow and QoS degradation.

To tackle the problems of long round-trip delays, many switch schemes are proposed to detect and control congestion. Their objectives are to protect network resources, minimize queue length and cell delay, and maximize throughput. Some of these switch schemes use queue length as a measurement unit for congestion detection. Single or duel thresholds on queue length are commonly used [Rob94, CXK96, CAXC97]. Differential queue length or mathematical equations are also developed by some researchers [CXK96, JKG+96]. Others use queue growth rate or prediction techniques [JT00, LH00, PcR97, HLP95].

Due to highly fluctuated traffic in ATM networks, congestion control based on static threshold values, or mathematical equations which are inflexible and abrupt may cause high variance in queue fluctuations. As a result, they cause high cell loss or low link utilization in some situations. On the other hand, prediction techniques appear to have the potential to control congestion more effectively. In particular, a fuzzy logic prediction [Qiu01, Qiu98] has been proven to outperform conventional autoregression predictions. Therefore, a novel fuzzy logic congestion control scheme based on this prediction technique is proposed. A key feature of the proposed scheme is its ability to target link utilization dynamically based on the predicted future buffer condition in the switch. Performance of the proposed fuzzy logic congestion control scheme is evaluated by intensive simulation using different network configurations and traffic types.

1.1.2 Connection Admission Control

Another aspect of traffic management in ATM networks is the connection admission control (CAC), which is also referred to as the *preventive control.* It is an open-loop control aiming to prevent network congestion as well as to guarantee the QoS of the existing connections. It also aims at maximizing the utilization of network resources. CAC is performed by the network during the connection setup phase to determine whether a connection request should be accepted or rejected. The decision is based on the availability of resources (bandwidth and buffer) and whether the QoS can be guaranteed for the requested connection without degrading the promised QoS of the existing connections. The main problem of a CAC scheme is VBR traffic fluctuations and uncertainty, and the difficulty for users to accurately describe the traffic characteristics of the requested connections.

There are many CAC approaches reported to date [PE96, SYT99]. Conventional CAC relies on analytical models in which parameters are based on *a priori* or user-specified traffic descriptors [GAN91, GH91, Kel91]. However an analytical model may not be able to capture the statistical nature of traffic sources. This is because VBR traffic characteristics such as sustainable cell rate (SCR), intrinsic burst tolerance (IBT) and cell delay variation (CDV) are difficult for users to specify precisely. Improper specification of traffic characteristics could lead to either degraded QoS or link under-utilization. To overcome these drawbacks, recent researchers have proposed measurement-based CAC [GT99, ZL98a, DJM97, JDSZ97] and have exploited the advantages of statistical multiplexing. Admission decisions are made based on the measured incoming traffic and network performance parameters such as link load, queue occupancy and cell loss ratio (CLR).

Measurement-based CAC schemes have the capabilities to capture the on-line traffic dynamics and system performances, and achieve a higher link efficiency. However, admission decisions and promised QoS to users may be compromised due to the inaccuracies and errors involved in the measurement. Computational intelligence techniques such as fuzzy logic control appears to be very useful in solving complex processes which cannot be described precisely by exact mathematical expressions, or when the available information cannot be interpreted precisely and with certainty.

The ability to forecast workload is also an important factor for efficient network performance. Hence, based on the concepts of free bandwidth and the adaptive weight factor in [ZL98b, ZL98a], this thesis proposes a novel measurement-based CAC using fuzzy logic prediction [Qiu01, QSW99] and inference. Its main aim is to exploit the advantages of fuzzy logic prediction and control on measurement-based CAC under different VBR traces. There are two distinct features of the proposed scheme. Firstly, instead of the user defining the traffic parameters, the traffic parameters are measured and predicted by an on-line fuzzy logic predictor. Secondly, QoS are targeted indirectly by a fuzzy inference adaptive weight factor. Admission decisions are then based on real-time measurement of aggregate traffic statistics with the fuzzy logic adaptive weight factor as well as the predicted traffic parameters.

Using different VBR traces, the benefits of fuzzy logic prediction are first demonstrated by applying the prediction technique to the conventional CAC and the proposed fuzzy logic measurement-based CAC. Then the performance of the proposed fuzzy logic CAC is investigated by extensive simulation.

1.1.3 Congestion Control for TCP/IP over ATM

Due to the popularity of the Internet services, and the adoption of ATM as a backbone on IP networks, it is important to provide seamless and efficient transport of TCP/IP data packets over ATM. Hence, the third aspect of traffic management under investigation in this thesis is the congestion control for TCP/IP traffic over ATM.

TCP/IP is a best effort service which requires no QoS guarantee. It is sensitive to losses but tolerant to delay. In ATM networks, UBR and ABR services are most commonly used to support TCP/IP.

TCP/IP data applications have to be fragmented by the end systems into cells in order to transport over ATM networks. However, the problems of fragmentation and reassembly of application packets and cell loss in the ATM layer can severely affect application performance and network utilization, especially during periods of network congestion. To prevent congestion collapse, TCP end systems implement closed-loop, window-based flow and congestion avoidance mechanism. At the network, ATM-UBR switches implement intelligent drop schemes to reduce retransmission of packets at the TCP layer, and ATM-ABR switches implement effective closed-loop congestion control schemes to reduce cell loss at the ATM layer.

Both end system polices and network policies interact with each other. These policies are affected by the large round-trip delays of the network. Effective throughput decreases as round-trip delays become longer since the time needed to recover the original data transmission rate after a slow start is prolonged. The fairness among connections is also at risk if the differences of round-trip delays among connections become large.

To tackle the problem of long round-trip delays, an intelligent cell drop scheme for TCP over UBR based on fuzzy logic prediction and control is proposed. The scheme is able to accept or drop a new incoming packet dynamically based on the predicted future buffer condition in the switch. It can thereby compensate the effect of long round-trip delays and react to congestion earlier. In addition, fairness is maintained by a fuzzy logic inference system using soft-computing. Consequently cell loss is kept to a minimum, and high TCP throughput with fairness is delivered.

TCP performance of the proposed fuzzy logic intelligent drop scheme is examined by extensive simulation using different segment sizes, buffer sizes and link distances. For TCP end systems, different congestion control mechanisms are used in the simulation. The effect of fuzzy logic prediction and control on TCP over ATM ABR congestion control is also investigated. The earlier proposed fuzzy logic congestion control scheme for the ABR service is applied to two ABR switch algorithms. The performance is evaluated in terms of effective throughput, fairness, cell loss and cell delay variation.

1.2 Organization of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 provides the necessary background material required for the rest of this thesis. It also establishes the foundation for the understanding of fuzzy sets and fuzzy logic theory, and an introduction to fuzzy logic prediction. It begins with the concept of ATM networks and service architecture including various service categories, QoS, components of traffic contract and the functionality of ATM adaptation layers. Since this research study also investigates the issues of transporting

TCP/IP data over ATM networks, a description of the Internet protocols is provided. Then fuzzy sets and fuzzy logic theory are presented, followed by the fuzzy logic predictor that is incorporated in the proposed fuzzy logic traffic controllers.

Chapter 3 provides an overview of the development of traffic flow and congestion control schemes for the ABR service. It first presents the functionality of the rate-based, closed-loop flow and congestion control framework proposed by the ATM Forum [ATM96]. The design issues of an ABR switch scheme are also addressed. A survey of reported switch schemes is conducted, and their design features and limitations are discussed. In the next chapter, some of the existing switch schemes are further enhanced and evaluated, and a comparative study is conducted between these switch schemes and the proposed fuzzy logic control scheme.

Chapter 4 proposes a novel fuzzy logic congestion control scheme for the ABR service. Its main aim is to address the problem of long round-trip delays in the feedback control loop. It is based on prediction in which link utilization is targeted dynamically one round-trip delay in advance. The performance of the proposed fuzzy logic control scheme is then compared with the conventional control scheme on two switch algorithms. Therefore, for the purpose of comparison, the BEMO [TQ98] switch algorithm is first enhanced by incorporating the conventional congestion control mechanism of the ERICA+ [JKG+96] switch algorithm. The queue control functions of the ERICA+ are also further studied and evaluated.

Chapter 5 focuses on the open-loop connection admission control. It first addresses CAC related issues, such as traffic modeling, connection traffic descriptors and design issues. A survey of various CAC approaches is then conducted. Their design features and limitations are discussed. To exploit the advantages of fuzzy logic prediction and control, a novel predictive fuzzy logic measurement-based CAC is proposed. Admission decisions are based on real-time measurement of aggregate traffic statistics with the fuzzy logic adaptive weight factor as well as the predicted traffic parameters. The measurement interval is first investigated. The benefits of fuzzy logic prediction are then evaluated by applying the prediction technique to the conventional CAC and the proposed measurement-based CAC. Finally, the performance of fuzzy logic measurement-based CAC is compared with the conventional CAC.

Chapter 6 concentrates on the congestion control for transporting TCP/IP application data over ATM networks. The architecture and transportation of TCP/IP over ATM are first discussed, followed by a discussion of the impact of cell loss in ATM networks. A survey of traffic management for TCP over UBR is then conducted. An intelligent fuzzy logic selective cell drop scheme based on the predicted future buffer condition in the switch and fairness among VCs is proposed. The performance of this proposed scheme is then evaluated by comparing it with other conventional UBR cell drop schemes. In addition, TCP performance over ABR is also investigated by comparing the proposed fuzzy logic ABR congestion control scheme (described in Chapter 4) with the conventional control scheme on two ABR switch algorithms.

Finally, chapter 7 provides summaries and conclusions for the thesis. In addition, this chapter also discusses and recommends further direction of research in related areas.

1.3 Contributions

This section provides the reader with a list of the major and other contributions in relation to this thesis. Major contributions of the thesis include the development of traffic control schemes based on fuzzy logic prediction and control which are listed below:

- Development of a novel fuzzy logic congestion control scheme for ATM ABR service. A main feature of the scheme is its ability to target link utilization dynamically based on the predicted future buffer in the switch. This is achieved by using a fuzzy logic predictor and a fuzzy logic target utilization factor generator. Simulation results have demonstrated that the proposed scheme exhibits a faster convergence time in rate and queue, lower buffer requirements and cell loss, and is able to maintain high link utilization over the conventional control scheme.

- Development of a novel fuzzy logic measurement-based CAC for VBR service. The proposed CAC incorporates fuzzy logic prediction, traffic and network performance measurements into the admission decisions process. The key features of the proposed CAC are on-line traffic parameters prediction and aggregate free bandwidth allocation, which is estimated by a fuzzy logic adaptive weight factor estimator. Simulation results have shown that the measurement-based approach incorporating fuzzy logic inference and using fuzzy logic prediction achieves higher link efficiency with guaranteed QoS when compared to the conventional CAC.

- Development of an intelligent fuzzy logic selective cell drop mechanism for TCP over ATM-UBR service. It consists of a fuzzy logic UBR queue predictor and a fuzzy logic selective cell drop factor generator. A key feature of the scheme is its ability to accept or drop a new incoming packet dynamically based on the predicted future buffer condition in the switch. Simulation results have shown that the proposed scheme has achieved significant improvement in efficiency and fairness over the conventional schemes.

Other contributions in this thesis which include the enhancement and investigation of some existing schemes, and the examination of the effects of fuzzy logic prediction and control on traffic control are listed as follows:

- Enhancement of the BEMO switch algorithm by integrating the algorithm with the congestion control strategy of the ERICA+ switch algorithm. It adopts the queue control function of the ERICA+ algorithm. Its main features are a fast convergence time, and its ability to interoperate with other switch schemes that do not mark the ER field of resource management (RM) cells.

- Investigation of the queue control functions employed by the ERICA+ switch algorithm after a minor error was discovered and communicated with the original author. The design of the queue control functions are analyzed and examined by simulation.

- Investigation of the effects of measurement intervals for the measurement-based CAC. Simulation results have demonstrated that longer measurement periods result in lower link utilization. However, shorter measurement periods result in high cell loss though link utilization can be high. This is because there is not enough statistical information on traffic flows to establish the appropriate admission decisions. Therefore the measurement interval has to be short enough to capture the significant statistical information of traffic flows and achieves a higher efficiency.

- Investigation of the effectiveness of fuzzy logic prediction on CAC by applying the prediction technique to the conventional CAC and the proposed measurement-based CAC. Simulation results have shown that fuzzy logic prediction significantly improves link efficiency while maintaining QoS requirements on both CAC schemes.

- Investigation of the effectiveness of the proposed fuzzy logic ABR congestion control scheme on TCP over ABR by comparing its performance with the conventional control scheme. Simulation results have shown that the proposed scheme has achieved higher TCP efficiency with lower cell delay variation and cell loss over the conventional scheme.

Chapter 2

Network QoS and Fuzzy Logic Systems

2.1 Introduction

This chapter provides a description of the background knowledge of network QoS. It also introduces the fuzzy logic predictor that is applied to the proposed fuzzy logic traffic controllers. Firstly, the concept of ATM networks and service architecture which enables multi-services and QoS guarantees over a single network is introduced. Various service categories supported by the ATM networks are then described. Bandwidth and performance requirements for these services are conveyed by means of a traffic contract. Therefore a description of the components in a traffic contract such as traffic descriptors and QoS parameters are also provided. These components define parameters that are negotiated during the connection establishment phase. It is possible to have different technologies and protocols transported over an ATM infrastructure. This is enabled by the functions provided by the ATM adaptation layers (AALs). Section 2.2.4 briefly describes the functions of the AALs.

Traditionally, IP provides a single level of service with no service guarantees. Due to the increasing demand of multimedia services, many researchers are currently investigating methods that provide multi-services and QoS guarantees over IP. In the meantime, ATM continues to grow in the backbone of IP networks because of its high optical transmission rates (155 *Mbps*, 622 *Mbps* and higher) and its attracting feature of providing different classes of service on the same fiber. Therefore, a chapter has been devoted later in this thesis to investigate the performance issues involved with running packet-based data communications over ATM networks. Section 2.3 of this chapter thus provides an overview of the Internet protocol and its supported protocols. It also briefly discusses the current research on providing QoS over IP networks.

Section 2.4 provides an overview of the theory of fuzzy sets and fuzzy logic. It also discusses the structure and various functional modules of a fuzzy logic controller. Then in Section 2.5, the fuzzy logic predictor is introduced. It is incorporated in the proposed fuzzy logic traffic controllers later in this thesis.

Finally, in Section 2.6, a brief description of the simulation software package used in this research study is provided. It also outlines the necessary modifications to the software for the specific needs of this research study.

2.2 ATM Networks and Quality of Service (QoS)

ATM technology was developed to deliver end-to-end QoS, and the ability to support virtually all types of traffic, such as voice, video and data applications from a single network infrastructure. The benefits of converging onto a single network are ease of management, support, monitoring, upgrading and is thus less expensive to maintain compared with separate networks for particular types of traffic.

ATM is a connection-oriented technology and information is carried in a fixed-length cell consisting of 48 bytes of payload and 5 bytes of header. The fixed-length cell enables high-speed switching and simplifies the engineering of queue and switch resources. Bandwidth is allocated on-demand and is statistically multiplexed which enables efficient used of limited resources (bandwidth, buffers, and so forth). ATM supports both permanent virtual circuit (PVC) and switched virtual circuit (SVC). PVC is created between two points with a dedicated bandwidth that is pre-assigned

	ATM Layer Service Category				
Attribute	CBR	rt-VBR	nrt-VBR	UBR	ABR
Traffic Descriptors:					
PCR and CDVT :	Specified				
SCR, MBS, CDVT :	NA	Specified		NA	
MCR	NA				Specified
QoS Parameters:					
peak-to-peak CDV	Specified		Unspecified		
maxCTD	Specified		Unspecified		
CLR	Specified			Unspecified	
Feedback	Unspecified				Specified

Table 2.1: Specifications of ATM Service Categories [ATM96].

manually. SVC is setup dynamically on a call-by-call basic using a call-setup mechanism. These virtual connections enable flexible traffic engineering to adapt to different traffic patterns.

ATM supports five different service categories. Each service category has a set of traffic parameters and performance QoS parameters that govern the traffic over a given virtual end-to-end connection. Table 2.1 defines the ATM service categories, their traffic parameters and QoS parameters.

Basically, QoS in ATM networks can be categorized into *guaranteed service* and *best effort service*. Guaranteed services are for real-time applications that are sensitive to delay and cell loss. Service classes such as constant bit rate (CBR) and variable bit rate (VBR) belong to this category. In contrast, best effort services are for non real-time applications such as data communication which can gracefully adapt to the bandwidth available in the network. Unspecified bit rate (UBR) and available bit rate (ABR) are the service classes of this category.

2.2.1 Service Architecture

The following functions are defined by the ATM Forum [ATM96] and the International Telecommunication Union (ITU-T) [ITU96] to manage and control traffic and congestion in ATM networks so as to meet the QoS requirements of different services:

- **Connection Admission Control (CAC):** defines a set of actions performed by the network during the call setup phase to accept or reject a new connection request. The decision is made based on the availability of network resources (e.g. bandwidth, buffer, etc.) along the traversing path of the new connection, the service category and the desired QoS without violating the QoS requirements of existing connections.

- **Feedback Control:** is a mechanism provided by the network to source end systems in order to regulate the transmission rate of sources according to the state of network elements.

- **Usage Parameter Control (UPC):** is a set of actions taken by the network to ensure an admitted connection remains within its requested contract and does not violate the negotiated parameters. Its main aim is to protect network resources from misbehaved users that can affect the QoS of other existing connections.

- **Cell Loss Priority Control:** The source end system of some service categories may generate cells with different priorities by marking the cell loss priority (CLP) bit. During congestion, the network may selectively discard low priority cells to protect the network performance of high priority cells.

- **Traffic Shaping:** is a mechanism that modifies the traffic characteristics of a stream of cells of a connection in order to achieve better network efficiency whilst meeting the QoS objectives.

- **Network Resource Management (NRM):** provides separation of traffic flows and allocation of network resources according to different service characteristics. This function deals with virtual paths on resource management in order to meet QoS requirements.

- **Frame Discard:** In the event of congestion, the network may intelligently discard all the cells from selected packets instead of dropping random cells from multiple packets. This improves data throughput because of a reduction in retransmission of unnecessary cells.

- **ABR Flow Control:** ensures the ABR connections adaptively and fairly share the bandwidth left-over by the higher priority services.

2.2.2 Service Categories

The following service categories are defined by the ATM Forum [ATM96]:

- **Constant Bit Rate (CBR):** This service is designed to support real-time applications such as voice. It is characterized by the peak cell rate (PCR), and requires guaranteed bandwidth and delay from the network during the entire life of the connection.

- **Real Time Variable Bit Rate (rt-VBR):** This service is intended for bursty real-time applications such as video and packetized voice which has strict end-to-end delay requirements from the network. It is characterized by PCR, sustainable cell rate (SCR) and maximum burst size (MBS).

- **Non-real Time Variable Bit Rate (nrt-VBR):** As opposed to rt-VBR service, this service does not guarantee any delay bounds and is used to support bursty real-time applications such as frame relay. Similar to rt VBR service, it is characterized by PCR, SCR and MBS.

- **Unspecified Bit Rate (UBR):** This service is introduced to transport traditional data such as IP over ATM. The connections share the remaining bandwidth left-over by CBR and VBR services and have no specific bandwidth requirements, delay or loss rate. Therefore, this service is often referred to as a "best-effort" service. During congestion, it relies on higher-layer protocols at the end system to perform cell loss recovery and retransmission mechanisms.

- **Available Bit Rate (ABR):** This service is also designed to transport data applications such as electronic mail and file transfer. The connections adapt to the available network capacity by using a well-defined rate-based flow and congestion control mechanism (Chapter 3). The transmission rates are adjusted by sources within the boundary of PCR and minimum cell rate (MCR) according to the network situations. Although the standard does not impose any delay and loss restrictions, it is desirable for switches to keep the delay and loss as low as possible.

2.2.3 Traffic Contract

At a connection setup phase, the user and the network negotiate a traffic contract that both parties have to adhere to throughout the entire life span of the connection. The traffic contract consists of source traffic descriptors, QoS requirements and the conformance definition (i.e. the definition of traffic behavior). The traffic descriptors define the traffic characteristics in which the source promises to maintain. The QoS parameters define the QoS that the network promises to guarantee. The following sections discuss the traffic descriptors and QoS parameters. The conformance definition is out of scope for this research study but details can be found in [ATM96, ITU96].

2.2.3.1 Traffic Descriptors

Source traffic descriptors are used to describe the traffic characteristics of an ATM connection. They are defined by one or more of the following parameters:

- **Peak Cell Rate (PCR):** The PCR defines the maximum emission rate of the source. It can be limited by the physical link speed or by the traffic shaper at the edge of the network.
- **Sustainable Cell Rate (SCR):** The SCR defines the upper bound of the average transmission rate of the conforming cells.
- **Maximum Burst Size (MBS):** The MBS defines the maximum number of cells that can be sent by the source at PCR while still conforming to the negotiated SCR.

- **Minimum Cell Rate (MCR):** The MCR defines the minimum bandwidth guaranteed by the network.

Besides the above four parameters, the cell delay variation tolerant (CDVT) parameter of a connection is also included in the source traffic descriptors. It defines the maximum cell delay variance of the connection and is used to verify the conformance of the traffic behavior to the source traffic descriptors.

2.2.3.2 QoS Parameters

QoS parameters are used to measure the end-to-end network performance of a connection at the ATM layer. The following QoS parameters are negotiated during the connection establishment phase:

- **Maximum Cell Transfer Delay (maxCTD):** The CTD is the delays accumulated by a cell between network entry and exit points. These delays include internal and external queuing and transmission delays at each node, propagation delay on each link and processing delay. The maxCTD specifies the upper limit of CTD.

- **Peak-to-peak Cell Delay Variation (peak-to-peak CDV):** Since the queuing delay varies from one cell to another, there will be a variation in the cell delay measurement. The peak-to-peak CDV defines the difference between the maximum CTD and the minimum CTD (which is the fixed delay, such as the propagation delay).

- **Cell Loss Ratio (CLR):** Cell loss occurs when the limited buffer is overflown due to simultaneous arrivals of large bursts from different connections, or through a network component failure. The CLR is defined as the ratio of lost cells to the total transmitted cells. It is based on per connection measurements.

2.2.4 ATM Adaptation Layers

In order for an ATM network to support many other services (e.g. frame relay, voice, TDM) with different traffic characteristics and system requirements, it is necessary to adapt the different technologies and protocols to the ATM layer. However the ATM network provides only cell-based transport, the ITU-T thus established the ATM adaptation layer (AAL) to map the application data streams into the cell-based payloads for the underlying ATM network. The functions provided by the AAL are end-to-end and service-dependent. The four AALs defined in the ITU-T specification [ITU93] are:

- **AAL1:** Support CBR service such as voice which has specific timing and strict delay requirements.
- **AAL2:** Support VBR service which is delay sensitive.
- **AAL3/4:** The AAL3 was designed to transport connection-oriented variable bit rate data services (e.g. X.25 or Frame Relay), and AAL4 was intended for connectionless traffic (e.g. IP). They have been merged to form a single AAL3/4.
- **AAL5:** Support connection-oriented variable bit rate data services. It is intended to reduce the overhead and the complexity of the AAL3/4.

The AAL is composed of two sub-layers:

- **Convergence Sublayer (CS):** This layer encapsulates higher-layer user application data, and provides transmission and error-detection of user data.
- **Segmentation and Reassembly Sublayer (SAR):** This layers segments the encapsulated frame from the CS into 48-octet ATM cell payloads. It also provides cell-based transmission and error-detection. At the destination end system, the SAR reassembles the cells into data units to be delivered to the higher layers.

2.3 The Internet

The Internet is a global interconnected network connecting millions of users worldwide via many computer networks using a simple standard addressing system and communication protocols defined by Internet Standards. It has been growing exponentially over the past decade due to the high demand in world wide web access, electronic mail services, remote computer access, etc. The Internet Engineering Task Force (IETF) is the body responsible for the specifications development and reviewing of Internet standards. Historically, the Internet offers a single level of service, i.e. "best effort" service with no service guarantees. Due to the emerging demand on multimedia applications which require service guarantees over the Internet, the IETF working groups are currently investigating how to provide QoS on Internet services.

The Transmission Control Protocol (TCP) and Internet Protocol (IP) are the most widely used protocols in the existing Internet. The following sections provide a brief overview of these protocols.

2.3.1 Internet Protocol (IP)

IP is a connectionless protocol that currently provides unreliable, best-effect packet delivery service with no guaranteed QoS. It is a network layer protocol which is responsible for routing packets from one host to another. IP packets are stamped with the addresses of the sender and the receiver before delivery, and are not guaranteed to be received in the same order as when created.

2.3.2 Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol that provides reliable end-to-end delivery of information for the application layer. It is a transport layer protocol and is supported underneath by the IP protocol. Information is first segmented into segments. Each segment is then numbered in sequence when transmitted. The destination TCP layer rearranges the received segments in proper order by keeping track of each sequence number. Data integrity is also checked by using 32-bit cyclic redundancy check (CRC). Segments are then acknowledged by the destination TCP layer. Error recovery and lost segments are retransmitted by the source TCP layer when its retransmission timer timeout.

TCP supports applications such as *email* (electronic mail), *telnet* (terminal access), file transfer protocol (FTP) and world wide web (WWW) access which require reliable transfer of information.

2.3.3 QoS in IP

The current Internet mostly uses IPv4 which does not provide QoS guaranteed for higher-layer applications. As a result, they are mainly for data communications applications, such as file transfer and e-mail services; they are not for delay constraint services like voice and video.

The IETF is now addressing this problem by proposing several new schemes to provide QoS in IP. One of the schemes is to improve IPv4 to IPv6 which includes a source identification (flow labelling) field. This allows unique identification of different data streams. Another approach is to use the integrated services model (IntServ) which was the earliest QoS delivery model developed for the Internet. It uses a signalling protocol, known as resource reservation protocol (RSVP, RFC2205 [BZB+97]) to signal bandwidth requirements. This scheme may work well with a small network. However it does not scale well in large networks due to its need to maintain state information for a vast amount of VCs. Therefore the differentiated services model (DiffServ, RFC2475 [BBC+98]) emerges next. DiffServ supports priority service in IP network by marking the type-of-service/precedence field in IPv4 and in the traffic class field in IPv6. A further development in switch/router architecture approach is to use the multi-protocol label switching (MPLS, RFC3031 [RVC01]). It expedites traffic flow by setting up an end-to-end layer-2 (actually it is between layer-2 and layer-3) *label path* through the network.

Hence each IP packet has a label attached which identifies the next hop in the network. This label also serves as an indication for particular QoS requirements.

At present, all these IP QoS schemes are only a work in progress. Moreover, sophisticated charging mechanisms for differentiated QoS do not exist. Currently, most Internet service providers (ISP) adopt ATM as a high-speed backbone for transporting Internet services. Hence, this thesis has devoted a chapter (see Chapter 6) in performance issues involved with running TCP-based data communications over ATM networks.

2.4 Fuzzy Logic Control Systems

In recent years, computational intelligence techniques, such as fuzzy logic controls, have been demonstrated to solve some complex, non-linear and uncertainty problems. The theory of fuzzy sets and fuzzy logic have emerged following the pioneering works of Zadeh [Zad73, Zad65].

Unlike traditional control systems which require exact mathematical equations and precise numerical values, fuzzy logic control systems provide an effective way to find solutions based on the approximate nature of human decision making. To achieve this, linguistically expressed knowledge of an expert about a given process is converted into mathematically defined control strategies. This is particularly crucial in control systems which have nonlinearity or time-varying properties, or where it is difficult to obtain precise measurements. As a result, the ability to define precise mathematical models for such control systems becomes difficult or even impossible. However it has been observed that the approximate nature of human knowledge and reasoning are generally able to overcome these difficulties and perform well under these conditions.

The concept of a fuzzy set is an extension of a traditional crisp set. A crisp set is dichotomous, i.e. yes-or-no type whereas a fuzzy set is vague, i.e. more-or-less type. A fuzzy set associates with all objects in a universe of discourse (i.e. the

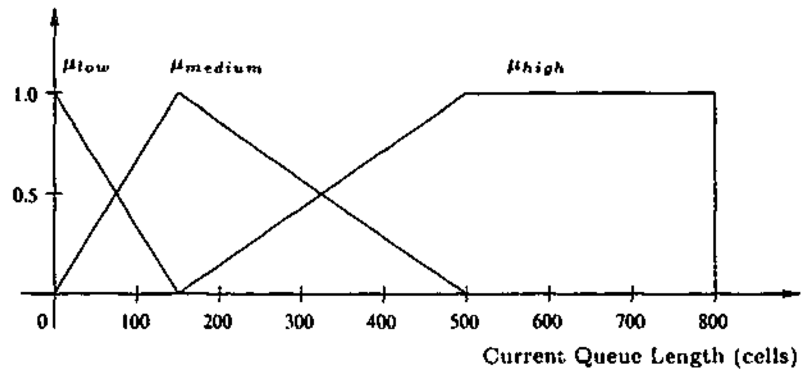


Figure 2.1: Membership functions of the linguistic values for representing the linguistic variable "current queue length".

input space) a membership function. A membership function is a curve that defines how strong an object belongs to the fuzzy set and is mapped to a membership value (i.e. the output space) between interval [0,1]. The curve of a membership function can be linear, triangular, trapezoidal, Gaussian, sigmoid or polynomial, and is often designated by μ. In this research study, the triangular and trapezoidal membership functions are adopted.

As an example, Figure 2.1 shows the membership functions of a "current queue length". From the figure, it shows that the "current queue length" has three membership functions: μ_{low}, μ_{medium} and μ_{high}, represented by triangular and trapezoidal curves. The membership value is *1.0* if the current queue length exceeds 500 cells in which a buffer is considered highly filled. The "current queue length" can also be described as a *linguistic variable*, and the values low, medium and high are *linguistic values* [Zad75].

The heart of a fuzzy logic control system is a list of IF-THEN rules describing the control actions for a given process based on the knowledge of an expert. Hence, a fuzzy logic control system is also referred to as a knowledge-based system or a rule-based system. The rules are statements characterized by linguistic variables and their linguistic values. These statements can be conjuncted with AND, OR,

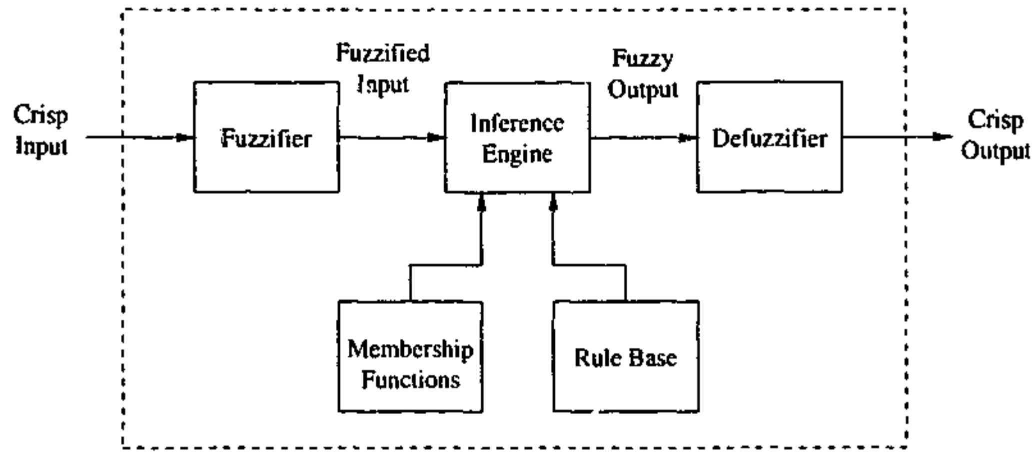


Figure 2.2: Generic Structure of a Fuzzy Logic Controller.

or NOT fuzzy operators if a rule has more than one part. For example:

IF current queue length is high
AND current queue growth rate is high
THEN target utilization rate is low

Note that "current queue length" and "current queue growth rate" are input linguistic variables, whereas "target utilization rate" is an output linguistic variable. The words "high" and "low" are an input and an output linguistic value respectively.

Fuzzy Logic Controllers

Fuzzy logic controllers are the application of fuzzy sets and fuzzy logic in control theory. A typical fuzzy logic controller consists of a fuzzifier, a defuzzifier, an inference engine, membership functions and a rule base as shown in Figure 2.2.

The operation of a fuzzy logic controller involves the following four phases:

1. **Fuzzification:** maps the crisp input values into fuzzy sets via their membership functions and linguistic values according to the fuzzy degree of membership of each input data.

2. **Inference:** produces new fuzzy sets based on the linguistic values from the

fuzzification process and the rules in the rule based, i.e. one fuzzy subset is assigned to each output variable of each rule.

3. **Aggregation:** combines the output fuzzy subsets of each rule from the inference engine into a single fuzzy set.

4. **Defuzzification:** maps the output fuzzy set from the aggregation process to a crisp control output value.

Mamdani's fuzzy inference method [MA75] is the most widely used inference method and it is the method that this research study chooses to incorporate in the proposed fuzzy logic control systems. Another well-known inference method is the so-called Sugeno or Takagi-Sugeno-Kang method [Sug85]. The Mamdani's method produces a distributed fuzzy set for the aggregation process. A centroid method is adopted during the defuzzification process to find a point representing the center gravity of the fuzzy set. In contrast, Sugeno's method produces a single spike output membership function, and a weighted average of a few data points is produced during the defuzzification process. In general, the Sugeno's method is suited for any inference systems in which the output membership functions are either linear or constant.

The aggregation method is commutative. Thus the order of executing the rules is not important. Usually *max* (maximum) or *sum* are used. In *max*, the combined output fuzzy set is constructed by taking the point-wise maximum over all of the fuzzy subsets of each rule from the inference engine. In *sum*, the combined output fuzzy set is simply the sum of each output fuzzy subsets from the inference engine. In this research study, the less computational intensive *max* method is used.

As for the defuzzification process, other than the centroid method and weighted average method mentioned previously, there are numerous other methods available, such as bisector, middle of maximum, largest of maximum and smallest of maximum [Jan98].

2.5 Fuzzy Logic Prediction

This section provides a description of the fuzzy logic prediction that is applied to the traffic controllers in this thesis. In [KK92], Khedkar and Keshav introduce an exponential average predictor based on fuzzy logic control. It is a variant of the Kalman predictor in which parameters can be adjusted according to the estimation errors with a fuzzy feedback system. It was originally designed to make one-step ahead prediction in an environment where system and observation noise are present. Based on this single step ahead prediction, a multi-step ahead prediction in which system and observation noise do not exist is introduced in [Qiu01, Qiu98]. Simulation results show that fuzzy logic prediction outperforms the traditional AR predictions. It has a smaller error mean and standard deviation compared to the AR predictions.

One-step ahead prediction can be expressed as:

$$\hat{Q}_{t,1} = \alpha\hat{Q}_{t-1,1} + (1-\alpha)Q_t \tag{2.1}$$

where $\hat{Q}_{t,1}$ is the one-step ahead predicted value of Q_{t+1} at time t, and $\hat{Q}_{t-1,1}$ is the past predicted value of Q_t at time $t-1$.

The α in Equation (2.1) can be considered as the weight given to the past history. The larger the value of α, the more weight is given to the past history. That is, if the prediction error is small, then the value of α should be large and vice versa. Since the distribution of Q_t is unknown, a fuzzy tuning system is the simplest solution that takes a prediction error as an input and α as an output. This is illustrated in Figure 2.3.

Since α is within the range of $[0,1]$, and the prediction error is unbounded, proportional (relative) error is used. It is defined as $|Q_t - \hat{Q}_{t-1,1}|/Q_t$. The fuzzy linguistic rules can be easily established for the inference of α with fuzzy linguistic values small, medium and large:

- if *proportional error* is large then α is small

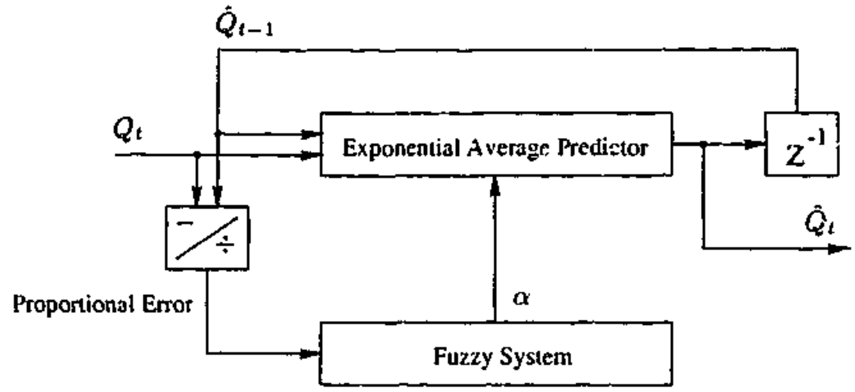


Figure 2.3: Fuzzy Logic Exponential Average Predictor.

- if *proportional error* is medium then α is medium
- if *proportional error* is small then α is large

The input/output membership functions of the fuzzy controller are shown in Figure 2.4. Triangular shape membership functions are adopted for both input and output linguistic variables. The memberships of small and large are centered at the two ends of the considered range with maximum support for all memberships. From the simulation, input and output values of 0.3 and 0.6 respectively give the best performance for the central values of medium linguistic values. The output variable, α is then defuzzified by a standard centroid defuzzifier, and scaled up to fit the response range [0,1].

By extending and modifying Equation (2.1), multi-step ahead prediction can be obtained as:

$$\hat{Q}_{t,L} = \alpha\hat{Q}_{t-1,L} + (1-\alpha)\hat{Q}_{t,L-1} \tag{2.2}$$

This is a L-step ahead prediction. Substituting L equal to one, Equation (2.2) becomes Equation (2.1). Equation (2.2) is a recursive expression. It can be implemented practically by a one-step ahead prediction with a delay unit. The procedure

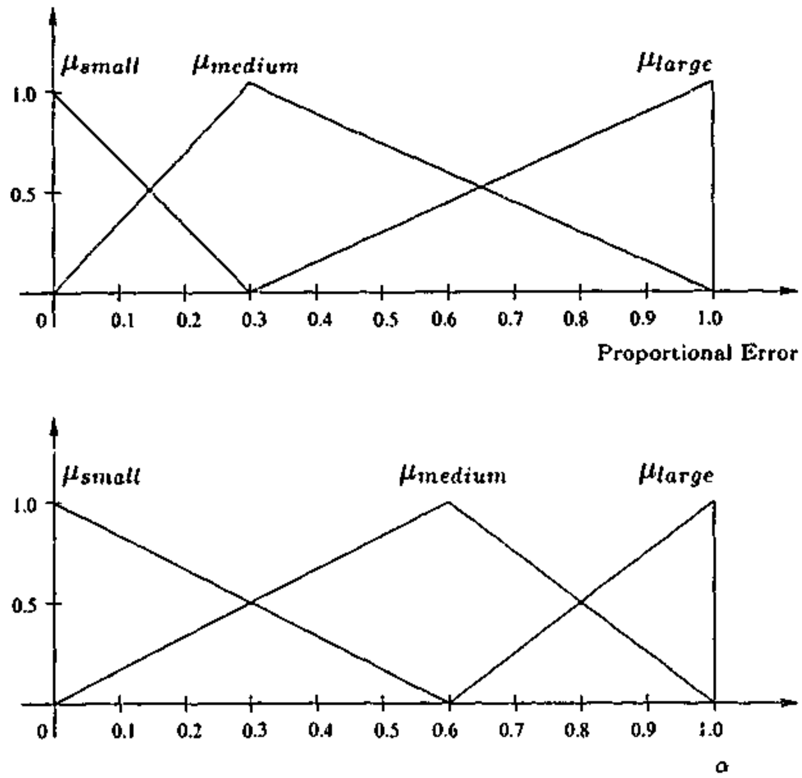


Figure 2.4: Membership functions of the linguistic values for representing the linguistic variables "*proportional error*" and α.

of operation can be summarized in an $(L+1) \times N$ array:

$$\begin{array}{llllll}
\boldsymbol{Q_t} & Q_{t+1} & Q_{t+2} & \cdots & Q_{t+N-2} & Q_{t+N-1} \\
\boldsymbol{\hat{Q}_{t-1,1}} & \boldsymbol{\hat{Q}_{t,1}} & \hat{Q}_{t+1,1} & \cdots & \hat{Q}_{t+N-3,1} & \hat{Q}_{t+N-2,1} \\
\boldsymbol{\hat{Q}_{t-2,2}} & \boldsymbol{\hat{Q}_{t-1,2}} & \boldsymbol{\hat{Q}_{t,2}} & \cdots & \hat{Q}_{t+N-4,2} & \hat{Q}_{t+N-3,2} \\
 & & & \cdots & & \\
\boldsymbol{\hat{Q}_{t-L,L}} & \boldsymbol{\hat{Q}_{t-L+1,L}} & \boldsymbol{\hat{Q}_{t-L+2,L}} & \cdots & \hat{Q}_{t-L+N-1,L} & \hat{Q}_{t-L+N,L}
\end{array}$$

This array consists of predicted results, current and future observed values. Those values which are available at time t are highlighted in bold. It can be observed that prediction from one to L steps ahead are all available at the same time.

This multi-step ahead fuzzy logic predictor is quite attractive because of its adaptive feedback mechanism which can respond to changes in system behavior dynamically. In addition, the three rules are able to cover the entire range of possible scenarios. Therefore, it would be of interest to apply this predictor in high-speed networks where traffic or network resources can be predicted far enough in advance so that one can control traffic flows or resource usage in time to avoid congestion and achieve high network utilization.

2.6 Simulation Modeling

Performance analysis for this research study was conducted using a combination of experimental design, simulation and analysis. An event-driven simulation software package was used to model various components and network configurations of an ATM network. It is originally developed at MIT [HR94] and is called the *netsim* simulator. It was then modified and extended by the National Institute of Standards and Technology (NIST) [GMH+98]. This simulation software package is written in the 'C' programming language under the Linux platform. It has a X-window graphic user interface (GUI) which provides an interactive modeling environment.

The simulator consists of two parts: a core events handling module that schedules and fires up events, and a set of components that communicate with one another by passing messages using a standard interface with the core events handling module. Source/destination end systems, physical links and switches are the examples of components.

The simulator has been modified based on the specific needs of this research study. In addition, the fuzzy inference engine has also been incorporated in the simulator by embedding the stand-alone source codes from the MATLAB fuzzy logic toolbox [Jan98]. The following are the modifications and additions:

- Modification of the source/destination component that generates Poisson and on-off traffic patterns.
- Addition of fuzzy logic prediction and congestion control modules to the switch component that performs the ABR fair rate allocation algorithms.
- Addition of conventional CAC schemes and the proposed fuzzy logic measurement-based CAC scheme to the switch component.
- Addition of the selective drop (SD) [GJK+98, GJK+97] packet discard scheme

and the proposed fuzzy logic selective cell drop scheme to the switch component.

2.7 Conclusions

This chapter provides the necessary background information for this research study, in particular, QoS and ATM service categories, Internet protocols, fuzzy sets and fuzzy logic theory, and the fuzzy logic predictor.

Subsequent chapters discuss in more detail the three aspects of traffic management: (1) flow and congestion control for ABR service, (2) connection admission control for VBR service and (3) congestion control for TCP/IP traffic over ATM. The fuzzy logic prediction and control are applied in these areas for the improvement of network efficiency and QoS of various services. Performance issues such as efficiency, fairness, buffer requirements are addressed and investigated.

The next chapter begins with the traffic flow and congestion control for the ABR service. It provides an overview of the development of ABR switch schemes, their design features and limitations.

Chapter 3

Traffic Flow and Congestion Control

This chapter provides an overview of the functionality of the rate-based, closed-loop flow and congestion control framework for the ABR service proposed by the ATM Forum [ATM96], and the development of ABR switch schemes. A survey of existing switch schemes is presented. Their fair share allocation algorithms, congestion detection techniques and control mechanisms are discussed. Some of these switch schemes are used for the purpose of comparison in the next chapter.

3.1 Introduction

To support a variety of multimedia services through BISDN/ATM networks, several types of service with different QoS requirements have been proposed by the ITU-T and ATM Forum [ITU96, ATM96]. CBR and VBR services are intended for real time applications such as voice and video. These services have constrained QoS requirements on cell loss ratio (CLR), cell transfer delay (CTD) and delay variation (CDV). In contrast, ABR service is intended for non-real time applications such as data communications. This service can tolerate unpredictable end-to-end cell delays, and accept reasonably low cell loss.

The ABR service guarantees a fair share of the available bandwidth left-over by the VBR and CBR services to an end system that adapts its transmission rate according to the feedback received from the network. Therefore a congestion control scheme is necessary at a switch for the support of ABR traffic to utilize the available bandwidth without causing congestion, and a fair rate allocation algorithm is needed to ensure fair allocation of that bandwidth among ABR connections. A flow control mechanism is also required to allow ABR sources to dynamically adapt to this time-varying bandwidth in the network.

The ATM Forum has adopted a closed-loop rate-based flow and congestion control scheme as the standard for the ABR service [BF95]. Resource management (RM) cells are defined to carry feedback information from the switches (or network) to the corresponding end systems. The source and destination behaviors are also standardized, but switch behavior is loosely defined which allows flexibility for future development. Nonetheless, the specifications defined by the ATM Forum [ATM96] require each congestion control switch to implement at least one of the following methods:

(1) *Explicit Forward Congestion Indication (EFCI) marking*: the switch sets the EFCI bit in the data cell header to indicate congestion, and relies on the destinations to convey congestion back to the sources by marking the congestion indication (CI) bit in the RM cells. This is the first-generation switch before the RM cell was fully defined.

(2) *Relative Rate (RR) marking*: the switch sets the CI or no increase (NI) bit in forward and/or backward RM cells to provide congestion status to each sources.

(3) *Explicit Rate (ER) marking*: the switch marks the ER field of the RM cells explicitly to convey the allowed rate to each source.

Several switch algorithms have been proposed and discussed within the ATM Forum and in the literature. The key issues of these algorithms are the computation of fair share of the available bandwidth in which each connection should converge, the congestion detection techniques, and the congestion control mechanisms to deal with the build up of switch queues during transients.

This chapter provides necessary background information on the rate-based ABR service and a survey of various existing switch schemes. Some of these switch schemes are used for the purpose of comparison in the next chapter. This chapter begins with the presentation of the rate-based flow and congestion control framework for the ABR service, followed by a brief description of its fundamental operation. Design issues associated with a switch scheme are then discussed. A commonly used fairness definition is then defined in Section 3.5. Finally, various switch schemes are presented and discussed in Section 3.6.

3.2 The Rate-based Congestion Control Framework

Figure 3.1 illustrates the main elements of an ABR service rate-based closed-loop flow and congestion controlled communication network [BF95]. The control loop requires the active cooperation of network elements, the source, the destination and the switch. Information about the sources and feedback from the switches (or networks) is carried by the RM cells. Traffic sources are thus able to dynamically regulate the transportation of data through the network between sources, and attempt to avoid congestion. The following sections describe the functions and some specific implementations of these network elements.

3.2.1 Source End System and Destination End System

Typical source end system (SES) and destination end system (DES) are situated at the end points of an ATM virtual connection, in the terminal adaptors, or network interface cards. Cells are generated from the SES. They then traverse through the network path, and are received by the DES. Sources are connected to their intended destination through the network bi-directionally, that is, a forward

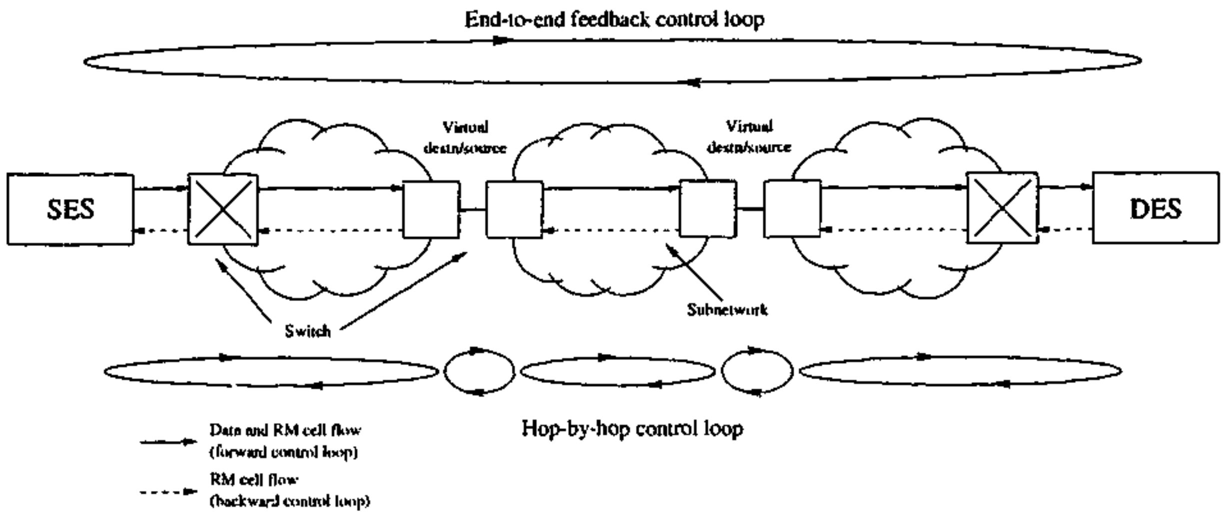


Figure 3.1: A Rate-based Closed-loop Flow and Congestion Controlled Communication Network.

(from SES to DES) and a backward (from DES to SES) path. An end system can be considered as both a SES and DES if a particular ABR service requires bi-directional transmission.

The main functionality of SES is its ability to schedule and adjust the rate of cells submitted into the network. It also generates forward RM (FRM) cells which are used to collect and convey the congestion state of each node along the traversing path to the DES.

The function of a DES is to keep track of the congestion state of a connection. Upon reception of a FRM cell, the DES transmits a backward RM cell (BRM, or turns around the FRM cell by setting the directional bit) back to the source. The content of the backward RM cell is the same as a FRM cell, except the CI or NI bit may be set to indicate congestion. A DES may generate a backward RM cell to indicate its congestion status without having received a forward RM cell. The rate is limited to 10 cells/second, per connection.

3.2.2 Network Switch

A network switch is able to route cells from sources to destinations by providing essential resources (e.g., port bandwidth, buffer space, etc.). These network resources are shared by multiple services and thus resource contention may arise.

As a result, a high level of cell loss or an excessive cell delay may occur. Hence a queuing structure is required to temporarily store cells. Queuing structures are generally required at the input port, multiplexers, switching fabric, demultiplexers and output port.

A queuing structure can be classified as

1. **Per-group queuing** where many connections share the same queue in a first-in-first-out (FIFO) arrangement. A typical group consists of connections belonging to the same service category, service class or conformance definition.

2. **Per-VC/VP queuing** where the cells of each VC or VP are queued independently.

The switch also implements a cell scheduling mechanism at each queuing structure for port bandwidth management, and a cell discard strategy for cell buffer management. In addition, a congestion detection mechanism is also implemented at the switch in order to monitor the level of congestion, and therefore provide congestion indication to the end systems.

3.2.3 Closed-loop Feedback Control

The ABR service is supported by an end-to-end closed-loop per-connection feedback flow control mechanism [ATM96]. End-to-end means the RM cells (which carry rate feedback) travel from the source to the destination and back to the source. Based on the feedback information from the network, traffic sources regulate their data transmission rate so that network congestion is controlled or even prevented, and the available bandwidth from the network can be highly utilized.

The adoption of feedback flow control mechanism is initiated by the successful experience gained with other data network protocols (e.g. TCP/IP) [Jac88, FR94].

However in TCP, the control is "window-based" and the sources limit their transmission to a particular number of packets.

Recent proposals have shown that an ABR service can also support flexible QoS over ATM backbones on the emerging multimedia applications [LMR97] and Internet technologies [FJR+99]. Hence the control loop at the ATM layer has to interact efficiently with other control protocols (e.g. TCP/IP) at higher levels.

3.2.4 Virtual Source/Virtual Destination

The large control latency resulting from a long round trip delay in a high speed ATM network makes it difficult to implement an end-to-end rate-based control scheme. Excessive QoS degradation or bandwidth under-utilization may result due to the delay control messages which are usually inaccurate for most current network conditions. One possible solution is to segment the end-to-end rate control loop into smaller hop-by-hop control loops and implement "Virtual source" (VS) and/or "Virtual destination" (VD) at the intermediate switch [Kea96] (as shown in Figure 3.1).

An intermediate switching element can thus reduce the size of the feedback loop by functionally behaving as a DES as well as a new SES. On one end of the segment, the switch functions as a DES, i.e. it receives data cells and turns around RM cells to the SES. On the other segment, the switch functions as a SES, i.e. it generates RM cells, regulates and schedules its data transmission rate.

With a shorter control loop, feedback to the source end systems is faster, and thus performance can be potentially improved. Another advantage is to isolate different networks from each other. For example, between a satellite network and a LAN. However VS/VD approach can be quite expensive to implement as it requires the conversion of per-class queue to per-VC queue and the inclusion of all source/destination related functions at each switch.

Acronym	Description	Negotiation	Default Value
PCR	Peak Cell Rate	down	-
MCR	Minimum Cell Rate	down to MCR_{min} if MCR_{min} is signalled.	0
ACR	Allowed Cell Rate	no	-
ICR	Initial Cell Rate	down	PCR
TCR	Tagged Cell Rate	constant	10 cells/sec
N_{rm}	Maximum number of cells between FRM cells	no	32
M_{rm}	Controls bandwidth allocation between FRM,BRM and data cells	constant	2
T_{rm}	Upper bound on the time between FRM cells	no	100 $msec$
RIF	Rate Increase Factor	down	1/16
RDF	Rate Decrease Factor	down	1/16
ADTF	ACR Decrease Time Factor	down	0.5 $msec$
TBE	Transient Buffer Exposure	down	16,777,215
CRM	Missing RM-cell count	computed after call setup	TBE/N_{rm}
CDF	Cutoff Decrease Factor	up	1/16
FRTT	Fixed Round Trip Time	accumulated	set by source to fixed source delay

Table 3.1: List of ABR service parameters.

3.3 Fundamental Operation

In ABR service, the SES creates a connection by sending a call request to the network. During the call setup phase, the SES declairs ABR-specific parameters. Table 3.1 shows some of the parameters which are defined by the source (e.g. PCR or MCR) as well as others that are provided by the network (RIF, RDF, N_{rm}). Complete definition of ABR service parameters can be found in [ATM96].

Once the SES is granted permission, it begins to transmit cells at the allowed cell rate (ACR). The ACR is initially set to initial cell rate (ICR) and is dynamically varied between the minimum cell rate (MCR) and peak cell rate (PCR).

Field	Description	Length	Notes
Header	ATM Header	5 bytes	PTI = 110 (binary)
ID	Protocol ID	1 bytes	1 = ABR service
DIR	Direction	1 bit	0 = forward, 1 = backward
BN	Backward Notification	1 bit	1 = non-source generated RM cells, 0 = source generated RM cells
CI	Congestion Indication	1 bit	1 = congestion, 0 = otherwise
NI	No increase	1 bit	1 = no additive increase allow
RA	Request/Acknowledge	1 bit	0 or set in accordance with I.371
Reserved		3 bit	0
ER	Explicit Rate	2 bytes	Limit the source ACR to a value that the network elements and the destination can sustain.
CCR	Current Cell Rate	2 bytes	Set by the source to its current ACR.
MCR	Minimum Cell Rate	2 bytes	Minimum cell rate of the connection.
QL	Queue Length	4 bytes	0 or set in accordance with I.371
SN	Sequence Number	4 bytes	0 or set in accordance with I.371
Reserved		30.75 bytes	6A(Hex) for the first 30 bytes and 0 for the last 6 bits
CRC-10	Cyclic Redundancy Check	10 bits	Verify the accuracy of the content.

Table 3.2: Resource Management (RM) Cell Fields.

In order to probe network congestion status, the SES sends a forward RM cell every $N_{rm} - 1$ data cells. Table 3.2 describes the RM cell structure. The source assigns the ACR value in the current cell rate (CCR) field of the RM cell, and its desired transmission rate (usually the PCR value) in the ER field. The RM cells traverse forward through the network, and are turned around by the destination in the backward path. Switches along the path notify the end systems of the congestion status by marking the EFCI bit in the data cells and the CI and NI bit in the RM cell, and/or reducing the ER value in the RM cells.

When the backward RM cell is received by the source, it modifies the ACR based on the information carried by the RM cell. The ACR is determined by:

$$ACR = \begin{cases} max(min(ACR + RIF \times PCR,\ ER),\ MCR) & if\ CI\ \&\ NI = 0, \\ max(min(ACR \times (1 - RDF),\ ER),\ MCR) & if\ CI = 1\ \&\ NI = 0/1, \\ max(min(ACR,\ ER),\ MCR) & if\ NI = 1\ \&\ CI = 0 \end{cases} \tag{3.1}$$

where RIF is the *rate increase factor* and RDF is the *rate decrease factor*.

3.4 Switch Scheme Design Issues

The parameter settings and proprietary techniques employed by the switch to set the EFCI/CI/NI bits or to calculate the ER values can impact the performance of the control loop. This section outlines the design issues which switch designers must consider for the design and implementation of a low cost, high efficiency and fair switch scheme [Jai96, ACA96].

Fairness: One of the main goals of a switch scheme is to allocate bandwidth fairly among connections at each link so that the link is fully utilized. A commonly used *max-min fairness* is defined in [BG87, Jaf81] and presented in Section 3.5.

Robustness: A switch algorithm should maintain good performance even in the presence of dynamic complex traffic fluctuations, network failures, loss of control messages and badly tuned parameters. It is also preferable to design a switch algorithm with few tuning parameters.

Responsiveness: A switch algorithm should respond quickly to load fluctuations in transient state and reach optimal rate in steady state rapidly. Fast access to the available bandwidth left-over by the higher priority CBR or VBR traffic, and rapid rate decrements under congestion are also the important design goals.

Convergence: A switch algorithm should be able to optimize both transient and steady state performance, i.e. bit rate and queue length should converge and stabilize to an optimal point from any initial conditions without causing large oscillations. Large oscillations cause undesired effects such as poor link utilization, low throughput and buffer overflow.

Scalability: Due to the rapid network growth (both in size and distance), a switch algorithm should not limit its application to a particular range of distance, speed, number of connections and number of switches. Buffer requirement and allocation, queuing and scheduling managements, and convergence time of the switch algorithm are some of the elements which have scalability implications.

Efficiency: The efficiency of a switch element is a measure of the actual achievable throughput divided by the available bandwidth. It is also a measure of how quickly a switch scheme acquires and releases the bandwidth dynamically that is unused by CBR and VBR.

Buffer Management: An efficient buffer management technique is one of the strategies to provide high quality of service. Buffer management involves the control on the usage of buffer space and the detection of congestion. Its objective is to protect network resources (e.g. buffer and bandwidth), and thus minimizing queue length and cell delay, and maximizing throughput. The design of an efficient buffer management technique has to take into account the switch architecture, implementation complexity, the time scale over which the control objective can be achieved, the desired stability and performance of the overall system.

Low Buffer Occupancy: Low buffer occupancy not only means reducing memory space requirements and costs, it also means minimizing cell transfer delay (CTD) and cell transfer delay variation (CTDV), and thus achieving a high quality of service.

Implementation Complexity: A switch scheme should be relatively simple to implement. It should not require expensive measurements and computation, neither should it require huge amounts of memory.

Early implementations of switch schemes, e.g. binary feedback scheme, used simple shift operation in congestion control. However, due to the inherent shortcomings of the binary approach and the advances in hardware technologies, the implementation of more sophisticated explicit rate allocation algorithms which compute fair rates for each connection are currently being considered.

Hence researchers are concentrating on switch schemes which have the properties of constant space ($O(1)$ space complexity), i.e. irrespective of the number of VCs being setup or active, and constant time ($O(1)$ time complexity), i.e. a constant number of steps to compute the fair rate when a RM cell is processed.

Interoperability: Due to the advances in hardware and switching technologies, ER switches are becoming popular since they perform better than the earlier generation EFCI switches. In the transitional period, EFCI and ER switches may coexist in the same network. Thus a switch scheme should operate efficiently in a multi-vendor and mixed EFCI-ER environment. Interoperability of newly developed ER switches and EFCI switches requires an extensive amount of further investigation.

3.5 Max-min Fairness Definition

The ABR service divides the bandwidth left-over by the VBR and CBR connections among the ABR connections while allocating a fair share for each ABR source. Different definitions of *fairness* exist. The most commonly used fairness criterion with zero MCR is defined in [BG87, Jaf81] and it is known as *max-min fairness*. In the context of non-zero MCR, various fairness definitions can be found in [Yin94]. The works presented in this thesis only deal with zero MCR.

At any given link in the network, there are two categories of connections competing for bandwidth:

1. **Bottlenecked Connections**: those connections which are unable to consume the fair (equal) share of bandwidth at the given link because of their smaller PCR, or limited bandwidth available at other links along the traversing route.

2. **Non-bottlenecked Connections**: those connections which are able to consume the fair share or/and an even higher bandwidth that is only limited at the given link.

Max-min fairness criterion aims to maximize throughput of all the connections by equally allocating the available bandwidth which is left-over by the bottlenecked connections to all non-bottlenecked connections at a given link. Thus the fair share of link l can be computed as follows:

$$
\begin{aligned}
\textit{Fair Share} &= \frac{C_l^A - \sum \textit{Bandwidth of bottlenecked connections elsewhere}}{N_l - \sum \textit{Bottlenecked connections elsewhere}} \\
\textit{Or:} & \\
&= \frac{C_l^A - \bar{C}_l}{N_l - \bar{N}_l}
\end{aligned}
\tag{3.2}
$$

where C_l^A is the total available bandwidth for ABR service on link l, $\bar{C}_l$ is the total bandwidth of connections bottlenecked elsewhere, N_l is the total number of connections traversing link l, and $\tilde{N}_l$ is the total number of bottlenecked connections elsewhere.

3.6 Survey of Switch Congestion Control Schemes

The main objectives of a switch scheme are high link utilization, fairness among all connections, acceptable level of low queuing delay and low cell loss. Several algorithms have been proposed and discussed within the ATM Forum and in the literature. The key issues of these algorithms are the computation of fair share of the available bandwidth in which each connection should converge, the congestion detection techniques, and the congestion control mechanisms to deal with the build up of switch queues during transients. This section presents some of the proposed switch schemes. A more detail survey of switch schemes can be found in [Kal97, ACA96].

Switches implementing EFCI and RR marking are binary switches. The following section first discusses these early generation switches. Binary switches are simple to implement but may result in unfairness, slow congestion response and rate oscillation. Hence more sophisticated mechanisms using ER marking are proposed. These switches are known as ER switches. They provide more flexibility, fast convergence time, reducing the rate oscillation and an increase in robustness against RM cell loss. ER switches can be classified into two approaches: approximate and exact fair rate computation. Section 3.6.2 and 3.6.3 discuss some of these approaches respectively. Recently, fuzzy logic and neural networks have shown great potential to deal with dynamic and uncertain network conditions. Section 3.6.4 presents some ABR switch implementations which use these approaches.

3.6.1 Binary Feedback Scheme

Earlier generation switches implement a binary feedback scheme by marking the EFCI bit in the payload type indicator (PTI) field of a data cells header when congestion is detected. These data cells are sent by the source with the EFCI bit initially set to zero. The source also generates a RM cell every $N_{rm} - 1$ data cells in order to probe the current network situation. Upon receiving the marked last data cell, the destination sets the CI bit in the RM cell. The RM cell is eventually returned to the source in the backward path. If the CI bit is set, the source reduces its transmission rate by RDF but not below the MCR, else it increases its rate by RIF without exceeding a negotiated PCR. The most representative binary feedback scheme is the proportional rate control algorithm (PRCA). It is so called because the rate decreases and chances for rate increases are proportional to the current transmission rate.

PRCA is simple and cost effective to implement as it only requires to set a bit in the cell header. However, the rate may take several round trips to converge since the source rate can only be increased or decreased by a fixed step size. This results in slow congestion response. If the transient period is long, the network queues might build up too. The algorithm also exhibits large rate and queue length oscillation in steady state.

In addition, PRCA was found to suffer from the "beat-down" fairness problem [BJ94, ACA96] in which connections traveling more hops are more likely to have the EFCI bit set than those connections traveling smaller number of hops. Thus, long path connections are unable to increase their rate, and are beaten down frequently by short path connections.

The fairness problem can be alleviated by employing selective or intelligent marking [Bar94] in which the current rate of the connection is incorporated in the decision of setting the EFCI bit in the cell. The switch computes a fair share. When congestion is detected, the switch sets the EFCI bit whose cells belong to those connections having a higher rate than this fair share.

3.6.2 Approximate Fair Rate Allocation Algorithms

This category of algorithms does not adhere strictly to the max-min fairness criterion equation (defined in Section 3.5) in fair share computation. Instead, algorithms in this category approximate the fair share rate based on the CCR value in RM cells and the queue length information.

To obtain the fair share, some of these algorithms maintain an exponential weighted average of fair share based on the CCR value and the queue length information, in the hope that this value gracefully converges to the fair share rate at steady state [Rob94, CXK96]. Whilst others use traffic load to make an approximate fair share convergence to fair share [JKG+96].

Due to approximate nature, these algorithms converge to fair share rate slowly after few round-trip times and excessive oscillations may result.

3.6.2.1 EPRCA Algorithm

EPRCA [Rob94] is the enhanced version of PRCA. It employs explicit rate setting and intelligent marking to alleviate the fairness problem as with PRCA. When the switch receives an RM cell, it computes a *mean allowed cell rate* ($MACR$) from CCR using an exponential weighted average. Fair share is then derived from a fraction of this $MACR$ for explicit marking.

$$\begin{aligned} MACR &= (1 - AV)\ MACR + AV \times CCR \\ \textit{Fair Share} &= DPF \times\ MACR \end{aligned} \tag{3.3}$$

where AV is an *averaging factor* and DPF is a *down pressure factor*. Typical values of AV and DPF are 1/16 and 7/8 respectively.

EPRCA detects congestion by maintaining two threshold values QT and DQT on queue length. When the queue length is below QT, all connections are allowed to increase their rates. When congestion is detected, i.e. the queue length exceeds QT, the switch performs intelligent marking by asking those connections whose ACR are higher than $MACR$ to reduce their rates, and others to increase. Therefore the switch may reduce the rate by setting the CI bit and/or by marking the ER field of an RM cell if CCR value is higher than the fair share. When the queue length exceeds DQT, the switch is considered severely congested, all connections have to reduce their rates regardless of their CCR values.

This algorithm also supports EFCI bit setting switches. In this case, the destination marks the CI bit in the BRM cell if it sees the last data cell has the EFCI bit set.

EPRCA has low implementation complexity as it operates with $O(1)$ in space and time. Another advantage of EPRCA is that it supports both binary feedback switches and explicit rate feedback switches. Therefore it bridges the gap between PRCA and explicit rate schemes.

In EPRCA, the convergence of fair allocation is enforced by several multiplier factors on the CCR value and mean ACR. Severe oscillations may result if these factors are not tuned properly, and thus rates are not able to converge to the fair share. In addition, the algorithm may introduce considerable unfairness if the mean ACR is not a good estimation of the fair share [CXK96]. This situation may arise if the sources set CCR values incorrectly or are not well behaved; connections underestimate or overestimate the fair share due to bottlenecks elsewhere or during the transient period.

3.6.2.2 DMRCA Algorithm

The Dynamic Max Rate Control Algorithm (DMRCA) [CXK96] is proposed by Chiussi et la. to improve the shortcomings of the EPRCA.

In DMRCA, the maximum rate of all the connections is used as opposed to the mean rate used by EPRCA. The switch monitors the maximum rate (MAX) of all connections arriving at the switch, and records the corresponding VC number (MAX_VC). The authors observe the maximum rate would elevate above the fair share, and then converge to the fair share in steady state. However excessive oscillations may result due to dynamic changes of the maximum rate, leading to transient instabilities. To smooth excessive oscillations in the MAX value, the switch computes an *adjusted maximum rate* ($AMAX$) using an exponential weighted average when a new maximum rate is detected.

$$AMAX = (1 - \alpha)\ AMAX + \alpha \times MAX \tag{3.4}$$

where α is an *averaging factor* and typical value is 1/16.

Similar to EPRCA, DMRCA maintains two queue thresholds *QT* and *DQT* for congestion detection. However, the DMRCA threshold is based on the function that describes the *degree of congestion* at the switch. When the queue length exceeds QT, the switch performs intelligent marking based on a marking threshold. This threshold is defined as a function of the queue length and the $AMAX$:

$$\textit{Marking Threshold} = AMAX \times\ F_n(\textit{Queue Length}) \tag{3.5}$$

where F_n*(Queue Length)* is a discrete non-increasing function of the queue length $(0 \leq F_n \leq 1)$.

When the queue length is higher than DQT, the rate of all connections are decreased rapidly by marking RM cells. The ER field of RM cells is thus set to $AMAX \times MRF$, where MRF is known as a *major reduction factor*.

DMRCA has a low implementation complexity. It improves fairness and efficiency over the EPRCA by incorporating the exponentially averaged maximum rate and an effective queue thresholding control function.

However, the convergence of rates depend heavily on the correct tuning of parameters and a fixed linear queue thresholding control function. Incorrect settings of the parameters may lead to longer oscillations and transient convergence times. In addition, different parameter settings and queue thresholding control functions are required for different network situations and round trip delays.

3.6.2.3 ERICA Algorithm

The Explicit Rate Indication for Congestion Avoidance (ERICA) switch algorithm [JKG+96] uses traffic load as a congestion indicator to make an approximate fair share converge to the actual fair share. It periodically monitors the load, capacity and the number of active connections at the switch averaging interval. The link utilization is typically targeted at 95% and 90% for LANs and WANs respectively.

To ensure every VCs gets a fair share of the available bandwidth, fair share can be computed as follow:

$$\textit{FairShare of each VC} = \frac{\textit{Target ABR Capacity}}{\textit{Number of Active VCs}} \tag{3.6}$$

If the link is not overloaded, and some sources do not use all of their *FairShare* due to constraints elsewhere, then the switch allocates the remaining bandwidth to those sources which can use it by *VCShare*:

$$\textit{VCShare} = \frac{\textit{VC's CCR(i)}}{\textit{Load Factor}} \tag{3.7}$$

where $CCR(i)$ is the current cell rate of the most recently received forward RM cell of VC i.

If the link is overloaded, the maximum of the *VCShare* or *FairShare* is given as the calculated explicit rate. To ensure the bottleneck ER reaches the source, the ER field of a BRM cell is updated by the minimum of the calculated ER and the ER value in the BRM cell.

The ERICA algorithm has proven to be very robust and insensitive to parameter variations. The rate is able to converge quickly with a shorter oscillation [JKG+96]. However its simple fair share computation has increased the switch burden on additional countermeasures against unfairness, transient overloads and heavy cell loss [ACA96]. These countermeasures also require $O(N)$ storage space to maintain a list of connection status and rates. One of the additional countermeasures to ensure ERICA converges to max-min fairness is the introduction of two variables, *MaxAllocPrevious* which stores the maximum allocation in the previous switch measuring interval, and *MaxAllocCurrent* which stores the maximum allocation in the current switch interval.

3.6.2.4 ERICA+ Algorithm

The ERICA+ algorithm [JKG+96] is an enhancement of the ERICA algorithm. It introduces the concept of 100% link utilization with controlled queue lengths, and estimation of the available bandwidth when CBR and VBR traffic have a higher priority over ABR traffic.

The ERICA+ algorithm uses a *dynamic queue control function* $f(Q)$ and a fixed non-zero queuing delay to control the target operating point. Thus the product of the queue control function $f(Q)$ and the available bandwidth determines the target ABR capacity:

$$\textit{Target ABR Capacity} = f(Q) \times \textit{Total ABR Capacity} \tag{3.8}$$

where

$$\textit{Total ABR Capacity} = \textit{Link Bandwidth} - \textit{CBR Capacity} - \textit{VBR Capacity} \tag{3.9}$$

The value of queue control function $f(Q)$ is dependent on the switch target queue length, Q_0, which is defined as follows:

$$Q_0 = \textit{Total ABR Capacity} \times T_0 \tag{3.10}$$

where T_0 is the fixed non-zero queuing delay (threshold value). Typical values are 100 *usec* for a LAN switch and 500 *usec* for a WAN switch on a 155 Mbps link.

Design of Queue Control Functions

Step, linear, hyperbolic and inverse hyperbolic queue control functions are proposed in [VJGF98, JKG+96] for the ERICA+ algorithm. Figure 3.2 depicts the various queue control functions. The design of the queue control functions have taken into consideration of the following conditions:

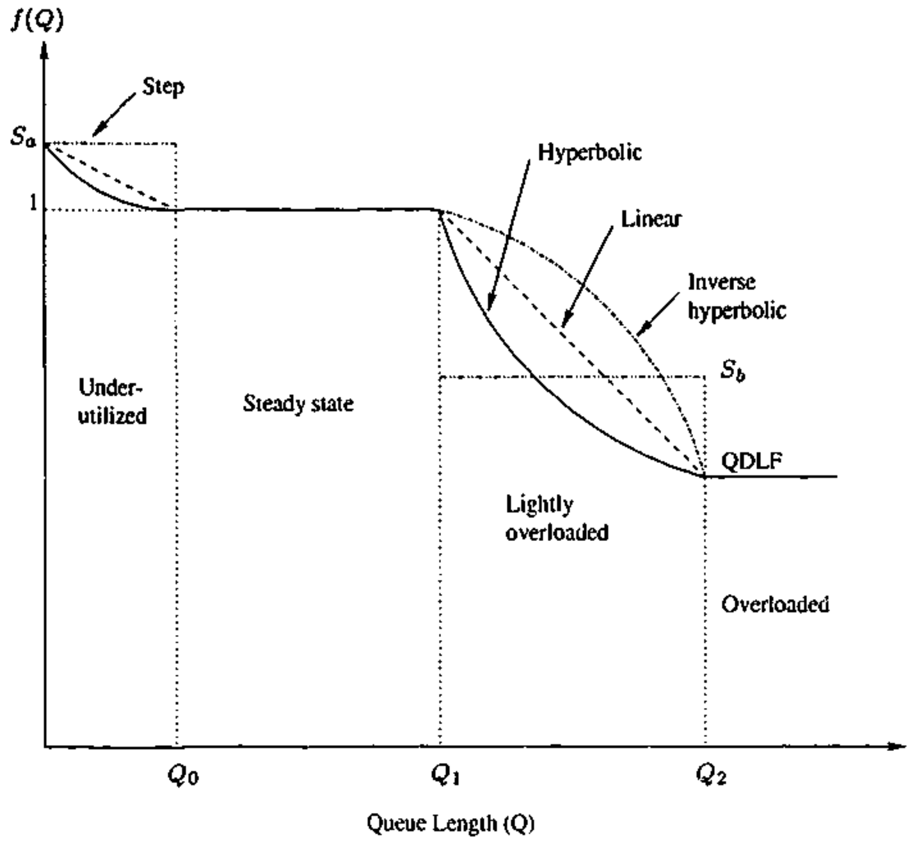


Figure 3.2: The Queue Control Functions of the ERICA+ Switch Algorithm.

- When the link is under-utilized, the current queue length (Q) is below the target queue length (Q_0), i.e. $0 \leq Q < Q_0$, and the source is encouraged to increase its rate. This implies that $f(Q)$ should be greater than 1.

- In steady state, $Q_0 \leq Q < Q_1$, $f(Q) = 1$ to maintain a constant queue length and achieve a max-min fairness rate.

- Under lightly overloaded conditions, $Q_1 \leq Q < Q_2$, the source rate should be decreased. This implies that $f(Q)$ should be less than 1.

- Under heavily overloaded conditions, $Q_2 \leq Q < \infty$, $f(Q)$ is limited by a *queue drain limit factor* ($QDLF$). This allows enough capacity to drain the large queue and also enables a minimum amount of bandwidth for forwarding normal traffic. Recommended values for $QDLF$ are 0.8 for LAN switch and 0.5 for WAN switch.

Linear, hyperbolic and inverse hyperbolic queue control functions require more computation power as they have division operations. Simulation results in [VJGF98] show that the step control function has more oscillations compared to others and may not be able to converge to a fair rate on certain network conditions. The authors also claim that the inverse hyperbolic queue control function has the best performance, and linear control function has the advantage of simplicity.

In Section 4.3, the performance of the hyperbolic and inverse hyperbolic queue control functions is further evaluated.

3.6.3 Exact Fair Rate Allocation Algorithms

As the name implies, this category of allocation algorithms computes "exact" fair share in a distributed manner by strictly adhering to the max-min fairness criterion equation defined in Section 3.5. With exact rate, these algorithms can achieve faster convergence time and reduce unnecessary oscillations in steady state.

To calculate the fair share, these algorithms need to keep track of every connection status and bandwidth, and store the information in a connection-based table. Due to different implementation complexities, the number of computations may differ among algorithms.

3.6.3.1 MIT Algorithm

The Massachusetts Institute of Technology (MIT) algorithm [Cha94] designed by Charny was an early introduction of computing explicit rate in a distributed manner. The algorithm was developed for the packet switched network before the existence of ATM ABR specifications. Therefore it did not use many of the features defined in the current ABR service specifications. Nevertheless, this algorithm has had a profound impact on the later development of fair rate allocation algorithms for ABR service.

Upon receipt of an RM cell, the switch computes an *advertised rate* (A_{max}) or fair share using Equation (3.2). It then compares the CCR value in the RM cell to the A_{max}. If the CCR value of a connection is less than the advertised rate, this connection is marked as bottlenecked, otherwise it is marked as non-bottlenecked.

Once the connection status is updated, the ER field of the RM cell is set to the minimum of the advertised rate and the ER value of the received RM cell. The minimum rate is then recorded in a table.

The algorithm has demonstrated the ability to adapt to dynamic changes in the network quickly. It has also shown that it converges to the max-min rates within $4M$ round-trips times, where M is the number of bottleneck connections in the network.

However, the computation of the advertised rate may require $O(N)$ operations, where N is the number of connections. It also needs $O(N)$ storage space to maintain a table of connection status and rates. In addition, the advertised rate is calculated based on the ER value in the RM cell, so if any switch along the traversing path does not mark the ER field, then this algorithm may not be able to provide fairness among different types of switch. The algorithm also does not provide any congestion control strategy to drain the large queue build up during transient periods.

3.6.3.2 FMMRA Algorithm

The Fast Max-Min Rate Allocation (FMMRA) algorithm [ACA97] was proposed by Arulambalam et la. Its main aim was to reduce the fair share computational complexity of the MIT algorithm from $O(N)$ to $O(1)$.

In FMMRA, the advertised rate or fair share for non-bottlenecked connections, γ, is computed by using Equation (3.2) and incorporates the change of bottleneck status and bandwidth requirement of each connection. Therefore, FMMRA algorithm achieves max-min fairness without inspecting the state of all the connections at the update time. Hence it reduces the computation complexity. The fair share or advertised rate, γ_l, of FMMRA algorithm can be expressed as:

$$\gamma_l(t^+) = \frac{C_l^A - \tilde{C}_l(t) - \Delta\lambda}{N_l - \tilde{N}_l(t) - \Delta\beta} \tag{3.11}$$

where t is the time when a BRM cell is received, t^+ is the update time. This advertised rate is only updated when a BRM cell is received. However the ER field in the RM cell is marked bi-directionally in order to achieve a fast convergence time.

When a BRM cell is received at the switch, the change of status of each connection, i, is determined by, $\Delta\lambda$, and $\Delta\beta$ as in Equation (3.12) and Equation (3.13).

$$\begin{aligned}\Delta\lambda &= \lambda^i_{l,new} - \lambda^i_{l,old} \\ &= \begin{cases} \lambda^{ER}_i - \lambda^i_l(t) & \text{if } \lambda^{ER}_i < \gamma_l(t) \quad ;bottlenecked \\ -\lambda^i_l(t) & \text{if } \lambda^{ER}_i \geq \gamma_l(t) \quad ;non-bottlenecked \end{cases}\end{aligned} \tag{3.12}$$

$$\begin{aligned}\Delta\beta &= \beta^i_{l,new} - \beta^i_{l,old} \\ &= \begin{cases} 1 - \beta^i_l(t) & \text{if } \lambda^{ER}_i < \gamma_l(t) \quad ;bottlenecked \\ -\beta^i_l(t) & \text{if } \lambda^{ER}_i \geq \gamma_l(t) \quad ;non-bottlenecked \end{cases}\end{aligned} \tag{3.13}$$

where λ and β are per-VC variables for marking the bottleneck bandwidth and bottleneck status elsewhere of a connection respectively.

Finally the new advertised rate which incorporates the status of the connection is updated as:

$$\gamma_{l,new} = \begin{cases} C^A_l & \text{if } N_l = 0, \\ \gamma_l(t) + \dfrac{\gamma_l(t)\Delta\beta - \Delta\lambda}{N_l - [\bar{N}_l(t) + \Delta\beta]} & \text{if } N_l > \bar{N}_l, \\ \gamma_l(t) & \text{if } N_l = \bar{N}_l. \end{cases} \tag{3.14}$$

An enhancement to the FMMRA algorithm with a congestion control mechanism was proposed in [CAXC97]. It is called enhanced-FMMRA (E-FMMRA) algorithm. The algorithm incorporates traffic load as in ERICA algorithm, and queue length information into the ER computation [CAXC97]. This is achieved by computing an *adjusted maximum ER* value (ER_{max}) using an exponential weighted average on the received BRM cell ER value and ER_{max}:

$$ER_{max} = (1-\alpha)\ ER_{max} + \alpha \times max\{RM(ER),\ \frac{ER_{max}}{load\ factor}\} \tag{3.15}$$

where α is an *averaging factor* and typical value is 1/8.

The E-FMMRA maintains two queue thresholds QT and DQT for congestion detection. When the switch is not congested, i.e. queue length is below QT, the load factor is used as a correcting factor, the ER field of the FRM and BRM cells are marked according to the Equation (3.16). Note: The FRM cell is always marked using this equation regardless of queue length.

$$RM(ER)_{new} = min\{RM(ER)_{old}, max\{A_{max}(t), \frac{ER_{max}}{load\ factor}\}\} \tag{3.16}$$

When the queue length is between QT and DQT, if the traffic load is greater than one, the ER field of BRM cell is set using Equation (3.16), otherwise the following equation is used. The main aim is to use the lowest possible rate in order to drain the build up of the queue.

$$RM(ER)_{new} = min\{RM(ER)_{old}, max\{A_{max}(t),\ ER_{max}\}\} \tag{3.17}$$

When the switch is highly congested, i.e. the queue length exceeds DQT, the rate of the connections should be reduced rapidly. Thus the ER field of BRM cell is set to the minimum of advertised rate and received RM cell value:

$$RM(ER)_{new} = min\{RM(ER)_{old},\ A_{max}(t)\} \tag{3.18}$$

In addition, the ER_{max} is updated by using the following more conservative equation instead of Equation (3.15) as ER_{max} is very high in this situation.

$$ER_{max} = max\{A_{max}(t), \frac{ER_{max}}{load\ factor}\} \tag{3.19}$$

The FMMRA algorithm has improved the fair share computation complexity from $O(N)$ to $O(1)$ though it still needs $O(N)$ space to keep track of the bottleneck status and bandwidth for each connection. The algorithm also improves the rate convergence time by marking RM cells in both forward and backward directions. In addition, the E-FMMRA has fully utilized the unused bandwidth from idle connections by using traffic load measurements in the ER computation. Since the FMMRA algorithm relies on the ER value of RM cells to compute the fair share, it is not sensitive to inaccuracies in CCR values. However, if any switches along the path does not mark the ER field, the algorithm may cause unfairness in the network. The congestion control mechanism is also considered quite complex and requires tuning of parameters and queue threshold values in order to achieve rate convergence.

3.6.3.3 BEMO Algorithm

The Bi-directional Explicit Rate Marking Optimizer (BEMO) [TQ98] is an improved version of the FMMRA algorithm [ACA97]. In [TQ98], it was found that the FMMRA algorithm offers the same advertised rate to both bottlenecked and non-bottlenecked connections. In addition, the advertised rate updating formula in Equation (3.14) does not precisely allocate the non-bottleneck bandwidth among non-bottlenecked connections. Simulation results show that this will lead to link over-utilization under certain circumstances.

Thus the improved advertised rate formula is:

$$\gamma_{l,new} = \begin{cases} C_l^A & \text{if } N_l = 0 \\ \dfrac{C_{nb,old} - \Delta\lambda}{N_{l,new} - \bar{N}_{l,new}} & \text{if } N_{l,new} > \bar{N}_{l,new} \\ C_{nb,old} - \Delta\lambda & \text{if } N_{l,new} = \bar{N}_{l,new} \end{cases} \tag{3.20}$$

where $N_{l,new}$ and $\bar{N}_{l,new}$ are the new total number of connections and new number of bottlenecked connections respectively; $C_{nb,old}$ is the non bottleneck bandwidth with respect to $\gamma_{l,old}$ and can be expressed by the following formula:

$$C_{nb,old} = \begin{cases} \gamma_{l,old}(N_{l,old} - \bar{N}_{l,old}) & \text{if } N_{l,old} > \bar{N}_{l,old} \\ \gamma_{l,old} & \text{if } N_{l,old} = \bar{N}_{l,old} \end{cases} \tag{3.21}$$

BEMO does not provide any congestion control mechanism to absorb excess traffic when the input rate exceeds the output link rate. In Section 4.2, a congestion control mechanism for the BEMO switch algorithm is proposed.

3.6.4 Fuzzy Logic and Neural Network Switch Schemes

In recent years, fuzzy logic and neural network controls have been demonstrated to have the capability to control complex systems that cannot be described precisely by mathematical processes. Several fuzzy logic and neural-fuzzy control schemes have been proposed for ABR rate-based closed-loop control. Most schemes compare their performance with the early generation switch schemes. To deal with the large round-trip delay in the feedback control loop, many proposed schemes employ prediction and/or monitoring of the queue growth rate in their control mechanisms. In this section, some of these schemes are briefly presented.

A Fuzzy Explicit Rate Marking (FERM) algorithm for ABR service is developed by Pitsillides et la. [PcR97]. The algorithm generates a fractional flow rate (FFR $\in$ [0,1]) by using a fuzzy congestion controller (FCC) which has two input variables: average ABR queue length and queue growth rate. Upon receipt of a BRM cell, the ER field is marked by the minimum of the ER value in the received RM cell and the product of the link cell rate with FFR. The performance of the FERM is compared with the EPRCA in a LAN and WAN environment with the presence of VBR and CBR traffic. A network model consisting of three switches with a uniform link speed is used in the simulations. Simulation results show that the FERM algorithm exhibits faster transient response, lower end-to-end delay and better network utilization, especially in higher link load situations. However the fairness issue is not well demonstrated, in particular if a connection is bottlenecked elsewhere with a different link speed.

In [KA98], Kuan and Andrew compare the performance of FERM algorithm with a threshold based non-fuzzy control scheme [RR95, Yin95]. The authors demonstrate that the FERM algorithm has effective control only if the number of sources and round-trip times are small. In [AK98], they propose three modifications to the original FERM algorithm to improve the performance on large round-trip times. The first approach is to add another input variable, ACR, to the FCC. The second approach is to use a hybrid fuzzy/classical algorithm in which the ER rate is decreased by fuzzy logic but autoregressively increased by $ACR + \eta\{F(q, dq) - ACR\}$ where η is an increment step size for ACR. The last approach is to lower the target queue length by modifying the fuzzy rules. By using an ABR and VBR sources with a switch, the hybrid fuzzy/classical algorithm has shown to have the best performance and has no cell loss at 20 *msecs* round-trip times compared to only 8 *msecs* for pure fuzzy logic approach. Nonetheless, it requires fine tuning of threshold parameters. The simulation model is also considered too simplistic. In addition, there is no discussion of the fairness issue in the proposed modification mechanisms.

A fuzzy logic prediction approach is proposed in [Qiu98, Qiu97]. The ABR queue length is estimated one round-trip delay ahead by a fuzzy logic predictor. The predicted queue length, together with current queue length and queue growth rate is provided to a fuzzy logic inference system which generates a rate factor ($R_f \in [-1, 1]$). This rate factor is applied to the ER value generated by the ERICA algorithm using the equation $ER = ER(1 + R_f)$. It has the capability to double the ER value in a BRM cell when the switch is underloaded, or decrease the rate to zero under heavily overloaded situations. With the existence of VBR and CBR traffic, simulation results show that the fuzzy logic prediction outperforms the conventional autoregression predictions, particularly under heavy load and long propagation delay in MAN and WAN environments. It exhibits low cell loss, queue delay and delay variation. However, the fairness issue and stability of the proposed scheme are not addressed in the literature.

In [LH00], Lee and Hou propose a neural-fuzzy congestion controller to regulate the ABR source rate. The proposed congestion controller consists of a neural-fuzzy network and a fuzzy inference engine. The functions of the neural-fuzzy network are to predict cell loss and generate the weights of the fuzzy rules for the fuzzy inference engine. The cell loss is predicted based on the current queue length, queue change rate and previous queue change rate. On the other hand, the fuzzy inference engine produces an explicit rate to control the source cell rate. The performance of the proposed controller was compared with EPRCA algorithm. Although the link distance between switches is not clear, simulation results show that the controller achieves maximum throughput and lower cell loss. It also has a lower queue length and stable convergence rate. Nevertheless, fairness should be further demonstrated by using complex network models with a large number of connections.

3.7 Conclusions

This chapter has presented the fundamental operation of a rate-based closed-loop flow and congestion control for the ABR service. The operation is based on a set of "behavior" rules that the sources, destinations and network switching elements comply with. One of the key behaviors of the switches is that they need to convey a fair share of the available bandwidth to the corresponding source via BRM cells. The source is then able to adjust its transmission rate into the network based on the feedback information received.

Two types of switch scheme are proposed to convey feedback information to the source: a binary feedback scheme and an explicit rate feedback scheme. Binary feedback scheme uses EFCI/CI/NI bits to convey network status, and the source adjusts its rate based on these bits and some other parameters. The ABR source converges slowly to the available bandwidth. On the other hand, the explicit rate feedback scheme provides network information by specifying an exact rate to the source which allows the ABR source to immediately utilize the available bandwidth. This feedback scheme consists of two kinds of algorithm: approximate fair rate computation algorithms and exact fair rate computation algorithms. These switch algorithms differ in their rate allocation, speed of convergence and implementation complexity.

Various congestion detection techniques and congestion control mechanisms are employed by the existing switch schemes. Some of these control schemes are based on single or dual thresholds for queue length, while others are based on differential queue length or mathematical equations.

An unconventional approach using fuzzy logic and/or neural networks on rate allocation and congestion control was also reviewed. Many such switch schemes employ prediction or monitor the queue growth rate to control congestion.

In the next chapter, some of these existing switch schemes are further enhanced and evaluated, and a comparative study is conducted between these switch schemes and the proposed fuzzy logic control scheme.

Chapter 4

Fuzzy Logic Inference and Prediction in ABR Service

The previous chapter has outlined the functions of each key element which constitutes the operation of a rate-based flow and congestion control framework. It also provides an overview of the existing switch schemes. Their fair share computation algorithms, congestion detection techniques and control mechanisms were discussed. Since conventional congestion control schemes based on static threshold values, or mathematical equations are inflexible and abrupt, this chapter proposes a novel scheme using fuzzy logic inference and prediction. The proposed fuzzy logic congestion control scheme is applied to the two switch algorithms, ERICA+ [JKG+96] and BEMO [TQ98]. The performance is then compared with the conventional control schemes by extensive simulation. Therefore, for later comparison purposes, an enhancement is first made to the BEMO [TQ98] switch algorithm by incorporating the algorithm with a conventional congestion control mechanism. Then the queue control functions proposed by the ERICA+ [VJGF98] switch algorithm on congestion control are further evaluated before the comparison.

4.1 Introduction

A network switch allocates bandwidth and buffers to route ATM cells. These resources are subjected to contention among competing sources for efficiency. Heavy contention may lead to congestion resulting in cell loss and excessive delay of ATM cells. In addition. traffic flow is highly uncertain in ATM networks, cells may be lost due to buffer overflows during transient periods. Therefore, in order to provide high QoS and absorb excess incoming traffic, a switch must have an efficient congestion detection technique and a control mechanism. Its objective is to protect the resources, and thus minimizing queue length and cell delay, and maximizing throughput. The design of an efficient congestion detection technique and control has to take into account the switch architecture, implementation complexity, the convergence time, the desired stability and the performance of the overall system.

As highlighted in the previous chapter, various congestion detection techniques and control mechanisms are employed by the existing switch schemes. They are based on either single or dual queue thresholds, differential of queue length, varying output port link utilization or mathematical equations. Due to highly fluctuated traffic in ATM networks, congestion control based on static threshold values, or mathematical equations which introduce discontinuities function may cause high variance in queue fluctuations. They cannot adapt well to traffic fluctuations, resulting in high cell loss or low link utilization in some situations.

Besides the traffic uncertainty in ATM networks, congestion control using feedback mechanisms has proven to be a challenging task, especially with large bandwidth delay products, i.e. large propagation delays in comparison to buffer dynamics. Consequently, feedback control messages may be delayed. They are thus usually inaccurate for most current network conditions. This can cause buffer overflow and QoS degradation. To address these problems, the application of predictive control techniques in the flow and congestion control for the ABR service is investigated. In addition, fuzzy logic control has emerged as a viable technique for dealing with dynamic systems which cannot be described precisely by exact mathematical processes. In [Qiu01, Qiu98], a fuzzy logic prediction has been proven to outperform conventional autoregression predictions. Therefore, a novel fuzzy logic congestion control scheme based on the fuzzy logic prediction technique in [Qiu01, Qiu98] is proposed. The prediction estimates the queue length one round-trip delay ahead.

The predicted queue length, together with the queue growth rate and current queue length is provided to a fuzzy inference system which produces a target utilization factor. Hence, based on prediction, link utilization is targeted dynamically one round trip propagation delay ahead.

The proposed fuzzy logic congestion control scheme is applied to an approximate fair rate computation algorithm (ERICA+ [JKG+96]) and an exact fair rate computation algorithm (BEMO [TQ98]). The performance of the fuzzy logic control scheme is then compared with the conventional scheme on the two switch algorithms. Simulation results demonstrate that the fuzzy logic control scheme significantly reduces convergence time and buffer requirement. It also exhibits lower queuing delay, delay variation and cell loss ratio.

The BEMO switch algorithm proposed in [TQ98] does not provide a congestion control mechanism to protect network resources. Therefore this chapter first presents an enhancement to the BEMO switch algorithm by incorporating the congestion control scheme of the ERICA+ [JKG+96]. Simulations are conducted later in the chapter for the purpose of comparison to the proposed fuzzy logic control scheme. A minor error was discovered on one of the queue control functions of the ERICA+. After communicating with the original author [Van99], it was corrected, and the performance of the ERICA+ queue control functions is further evaluated in Section 4.3. To exploit the effectiveness of fuzzy logic prediction and control, a fuzzy logic congestion control scheme for the ABR service is proposed in Section 4.4. Then, an extensive simulation study is performed on the proposed scheme. The proposed fuzzy logic control scheme is first applied to the ERICA+ and BEMO switch algorithms, and its performance is then compared with the conventional control approach. Network models which consist of single-class traffic and multi-class traffic are used. Simulation results are presented and discussed in Section 4.5.

4.2 Enhancement to the BEMO Switch Algorithm

When the BEMO switch algorithm [TQ98] was first introduced, it was mainly aimed at the investigation of the fairness issue. It does not implement any congestion control mechanism to protect network resources. Therefore in this section, an enhancement to the BEMO switch algorithm is proposed by incorporating a conventional congestion control scheme. The enhanced BEMO algorithm adopts the ERICA+ congestion control strategy in which a dynamic queue control function and a fixed non-zero queuing delay are used to control the target operating point. Similar to the ERICA+ [JKG+96], it also measures the traffic load and ABR input rate periodically at the switch averaging interval.

FMMRA and BEMO are the representative algorithms which use the ER value in the RM cell to compute max-min fair share in the forward and backward direction. If a switch along the path does not use the ER approach or does not mark the forward RM cell, these algorithms may not converge to max-min fairness. In addition, if some connections become idle, the link will be under-utilized since the unused bandwidth is not reallocated. Thus these algorithms lose their responsiveness.

To overcome these limitations, the enhanced BEMO algorithm incorporates traffic load and queue length information into the ER calculation as in the ERICA+ algorithm. Traffic load (ρ) is a measurement of how well ABR bandwidth is utilized. It is the ratio of the ABR input rate and the target ABR capacity at a switch. If a link is underloaded, and some connections are not fully utilizing their fair share, then the unused bandwidth is reallocated to those connections which can use it by the ratio of $CCR(i)/\rho$. This is similar to the term $VCShare$ in the ERICA+.

Procedures 4.1 and 4.2 show the pseudo codes of the enhancement. From Procedure 4.1, it can be seen that the RM cell is marked both in forward (line 13) and backward directions (line 29), but the advertised rate is only updated in the backward direction (lines 30–42).

Unlike the ERICA+, the enhanced BEMO algorithm does not need to keep track

Procedure 4.1 Pseudo code of the enhanced BEMO algorithm.

```
if (receiving forward RM cell) then
  CCR[i]=CCR_In_FRM_Cell;
  if (Seen_FRM_cell_In_This_Interval[i]) then
    ER_Calculated = Last_Allocated_FRM[i];
  else
    Amax=Offered_Advertised_Rate();
    ER_Calculated=Max(τ_f×Amax, CCR[i]/Load_Factor);
    ER_Calculated=Min(ER_Calculated, Target ABR Capacity);
    Last_Allocated_FRM[i]=ER_Calculated;
    Seen_FRM_cell_In_This_Interval[i]=TRUE;
  end if
  /* Perform FRM ER marking */
  ER_In_FRM_Cell=Min(ER_In_FRM_Cell, ER_Calculated);
  Latest_FRM_ER[i]=ER_In_FRM_Cell;
end if

if (receiving Backward RM cell) then
  Amax=Offered_Advertised_Rate();
  if (Seen_BRM_cell_In_This_Interval[i]) then
    ER_Calculated = Last_Allocated_BRM[i];
  else
    ER_Calculated=Max(τ_f×Amax, CCR[i]/Load_Factor);
    ER_Calculated=Min(ER_Calculated, Target ABR Capacity);
    Last_Allocated_BRM[i]=ER_Calculated;
    Seen_BRM_cell_In_This_Interval[i]=TRUE;
  end if
  /* Perform BRM ER marking */
  ER_In_BRM_Cell=Min(ER_In_BRM_Cell, ER_Calculated);
  ER_In_BRM_Cell=Min(ER_In_BRM_Cell, Latest_FRM_ER[i]);
  if (ER_In_BRM_Cell < Amax) then
    /* connection i is bottlenecked */
    N_b=N_b - BOTT_STATE[i] + 1;
    BOTT_STATE[i]=1;
    C_b=C_b - BOTT_BW[i] + ER_In_BRM_Cell;
    BOTT_BW[i]=ER_In_BRM_Cell;
  else
    /* connection i is non-bottlenecked */
    N_b=N_b - BOTT_STATE[i];
    BOTT_STATE[i]=0;
    C_b=C_b - BOTT_BW[i];
    BOTT_BW[i]=0.0;
  end if
end if
```

Procedure 4.2 Pseudo code of the Offered_Advertised_Rate() sub-routine.

$\Delta\lambda$=−BOTT_BW[i];
$\Delta\beta$=−BOTT_STATE[i];
tmp_C_b=C_b + $\Delta\lambda$;
tmp_N_b=N_b + $\Delta\beta$;
if (N_l == tmp_N_b) **then**
  /* all connections are bottlenecked */
  $Amax$=C_l^A − tmp_C_b;
else
  $Amax$=(C_l^A − tmp_C_b)/(N_l − tmp_N_b);
end if

of maximum previous and current allocation of ER value of the RM cell in order to achieve max-min fairness. This is because BEMO fair share computation strictly adheres to the max-min fairness criterion. Therefore, the calculated explicit rate (lines 7 and 22) is just the maximum of target utilization (τ) times the advertised rate ($Amax$), and the $CCR(i)/\rho$.

Procedure 4.2 calculates the advertised rate offered to non-bottlenecked connections. A connection has to be first assumed that it is non-bottlenecked. Thus if it is previously bottlenecked, its status and rates have to be reset. Then the new advertised rate is computed. By doing so, BEMO is able to allocate the available bandwidth fairly among connections.

The performance of the proposed BEMO conventional congestion control scheme will be studied in Section 4.5.

4.3 Evaluation of the Queue Control Functions of the ERICA+ Algorithm

Section 3.6.2.4 has illustrated that the ERICA+ switch algorithm employs some sophisticated queue control functions. In [VJGF98], extensive analytical and simulation studies were conducted on the step, linear, hyperbolic and inverse hyperbolic queue control functions with a complex network configuration. The authors concluded that the inverse hyperbolic queue control function is the most favored. It is obvious that step and linear queue control functions do not perform as satisfactory as the hyperbolic functions. After a minor error was discovered and communicated with the original author [Van99], this section demonstrates that the hyperbolic function is more preferred.

Equation (4.1) and (4.2) are the hyperbolic and inverse hyperbolic queue control functions respectively [VJGF98, Van99].

Hyperbolic Queue Control Function

The hyperbolic queue control function ($f(Q)$) is defined as:

$$f(Q) = \begin{cases} \dfrac{b\,Q_0}{(b-1)\,Q + Q_0} & \text{if } 0 \le Q \le Q_0 \\ 1 & \text{if } Q_0 < Q \le Q_1 \\ \dfrac{a\,Q_1}{(a-1)\,Q + Q_1} & \text{if } Q_1 < Q \le Q_2 \\ QDLF & \text{if } Q_2 < \ Q < \infty \end{cases} \tag{4.1}$$

where $Q_1 = 2Q_0$ or $4Q_0$, $Q_2 = 26Q_0$, a and b are set to 1.15 and 1.05 respectively.

Inverse Hyperbolic Queue Control Function

The inverse hyperbolic queue control function ($f(Q)$) is defined as:

$$f(Q) = \begin{cases} \dfrac{b\,Q_0}{(b-1)\,Q + Q_0} & \text{if } 0 \le Q \le Q_0 \\ 1 & \text{if } Q_0 < Q \le Q_1 \\ 2 - \dfrac{a\,Q_1}{(a+1)\,Q_1 - Q} & \text{if } Q_1 < Q \le Q_2 \\ QDLF & \text{if } Q_2 < \ Q < \infty \end{cases} \tag{4.2}$$

where $Q_1 = 2Q_0$ or $4Q_0$, $Q_2 = 26Q_0$, a is set to 36 or 16.5 for $Q_1 = 2Q_0$ or $4Q_0$ respectively, b is set to 1.05.

Assuming the link rate is 100 Mbps and queuing delay, T_0, is set to 500 *usec* for a WAN configuration, the target queue length, Q_0, is 118 cells. Figures 4.1 and 4.2 show the plotting of the respective queue control functions when $Q_1 = 2Q_0$ and $4Q_0$. For comparison purpose, Figure 4.3 shows the plotting of both queue control functions when $Q_1 = 4Q_0$.

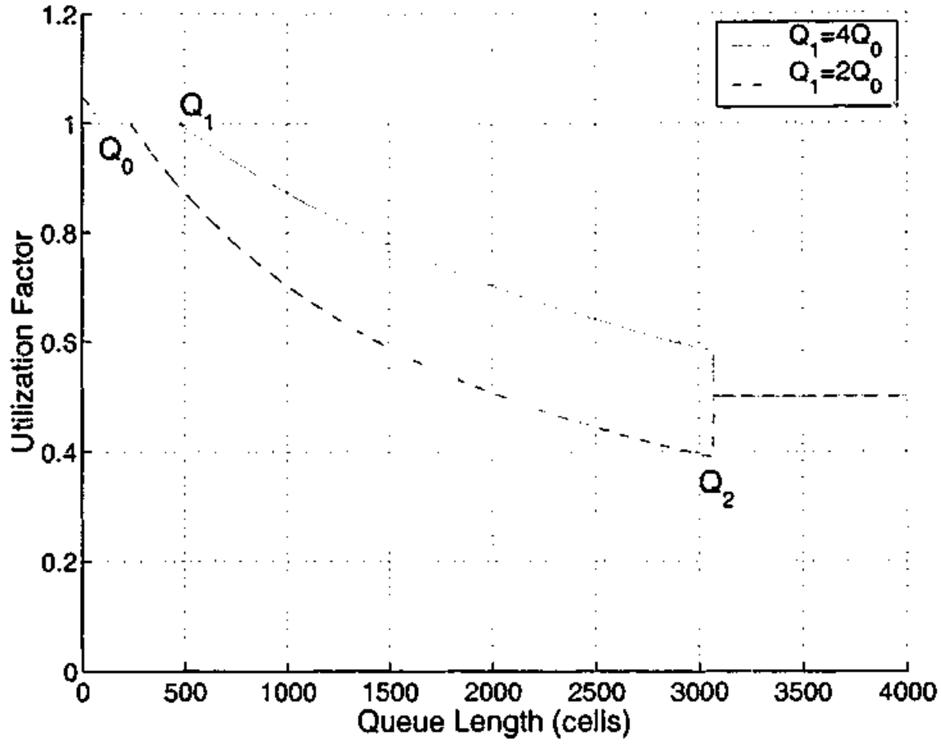


Figure 4.1: The Hyperbolic $f(Q)$. Both cases do not have a smooth transition after $Q > Q_2$, i.e. under heavily load region. This implies that sudden changes will give rise to oscillations. Nonetheless, between Q_1 and Q_2, the utilization factor is decreasing hyperbolically.

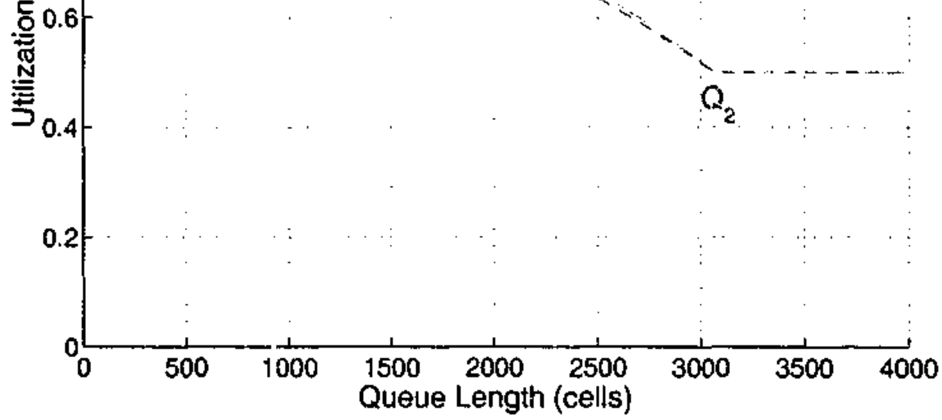


Figure 4.2: The Inverse Hyperbolic $f(Q)$. Both cases have a smooth transition after $Q > Q_2$, i.e. under heavily load region. However, between Q_1 and Q_2, the utilization factor is increasing hyperbolically instead of decreasing.

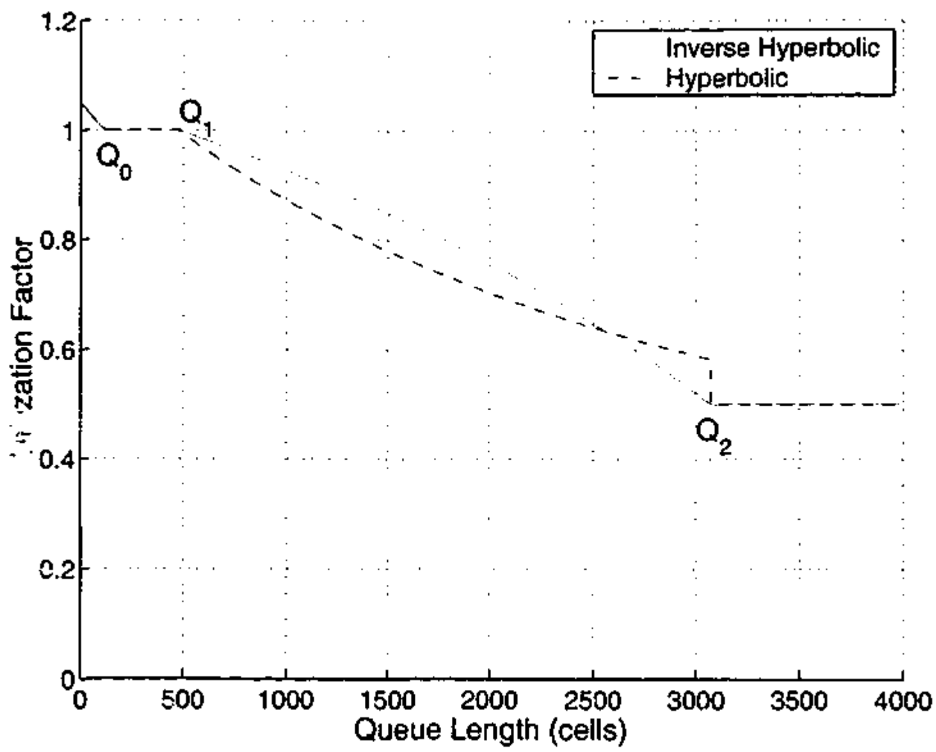


Figure 4.3: The Hyperbolic and Inverse Hyperbolic $f(Q)$ when $Q_1 = 4Q_0$. Between Q_1 and Q_2, the utilization factor of the inverse hyperbolic $f(Q)$ is increasing hyperbolically.

From the figures, it can be observed that the hyperbolic queue control function does not have a smooth transition under the heavily load region (i.e. $Q > Q_2$). This implies that sudden changes in control will give rise to undesired oscillations. On the other hand, the inverse hyperbolic queue control function does not have this problem. However, under the lightly overloaded region (i.e. between Q_1 and Q_2), the inverse hyperbolic control function increases the target operation point instead of decreasing the operation point. This results in a further build up of queue instead of draining the large queue when a switch is overloaded.

A simple two switches with five VCs network configuration is used to demonstrate that the hyperbolic queue control function is preferred. Figure 4.4 shows the simple two-switch configuration. It is a WAN configuration in which the distance between switches is 1000 Km; and all links connecting sources or destinations to switches have length equal to 100 Km. All links are 100 Mbps. It is given ICR=2 Mbps, RIF=1/8 and $Q_1 = 4Q_0$.

At sources S1 and S3, the PCR is set to 5 Mbps so that connections 1 and 3 are bottlenecked. Connections 2, 4 and 5 are non-bottlenecked connections as the

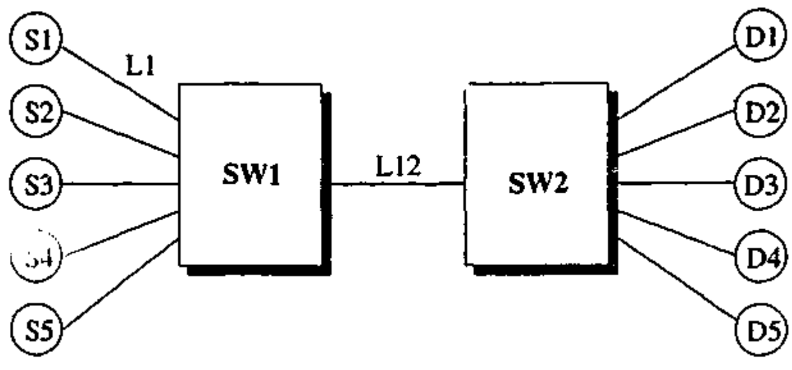
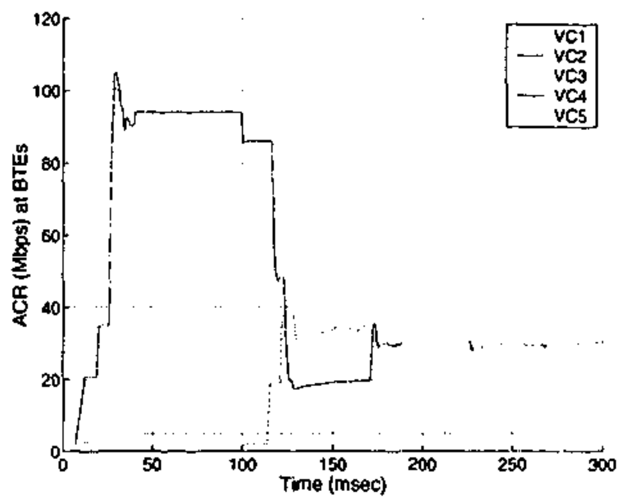


Figure 4.4: Network Model for the Two-Switch Configuration.

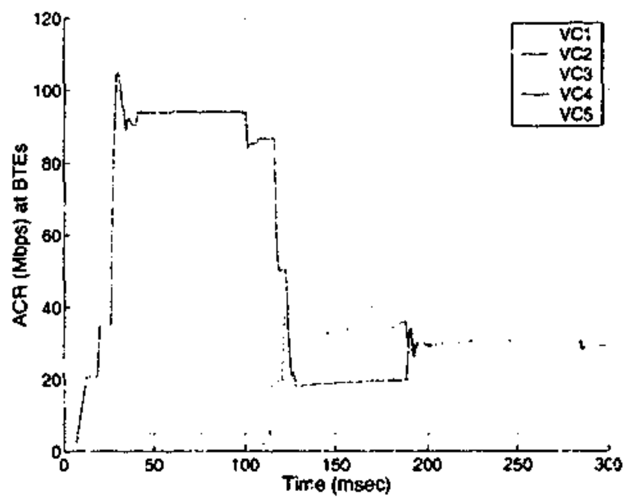
source PCR is set to 150 Mbps. Sources S1, S2 and S3 are also assumed to start sending cells at time, $t = 0$, whereas S4 and S5 start at time, $t = 100$ *msec*. This implies that connections 1 and 3 always get 5 Mbps on link L12; connection 2 gets 90 Mbps until time, $t = 100$ *msec*. At time $t > 100$ *msec*, L12 becomes a bottleneck link. Thus in steady state, connections 2, 3 and 4 get an equal share of the remaining bandwidth, i.e. each gets 30 Mbps.

Figures 4.5(a) and 4.5(b) show the simulation results of the allowed cell rate for each connection when using the hyperbolic and inverse hyperbolic queue control functions respectively. All rates are able to converge to the fair share after 100 *msec*. Comparing the figures, the hyperbolic queue control function has a faster rate convergence time than the inverse hyperbolic queue control function. Figure 4.5(c) shows the buffer requirements at switch SW1. From the figure, it can be seen that the inverse hyperbolic queue control function has a higher peak and average queue length compared to the hyperbolic queue control function. This confirms that as long as the queue reaches or converges on the lightly overloaded region, i.e. between Q_1 and Q_2 as shown in Figure 4.3, the inverse hyperbolic queue control function requires higher buffer requirements.

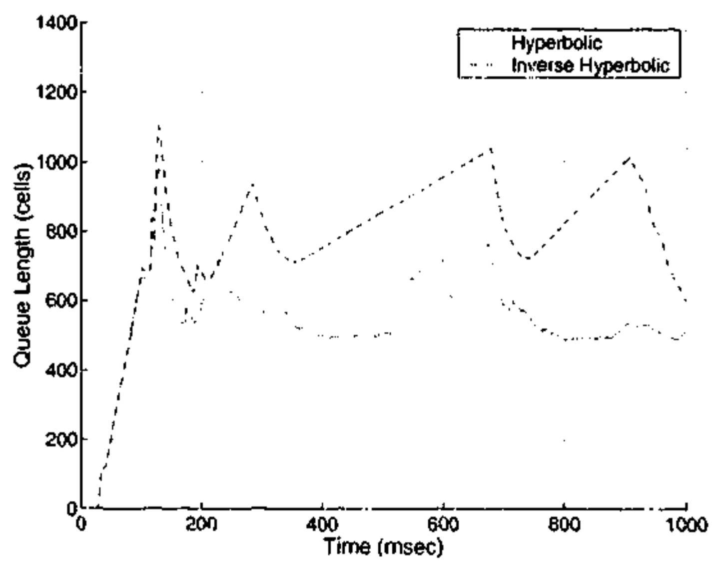
A more realistic situation is simulated in which distance between sources and destinations to switches is 10 Km. Figures 4.6(a) and 4.6(b) show the simulation results of the allowed cell rate for each connection when using the hyperbolic and inverse hyperbolic queue control functions respectively. Comparing the figures, the hyperbolic queue control function again has a faster rate convergence time than



(a) The ACR when using the Hyperbolic $f(Q)$. The rate is converged at 180 *msec*

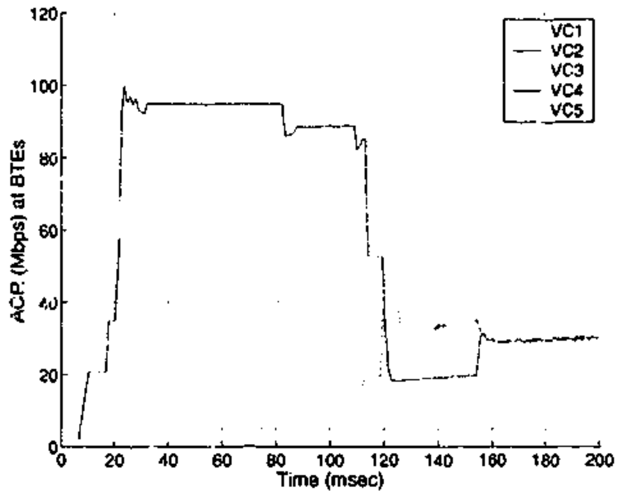


(b) The ACR when using the Inverse Hyperbolic $f(Q)$. The rate is converged at 190 *msec* which is much slower compared to the hyperbolic $f(Q)$.

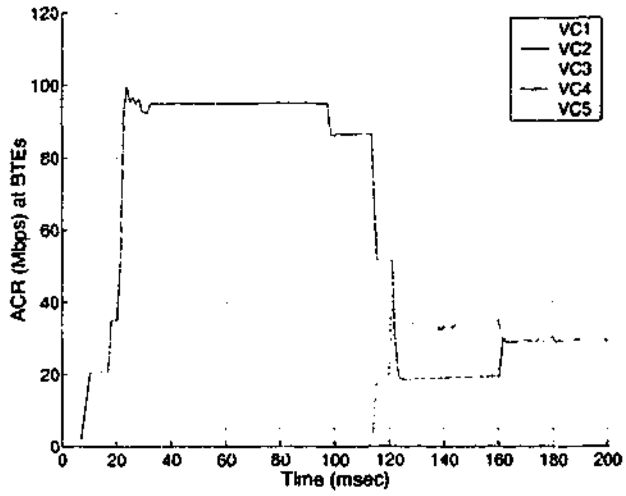


(c) Output Queue Size at Switch SW1.

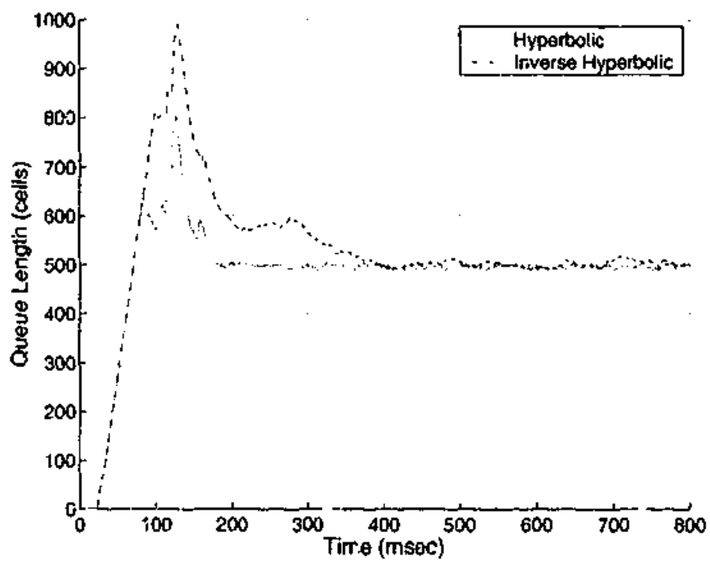
Figure 4.5: Simulation results of the Hyperbolic and Inverse Hyperbolic Queue Control Functions with the Two-Switch Configuration (distance between source/destination to switches is 100 Km).



(a) The ACR when using the Hyperbolic $f(Q)$. The rate is converged at 155 *msec*



(b) The ACR when using the Inverse Hyperbolic $f(Q)$. The rate is converged at 160 *msec* which is much slower compared to the hyperbolic $f(Q)$.



(c) Output Queue Size at Switch SW1.

Figure 4.6: Simulation results of the Hyperbolic and Inverse Hyperbolic Queue Control Functions with the Two-Switch Configuration (distance between source/destination to switches is 10 Km).

the inverse hyperbolic queue control function. Figure 4.6(c) shows the buffer requirements at switch SW1. Again, the figure demonstrates that a higher peak and average queue length is required by the inverse hyperbolic queue control function before the queue converged. A higher queue length implies higher cell delay variation, cell loss ratio and convergence time. Hence it can be concluded that the hyperbolic queue control function is preferred as it has a faster convergence time with lower buffer requirements. Therefore all simulation studies in this research work will be based on the hyperbolic queue control function.

4.4 The Proposed Fuzzy Logic Congestion Control Scheme

Conventional congestion control schemes implement binary, static thresholds values [Rob94, CXK96, CAXC97] and/or mathematical equations [CXK96, JKG+96] on queue control and are therefore inflexible and abrupt. They cannot adapt well to traffic fluctuations especially under large propagation delays, resulting in high cell loss, rate and queue oscillations and low link utilization. To address the problems of traffic fluctuations and long round-trip delays, this section presents a novel congestion control scheme using fuzzy logic prediction and control as opposed to the mathematical equations approach proposed in [JKG+96].

In the conventional congestion control scheme [JKG+96], the ABR capacity is targeted dynamically by a complex queue control function. This function is determined by the input rate of higher priority services, link rate, queue delay, current and target queue length and some other constant factors. There are so many parameters for a network administrator to adjust. Improper selection of parameters may lead to link under-utilization or QoS degradation.

In contrast, the proposed fuzzy logic control scheme does not require predefined threshold values or complex mathematical equations. It has the following features compared to the conventional approach:

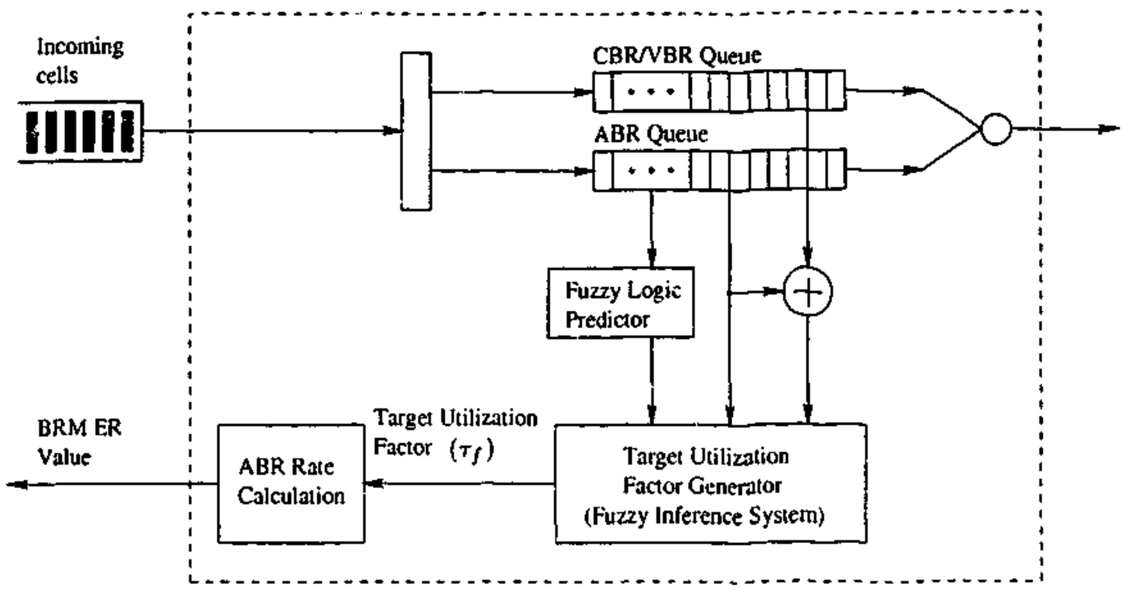


Figure 4.7: The Fuzzy Logic ABR Service Congestion Controller.

- The conventional control scheme uses a fixed threshold value for queuing delay which translates to target queue length. However the fuzzy logic control scheme uses a fuzzy logic predictor to predict queue length one round-trip delay ahead;
- The fuzzy logic control scheme does not use equations to control the target ABR capacity as in the conventional scheme, it uses soft-computing of a fuzzy inference system which adapts to the traffic fluctuations.

The Fuzzy Logic ABR Service Congestion Controller

Figure 4.7 shows the proposed fuzzy logic congestion controller. It consists of a fuzzy logic predictor and a fuzzy logic target utilization factor generator.

Due to the complex dynamic traffic fluctuations in high-speed networks, class-based queuing is necessary in order to isolate traffic flow and maintain QoS of each class. Therefore separate output queues for different categories of services, i.e. the CBR/VBR and the ABR, are considered. Since CBR and VBR traffic have strict tolerances for delay, jitter and cell loss ratio, they are served at a higher priority than ABR traffic.

The predictor predicts the ABR queue one round-trip delay ahead. This predicted queue value, together with the total queue growth rate and current ABR queue length is provided to the fuzzy logic factor generator which produces a target utilization factor (τ_f). This factor varies the target ABR capacity dynamically according to the buffer condition one round-trip delay ahead.

$$\textit{Target ABR Capacity} = \textit{Target Utilization Factor} \times \textit{Total ABR Capacity} \quad (4.3)$$

When a BRM cell (and FRM cell in the BEMO case) is received by the switch, the explicit rate switch algorithm allocates the explicit rate fairly among ABR sources based on this dynamic capacity. Hence the computed ER value takes into consideration the buffer situation one round-trip delay in advance. As a consequence, the buffer is likely to be maintained at an appropriate size instead of overflown or empty.

As shown in Figure 4.7, the fuzzy logic target utilization factor generator is integrated with the fuzzy logic predictor. The output variable, target utilization factor (τ_f), is governed by three input variables, i.e. the current and predicted ABR queue length, total queue growth rate; and the linguistic information stored in the rule base.

Figure 4.8 shows the membership functions of the linguistic variables for the fuzzy logic target utilization inference system. Triangular and trapezium shapes are adopted for both input and output membership functions. The linguistic values of the target utilization factor are low, moderate_low, medium, moderate_high and high. It is in the range of $[0, 2]$. The design of the fuzzy linguistic rules for the inference of τ_f takes into consideration the following conditions:

- During transient overload, the queue growth rate is normally high, current and predicted queue length are also increasing rapidly, so the target utilization should be low or moderate_low, in order to drain the queue.
- Under lightly loaded condition, the queue growth rate is slow, and current and predicted queue length are small, so the target utilization factor should

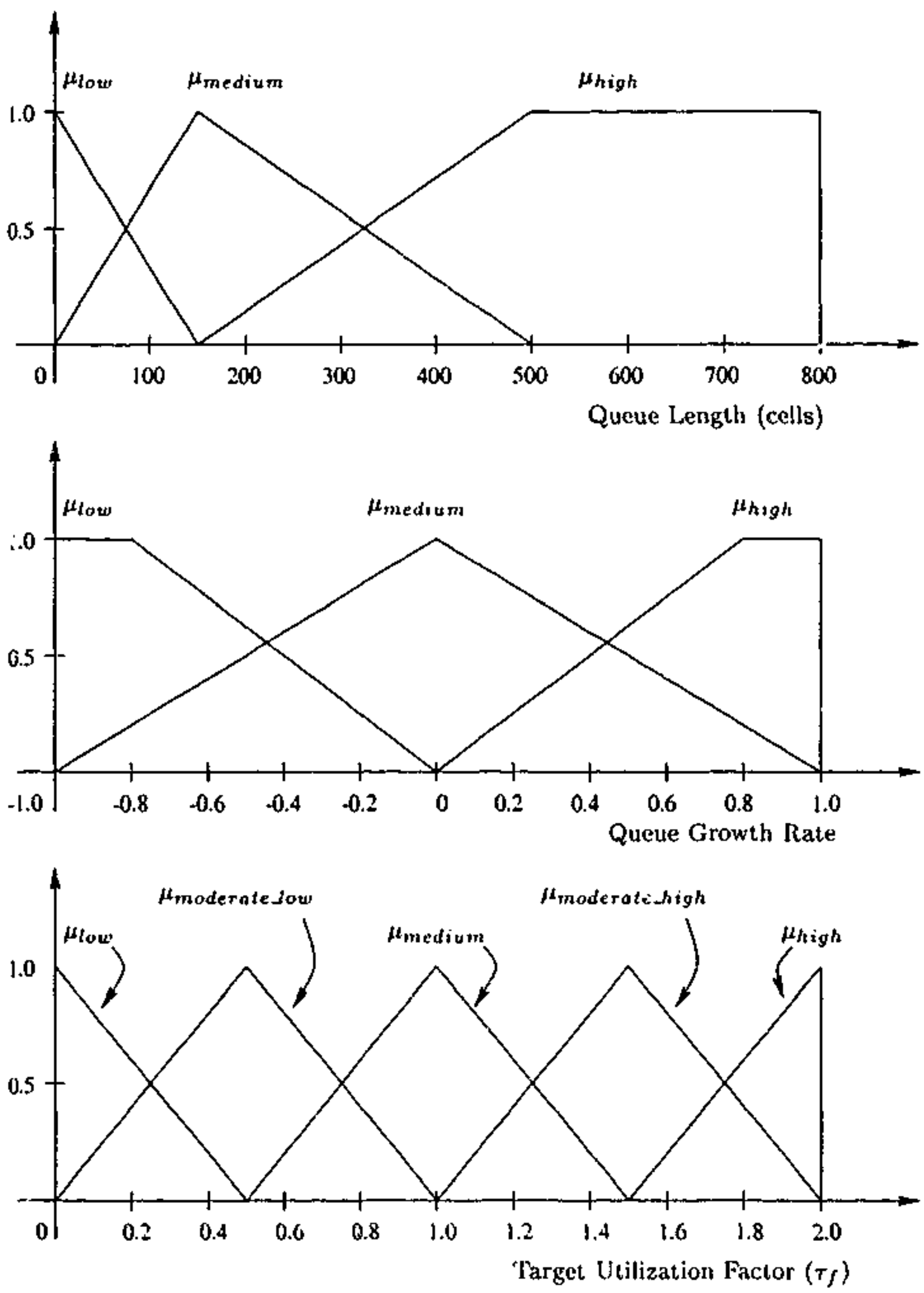


Figure 4.8: Membership functions of the linguistic values for representing the linguistic variables "*current ABR queue length*"(same for predicted queue), "*total queue growth rate*", and target utilization factor (τ_f). Note: The linguistic values shown here are used in Section 4.5.2 simulation.

if q is	**and qg is**	**and pq is**	**then τ_f is**
low	low	low	high
low	low	medium	moderate_high
low	low	high	medium
low	medium	low or medium	medium
low	medium	high	moderate_low
low	high	low or medium	medium
low	high	high	moderate_low
medium	low	low	moderate_high
medium	low	medium or high	medium
medium	medium		medium
medium	high		moderate_low
high	low		moderate_low
high	medium	low	moderate_low
high	medium	medium or high	low
high	high	low	moderate_low
high	high	medium or high	low

q:current ABR queue; qg: total queue growth rate;
pq: predicted ABR queue

Table 4.1: Linguistic Rules for the Fuzzy Logic ABR Service Congestion Controller.

be high, to allow the scheme to maintain a small queue length to avoid link under-utilization.

- In steady state, the target utilization is medium so that the rate and queue can be maintained at a constant level.

Since the fuzzy system has three inputs, it has a total of 27 linguistic rules. After optimization, 16 independent rules are adopted. Table 4.1 shows the linguistic rules for the fuzzy logic congestion controller.

4.5 Performance Evaluation

In this section, simulation results are presented to illustrate the performance of the fuzzy logic congestion control scheme and its advantages over the conventional scheme. The fuzzy logic congestion control scheme is first applied to the ERICA+ and BEMO algorithms. The performance is then compared with the conventional congestion control scheme for the two switch algorithms.

It is important to note that the comparison is not between the two switch algorithms, instead it is made between the fuzzy logic and conventional congestion control scheme for the two switch algorithms.

Firstly, only single-class traffic is considered. Then in Section 4.5.2 multi-class traffic is used.

For all cases simulated, switches are assumed to be non-blocking, output-buffered and apply a FIFO service discipline of their queues. All links are assumed to have a propagation delay of $5\mu s$/Km. ABR sources are assumed to be well-behaved, and always transmit cells at the maximum allowed cell rate (ACR). The following simulation parameters are kept constant throughout the study: switch averaging interval is 100 cells and 250 *usec*, RDF=1/16, RIF=1 and N_{rm}=32.

4.5.1 Single-class Traffic

Three network topologies with ABR class only traffic are used for studying the fairness and convergence time of the schemes: the simple two-switch configuration, the *Parking-lot* configuration [CXK96, LF95] and the Generic Fairness Configuration-2 (GFC2) [Sim94]. The justification of using these models will be indicated later. In each case study, the buffer requirements at the switch are characterized in order to achieve zero cell loss and maintain a non-zero (small) queue in steady state.

4.5.1.1 Two-Switch Configuration

The two switches with five VCs configuration from Section 4.3 is used here to study the transient response and fairness of the control schemes when some sources start late. It has been used in the literature [ACA96, ACA97]. The parameters of the switches and sources have a similar setup except all links connecting sources or destinations to switches are set to 10 Km.

Figures 4.9 and 4.10 show the simulation results of the allowed cell rate for each connection and buffer requirements at switch SW1 for the conventional and fuzzy

		ERICA+		BEMO	
		Conventional	Fuzzy	Conventional	Fuzzy
Convergence Time ($msec$)	ACR	155	130	122	120
	SW1	200	145	271	120
SW1 Queue (cells)	MAX	809	279	235	100
	AVG	500	200	235	100

Table 4.2: Comparison of Convergence Time and Buffer Requirement for the Two-switch Configuration.

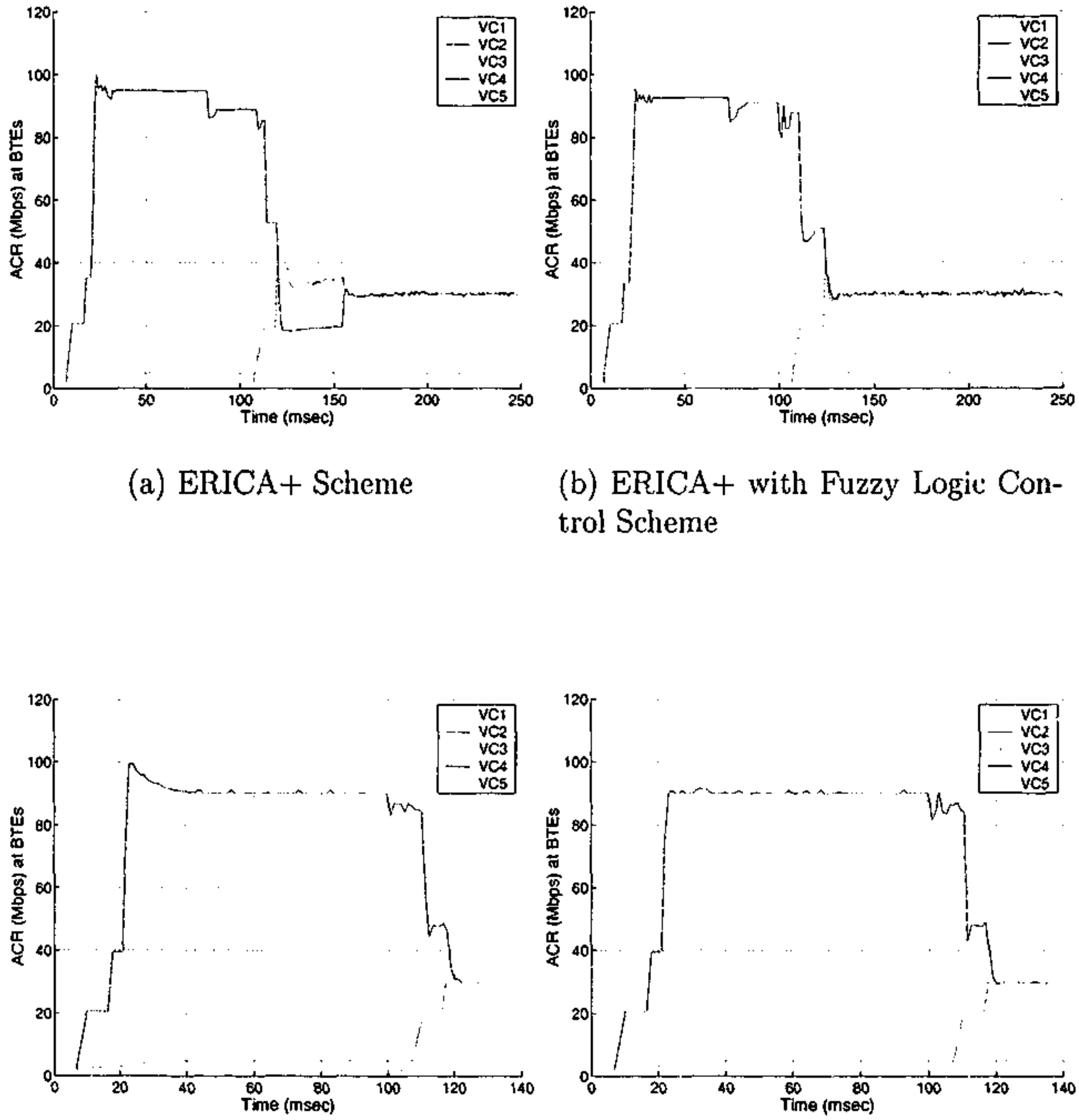
logic control schemes. The convergence time and buffer requirements are tabulated in Table 4.2.

From Figure 4.9, it can be observed that all connections are able to converge to the appropriate rate. In Figures 4.9(a) and 4.9(c), the conventional scheme shows a temporary over-allocation of rate on VC2 before feedback is established. On the other hand, the fuzzy logic control scheme does not suffer from a large rate oscillation before feedback is established. The ERICA+ scheme also exhibits large rate oscillation when connections 3 and 4 become active at time, $t = 100$ $msec$. A large rate oscillation generally results in buffer overflow problems.

From Table 4.2, it is shown that when connection 3 and 4 become active at time, $t = 100$ $msec$, the fuzzy logic control scheme is able to reduce the fair share allocation convergence time by 25 $msec$ and 2 $msec$ for the ERICA+ and BEMO algorithms respectively.

Figures 4.10(a) and 4.10(b) show that queue is built up slowly as the ICR is set to 2% of the link capacity. At transient, Figure 4.10(a) shows that there is a significant reduction in peak queue length and queue oscillation in the ERICA+ that incorporates the fuzzy logic control scheme. This results in a faster queue convergence time as indicated in Table 4.2. In steady state, the ERICA+ with the fuzzy logic control scheme maintains a queue size of 200 cells compared to 500 cells in the ERICA+ scheme.

Figure 4.10(b) shows that at transient, BEMO with the conventional control scheme has a higher queue length and converges slowly when the system is approaching



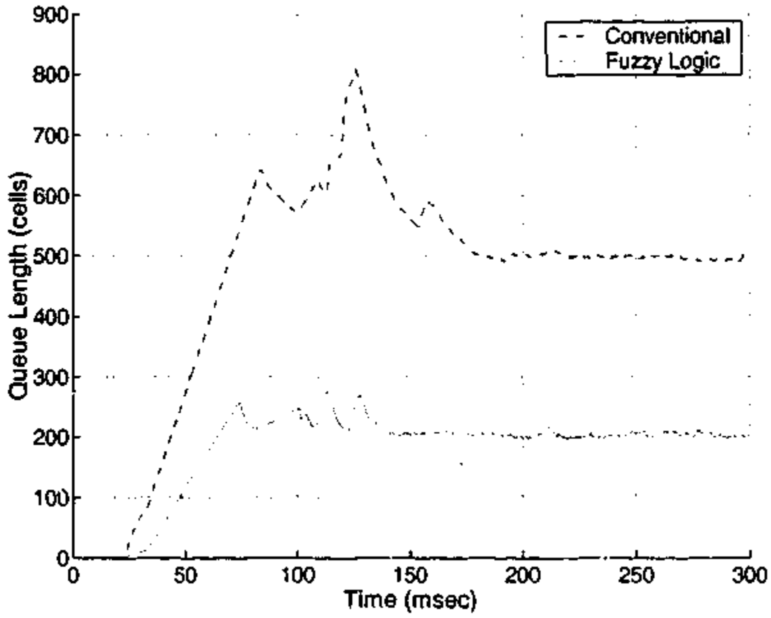
(a) ERICA+ Scheme

(b) ERICA+ with Fuzzy Logic Control Scheme

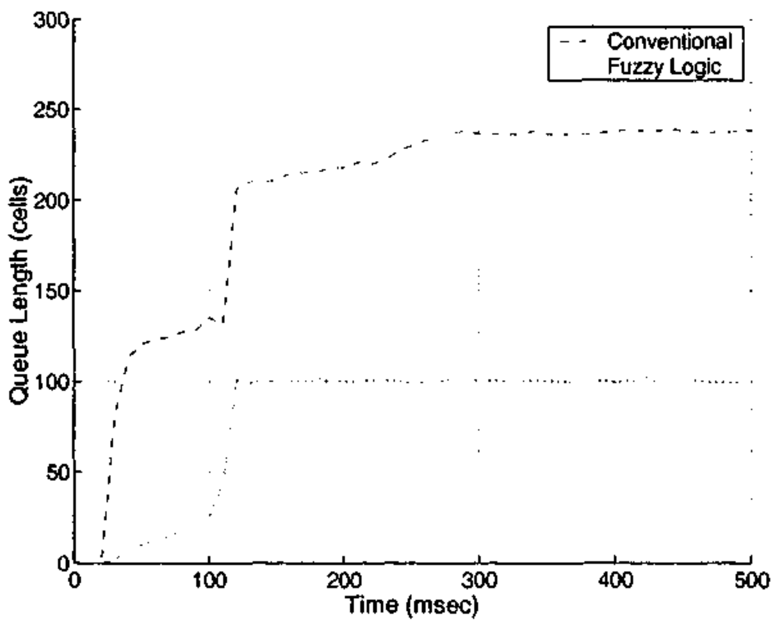
(c) BEMO with Conventional Control Scheme

(d) BEMO with Fuzzy Logic Control Scheme

Figure 4.9: The Allowed Cell Rate of the Two-Switch Configuration.



(a) ERICA+ Scheme and ERICA+ with Fuzzy Logic Control Scheme



(b) BEMO with Conventional Scheme and Fuzzy Logic Control Schemes

Figure 4.10: The Output Queue Size at Switch SW1 for the Two-Switch Configuration.

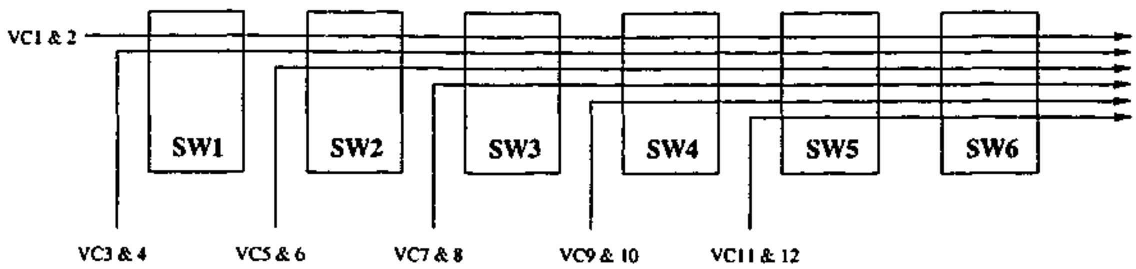


Figure 4.11: Network Model for the *Parking-lot* Configuration.

steady state. In steady state, BEMO with the fuzzy logic control scheme maintains a queue size of 100 cells compared to 235 cells in the conventional scheme.

Hence the fuzzy logic control scheme is able to achieve faster convergence time, smaller queue oscillation and smaller buffer requirements. With a small queue oscillation, the cell delay variation and cell loss ratio are kept to minimum. Smaller queue also means smaller queuing delay. Thus the fuzzy logic control scheme is capable of improving the overall QoS of the ABR service.

4.5.1.2 *Parking-lot* Configuration

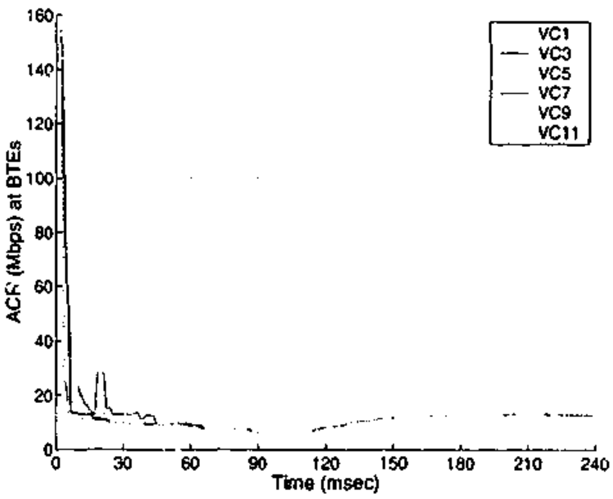
Figure 4.11 shows the *Parking-lot* configuration. This configuration is effective to illustrate the fairness problem, and has been demonstrated in literature [CXK96, LF95].

It consists of 6 switches with 12 VCs. VC1 to VC4 start at switch SW1; two of the remaining VCs start at each one of the other switches and all VCs exit at switch SW6. The distance between switches is 100 Km; and the links connecting sources or destinations to switches are 1 Km. All links are 155.52 Mbps. It is given PCR=155.52 Mbps, ICR=155.52 Mbps, RIF=1.

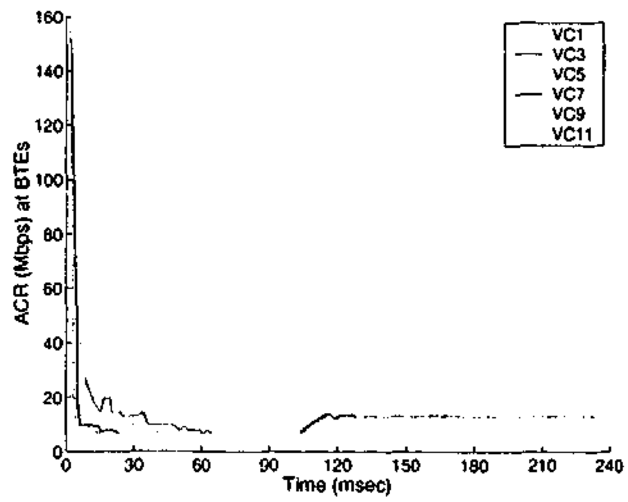
Figure 4.12 shows the allowed cell rate for the odd-numbered connections of the conventional and the fuzzy logic control schemes. Since there are a total of 12 VCs, each VCs gets a fair share of 12.96% of the available bandwidth. The simulation results show that both schemes are able to converge to the fair share. The rate convergence time is tabulated in Table 4.3.

		ERICA+		BEMO	
		Conventional	Fuzzy	Conventional	Fuzzy
Convergence Time (*msec*)	ACR	165	125	140	100
	SW5	400	130	150	110
SW5 Queue (cells)	MAX	5537	3783	3434	2228
	AVG	800	356	370	100

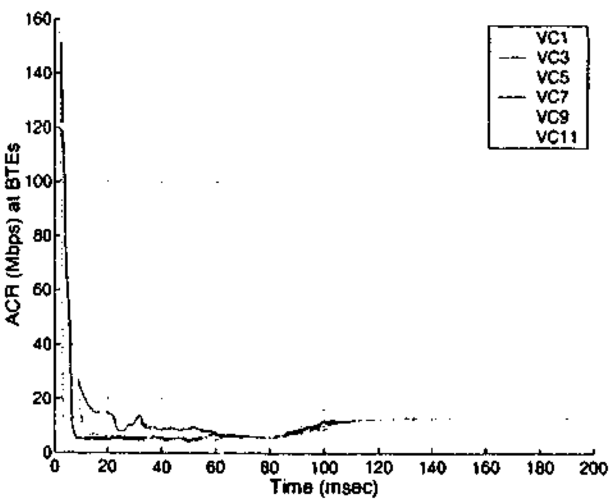
Table 4.3: Comparison of Convergence Time and Buffer Requirement for the *Parking-lot* Configuration.



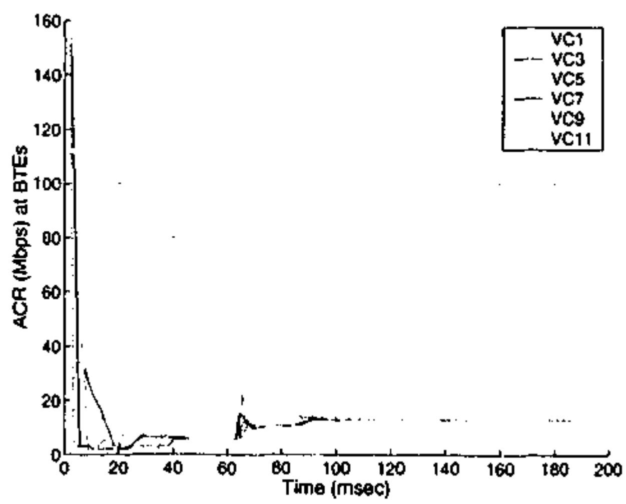
(a) ERICA+ Scheme



(b) ERICA+ with Fuzzy Logic Control Scheme

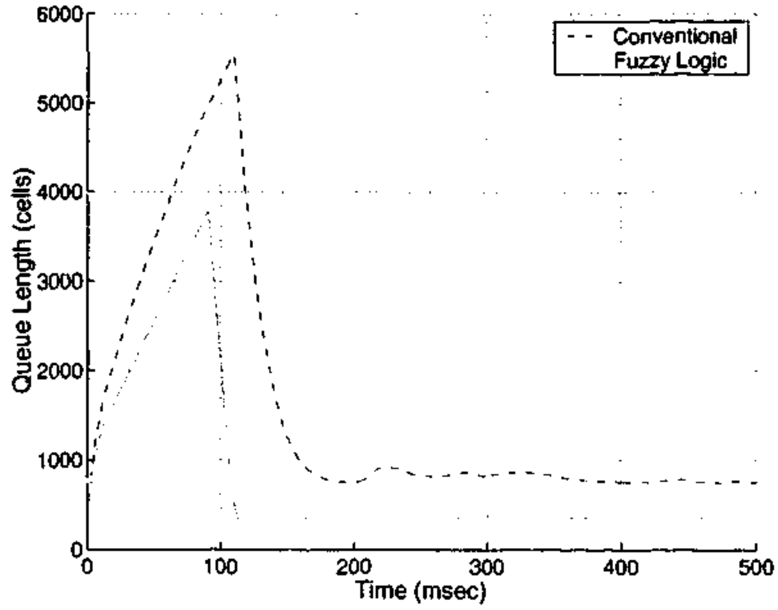


(c) BEMO with Conventional Control Scheme

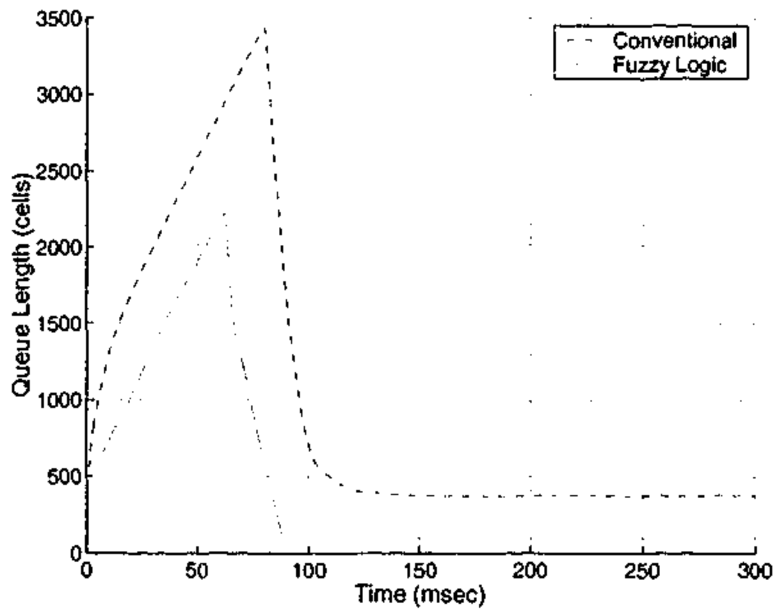


(d) BEMO with Fuzzy Logic Control Scheme

Figure 4.12: The Allowed Cell Rate of the *Parking-lot* Configuration.



(a) ERICA+ Scheme and ERICA+ with Fuzzy Logic Control Scheme



(b) BEMO with Conventional Scheme and Fuzzy Logic Control Schemes

Figure 4.13: Output Queue Size at Switch SW5 of the *Parking-lot* Configuration.

From Table 4.3, it is shown that the ERICA+ scheme converges at 165 *msec* whereas the fuzzy logic control scheme converges at 125 *msec*: and BEMO with the conventional scheme converges at 140 *msec* whereas the fuzzy logic control scheme converges at 100 *msec*. Therefore the fuzzy logic control scheme for both switch algorithms has a 40 *msec* faster rate convergence time.

Figure 4.13 shows the buffer requirements at switch SW5 for both schemes. Switch SW5 is the most congested switch, the queue length is large. Since the ICR is set to 100% of the link capacity, the queue is built up very fast. Table 4.3 tabulates the queue convergence time and buffer requirements.

Figure 4.13(a) and 4.13(b) show that at transient, the conventional scheme has a relatively high peak queue length. The peak queue length of the ERICA+ scheme is 5537 cells whereas the ERICA+ with fuzzy logic control scheme is 3783 cells. BEMO with the conventional and fuzzy logic control scheme have a peak queue length of 3434 cells and 2228 cells respectively. Hence the fuzzy logic control scheme has a smaller peak queue length compared to the conventional scheme for the respective switch algorithms. This is due to the fast convergence time on the fuzzy logic control scheme which is shown in Table 4.3, and thus the peak queue length built up at the switch is minimized.

After feedback is established, the queue length is controlled and maintained for both schemes. The ERICA+ with fuzzy logic control scheme maintains a queue size of 356 cells compared to 800 cells in the ERICA+ scheme. BEMO with the fuzzy logic control scheme maintains a queue size of 100 cells compared to 370 cells in the conventional scheme. Thus the fuzzy logic control scheme is able to maintain a smaller queue size.

Again, the fuzzy logic control scheme is shown to be able to optimize both the transient and steady state responses, i.e. converges quickly to a solid steady state from the initial conditions, and drains the queue produced in the transient phase rapidly.

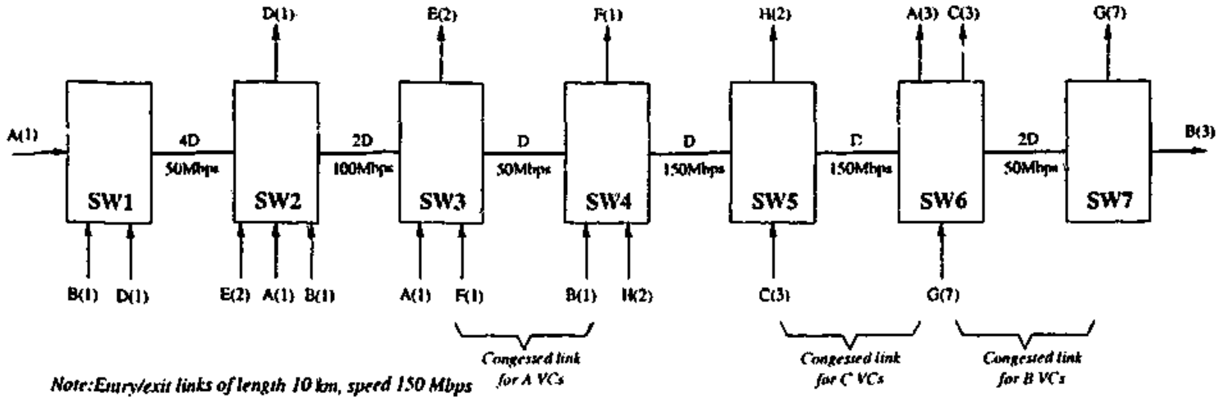


Figure 4.14: Network Model for the Generic Fairness Configuration-2 (GFC2).

Group	No. of VCs	Max-Min Rate (Mbps)
A	3	10.0
B	3	5.0
C	3	35.0
D	1	35.0
E	2	35.0
F	1	10.0
G	7	5.0
H	2	52.5

Table 4.4: Max-Min Rates for the GFC2 Network.

4.5.1.3 GFC2 Configuration

Figure 4.14 shows the network model used in the simulation. It is also known as Generic Fairness Configuration-2 (GFC2) [Sim94]. It has embedded the *parking-lot* and the *chain* configurations. This network model is widely recognized for the study of convergence time, robustness as well as fairness of a congestion control scheme.

The GFC2 network consists of 7 switches and 22 connections. These connections are grouped into eight classes (A-H) in a WAN configuration. The link distance, D, is assumed to be 100 Km, and all links connecting sources or destinations to switches have length equal to 10 Km. It is given ICR=PCR=150 Mbps. The converged max-min fairness rate for the different group of VCs are tabulated in Table 4.4.

Figures 4.15 and 4.16 show the allowed cell rate and the queue sizes for both the

Convergence Time (*msec*)	ERICA+		BEMO	
	Conventional	Fuzzy	Conventional	Fuzzy
ACR	300	210	230	210
SW5 Queue	1600	600	750	240

Table 4.5: Comparison of Convergence Time for the GFC2 Network.

conventional and fuzzy logic congestion control schemes. The simulation run time is 3 seconds.

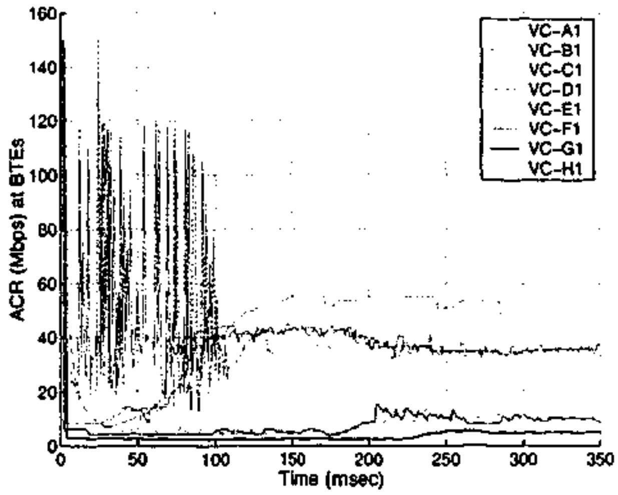
From Figures 4.15(a) - 4.15(d), it can be observed that both schemes are able to converge to the expected max-min fair share rates as shown in Table 4.4. In Figures 4.15(a) and 4.15(b), the ERICA+ has a temporary over allocation of rate before feedback is established. On the other hand, due to exact fair share computation, BEMO does not suffer from any prolonged large rate oscillation in the initial conditions. In steady state, BEMO is shown to have a more stable rate.

Figures 4.16(a) - 4.16(d) show the buffer requirements for both schemes. Since the ICR is set to 150 Mbps, the queue is built up very fast. Once the feedback is established, queues are controlled and maintained for both schemes.

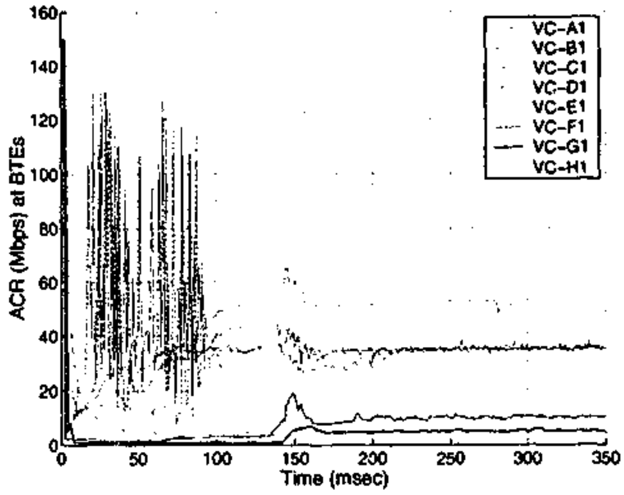
Table 4.5 shows the convergence time of the ACR and SW5 queue for the respective explicit rate algorithms. The fuzzy logic control scheme significantly reduces the rate allocation convergence time by 90 *msec* and 20 *msec* for the ERICA+ and BEMO algorithms respectively. It also exhibits a faster queue convergence time for both schemes.

Table 4.6 shows the maximum, average and deviation of queue length of switch SW1 to switch SW6 for both schemes. There are slight differences in the maximum queue length since it depends on the ICR, the distance of the feedback path and the queue control functions. Once the feedback is established, the fuzzy logic control scheme is shown to have a lower maximum queue length, especially the SW3, SW4, SW5 and SW6 queues.

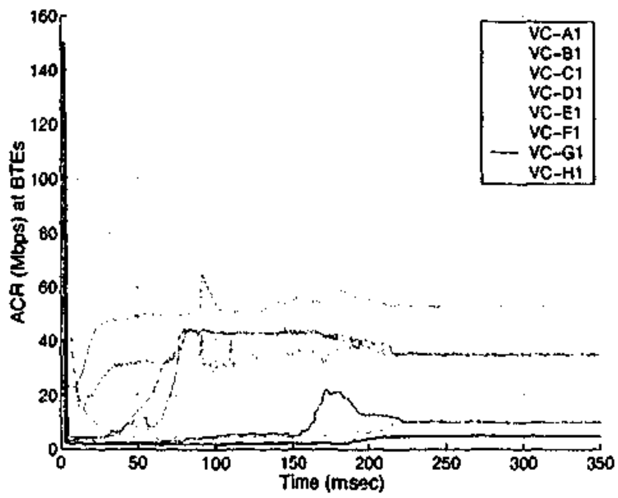
The average queue length and queue deviation are computed after 400 *msec* when the rates have converged. The simulation results show that the fuzzy logic control



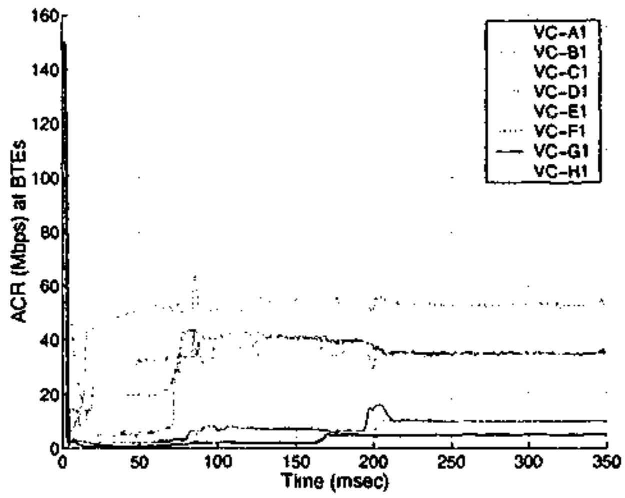
(a) ERICA+ Scheme



(b) ERICA+ with Fuzzy Logic Control Scheme

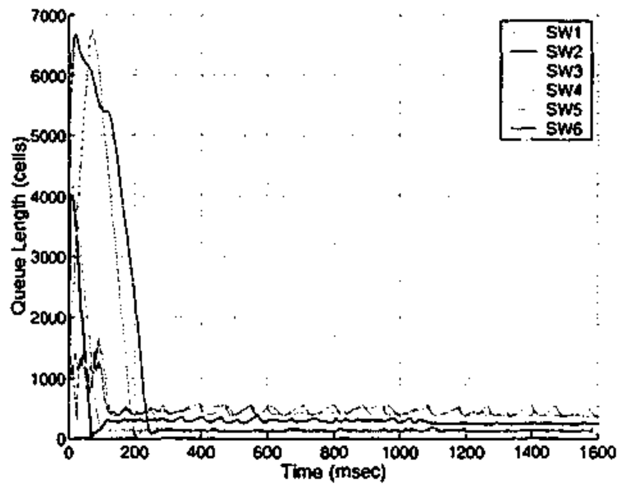


(c) BEMO with Conventional Control Scheme

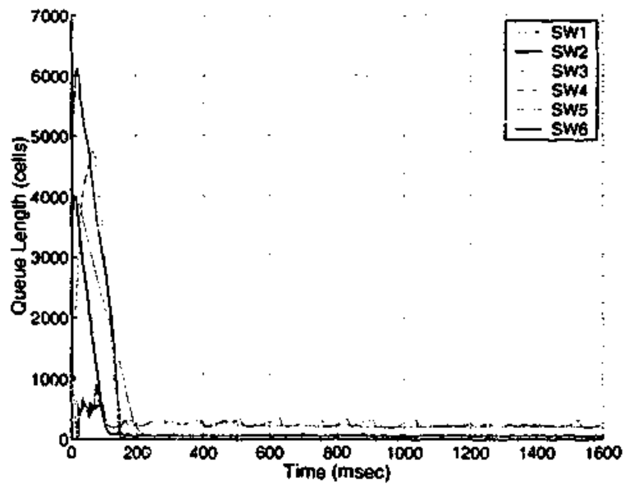


(d) BEMO with Fuzzy Logic Control Scheme

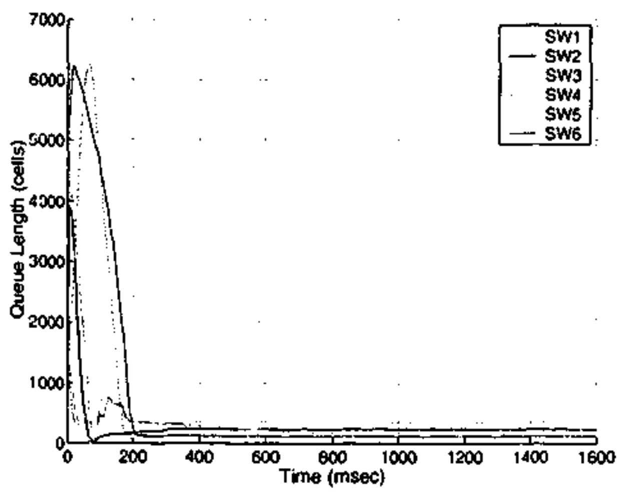
Figure 4.15: The Allowed Cell Rate of the GFC2 Network.



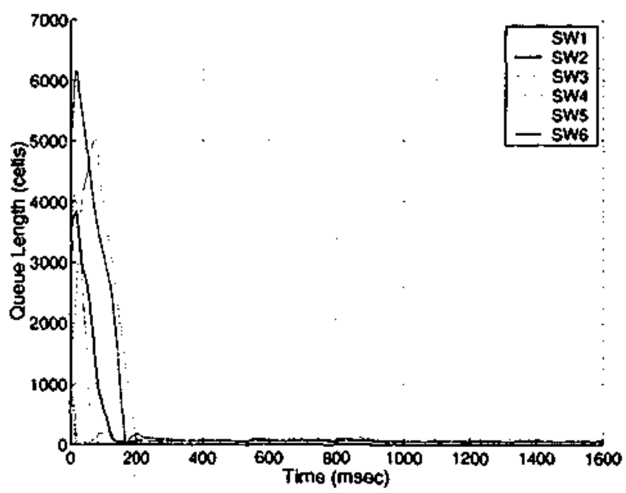
(a) ERICA+ Scheme



(b) ERICA+ with Fuzzy Logic Control Scheme



(c) BEMO with Conventional Control Scheme



(d) BEMO with Fuzzy Logic Control Scheme

Figure 4.16: Output Queue Size of the GFC2 Network.

Queue (cells)	**ERICA+**		**BEMO**	
	Conventional	**Fuzzy**	**Conventional**	**Fuzzy**
SW1	4179†	4175	4116	4110
	125.15‡	57.03	116.73	49.29
	3.40*	3.16	2.45	2.13
SW2	4023	4013	3898	3887
	265.62	57.64	237.12	51.16
	27.69	5.18	2.26	1.84
SW3	6771	4746	6265	5035
	134.50	60.64	120.36	51.99
	9.62	8.44	2.69	2.60
SW4	1424	1183	1320	1154
	445.02	253.90	348.78	97.14
	50.67	41.83	6.11	2.94
SW5	1677	1049	1169	862
	400.90	214.11	181.48	97.18
	49.97	12.70	23.69	2.52
SW6	6673	6129	6233	6163
	137.58	58.62	121.97	58.56
	15.39	7.94	3.32	3.30

†maximum; ‡mean; *queue deviation

Table 4.6: Comparison of Queue Length for the GFC2 Network.

scheme has lower average queues. It also exhibits a much more stable queue length when compared to the conventional scheme as the standard deviation is smaller.

Hence the fuzzy logic control scheme is able to adapt well to the complex network configuration and traffic fluctuations, resulting in smaller rate oscillation, faster convergence time in rates and queues during transient response, stable and smaller buffer requirements in steady state.

4.5.2 Multi-class Traffic

The two switches with five VCs network topology in Figure 4.17 is used to compare the QoS performance of the conventional and fuzzy logic congestion control schemes with multi-class traffic.

The network consists of one VBR and CBR sources, and three ABR sources. It is a WAN configuration. The distance between switches can be varied from 100 Km

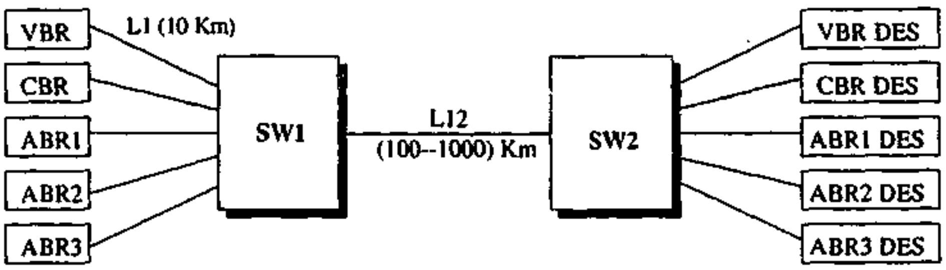


Figure 4.17: Network Model for the Two-Switch Configuration.

to 1000 Km; and the links connecting sources or destinations to switches are all 10 Km in distance. All links are 155.52 Mbps.

CBR/VBR traffic is set to 80% of the link rate. The CBR source transmits cells constantly at one-third of the link rate which is 51.84 Mbps. The VBR source is modeled as a Poisson distributed on-off source which alternates between active and idle periods, and transmits cells at a variable rate with a mean bit rate of 72.5 Mbps during active periods. The duration of active and idle periods are 2000 μs and 15000 μs respectively. The ABR sources are given PCR=ICR=155.52 Mbps. They become a variable quantity with the existence of CBR/VBR traffic, i.e. they share the remaining bandwidth left over by the CBR/VBR traffic.

An overflow or empty output buffer will affect either the efficiency or QoS. Ideally for optimal network efficiency and QoS, the buffer should only have a queue length of one cell. However with the existence of VBR traffic, a queue length of one cell is too small. Whenever the VBR queue is drained, it takes some time to get ABR cells from the source. Hence it is better to maintain a larger ABR queue in the switch, so that the queued cells use the available capacity while the RM cells traverse the link and inform the source to increase its rate. Thus the ABR output buffer is set to a maximum of 800 cells.

Since CBR/VBR traffic has a higher priority than ABR traffic, and this simulation study is to investigate network efficiency and QoS of the ABR service, the CBR/VBR queue is thus assumed to have an infinite buffer length.

Figure 4.18 shows the behavior of the CBR/VBR and ABR queues respectively. It can be observed that both queues are very "bursty". Since the ABR queue has a

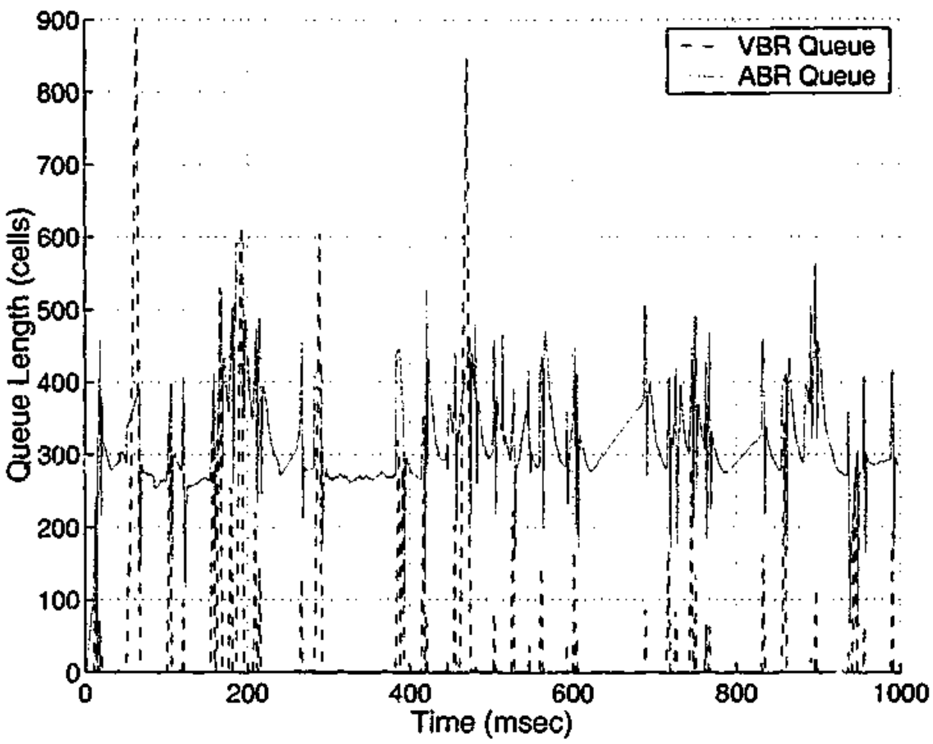


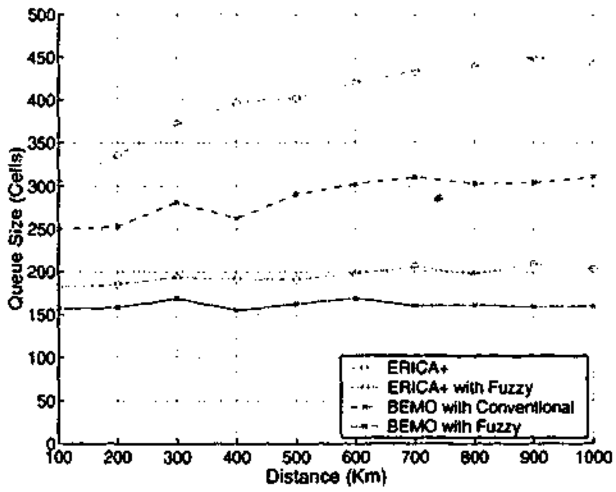
Figure 4.18: The behavior of the CBR/VBR and ABR queues when using the ERICA+ scheme with link distance 100 Km.

lower priority than the CBR/VBR queue, it is built up when there is CBR/VBR traffic; and it starts to decrease when the CBR/VBR queue is drained.

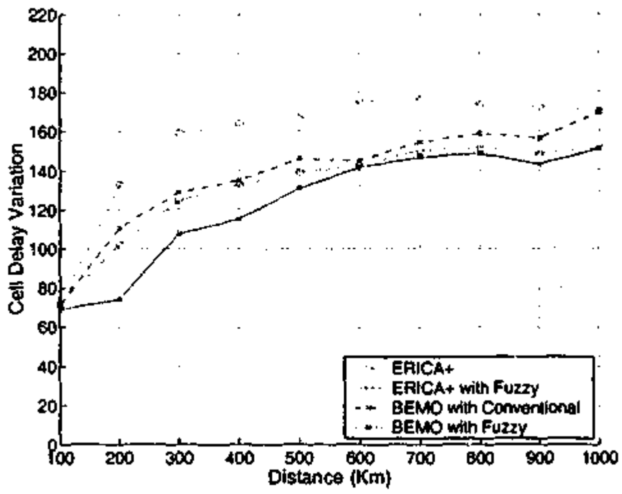
Figure 4.19 shows the performance of the conventional and fuzzy logic congestion control schemes on the respective explicit rate allocation algorithms with different link distances (L12). The two-point definition is used for CDV before and after the switch. The maximum of CTD and CDV are affected by the control threshold, thus average queuing delay and delay variation are more representative for the performance in this case study.

Figure 4.19(a) shows the average ABR queue at switch SW1 with different link distances. The fuzzy logic congestion control scheme has a smaller average queue length compared to the conventional scheme on both switch algorithms. It also has a lower CDV. This is evident from Figure 4.19(b).

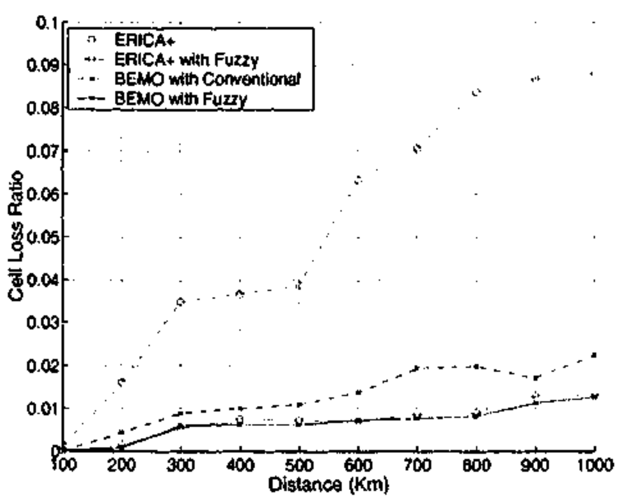
Figure 4.19(c) shows the CLR of the ABR queue with varying distances. There is no cell loss at 100 Km for both schemes as the buffer is not full. Beyond 200 Km, the buffer is overflown for both schemes, and cells are lost. The conventional congestion control scheme is shown to have a relatively high CLR especially when



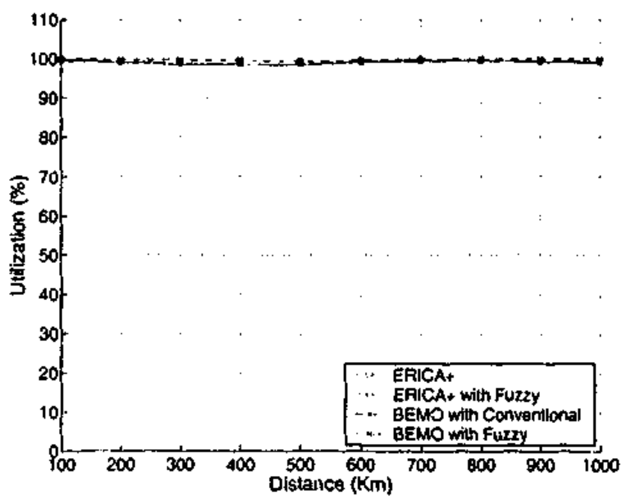
(a) Average ABR Queue



(b) ABR Queue Cell Delay Variation (CDV)



(c) Cell Loss Ratio of ABR Queue



(d) Average Link Utilization

Figure 4.19: Simulation results of the Two-Switch Configuration with Multi-class Traffic.

the link distance is large. This is due to the increasing number of cells in the ABR queue as the link distance is increased.

Figure 4.19(d) shows the link utilization on link L12 for both schemes. Overall, both schemes exhibit high link utilization. The fuzzy logic control scheme has a slight trade-off of link utilization for better CTD, CDV and CLR. However the link utilization is still above 98% of the link capacity.

Hence it can be seen that the results of the fuzzy logic congestion control scheme outperforms the conventional scheme in all aspects of QoS measurements.

4.6 Conclusions

This chapter has proposed a novel fuzzy logic congestion control scheme which does not use static threshold values or complex queue control functions. To address the problem of large propagation delays in the feedback control loop, the link utilization is targeted dynamically based on the predicted future buffer condition in the switch. This is achieved by using a fuzzy logic predictor and a fuzzy logic target utilization factor generator. The predictor estimates the ABR queue length one round-trip delay ahead. The predicted queue length, together with the queue growth rate and current queue length is provided to a fuzzy logic inference system which produces a target utilization factor. This factor is then incorporated in the ER value of BRM cells which convey congestion control information to the source.

The proposed fuzzy logic control scheme was applied to an approximate fair rate switch algorithm and an exact fair rate switch algorithm, namely, the ERICA+ switch algorithm and the BEMO switch algorithm. The performance of the fuzzy logic control scheme was then compared with the conventional scheme on the two switch algorithms. Three network models: two-switch configuration, *parking-lot* configuration and GFC2 configuration, with ABR class only traffic were used in the simulation studies. The simulation results show that the fuzzy logic congestion control scheme is able to optimize both the transient and steady state responses.

It exhibits a faster convergence time, lower buffer requirements, smaller rate and queue oscillation during transient periods, and is much more stable in rate and queue when in steady state. Using a two-switch configuration network model in the presence of VBR/CBR and ABR traffic, the fuzzy logic congestion control scheme exhibits lower buffer requirements, low cell loss for large round-trip delays, while still being able to maintain high link utilization. Hence, the overall requirements of QoS are improved. The proposed fuzzy logic congestion control scheme which can adapt well to complex network configurations and traffic fluctuations is thus proved to be a more effective mechanism than the conventional scheme. Although the proposed fuzzy logic control scheme was applied to the ERICA+ and BEMO switch algorithms, the same control strategy can also be successfully applied to any other switch algorithms on end-to-end feedback networks.

For the purpose of comparison to the proposed fuzzy logic control scheme, the BEMO switch algorithm has been enhanced by combining the exact fair rate computation algorithm with the conventional congestion control strategy of the ERICA+ switch algorithm. Hence, a queue control function and a fixed non-zero queuing delay are used to control the target link utilization, and thus control congestion. However, unlike the ERICA+ congestion control scheme, due to the exact fair rate computation, the enhanced BEMO algorithm does not need to keep track of maximum previous and current allocation of ER values in the RM cells in order to achieve max-min fairness. Besides providing fast convergence time, the enhanced BEMO is able to interoperate with other switch schemes which do not mark the ER field of RM cells.

In the ERICA+, mathematical equations are used to control queue and delay. Step, linear, hyperbolic and inverse hyperbolic queue control functions have been proposed and evaluated. From the simulation results, it has been established that the inverse hyperbolic queue control function has the best performance. This chapter has presented a further investigation of the performance of the hyperbolic and inverse hyperbolic queue control functions after a discovery of a minor error. An analytical study shows that the hyperbolic queue control function is preferred as it has a decreasing function when a switch is lightly overloaded. Using a two-switch network model, simulation results reveal that the hyperbolic queue control function performs better in terms of rate convergence time and buffer requirements.

Traffic flow and congestion control schemes proposed and investigated so far are closed-loop reactive control. The next chapter focuses on the open-loop preventive congestion control. It is performed at the connection establishment phase and is also known as connection admission control.

Chapter 5

Fuzzy Logic Measurement-based Connection Admission Control

The previous chapter has proposed methods of alleviating network congestion which are carried out during the life time of a connection. This chapter proposes techniques for preventing congestion which are performed at connection setup phase. It is known as connection admission control (CAC). Various existing CAC approaches are first investigated. The advantages of fuzzy logic prediction and control are exploited, and a novel predictive fuzzy logic measurement-based CAC is proposed. The performance evaluation is also provided.

5.1 Introduction

ATM networks support multi-class services with widely different traffic characteristics and QoS requirements. In order to maximize network efficiency and guarantee QoS, these networks adopt the concept of statistical multiplexing of traffic streams. At times, however, due to statistical fluctuations of traffic streams in the network, congestion can occur and the promised QoS can deteriorate. CAC is an important preventive measure against congestion. The main role of a CAC is to determine whether a new connection request should be accepted or rejected at the connection establishment phase. The decision is based on the availability of resources (bandwidth and buffer) and whether the QoS can be guaranteed for the new connection without degrading the promised QoS of the existing connections.

Unlike ABR and UBR services, CBR and VBR services are not delay tolerant and require real-time delivery, thus CAC gives priorities to CBR and VBR connections. In other words, bandwidth is given to real-time connections, and the remaining bandwidth is for non real-time connections. This ensures high link utilization and maintains QoS objectives for different services. Since CBR connections deliver cells at a constant rate, the CAC method for CBR service is based on peak rate allocation. It can then be divided into the methods that neglect CDV and the methods that take into account for CDV. Further reading can be found in [RMV96]. This chapter focuses on CAC approaches for VBR service since bandwidth requirements for VBR sources are highly fluctuated and uncertain. It is a challenge to design an appropriate CAC scheme that achieves high link utilization with guaranteed QoS.

The CAC algorithms are not specified by the ITU-T or ATM Forum [ITU96, ATM96] because a CAC scheme is more specific to each switch architecture, queuing and scheduling implementation. Since connections are setup and torn down in real-time, CAC algorithms cannot be computationally intensive as they are executed in real-time and on every connection. Computational intensive algorithms directly affect the connection setup rate and setup delay.

There are many CAC approaches reported to date [PE96, SYT99]. Conventional CAC relies on analytical models in which parameters are based on *a priori* or user-specified traffic descriptors [GAN91, GH91, Kel91]. This approach views each connection independently at a queuing point. The analytical model translates the traffic parameters of each connection into an *effective bandwidth* such that the QoS constraints are satisfied. This concept of effective bandwidth has an additive property in which the total effective bandwidth of N superposition connections is equal to the sum of their individual effective bandwidths. This method could be very conservative as the actual bandwidth needed by N connections could be far less than the sum of the bandwidths of N connections. In addition, VBR traffic characteristics such as Sustainable Cell Rate (SCR), Intrinsic Burst Tolerance (IBT) and Cell Delay Variation (CDV), are difficult for users to specify precisely. Hence, an analytical model may not be able to capture the statistical nature of traffic sources. This could lead to either degraded QoS or link under-utilization. To overcome these drawbacks, recent researchers have proposed measurement-based CAC [GT99, ZL98a, DJM97, JDSZ97] and have exploited the advantages of statistical multiplexing. Admission decisions are made based on the measured incoming traffic and network performance parameters such as link load, queue occupancy and cell loss ratio (CLR).

Measurement-based CAC schemes have the capabilities to capture the on-line traffic dynamics and system performances, and achieve a higher link efficiency. However, admission decisions and promised QoS to users may be compromised due to the inaccuracies and errors involved in the measurement. Computational intelligence techniques such as fuzzy logic control are useful in solving complex processes which cannot be described precisely by exact mathematical expressions, or available information cannot be interpreted precisely and with certainty.

The ability to forecast workload is important for efficient network performance. In [Qiu01, QSW99], a fuzzy logic predictor is used to model and predict VBR traffic. The predictor offers a viable alternative to analytical models. Simulation results show that the predictor has a smaller error mean and standard deviation compared to conventional autoregressive (AR) predictions.

Based on the concepts of free bandwidth and the adaptive weight factor in [ZL98b, ZL98a], this chapter proposes a novel measurement-based CAC using a different approach. The proposed CAC adopts the concept of statistical multiplexing in combination with the application of fuzzy logic prediction [Qiu01, QSW99] and inference. Its main aim is to exploit the advantages of fuzzy logic prediction and control on measurement-based CAC under different VBR traces.

Using a conventional CAC [GAN91], the benefits of fuzzy logic prediction are first evaluated. Therefore, three methods are used to obtain the traffic descriptors of a new connection in the conventional CAC, *a priori*, measured non-predictive and fuzzy logic predictive traffic descriptors. With the promising performance of fuzzy logic prediction in the conventional CAC, it is then incorporated in the proposed fuzzy logic measurement-based CAC to provide further benefits.

This chapter first discusses CAC related issues, such as traffic modeling, connection traffic descriptors and design issues. It then describes some of the CAC approaches proposed in the literature. Applications of fuzzy logic prediction and control on CAC are proposed in Section 5.4. Fuzzy logic prediction is first applied to a conventional CAC for the purpose of evaluation and comparison. Then, a novel fuzzy logic CAC scheme is proposed in Section 5.4.2. Performance of these proposals is evaluated. A simulation model which consists of homogeneous and heterogeneous traffic is first discussed. In Section 5.5.2, the effects of the measurement interval are investigated by comparing the link utilization and cell loss ratio against the measurement periods. Subsequently, the effects of fuzzy logic prediction are evaluated using the conventional CAC scheme and the proposed measurement-based CAC scheme. The performance of the proposed fuzzy logic CAC is examined by comparing it to the conventional CAC.

5.2 Traffic Modeling, Descriptors and Design Issues for CAC

This section discusses some of the issues related to CAC, which include traffic modeling, connection traffic descriptors and design issues.

5.2.1 Traffic Modeling

The decision to admit or reject a new connection is evaluated by cell-level network performance. To achieve this, traffic characteristics for the new connection as well as the existing connections have to be known. Therefore, traffic models are required for performance evaluation (e.g. loss probability or delay) and dimensioning. These models can be described mathematically by a few statistical parameters, and are able to capture the significant statistical information of the actual traffic.

Many mathematical traffic models for voice, data and video have been proposed to date [Ada97]. They can be classified into two categories: non self-similar which has significant correlations for small lags (e.g. on-off process, Markov modulated process and autoregressive process); and self-similar which shows long-range dependence (e.g. fractional Brownian motion).

The traffic represented by an on-off process has alternate *on* and *off* periods, and the length of each period is independent. The distribution of the *on* and *off* periods is also independent. Cells are generated during the *on* period. The on-off process is often used to model the arrival process of a segmented packet into cells. A Markov modulated process is a process in which arrival rates are modulated by a finite number of states. This process is grouped into continuous and discrete time intervals, and the possibility of group arrivals. The most commonly used Markov modulated process are *Markov modulated Poisson process* (MMPP) and *Markov modulated fluid process* (MMFP). These processes are commonly used to model the arrival process of a single source or multiple sources traffic. The autoregressive (AR) process is another way to represent a traffic model based on a Gaussian distribution. A first-order AR model is given by:

$$A_n = aA_{n-1} + b\sigma \tag{5.1}$$

where A_n is the arrival rate during the nth interval, σ is a Gaussian white noise, and a and b are real numbers with $|a| < 1$. This model is proposed in [MAS+88] to characterize a video stream.

There are many approaches to generate self-similar traffic, fractional Brownian motion (FBM) [MN68] is one of the approaches. The FBM is a continuous zero mean Gaussian process. If the arrival rate within $(0, t)$ interval is Z_t, then $Z_0 = 0$, the mean bit rate is $E\{Z_t\} = 0$; for all t, $E\{Z_t^2\} = t^{2H}$ where H is the Hurst parameter. If a process A_n is modeled by FBM, then

$$S_n = nE\{A_n\} + \sigma Z_n \tag{5.2}$$

where S_n is the sum of the A_n values from interval 1 to n, σ is the standard deviation of the arrival rate in one interval. Thus, Equation (5.2) is determined by the means, the standard deviation and the Hurst parameter. Note the Hurst parameter of Z_t is the same as the A_n process.

Although some traffic models do capture the statistical significance of traffic characteristics, there is no one model which could describe all traffic streams. Moreover the arrival process of some traffic models, e.g. the Markov modulation process, is state-dependent; others may tend to over-estimate (lower loss and delay) or under-estimate traffic performance [NAZH95].

5.2.2 Connection Traffic Descriptors

At a connection establishment phase, the user and the network negotiate a traffic contract which consists of source traffic descriptors, QoS requirements and the conformance definition (i.e. the definition of traffic behavior). This contract enables an efficient network operation while at the same time ensuring that the QoS requirements of each connection are met. A connection admission control makes use of this contract, together with the cell delay variation tolerance (CDVT), to decide whether to admit or reject the connection.

This contract and the CDVT are also known as *connection traffic descriptors*, and have been standardized by the ITU-T and ATM Forum [ITU96, ATM96] for different categories of service. As illustrated in Chapter 2, the traffic descriptor for VBR traffic is a set of traffic parameters which consists of peak cell rate (PCR). sustainable cell rate (SCR), maximum burst size (MBS) and the CDVT. The QoS requirements for rt-VBR are CLR, CTD and CDV, whereas nrt-VBR traffic just requires CLR.

5.2.3 Design Issues

The decision to admit or reject a new connection can impact the performance of networks. This section outlines the design issues which CAC designers must consider for the design and implementation of a low cost, fast and high efficiency connection admission controller [SYT99].

Bandwidth Efficiency and CLR: CLR is a performance measure specified by a user, and bandwidth efficiency is a performance measure within a network. One of the main goals of CAC is to achieve high link efficiency while maintaining the CLR requirement.

Implementation Complexity: Due to high-speed and high-capacity networks, and admission control is a real-time traffic control procedure performed at connection setup time, admission processing time has to keep to a minimum. In other words, admission control should be as simple as possible so that it can be performed in real-time. Admission control implementation using complex hardware and/or software mechanisms should also be avoided.

Scalability: Due to the rapid network growth (both in size and distance) and that many connections are multiplexed onto a high-speed link, per-connection accounting and computational complexity that is proportional to the number of connections should be avoided.

Dependency on Traffic Model: Due to the rapid emergence of new applications, it is difficult or even impossible to keep up with the modeling of every new traffic flow. Hence admission control which depends on traffic modeling should be avoided; otherwise the applicability is limited.

Statistical Allocation: For a non-statistical allocation, the sum of the connection peak rates is not allowed to exceed the output link rate. It has the advantages of being simple to implement, minimal cell delay and no cell loss due to the fact that a buffer is hardly overflown. However high link utilization cannot be achieved if a large number of traffic flows are bursty. In contrast, a statistical allocation allows the sum of the connection peak rate to exceed the link rate. High link utilization gain can be achieved and many connections can be admitted by carefully balancing the increased complexity in the traffic controls, which must ensure the accompanying cell loss and delay meet the QoS specifications. Therefore, the statistical allocation is preferred as a service provider could maximize profit margins while achieving high link utilization.

5.3 Classification of CAC Approaches

Many CAC schemes have been widely studied and analyzed in the literature to date [PE96, SYT99]. In this section, four different approaches for implementing CAC are discussed. The first approach is based on peak rate allocation, and it is the simplest and the most conservative. The second is the effective bandwidth approach in which each connection is viewed independently at a queuing point. These two approaches are mostly model-based CAC as *a priori* traffic descriptors are required in terms of the parameters of an analytical model. In contrast, the third approach is based on measurements in which no assumption about traffic characteristics is made. Finally, the unconventional approach and also the fourth approach is based on fuzzy logic and/or neural networks.

5.3.1 Peak Rate Allocation

This is the simplest CAC and is also known as non-statistical allocation. A connection is admitted if the sum of the peak rates of all existing connections plus the peak rate of the new connection is less than the output link capacity on every link of the new connection traversing route, otherwise it is rejected.

$$\sum_{i} P_i \leq \textit{Link capacity} \tag{5.3}$$

where P_i is the peak rate associated with connection i.

Peak rate allocation is very inefficient and conservative because the peak rate is several times higher than the mean rate. In [WK90], it is shown that the link utilization can be as low as 5% when the *burstiness* of a VBR connection is small. The burstiness is defined by the ratio of SCR/PCR.

5.3.2 Effective Bandwidth

The effective bandwidth approach views each connection independently at a queuing point. Traffic parameters of each connection are translated into an *effective bandwidth* or *equivalent bandwidth* such that the QoS can be guaranteed if

$$\sum_i \alpha_i \leq Link\ capacity \tag{5.4}$$

where α_i is the effective bandwidth associated with connection i. [illegible]ally lies between SCR_i and PCR_i, i.e. $SCR_i \leq \alpha_i \leq PCR_i$. This ar[illegible]le of producing the statistical gain which is defined as the ratio [illegible]. A new connection is rejected if the sum of the effective band[illegible] existing connections plus the one yet to be admitted) exceeds the total ca[illegible] at least one of the network links along its traversing route; otherwise it is accepted.

The usefulness of effective bandwidth depends on the following two properties:

1. *Independence.* Effective bandwidth of a given connection is only related to the statistical properties of that connection and the buffering capacity in the network.

2. *Additivity.* The total effective bandwidth of the superposition of N connections is equal to the sum of their individual effective bandwidths.

With the independence and additive properties, a CAC simply adds or subtracts the effective bandwidth of a connection from the total effective bandwidth during a connection setup or release process. However due to the independence property, the effective bandwidth approach could be very conservative compared to the statistical multiplexing approach. This is because the actual bandwidth needed by N connections could be far less than the sum of bandwidths of N connections. The following sections discuss the effective bandwidth for rate envelope multiplexing and rate sharing.

5.3.2.1 Effective Bandwidth for Rate Envelope Multiplexing

A rate envelope multiplexing (REM) [RMV96] based CAC admits connections such that the total peak cell rates exceeds the link capacity. This is based on the assumption that not all connections transmit at their peak cell rate at the same time. It is also known as *zero buffer* or *bufferless approximation* because it assumes that there is little or no buffering available to real-time traffic. This makes the derivation of CLR simpler as no queuing analysis is required. Therefore, connections are admitted so that the aggregate arrival rate (AR, i.e. rate envelope) of the connections, or the total amount of work arriving at a queuing point, is less than the link capacity C with high probability. The CLR can be approximated by:

$$CLR = \frac{E\{(AR - C)^+\}}{E\{(AR)\}} \tag{5.5}$$

The CLR in this equation is simply the ratio of amount of work lost to the amount of work arrived. The operator $(.)^+$ considers only the positive differences (i.e. when $AR > C$) and is zero when $AR < C$.

The effective bandwidth for REM can be obtained by estimating the CLR using Equation (5.5) for a given distribution of AR such that the estimated CLR is not higher than the required CLR.

In [Kel91], Kelly developed the effective bandwidth for a multi-class system of sources by extending the Equation (5.5) :

$$AR = \sum_{j=1}^{J} \sum_{i=1}^{n_j} AR_{ji} \tag{5.6}$$

where n_j is the number of sources in class j, and AR_{ji} is the aggregate load generated by source i in class j.

The CLR for REM is then estimated by large deviation approximations:

$$log\ P(AR \geq C) \approx \inf_s \left[\sum_{j=1}^{J} (n_j M_j(s) - sC) \right] \tag{5.7}$$

where $M_j(s)$ is a logarithmic moment-generating function of the random variable AR_{ji}. The constraint on tail behavior $logP(AR \leq C) \leq -CLR$ is satisfied if the right-hand side component is less than or equal to $-CLR$. Let s attain an infimum at s^*, then the acceptance region is

$$A(n^*) = \left\{ n : \sum_{j=1}^{J} \alpha_j^* n_j + \frac{CLR}{S^*} \leq C \right\} \tag{5.8}$$

where $\alpha_j^* = M_j(s^*)/s^*$ is the effective bandwidth of a source of class j.

5.3.2.2 Effective Bandwidth for Rate Sharing

REM-based CAC assumes that the total amount of work arrived at a queuing point does not exceed the link capacity. However, this assumption may not be true for bursty traffic which has a much smaller mean rate compared to its peak rate. For example, VBR connections with $SCR << PCR$, transmitting cells at their PCR simultaneously, the sum of PCR can be much higher than the link capacity. To guarantee QoS requirements of the connections, such as CLR, buffering is required to temporarily store the large bursts and the link capacity is shared among the contending connections. Therefore there is a need for a CAC to take into account the buffering capabilities of the switches.

The techniques of effective bandwidth for rate sharing can thus be classified into lossless and loss-tolerant models. For a lossless model, since there is no cell loss, the loss probability is not explicitly considered. As for a loss-tolerant model, the effective bandwidth can be derived from a cell loss probability formula which takes into account the realistic models of the traffic streams as well as the buffering capabilities of switches. One approach is to keep a record of CLR against the buffer size produced by every modeled traffic source. However this is impossible if a large range of traffic sources is involved. Hence, an analytical approximate approach in which the asymptotically exponential of queue length distribution is developed.

Assuming a traffic stream i is fed to a single server queue of size B, the loss probability when the buffer overflown is given by:

$$P(queue\ length \geq B) \approx e^{-f(\alpha_i)B} \tag{5.9}$$

where α_i is the effective bandwidth of connection i needed to guarantee a given CLR, i.e. $P(queue\ length > B) \leq CLR$, and $f()$ is some function of α_i. With these two equations, the effective bandwidth of stream i can be obtained by:

$$\alpha_i = f^{-1}(-\ log\ CLR/B) \tag{5.10}$$

Kelly [Kel91] considers the effective bandwidth with the existence of an infinite buffer, i.e. lossless performance. The queue distribution is modeled by a $M/G/1$ process, and a source of class j with a Poisson distribution rate r_j and length G_j are assumed. Then the effective bandwidth of this source is expressed as:

$$\alpha_j = r_j \left[\mu_j + \frac{1}{2L}(\mu_j^2 + \sigma^2) \right] \tag{5.11}$$

where μ_j and σ_j are the mean and variance of G_j respectively, and L is the buffer size.

Gibbens and Hunt [GH91] develop an equivalent bandwidth for an exponentially distributed on-off source which on period is $1/\mu_i$ and off period is $1/\lambda_i$. Assuming the source transmits at a constant rate γ_i during the on period and $(log\ CLR)/B = \zeta$, the equivalent bandwidth α_i is given as:

$$\alpha_i = \frac{(\zeta\gamma_i + \mu_i + \lambda_i) - \sqrt{(\zeta\gamma_i + \mu_i - \lambda_i)^2 + 4\lambda_i\mu_i}}{2\zeta} \tag{5.12}$$

Simulation results indicate that the effective bandwidth of a connection is close to its peak cell rate for small buffers, and is around its mean bit rate for large buffers.

An equivalent capacity based on fluid-flow approximation for a two-state Markov source is proposed by Guérin et la. [GAN91]. Assuming the source has a peak rate R, utilization ρ and mean burst length b. With buffer capacity B, the equivalent capacity α_i can be estimated by:

$$\alpha_i = R \left[\frac{y - B + \sqrt{(y-B)^2 + 4B\rho y}}{2y} \right] \tag{5.13}$$

where $y = \alpha b(1-\rho)R$ and $\alpha = ln(1/CLR)$.

5.3.3 Measurement-based CAC

The CAC approaches described so far require an *a priori* traffic descriptor in terms of the parameters of an analytical model. In reality it is hard or even impossible for a user to specify or a control system to model traffic sources accurately, particularly with the vast varieties of traffic sources and rapid emergence of new applications. In addition, an analytical model may not completely capture the significant statistical information of traffic sources. These shortcomings could lead to degraded QoS or link over- or under-utilization. Moreover, traffic modeling usually involves complex and intensive computations and thus are not practical for real-time connection admission control.

To overcome the shortcomings of model-based CAC, researchers have been investigating measurement-based CAC. The measurement-based CAC does not make any assumption about the traffic characteristics. Instead, the decision to admit or reject a new connection is based on the real-time measurement of incoming traffic and network performance parameters such as link load, queue occupancy and cell loss ratio.

The following section briefly describes some existing measurement-based CAC schemes. Then in the later section, a measurement-based CAC which the proposed CAC in Section 5.4.2 is based on is presented in detail.

5.3.3.1 Exising Measurement-based CAC Schemes

Saito and Shiomoto [SS91] propose a CAC based on the upper bound of cell loss probability which is derived from the measured number of cells arriving at the output buffer during a fixed interval and the traffic parameters specified by users. If the estimated cell loss ratio is less than the desired QoS, the connection is accepted, otherwise it is rejected. The proposed scheme is demonstrated to have better link utilization compared to a CAC which only uses traffic parameters declaired by users.

Gibbens et la. [GKK95] introduce a decision-theoretic approach on a CAC in which burst-level (burstiness factor) and call-level (connection arrival rate) dynamics are incorporated in the framework. The decision to accept a new connection is based on whether the current measured load is less than a precomputed threshold. The choice of thresholds is provided by Bayesian decision theory. Compared with an optimal scheme which the burstiness factor and connection arrival rate are known, the proposed scheme is shown to perform as efficiently as the optimal scheme in terms of utilization and cell loss ratio.

An optimal framework that uses a linear Kalman filter for the estimation of the aggregate equivalent bandwidth required by the connections is proposed in [DJM97]. The framework requires the inputs of traffic descriptors from both the users and the aggregate bandwidth measurements in each output port of the switches. To meet the QoS objectives, the framework takes into account the connection level dynamics and provides information for evaluation of bandwidth to be reserved for possible estimation error. A new connection is accepted if the reserved bandwidth is less than the link capacity. The proposed scheme is shown to cope very well with unpredicted changes in traffic parameters. As a result, high link utilization is achieved while providing the required QoS.

Jamin et la. [JDSZ97] propose a measurement-based CAC for *predictive service*. Note that the predictive service is proposed in [SCZ93, CSZ92] and has a fairly, but not absolutely reliable bound on packet delivery times property. It is most suitable for those applications willing to tolerate occasional delay violations. The proposed CAC involves the measurements of the maximum delay and maximum rate utilization over a measurement interval. The good performance of the scheme depends on some tuning parameters, such as, sampling window size, measurement window size, utilization target and back-off factor. Simulation results show that the proposed scheme is able to provide a reliable delay bound for predictive service and achieves higher network utilization as compared with the guaranteed service, in particular, bursty sources.

In [GT99], Grossglauser and Tse use an analytical framework to study the impact of estimation errors, measurement memory, dynamics and separation of time-scales on a measurement-based CAC. Performance analysis of a large system is quantified using heavy traffic and Gaussian approximations at the call level.

5.3.3.2 CAC based on Overall Link Free Bandwidth

In [ZL98b, ZL98a], Zukerman and Tse propose a CAC based on the concept of *Overall Link Free Bandwidth* (OLFB). This free bandwidth consists of two components and is adjusted by an *adaptive weight factor* (W_i) according to the network conditions:

$$OLFB = W_i \cdot RSFB + (1.0 - W_i) \cdot PRAFB \tag{5.14}$$

where

1. Peak Rate Allocation Free Bandwidth (PRAFB): this is the difference between the total link bandwidth available and the sum of the PCR of all the connections on the same link.

2. Rate Sharing Free Bandwidth (RSFB): this is the difference between the total link bandwidth available and the *equivalent bandwidth* of the link. This equivalent bandwidth is computed by using an analytical equation which is based on the measured arrival rate and the aggregate traffic on the output buffer.

When the measured traffic level is higher than what was predicted by the CAC, there is a risk of violating the QoS requirements, thus more weight is given to the conservative PRAFB component. In contrast, if the measured traffic is lower than expected, then the more daring approach is adopted. In this case, more weight is given to the RSFB component.

The adaptive weight factor is determined by the link service rate (B_i) and the amount of work arriving (X_i) at the end of a sampling interval i. When the amount of work arriving is more than the predefined link threshold value (H), the weight factor is decreased by:

$$W_{i+1} = max\left\{0.0, \quad W_i - \frac{X_i - H}{B_i + Buffer - H}\right\} \tag{5.15}$$

On the other hand, if the amount of work arriving is less than the link threshold value, the adaptive weight factor is increased by:

$$W_{i+1} = min\{1.0, \quad W_i + F\} \tag{5.16}$$

where F is a fixed number and typical value is 0.01.

Using two types of traffic, both are modeled as a Poisson process with independent exponentially distributed holding time, CLR is fixed at 10^{-5}, and link rate is 51.84 Mbps, simulation results show that the proposed scheme achieves 70% efficiency while maintaining QoS requirements.

5.3.4 Fuzzy Logic and Neural Network CAC Schemes

Measurement-based CAC schemes have the abilities to capture the on-line traffic dynamics and system performances, and achieve higher possible link efficiency. However, measuring the statistics of traffic traces or system performance parameters requires additional processing power in the network. Hence, the measurement statistics have to be selected with caution to avoid an excessive processing burden on the network. In addition, admission decisions and promised QoS to users may be compromised due to the inaccuracies and errors involved in the measurement. In recent years, computational intelligence techniques, such as fuzzy logic controls and neural networks, have been applied to admission control with some success. Some of these techniques which apply to this area are described in this section. A neural network approach is proposed in [Hir90]. A traffic controller that handles both congestion control and admission control is presented in [CC96]. In [BLCT97, UH97], fuzzy logic admission controls based on cell loss ratio are developed, and in [MLN97], an adaptive CAC using a fuzzy estimator to estimate traffic characteristics and a neural network to estimate the cell loss probability is proposed.

An adaptive admission controller using a back-propagation neural network is proposed by Hiramatsu [Hir90]. The neural network learns the relation between offered traffic and service quality for single-bit-rate and multiple-bit-rate traffic. In the single-bit-rate traffic case, the history of observed cell arrival rates (or cell arrival pattern) is fed into the neural network. The cell arrival pattern is then classified into either the high-loss-rate group or the low-loss-rate group. The admission decision is then based on whether the corresponding cell loss rate for an input cell arrival pattern is greater than the target loss rate. In the multiple-bit-rate case, traffic is first divided into classes according to the declared traffic parameters, and then the optimum call admission boundary is obtained by the neural network through learning. The neural network based admission controller is shown to adapt well to changes in the arrival patterns and learn decision boundaries for various bit-rate distributions. In the single-bit-rate simulation, the calls are considered as a uniform distribution, and changes in call characteristics are achieved by changing the bit rate fluctuation of calls. In the multiple-bit-rate simulation, two-bit-rate classes are considered. The distribution of bit rates in each class has the same shape but different widths.

Cheng and Chang [CC96] present a fuzzy traffic controller that handles both congestion control and admission control. The input traffic is classified into real-time and non real-time traffic, and each has its own buffer. The system also reserves a portion of link bandwidth for the real-time traffic. Congestion control is managed by a *fuzzy congestion controller* whose inputs are the measured queue length, queue growth rate and cell loss. The controller yields a control action y depending on the congestion status of the system. A negative value of y denotes congestion situations whereas a positive value indicates congestion free periods. A *coding rate manager* and a *transmission rate manager* then inform the real-time and non real-time sources respectively to perform the selective discard or restoration function, and to reduce or increase the rate respectively, according to the value of the control action y. As for the connection admission control, a *fuzzy bandwidth predictor* predicts the equivalent capacity C_e required for the new call from the declared traffic parameters, such as peak rate, mean rate and peak rate duration. In the meanwhile, a *network resource estimator* computes the available network bandwidth C_a by subtracting and adding the bandwidth of call that has ended or just being admitted. A *fuzzy admission controller* is then decides whether to accept or reject the incoming call based on the p_l, y, and C_a values. Video, voice and data traffic are used in the simulations. The overall loss probability is defined as a weighted sum of the real-time and non real-time loss probabilities. Simulation results show that the fuzzy logic controller improves channel utilization by 11% compared to the Guérin's equivalent bandwidth approach [GAN91], and has a 4% lower cell loss rate compared to the two-threshold congestion control method.

Bensaou et la. [BLCT97] develop a fuzzy logic algorithm to predict the CLR in large-size systems. The fuzzy approximation (FA) algorithm estimates the CLR based on both information of small-size systems (e.g. small buffer size or small service rate), and the asymptotic behavior of the CLR of large-size systems. A CAC is then proposed which comprises two components. The first one consists of a set of virtual buffers with reduced service capacity to observe high cell loss with a small variance within a short measurement interval. The second component consists of a FA and a decision process. The FA algorithm determines the available bandwidth and the bandwidth required for the ongoing calls, and then the decision process makes a decision to accept or reject a new call based on the outputs of the algorithm. Simulation results reveal that the FA CAC performs much better than the conservative effective bandwidth approach in terms of reserved bandwidth against a number of sources.

A fuzzy logic CAC based on the possibility distribution of cell loss as a function of the number of calls per class is proposed by Uehara and Hirota [UH97]. With fuzzy α-cut theory, the range of the consequent (i.e. the CLR) are adjusted on-line by a learning algorithm so as to include the newly observed CLR. Energy functions are proposed for the learning algorithm. To make the estimation of upper bound of

CLR possible without over estimating the CLR, error back-propagation algorithm is used to adjust the α-level values. When observed CLR data is not available for the number of calls in a given class before they are admitted, new rules are generated on-line using exponential extrapolation from the neighboring ones. On-off and CBR traffic are used in the simulation studies. Simulation results shows that the proposed scheme provides good approximation of the upper bound of CLR for the admitted number of calls in a traffic class.

Mehrvar and Le-Ngoc [MLN97] propose an adaptive CAC scheme using a fuzzy estimator to estimate the traffic descriptor, i.e. traffic intensity and burstiness (Hurst parameters), from the aggregate traffic. The aggregate traffic consists of voice, video and data traffic. Based on the estimated traffic intensity and burstiness, the cell loss probability is then estimated by an Artificial Neural Network (ANN) [MLNH96]. The ANN is trained off-line for different types of traffic. Training data is obtained from the Markov modulated Poisson process (MMPP) for voice and video traffic and the Pareto modulated Poisson process (PMPP) for data traffic. If the estimated CLR is less than the required CLR, the new connections are admitted, otherwise they are rejected. Simulation results show that the proposed CAC is able to admit appropriate number of connections while maintaining the CLR requirement.

5.4 Application of Fuzzy Logic Prediction and Control to CAC

The neural networks and fuzzy logic controls on CAC described in the previous section have certain shortcomings. The back-propagation neural network approach proposed in [Hir90] requires a few thousand seconds on-line learning which is considered too long and impractical in high-speed network environments. The long training time is also a problem with the fuzzy-neuro approach [MLN97]. Moreover, in [MLN97], the cell loss probability of traffic flows is estimated by an ANN training data obtained from traffic modeling. In reality, it is difficult or even impossible for an analytical model to completely capture the significant statistical information of a traffic flow due to inaccuracies and uncertainties in the knowledge about traffic characteristics. The effective bandwidth approach using fuzzy logic proposed by Cheng and Chang [CC96] is conservative as the bandwidth is added or subtracted independently on a call setup or termination. In [BLCT97, UH97], on-line CLR measurement or estimation may take too long to be practical given the stringent QoS (e.g. in the range of 10^{-6} to 10^{-10}), and usually CLR data does not exist before calls are accepted. It is also a risky approach as it may be too late to take any action once the damage is done [ZL98a].

Based on the concepts of free bandwidth and the adaptive weight factor in [ZL98b, ZL98a] (also in Section 5.3.3.2), Section 5.4.2 proposes a novel measurement-based CAC using a different approach. Similar to [ZL98b, ZL98a], the proposed CAC adopts the concept of statistical multiplexing on equivalent bandwidth derivation. However in [ZL98b, ZL98a], the equivalent bandwidth is computed by using an analytical equation which is based on the aggregate traffic on the output buffer and the arrival rate. In addition, the adaptive weight factor is composed of mathematical expressions with some hard threshold values. In contrast, the proposed CAC does not use analytical models or hard threshold values for admission control. It uses a fuzzy logic predictor combined with a fuzzy inference system, and is suitable for on-line operation. To demonstrate the benefits of fuzzy logic prediction, in Section 5.4.1, it is applied to the conventional equivalent capacity CAC [GAN91]. Simulation results show that fuzzy logic prediction improves network efficiency of the conventional CAC. With the promised performance, fuzzy logic prediction is then applied to the proposed fuzzy logic measurement-based CAC. The performance of the proposed CAC is then examined.

5.4.1 Fuzzy Logic Prediction in the Conventional CAC

Figure 5.1 shows the application of fuzzy logic prediction to the conventional equivalent capacity CAC [GAN91]. It consists of a fuzzy logic traffic predictor, an equivalent capacity estimator, an available free bandwidth estimator and a connection

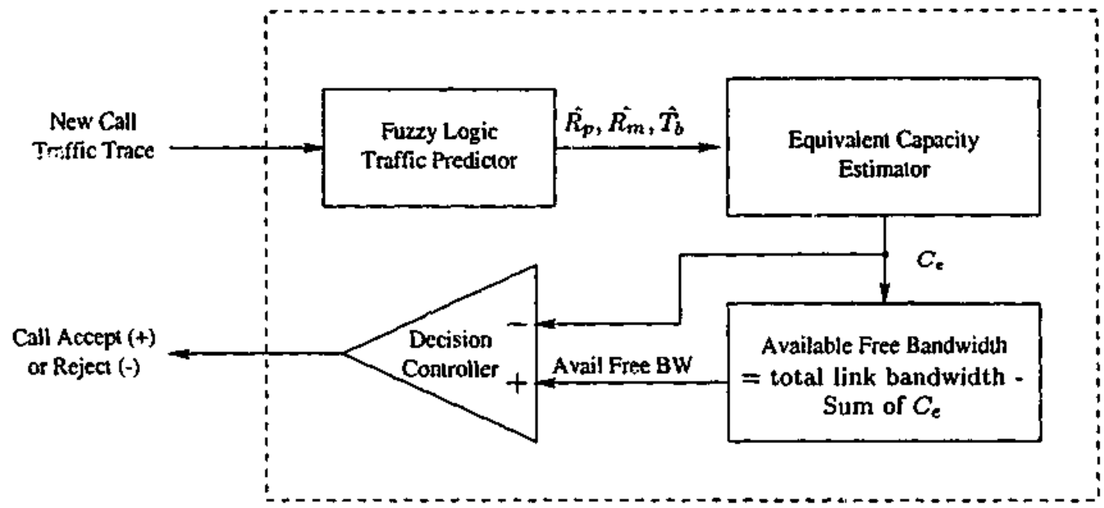


Figure 5.1: Application of Fuzzy Logic Prediction to the Conventional CAC.

admission decision controller.

When a new connection request arrives, the new traffic trace is predicted using the fuzzy logic predictor. To be compatible with the input requirements of the conventional CAC, the predicted peak rate ($\hat{R_p}$), the mean rate ($\hat{R_m}$) and the mean burst length ($\hat{T_b}$) are obtained at the end of a measurement interval. Note that the predicted mean rate together with the predicted peak rate is used to derive the input parameter *utilization* of the conventional CAC. It is the fraction of time the source is active, and can be measured by the ratio of mean rate against peak rate.

With the input parameters, the equivalent capacity estimator then computes the required equivalent capacity C_e for the new connection based on the availability of system resources (e.g., buffer capacity and overflow probability). If this equivalent capacity is higher than the minimum available free bandwidth, the new connection is rejected, otherwise it is accepted.

The available free bandwidth is simply the difference between the total link bandwidth and the sum of each individual equivalent capacity of all existing connections. It does not taken into consideration of the concept of statistical multiplexing.

Simulation studies show that fuzzy logic predicted traffic descriptors improve network efficiency of the conventional CAC compared with user-specified traffic descriptors and non-predictive traffic descriptors (see Section 5.5.3).

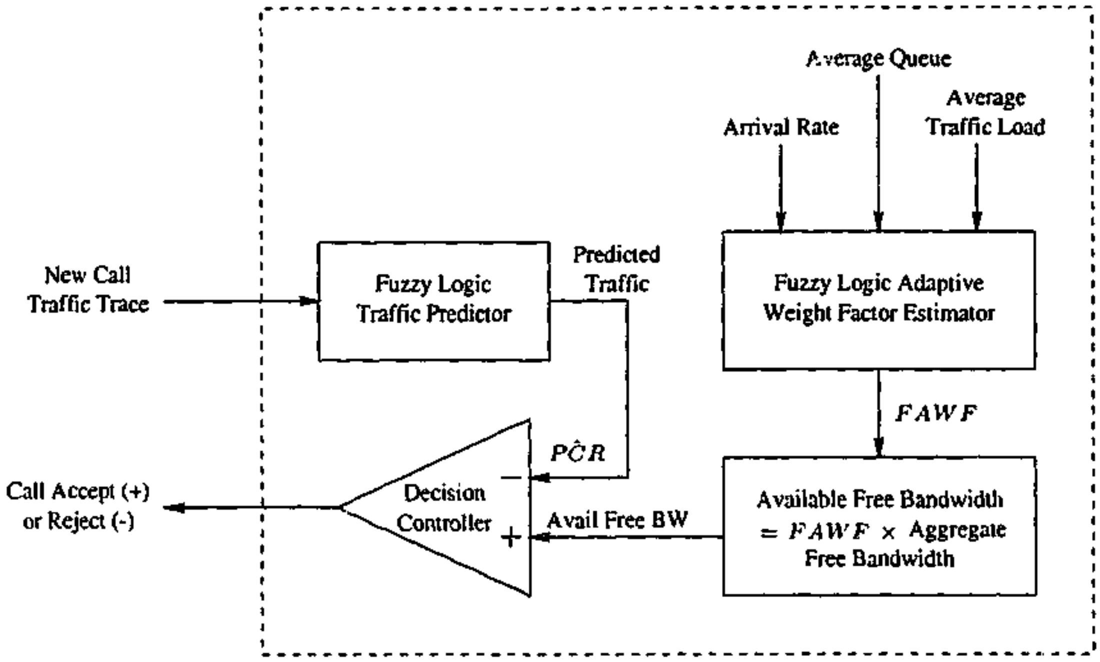


Figure 5.2: The Predictive Fuzzy Logic Measurement-based Connection Admission Controller.

5.4.2 The Proposed Predictive Fuzzy Logic Measurement-based CAC Scheme

With the promised performance of fuzzy logic prediction on the conventional CAC, it is then incorporated in the proposed fuzzy logic CAC. Figure 5.2 shows the proposed predictive fuzzy logic measurement-based connection admission controller. It consists of a fuzzy logic traffic predictor, a fuzzy logic adaptive weight factor estimator and a connection admission decision controller.

When a new connection request arrives, the new traffic trace is predicted using the fuzzy logic predictor. Predicted Peak Cell Rate ($P\hat{C}R$) is obtained at the end of a measurement interval. This predicted PCR is compared with the minimum available free bandwidth on all relevant links which the new connection will traverse. If it is higher than the minimum available free bandwidth, the new connection is rejected, otherwise, the connection is accepted. PCR is considered here as it is simpler to determine compared with the mean bit rate or burst length.

The available free bandwidth is computed based on the rate sharing (RS) approach in which connection admission decisions are based on the sharing of buffer and link capacity. In addition, a "Fuzzy Logic Adaptive Weight Factor" ($FAWF$) is introduced as an additional control. This control factor is adaptive, meaning that the value of the available free bandwidth at any congestion node will be reduced during periods of congestion. On the other hand, when congestion is alleviated, the free bandwidth will be increased and will accept additional new connections. Thus, the available free bandwidth can be defined as:

$$\begin{aligned} \textit{Available Free Bandwidth} &= \textit{Adaptive Weight Factor} \\ &\quad \times \textit{Aggregate Free Bandwidth} \end{aligned} \tag{5.17}$$

The aggregate free bandwidth is simply the difference between the total link capacity and the measured aggregate bandwidth of the link. This is equivalent to the RSFB component in Equation (5.14). Notice that the available free bandwidth in Equation (5.17) does not consist of the PRAFB component as in Equation (5.14). This is because adding PCR individually is a very conservative approach. Consequently, it does not take long for the PRAFB component to become zero, and therefore redundant under a lightly loaded network with a moderate number of admitted connections. Moreover, when the network is not congested, more weight is given to the RSFB. As a result, the OLFB is mainly dependent on the RSFB rather than the PRAFB. Therefore, the proposed CAC does not take the PRAFB component into consideration as it consumes unnecessary computational power in the system.

The Fuzzy Logic Adaptive Weight Factor Estimator

Cells will be lost when the total amount of work which arrives at a node is higher than the total amount of work that can be served by an outgoing link plus the work that can be buffered. Hence the arrival rate of the input traffic, the outgoing traffic load and queue occupancy can be used to approximate the lower bound of the CLR. However on-line CLR measurement may take too long to be practical given the stringent QoS (e.g. CLR in the range of 10^{-6} to 10^{-10}). It is also a very risky approach as it may be too late to take any action once the damage is done [ZL98a]. Therefore an indirect approach is considered in which available free bandwidth is "targeted" by the measured aggregate bandwidth and an adaptive weight factor.

As shown in Figure 5.2, the arrival rate, traffic load and queue length are fed into a fuzzy logic inference system which produces a weight factor. This weight factor is also known as the fuzzy logic adaptive weight factor ($FAWF$) as it adjusts the available free bandwidth adaptively according to the traffic conditions as expressed by Equation (5.17).

The $FAWF$ estimator is governed by three input variables and the linguistic information stored in the rule base. Triangular and trapezoidal shapes are adopted for both input and output membership functions. Figure 5.3 shows the membership functions of the input and output linguistic variables. The output of the estimator is defuzzified using a standard centroid defuzzifier, and the range is set to [0,1].

The design of the fuzzy linguistic rules for the inference of $FAWF$ takes into consideration of the following conditions:

- Under severe congestion: when the arrival rate of the input traffic increases rapidly, the queue length and traffic load at the output link are normally high, the adaptive weight factor should be low or moderate_low, to reduce the available free bandwidth.

- Under a light load condition: when the queue length is low, the input traffic and the outgoing link load are low, the adaptive weight factor should be high, so that the proposed CAC can increase the amount of available free bandwidth in order to accept more connections.

- Under moderate load condition: the weight factor is medium so that the proposed CAC keeps accepting new connections as long as the QoS requirements are satisfied.

Since the fuzzy system has three inputs, it has a total of 27 linguistic rules. Table 5.1 shows the linguistic rules for the $FAWF$ estimator.

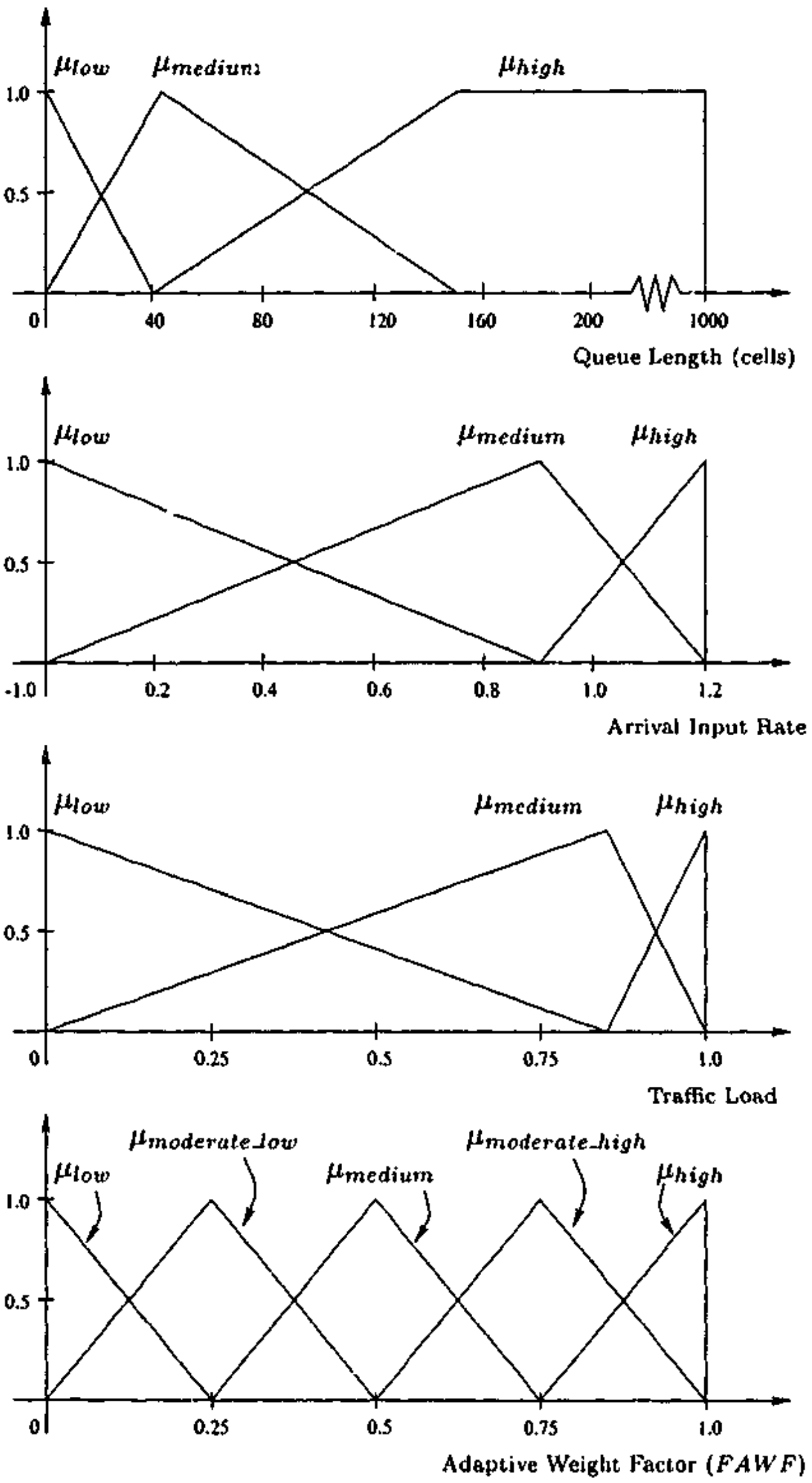


Figure 5.3: Membership functions of the linguistic values for representing the linguistic variables "*current VBR queue length*", "*arrival rate of the input traffic*", "*measured traffic load*" and the adaptive weight factor ($FAWF$). Note: The linguistic values shown here are used in Section 5.5 simulation for type 1 traffic when the buffer size is 1000 cells.

if q is	and load is	and Arrival_Rate is	then $FAWF$ is
low	low	low	high
low	low	medium	moderate_high
low	low	high	medium
low	medium	low or medium	medium
low	medium	high	moderate_low
low	high	low or medium	medium
low	high	high	moderate_low
medium	low	low	moderate_high
medium	low	medium or high	medium
medium	medium		medium
medium	high		moderate_low
high	low		moderate_low
high	medium	low	moderate_low
high	medium	medium or high	low
high	high	low	moderate_low
high	high	medium or high	low

q:current VBR queue; load: traffic load;
Arrival_Rate: Arrival rate of the input traffic

Table 5.1: Linguistic Rules for the Fuzzy Logic Adaptive Weight Factor Estimator.

5.5 Performance Evaluation

In this section, simulation results are presented to illustrate the benefits of fuzzy logic prediction on both the conventional CAC and proposed fuzzy logic CAC. It also shows the advantages of fuzzy logic measurement-based CAC over the conventional model-based analytical CAC.

The simulation model and setting of parameters are first described in the following section. Then, setting of measurement time scale is examined in 5.5.2. In Section 5.5.3, the performance of fuzzy logic prediction is first evaluated by applying the prediction to the conventional CAC. Then in Section 5.5.4, the performance of the proposed fuzzy logic measurement-based CAC is examined. This includes the effects of fuzzy logic prediction on the proposed CAC, and the advantages of fuzzy logic measurement-based CAC compared to the conventional model-based CAC.

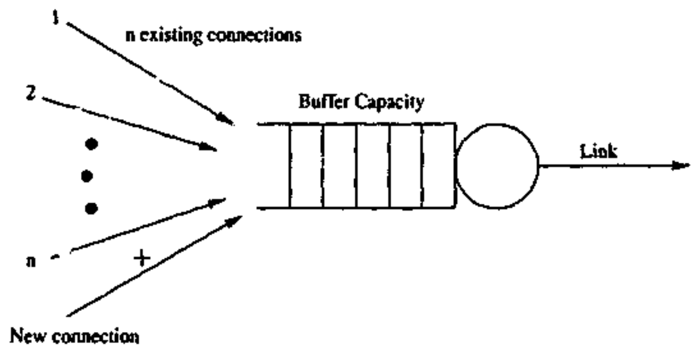


Figure 5.4: Simulation Model for the CAC.

Parameter	T1	T2
Mean bit rate ($Mbps$)	2	5
Active period ($usec$)	1000	5000
Idle period ($usec$)	1000	5000

T3=T1(1) and T2(1)

Table 5.2: Traffic Types Parameter Settings.

5.5.1 Simulation Model and Parameters Setting

Figure 5.4 shows the simulation model used in the study. As shown in the figure, there are n existing connections, feeding into a single server queue. A FIFO queuing discipline is assumed for the output buffer. The specified QoS requirements (e.g. CLR) of these n connections are currently maintained by the system. Meanwhile, switches constantly monitor the traffic statistics of calls that have been already accepted. When a new connection request arrives, the fuzzy logic predictor predicts the PCR of the new connection. To ensure that the same QoS for the $n+1$ (including the new arrival) connections are met, this new connection will be admitted if the predicted PCR is lower than the minimum available bandwidth of all relevant links.

Three types of VBR traffic trace (T1–T3) are used. All VBR sources are modeled as a Poisson distributed on-off source which alternate between active and idle periods, and transmit cells with an independent exponentially distributed mean bit rate during active periods. Table 5.2 shows the parameters of traffic types used in the simulation. Note that type 3 (T3) traffic is a combination of type 1 (T1) and type 2 (T2) traffic. For remaining parameters, CLR is fixed at 10^{-6}, the link rate is set at 155.52 $Mbps$, and the buffer size is set at 100 cells or 1000 cells.

5.5.2 The Measurement Window

When a new connection request arrives, its traffic characteristics are measured within a monitoring interval. This interval has to be short enough to be practical. Fuzzy logic CAC with predictive traffic descriptor is used as an example to illustrate the effects of the monitoring interval on the variation of utilization, cell loss ratio and number of connections.

Figure 5.5 shows the relationship between the monitoring period verses utilization. Different line types represent simulations with different buffer sizes. The dashed lines represent a buffer size of 1000 cells, and the solid lines represent a buffer size of 100 cells. As shown in the figure, utilization is significantly reduced by increasing the length of the monitoring period. In addition, if the monitoring period is too small, there is not enough traffic statistic to establish the peak cell rate information. This leads to an increase in the number of connections admitted and therefore cell loss has occurred. This is particularly true for type 2 (T2) traffic which has higher peak rate and long burst length. When the monitoring period is 10 *msec*, it has a CLR of 1.0698e-3 and 1.1765e-3 for a buffer size of 100 and 1000 cells respectively. However, if the monitoring period is beyond 20 *msec*, cell loss ratios are zero for all sources with the expense of lower utilization. Therefore in order to meet 10^{-6} cell loss requirements, 20 *msec* monitoring period is used in all simulations.

The statistical information of the aggregate traffic, traffic load and queue length are collected for the proposed fuzzy logic CAC. To ensure stability and reduce variance, an average value of this information is used instead of instantaneous information. Sampling of data is done in the background and is a long term measurement process. An average is obtained by sampling the queue length (for example) over a measurement interval and taking the average of samples. The average value is updated at each sampling period, meaning, the oldest sample is discarded and new average is obtained from the current and latest past samples. Empirically, it is found that the proposed fuzzy logic CAC is quite robust to different choices of samples and measurement intervals. Nevertheless, it is noted that setting of the

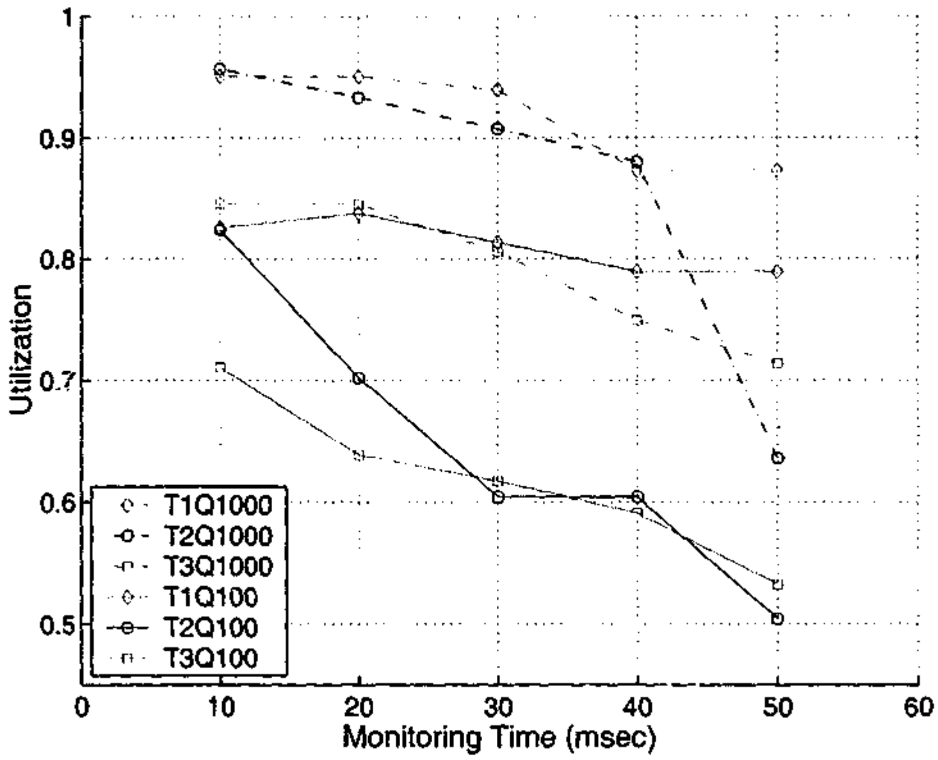


Figure 5.5: Monitoring Period vs Utilization.

measurement time scale is an important issue for measurement-based CAC, and further discussions can be found in [GT99, JDSZ97].

5.5.3 The Effects of Fuzzy Logic Prediction on the Conventional CAC

To demonstrate the benefits of fuzzy logic prediction, it is applied to the conventional equivalent capacity CAC [GAN91]. Therefore, for comparison purpose, three methods are used to obtain the traffic descriptors of a new connection, namely, *a priori*, measured non-predictive and measured fuzzy logic predictive. Traffic descriptors in this case are composed of peak rate, mean rate and burst length.

Below is a list of the different simulation scenarios in which the efficiency of the network and the benefits of fuzzy logic prediction are undertaken for investigation:

Conventional Equivalent Capacity (or model-based CAC) with:

1. *a priori* traffic descriptors.

2. measured non-predictive traffic descriptors.

Buffer Size (cells)	**Traffic Types**	**Pred vs *a priori***	**Pred vs Non-pred**
100	T1	0.1315	0.1021
	T2	0.2047	0.1033
	T3	0.1269	0.1128
Column Average		0.1544	0.1061
1000	T1	0.1847	0.1571
	T2	0.3995	0.2630
	T3	0.2281	0.1514
Column Average		0.2708	0.1905

Table 5.3: Efficiency improvement of the fuzzy logic predictive traffic descriptors over the *a priori* and the non-predictive traffic descriptors on the conventional CAC.

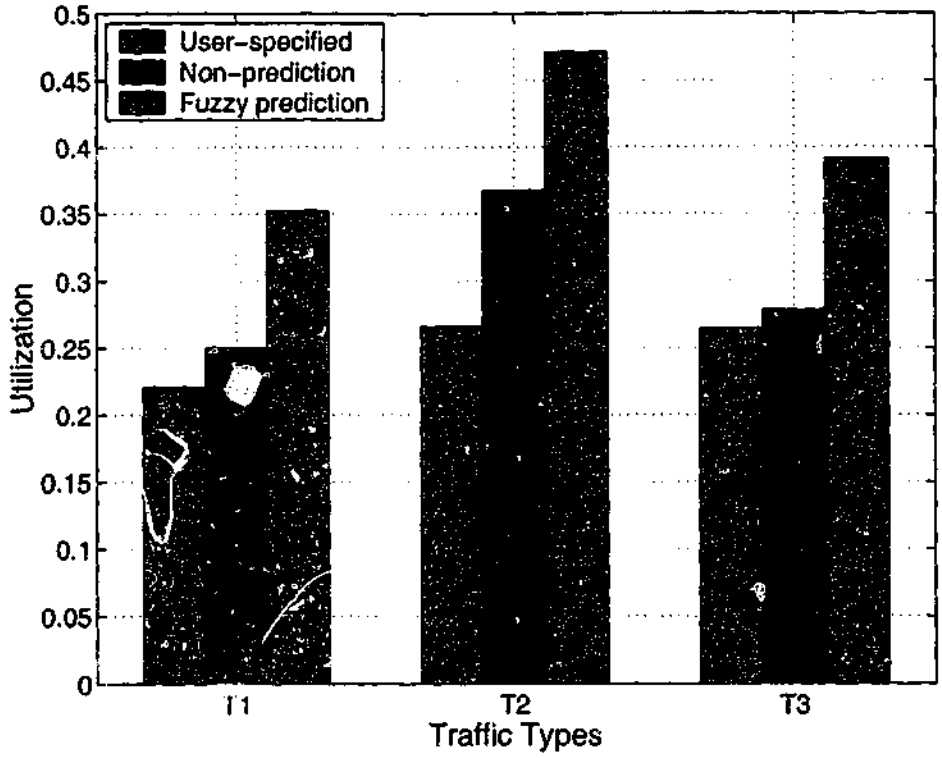
3. measured predictive traffic descriptors.

Figure 5.6 illustrates the efficiency achieved for the conventional CAC with various traffic descriptors and buffer capacity under different traffic situations. There are no cell loss in all cases. The figure clearly indicates that CAC scheme using predictive traffic descriptors achieves highest link utilization compared to the *a priori* and measured non-predictive traffic descriptors.

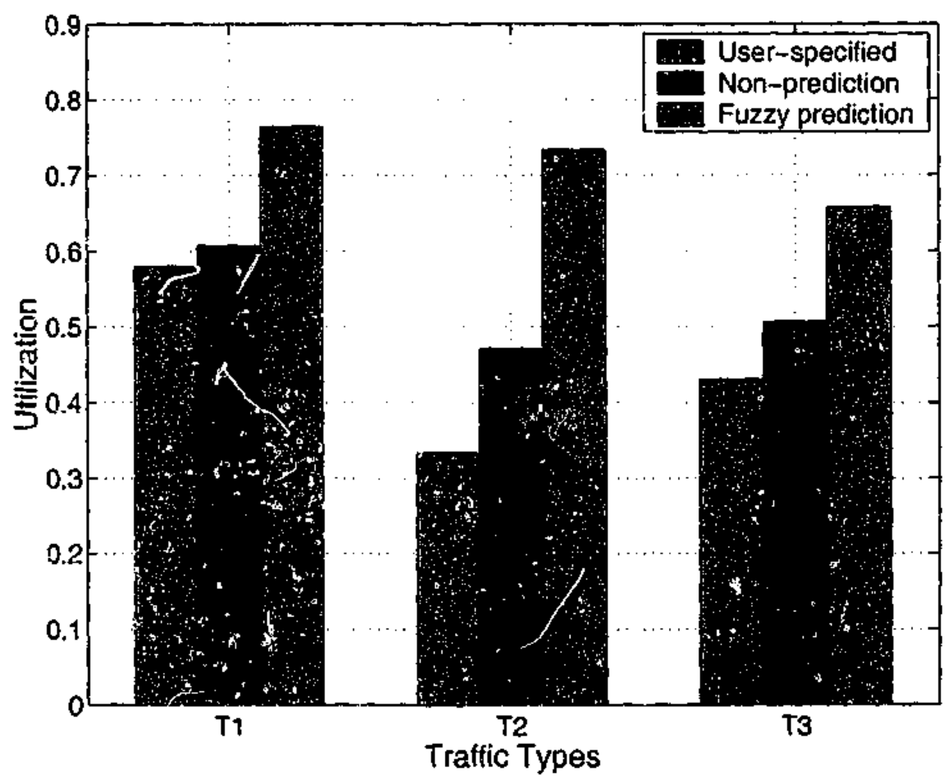
Table 5.3 tabulates the amount of efficiency improved for the fuzzy logic predictive traffic descriptors over the *a priori* and the non-predictive traffic descriptors on the conventional CAC. As shown in the table, when the buffer size is 100 cells, the predictive traffic descriptors improve efficiency by about 15% and 11% compared to the *a priori* and measured non-predictive traffic descriptors respectively; when the buffer size is 1000 cells, the improvement is approximately 27% and 19% respectively. Hence, it is demonstrated that fuzzy logic prediction significantly improves network efficiency while maintaining QoS.

5.5.4 Performance Evaluation of the Proposed Fuzzy Logic Measurement-based CAC

This section presents the simulation results to examine the performance of the proposed fuzzy logic CAC, the effect of fuzzy logic prediction and the advantages



(a) Buffer size 100 cells.



(b) Buffer size 1000 cells.

Figure 5.6: The efficiency achieved for the conventional CAC with various traffic descriptors and buffer capacity under different traffic traces.

of fuzzy logic measurement-based CAC over the conventional CAC.

The effect of fuzzy logic prediction on the proposed CAC is first evaluated in the following section by comparing the performance of the predictive traffic descriptor to the *a priori* and measured non-predictive traffic descriptor. Then, in Section 5.5.4.2, the performance of the proposed fuzzy logic measurement-based CAC is examined and compared with the conventional CAC.

5.5.4.1 The Effects of Fuzzy Logic Prediction

With the promising performance of fuzzy logic prediction on the conventional CAC (shown in Section 5.5.3), it is incorporated in the proposed fuzzy logic measurement-based CAC. Likewise, for comparison purpose, three methods are used to obtain the traffic descriptor of a new connection, namely, *a priori*, measured non-predictive and measured fuzzy logic predictive. However, in this case, the peak cell rate (PCR) is the only traffic descriptor required. PCR is considered rather than the mean bit rate or burst length as it is simpler to determine.

Below is a list of the different cases in which the efficiency of the network and the benefits of fuzzy logic prediction and control are undertaken for investigation:

Proposed Fuzzy Logic CAC (or real-time measurement-based CAC) with :

1. *a priori* traffic descriptor.
2. measured non-predictive traffic descriptor.
3. measured predictive traffic descriptor.

Figure 5.7 illustrates the efficiency achieved for the proposed fuzzy logic CAC with various traffic descriptors and buffer capacity under different traffic situations. There is no cell loss in all cases. The figure shows that fuzzy logic predicted traffic descriptor improves network efficiency compared to the *a priori* and measured non-predictive traffic descriptors.

Buffer Size (cells)	**Traffic Types**	**Pred vs *a priori***	**Pred vs Non-pred**
100	T1	0.1361	0.0484
	T2	0.0660	0.0330
	T3	0.0822	0.0267
Column Average		0.0948	0.0360
1000	T1	0.1369	0.0323
	T2	0.1086	0.0531
	T3	0.0411	0.0296
Column Average		0.0955	0.0383

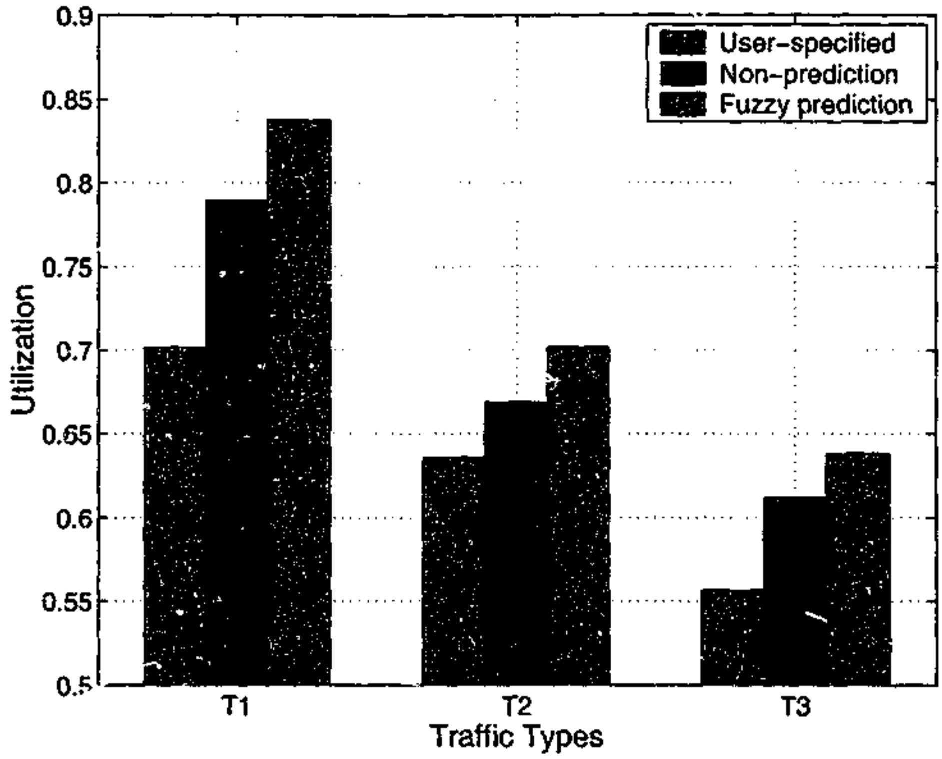
Table 5.4: Efficiency improvement of the fuzzy logic predictive traffic descriptor over the *a priori* and the non-predictive traffic descriptor on the fuzzy logic measurement-based CAC.

Table 5.4 tabulates the efficiency improvement for the fuzzy logic predictive traffic descriptors over the *a priori* and the non-predictive traffic descriptors on the proposed fuzzy logic measurement-based CAC. The improvement is less impressive but still distinctive for the fuzzy logic CAC as compared to the conventional CAC. The overall improvement in efficiency when using the fuzzy logic predictive traffic descriptor is about 10% and 4% compared to the *a priori* and measured non-predictive traffic descriptors respectively. This is mainly because statistical multiplexing gain has fully exploited in the fuzzy logic CAC.

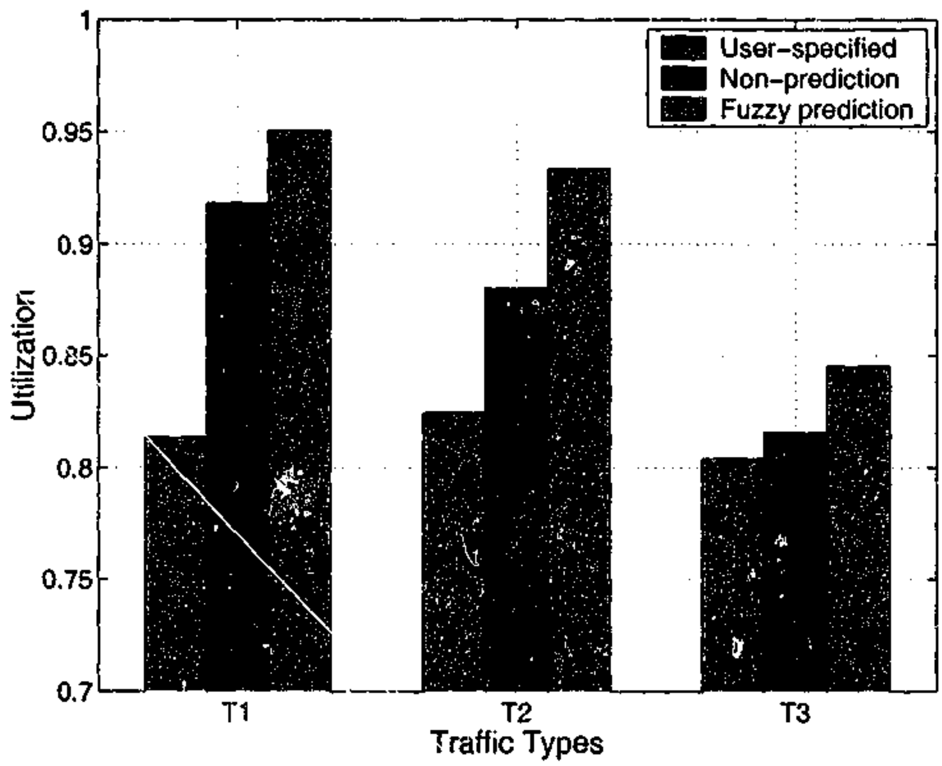
In summary, the fuzzy logic predictor provides a soft and accurate estimation of traffic parameter(s) for new connections. As a result, the performance of the CAC is significantly improved.

5.5.4.2 Comparison of the Conventional CAC and the Fuzzy Logic Measurement-based CAC

Comparing the performance of both the conventional CAC and the proposed fuzzy logic measurement-based CAC, Figure 5.7 shows that the proposed fuzzy logic CAC produces consistently higher efficiency under all traffic conditions compared with the conventional CAC (Figure 5.6). In Figure 5.6, efficiency achieved for the conventional CAC is less than 0.5 and 0.8 for a buffer size of 100 cells and 1000 cells respectively. However, in Figure 5.7, efficiency achieved for the proposed



(a) Buffer size 100 cells.



(b) Buffer size 1000 cells.

Figure 5.7: The efficiency achieved for the fuzzy logic measurement-based CAC with various traffic descriptors and buffer capacity under different traffic traces.

Traffic Types	Queue Size 100 cells	Queue Size 1000 cells
T1	0.4853	0.1860
T2	0.2312	0.1995
T3	0.2473	0.1871

Table 5.5: Efficiency improvement of the fuzzy logic measurement-based CAC over the conventional CAC with predictive traffic descriptor(s).

fuzzy logic CAC is more than 0.5 and 0.8 for a buffer size of 100 cells and 1000 cells respectively. In particular, the proposed fuzzy logic CAC with predictive traffic descriptor achieves highest efficiency among all cases. For example in Figure 5.7(b), it can be observed that an efficiency of over 90% is achieved for types 1 (T1) and 2 traffic (T2) and 80% for type 3 traffic (T3). It also indicates that regardless of CAC schemes, as the size of the buffer increases, higher efficiency is achieved.

Table 5.5 compares the efficiency improvement between the fuzzy logic CAC and conventional CAC using predictive traffic descriptor(s). On type 3 (T3) traffic, for example, efficiency is improved by 25% and 19% when the buffer size is 100 and 1000 cells respectively.

The improvement is due to the fact that network performance parameters are used as input linguistic variables. As a result, the fuzzy logic CAC uses more information for making connection admission decisions than the conventional CAC. In addition, because it does not have the additive property, the proposed fuzzy logic CAC better exploits the advantages of statistical multiplexing.

Table 5.5 also indicates that when the buffer size is small, the efficiency improvement between the fuzzy logic CAC and the conventional CAC yields a much greater percentage as compared to the larger buffer size. This is because the effective bandwidth of the conventional CAC tends to approach the PCR when the buffer size is small, and closes to the SCR when the buffer size is large. Hence, the link is very much under-utilized when the buffer size is small.

5.6 Conclusions

This chapter has addressed the issues related to CAC design and investigated various existing CAC approaches. Conventional admission controls are based on analytical models which require *a priori* traffic descriptors as input parameters. However it is often hard or even impossible to model every network situation and each traffic flow, especially with the rapid emergence of new applications. In addition, due to unavailability or inaccuracies and uncertainties in the knowledge about traffic characteristics, e.g. mean bit rate or peak duration, during a connection setup phase, an analytical model may not completely capture the significant statistical information of traffic flows. On the other hand, measurement-based CAC using a statistical allocation approach assumes fewer parameters and is able to achieve a higher link utilization. Nonetheless, computational intelligent techniques such as fuzzy logic controls, are quite attractive because of the difficulties in declaring traffic descriptors, highly fluctuated and uncertain traffic, errors and uncertainties involved in measurement-based CAC.

Based on the concepts of free bandwidth and the adaptive weight factor in [ZL98b, ZL98a], this chapter has proposed a novel measurement-based CAC using a fuzzy logic approach. Unlike conventional CAC, the proposed CAC does not use complicated analytical models or *a priori* traffic descriptors. Instead, traffic parameters are predicted by an on-line fuzzy logic predictor [Qiu01, QSW99]. QoS requirements are targeted indirectly by an adaptive weight factor. This weight factor is generated by a fuzzy logic inference system which is based on arrival traffic, queue occupancy and traffic load of the outgoing link. Admission decisions are then based on real-time measurement of aggregate traffic statistics with the fuzzy logic adaptive weight factor as well as the predicted traffic parameters.

The performance of the proposed CAC scheme has been investigated by simulations. Both homogeneous and heterogeneous traffic were used in the simulations. The effects of the measurement interval were first investigated by comparing the link utilization and cell loss ratio against the measurement periods. Simulation results reveal that longer measurement periods result in high computational power, massive storage space and low link utilization. On the other hand, due to lack of statistical information on traffic flows, small measurement periods result in more connections being admitted and thus high cell loss. Therefore the measurement interval not only has to be short enough to be practical, but also has to be able to capture the significant statistical information of traffic flows. The advantages of fuzzy logic prediction on CAC were then evaluated by applying the prediction technique to the Guérin's equivalent bandwidth CAC [GAN91] and the proposed measurement-based CAC. Simulation results show that fuzzy logic prediction significantly improves link efficiency while maintaining QoS requirements on both CAC schemes. In addition, the measurement-based approach incorporating fuzzy logic inference and using fuzzy logic prediction is shown to achieve higher network utilization while maintaining QoS objectives.

The next chapter focuses on the congestion control associated with transporting Internet application data over ATM networks.

Chapter 6

Fuzzy Logic Congestion Control for TCP/IP over ATM

The previous chapters have dealt with issues and solutions on reactive and preventive congestion control in ATM networks. This chapter focuses on the congestion control for transporting TCP/IP application data over ATM networks. It proposes an intelligent selective cell drop scheme for TCP over ATM UBR based on fuzzy logic prediction and control. The performance of the proposed scheme is compared with the conventional cell drop schemes by extensive simulation. The effects of fuzzy logic prediction and control on TCP over ATM ABR are also investigated. The fuzzy logic ABR service congestion control scheme proposed in Chapter 4 is used for this study. The performance is then evaluated in comparison with the conventional congestion control scheme.

6.1 Introduction

Over the past decade, the Internet has been growing exponentially. It is built based on IP connectionless protocol which does not provide QoS guarantees. On the other hand, ATM has been designed to support different types of service (Chapter 2) and offers guaranteed QoS to connections. Recently, there has been an increasing demand in multimedia applications which require guaranteed QoS. Since the vast majority of computers are based on IP protocol, to provide guaranteed QoS, ATM is adopted by large Internet service providers (ISPs) as a high-speed backbone. TCP/IP data applications are thus transported on ATM networks by best effort ATM ABR and UBR services.

An application packet has to be fragmented by the source into cells in order to transport over ATM networks. However, the problems of fragmentation and reassembly of the application packet and cell loss in ATM networks can severely affect application performance and network utilization, especially during periods of network congestion. Hence, there have been many recent investigations into how TCP/IP can operate efficiently over the ATM infrastructure [HAOS01, LB99, SHL98, GJK+97, JY96, RF95]. An important issue is to prevent congestion collapse, minimize congestion levels and guarantee fairness.

End system policies and network policies are available to manage TCP/IP traffic over ATM. At the end system, the transport layer of TCP can implement closed-loop, window-based congestion avoidance mechanisms to improve its throughput and prevent congestion collapse. At the network, switches can implement various mechanisms to optimize resource utilization, fairness and higher layer throughput. For example, ATM ABR service can implement effective closed-loop congestion feedback mechanisms to reduce cell loss, and UBR service can implement intelligent drop schemes to minimize incomplete packet transmission rate.

The end system polices interact with the network policies, and both of these policies are affected by the large round-trip delays of the network. Effective throughput decreases as round-trip delays get longer since the time required to recover the original data transmission rate after a slow start is prolonged. The fairness among connections is also at risk if the differences between round-trip delays among connections become large.

To tackle the problem of long round-trip delays, this chapter proposes a cell drop scheme for TCP over UBR based on fuzzy logic prediction and control. The scheme is able to accept or drop a new incoming packet dynamically based on the predicted future buffer condition in the switch, and it can thereby react to congestion earlier.

In addition, fairness is maintained by a fuzzy logic inference system using soft-computing. Consequently cell loss is kept to a minimum and high end-to-end performance for the TCP protocol is delivered.

This chapter also presents an investigation on the effect of fuzzy logic prediction and control on TCP over ATM ABR congestion control. The fuzzy logic ABR service congestion control scheme proposed in Chapter 4 is applied to the ABR switch algorithms, and its performance is evaluated in terms of effective throughput, fairness, cell loss and cell delay variation.

This chapter first describes the architecture and transportation of TCP/IP over ATM networks. The impact of ATM QoS requirements on TCP performance is discussed in Section 6.3. End system policies and network policies are used to manage TCP/IP traffic over ATM. Section 6.4.1 describes some end system polices such as the TCP/IP congestion control mechanism and its enhancements. Intelligent drop schemes for UBR service and rate-based closed-loop feedback control for ABR service are the network polices. Section 6.4.2 presents a comprehensive survey of intelligent drop schemes for the UBR service. Some of these schemes are used later for comparison purposes in performance evaluation of the proposed cell drop scheme. Then, in Section 6.4.3, the performance issues of running TCP over ABR rate-based closed-loop feedback control are addressed.

To exploit the advantages of fuzzy logic prediction and control on cell drop, a fuzzy logic selective cell drop scheme (FSCD) for TCP over UBR is proposed in Section 6.5. A simulation model and the setting of parameters are then described. TCP performance over UBR is first studied by comparing the proposed scheme with the well known conventional schemes described in Section 6.4.2. Different TCP congestion control options are used in the simulation and the results are presented in Section 6.7.2. Subsequently, the effect of fuzzy logic prediction and control on TCP performance over ABR is investigated in Section 6.7.3. The fuzzy logic ABR service congestion control scheme described in Chapter 4 is applied here for ABR traffic control. TCP performance is then compared with the conventional ABR switch algorithms by simulation. Finally a comparison of performance and

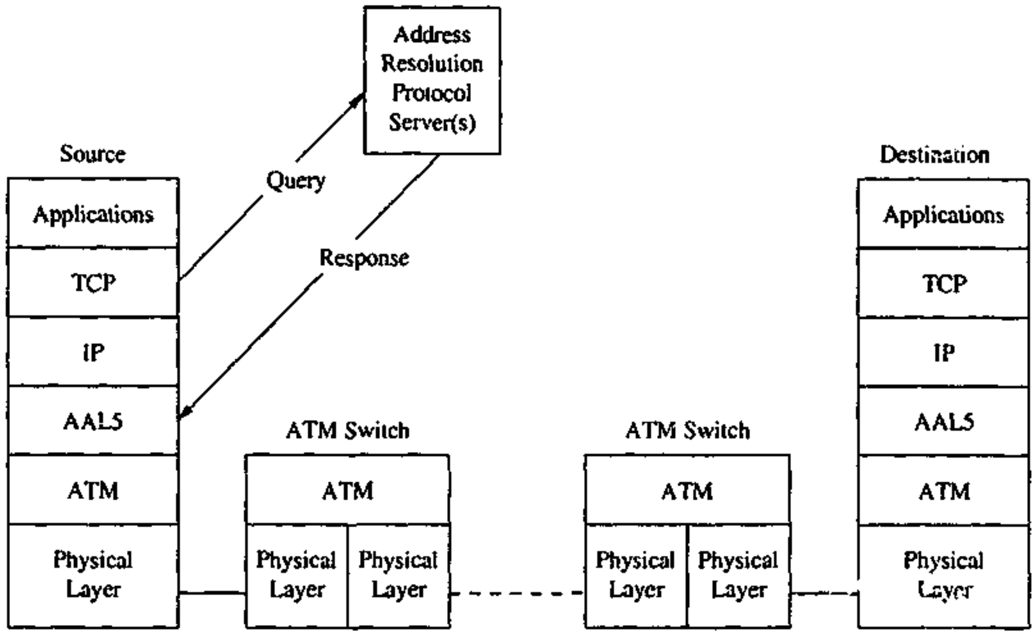


Figure 6.1: The Architecture of TCP/IP over ATM.

cost between various TCP over ABR and UBR control mechanisms is presented in Section 6.7.4.

6.2 Transportation of TCP/IP over ATM

The interoperability and transportation of TCP/IP over ATM are standardized by the Internet Engineering Task Force (IETF) [LH98, Lau94]. The general architecture of TCP/IP over ATM protocols is illustrated in Figure 6.1 (adapted from [Goy99]). Basically, IP treats ATM just like another link layer for IP packet transports, and enjoys all ATM functions and features including the guaranteed QoS. The application and TCP/IP are on top of the AAL and ATM layer on each source and destination. The source and destination end systems can be hosts or routers. They are configured with both IP and ATM addresses. A source IP may establish a switched VC (SVC) with the destination by querying the ATM address resolution protocol (ARP) server for an IP/ATM address binding. However, if permanent VCs (PVCs) are used, an ARP server is not required.

To transport IP application packets over ATM networks, the packets must be segmented and encapsulated into ATM cells. Figure 6.2 shows the structures of the protocol data units (PDUs) at each layer [Goy99]. The application packet

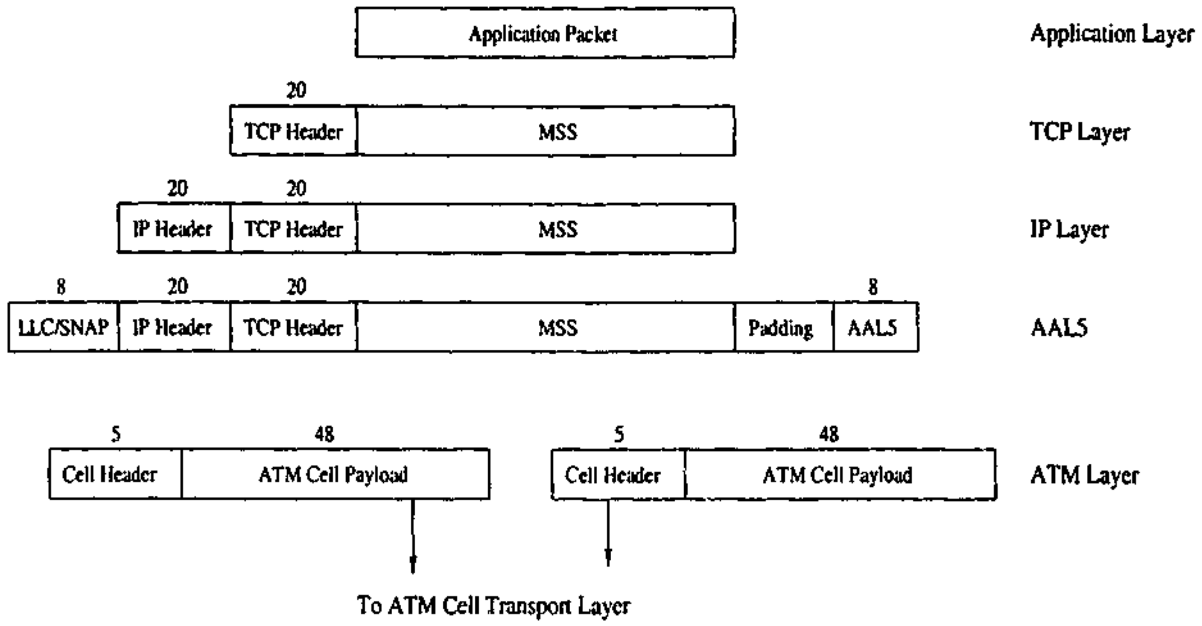


Figure 6.2: The Structures of Protocol Data Units at Each Layer.

is first segmented into smaller TCP segments. The TCP maximum segment size (MSS) is determined by the maximum transmission unit (MTU) of the underlying link layer. The segment is then attached by a 20 byte TCP header and IP header when it passes the TCP layer and IP layer respectively. Next, the IP packet passes to the AAL layer. The AAL consists of two internal sublayers: a convergence sublayer (CS) and a segmentation and reassembly sublayer (SAR). The IP packet is encapsulated by the CS layer specified by the RFC1483 [Hei93]. The layer attaches an 8 byte logical link control (LLC)/subnetwork attachment point (SNAP) header which contains the protocol ID for the higher layer. This header enables IP packets to be distinguished from packets of other network layer protocols. The packet is then further padded and appended an 8 byte trailer to form an AAL5 frame. The SAR layer then segments the AAL5 frame into 48-octet segments, filling the payloads of successive ATM cells. The padding ensures that the AAL5 trailer falls in the last eight octets of the last ATM cell. Each ATM cells has a 5 byte header which has an end-of-message (EOM) bit. This bit is an indication of AAL5 frame boundaries. Hence this bit is set to one on the last cell in the AAL5 frame, every other cell is set to zero.

6.3 The Impact of ATM QoS Requirements on TCP Performance

ATM QoS requirements are typically expressed in terms of loss and delay within a negotiated rate across the network. Most existing TCP/IP data applications are sensitive to losses but tolerant to delay. During congestion, a single cell loss at the ATM layer results in an entire packet loss at the higher layer on the destination TCP. Hence, cell loss in an ATM network can severely affect the TCP performance and the efficiency of the network. The remainder of this section discusses the impact of cell loss and its effect on TCP performance [HA00].

TCP performance can be affected by fragmentation. This is because a packet has to be fragmented into cells in order to be transported over ATM networks. If a cell loss occurs in the ATM layer, all subsequence cells from the packet are useless to the higher layer of the destination. However they are still transmitted across the network, clogging the congested link. As a result, bandwidth is wasted and TCP throughput is reduced.

The lack of error correction or recovery at the ATM and AAL5 layers can degrade TCP performance. Although ATM protocol has error checking on the cell header to prevent misrouting of cells, it does not perform any error checking on cell payload or the retransmission of lost cells. In addition, AAL5 simply discards the entire packet if a CRC error is detected. As a result, the destination TCP does not send an acknowledgment to the source since it does not receive the segment which has already been discarded at the AAL5 layer. The source TCP eventually timesout and retransmits the packet. This results in the unnecessary retransmission of many cells which have already arrived at the destination previously, causing further congestion and cell loss in the network.

TCP performance can be affected by the way the AAL5 reassembles cells at the destination. AAL5 frame boundary is indicated by marking the EOM bit in the header of the last cell of the AAL5 frame. If the last cell is lost, the AAL5 SAR layer at the destination continues to reassemble cells into subsequence AAL5 frame(s).

This process only stops when the last cell of an AAL5 frame is detected or a maximum threshold is reached. Consequently, a loss of one last cell in an AAL5 frame may cause a loss of more than one AAL5 frame, resulting in reduced TCP throughput.

The TCP congestion control protocol and related parameter settings can affect TCP performance. A source TCP detects congestion by using a retransmission timeout (RTO) timer for the last unacknowledged segment. Before the timer expires, the source TCP does not send any new segments if its allowed window is reached. Significant time is wasted waiting for the retransmission timer to expire. Hence the link is idle for a long period of time causing low TCP throughput. When the RTO timer expires, the source TCP enters slow start phase and retransmits the lost segment. It then waits for the acknowledgment of the retransmitted segment from the destination TCP. For long distance connections, the delay in receiving acknowledgment can be quite high, resulting in low TCP throughput. The situation becomes worse if an acknowledgment packet is lost in the congested link. In addition, if cell loss frequently occurs, the connection is frequently forced back to slow start phase. Each time, the link has to take several round-trip times to effectively utilize the bandwidth.

The size of a buffer at ATM switches also has a significant effect on cell loss rate for TCP/IP over ATM. The smaller the buffer, the higher the cell loss rate, resulting in significant loss of TCP throughput. In [KS94], Keung and Siu show that a small percentage of cell loss (less than 1%) can cause a throughput degradation of over 64% on a two-node network with five TCP connections.

Several mechanisms have been proposed to address the issue of cell loss on TCP over ATM and its effect on TCP performance. The following section discusses these mechanisms.

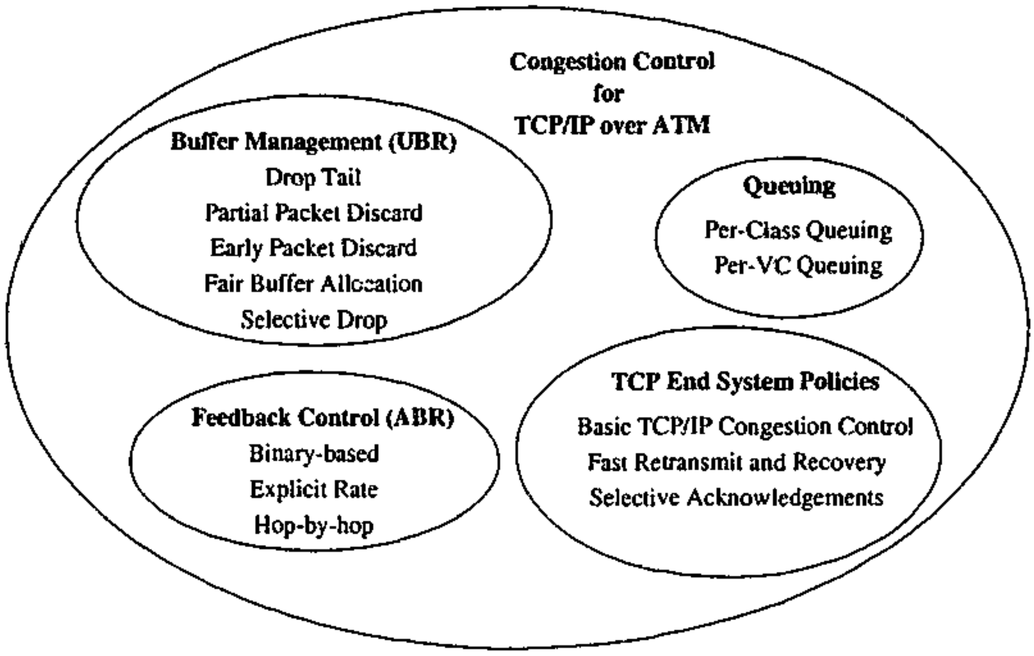


Figure 6.3: Congestion Control for TCP/IP over ATM.

6.4 Congestion Control for TCP/IP over ATM

This chapter focuses on the issues and solutions related to congestion control for TCP/IP over ATM. Figure 6.3 illustrates four categories of design option available to networks and TCP end systems for congestion control [Goy99]. These four categories of design option are:

- Buffer management (for UBR service)
- Queuing
- Feedback control (for ABR service)
- TCP end system policies

This chapter proposes solutions in relation to the buffer management for UBR service and feedback control for ABR service later in the chapter. The performance of the proposed solutions is evaluated using different TCP end system policies.

UBR service supports TCP/IP-based data communications by implementing FIFO queuing and a simple dropping policy. This simple dropping policy randomly drops cells from any frames. Since dropping cells without any regard to the AAL5 frame delineation can degrade TCP performance, more intelligent buffer management policies using a packet-based (or frame-based) discard strategy have been developed to improve TCP throughput and fairness on UBR service. These mechanisms aim to reduce cell losses in packets by minimizing incomplete packet transmission rate. Some of these mechanisms also adopt per-VC accounting to further improve fairness among connections. Section 6.4.2 discusses various buffer management policies on TCP over UBR.

Another TCP/IP over ATM implementation option is to use the ABR service category. ABR class connections are closed-loop and rate-based. Upon congestion, sources are informed by the feedback from the network to decrease their transmission rate. Consequently, cell loss in the network is reduced. Earlier generation switches use a binary feedback scheme to indicate congestion. Later generation switches require sophisticated mechanisms and use the ER field of the RM cells to convey the allowed rate to each source. Switches can also implement virtual source/virtual destination (VS/VD) by segmenting the end-to-end control loop into smaller hop-by-hop control loops. This approach can be quite expensive to implement as it requires the conversion of per-class queue to per-VC queue and the inclusion of all source/destination related functions at each switch. Section 6.4.3 discusses the effect of ABR rate control on TCP performance.

Besides network based solutions, effective TCP/IP end-to-end flow and congestion control can be implemented at the end systems to improve TCP performance over ATM. Mechanisms such as the fast retransmit and recovery (TCP Reno) [Jac88, Jac90, Ste97], together with the basic TCP slow start and congestion avoidance can be used to quickly recover from isolated segment losses. As for fast recovery from multiple segment losses, the selective acknowledgments (SACKS) is proposed in [MMFR96]. The following section describes these mechanisms.

6.4.1 TCP Flow and Congestion Control

TCP is currently the most widely used transport protocol in the Internet. It was originally specified in RFC793 [Pos81] but has since been enhanced in several ways. RFC1122 [Bra89] recommends that a TCP must implement slow start and congestion avoidance to ease congestion. The fast retransmit and recovery mechanism (TCP Reno) was implemented after RFC1122 to enable fast recovery from isolated segment losses. These four algorithms were developed by V. Jacobson [Jac88, Jac90, Ste97]. In 1996, RFC2018 [MMFR96] was proposed with the option of selectively acknowledged (SACK) out-of-order segments in order to quickly recover from multiple segment losses. The following sections provide a brief overview of the basic TCP flow and congestion control mechanism and some of these enhancements. Since basic TCP flow and congestion control mechanism, and TCP Reno are more commonly used compared to SACK, they will be used in the simulation studies on the proposed schemes. On going research, standardization and deployment of TCP congestion control mechanism can be found in [Flo01, ADG+00].

6.4.1.1 Basic TCP Flow and Congestion Control Principle

TCP is a reliable end-to-end window-based flow and congestion control protocol. It uses acknowledgments and retransmission for segment transmission and error recovery. The flow control is enforced by two windows: a congestion window (cwnd) and a receiver's advertised window (rcvwnd). cwnd is maintained by the sender and is a sender's perception about the network current capacity. rcvwnd is kept at the receiver, and is an indication of receiver's current buffering capacity. The current value of rcvwnd is carried to the sender every time the receiver sends an ACK. At any given instant, the sender never sends more than the minimum of cwnd and rcvwnd segments into the network.

The basic TCP congestion control scheme consists of the "slow start" and "congestion avoidance" phases [Jac88]. These two phases are identified by a variable ssthresh which is maintained at the sender. The slow start mechanism effectively increases the window size to probe the available bandwidth. It starts by the sender sending one segment of data into the network, that is, cwnd is initially set to one. When the segment is acknowledged by the receiver, the cwnd is increased to two segments. When these two segments are acknowledged, the cwnd is increased to four segments, and so on. The window is increased exponentially. Eventually either cwnd reaches ssthresh which is normally initialized to rcvwnd, or congestion occurs thereby segments are dropped at switches due to buffer overflow.

There are two ways to detect congestion. The sender detects congestion by maintaining a retransmission timeout (RTO) timer for the last unacknowledged segment. This timer is calculated based on the smoothed average and mean deviation of measured samples of round-trip times (RTT) [Jac88]. It is doubled each time (*exponential back off*) a segment is retransmitted due to the expiration of the time-out timer and is reset each time an ACK for a new segment is received. On the other hand, if the receiver detects a missing in-order segment from the arriving segments, it sends duplicate ACKs for each out-of-order segment received.

Therefore when congestion occurs, the sender retransmits the unacknowledged segment and sets the ssthresh to half of cwnd. More precisely, ssthresh is set to the minimum of cwnd and rcvwnd, but at least two segments. In addition, cwnd is set to one segment size. Since the cwnd is less than ssthresh, the sender is forced back to the slow start phase. Once ssthresh is reached, the sender enters the less aggressive congestion avoidance phase. During the congestion avoidance phase, the cwnd is increased additively by 1/cwnd each time a segment is acknowledged.

Figure 6.4 shows an visual description of slow start and congestion avoidance [Ste94]. The units of ssthresh and cwnd are assumed in segments instead of bytes. cwnd is initialized to 32 segments. When congestion occurs, ssthresh is set to half of cwnd which is 16 segments and cwnd is set to one segment. The sender enters slow start phase and retransmits the lost segment. cwnd is increased exponentially each time a new segment is acknowledged. This continues until cwnd reaches ssthresh where the congestion avoidance phase starts. At this phase, cwnd is increased linearly with a maximum of one segment per round-trip time regardless of number of ACKs

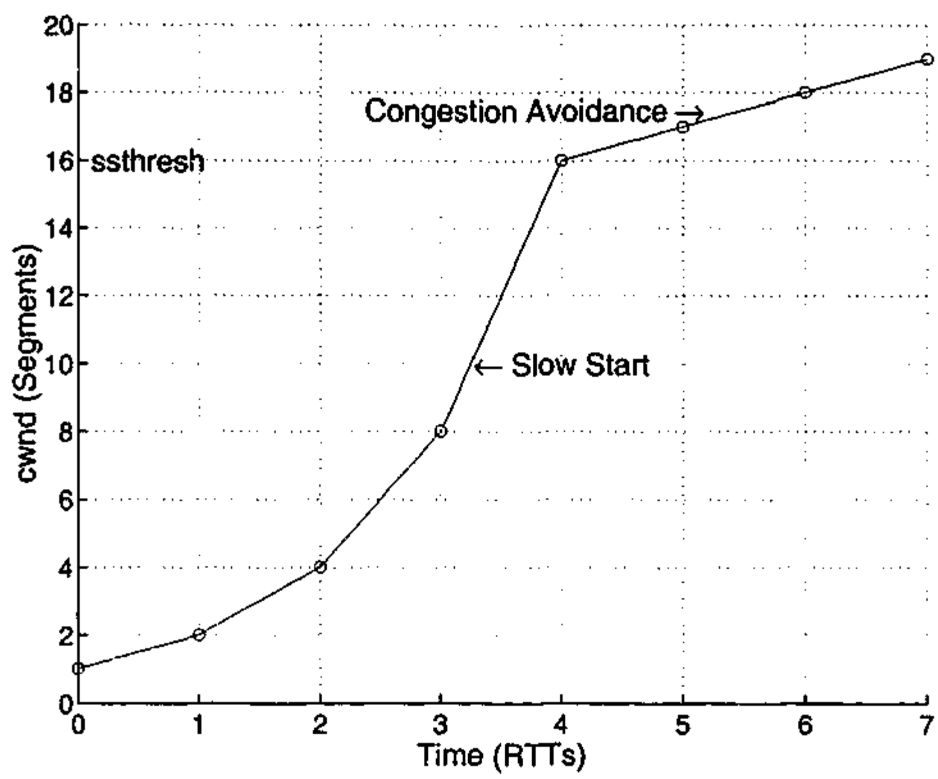


Figure 6.4: Visualization of Slow Start and Congestion Avoidance.

received in that round-trip time.

6.4.1.2 TCP Reno: Fast Retransmit and Recovery

Most TCP implementations use a coarse granularity (typically 500 *msecs*) timer for the retransmission timeout. When congestion occurs, significant time may be lost waiting for the timeout timer to trigger. During this time, the sender does not send any new segments if its allowed window is reached. It also does not retransmit the lost segment even if duplicate ACKs are received. Moreover, once the timeout timer expires, the sender retransmits the lost segment and is forced back to the slow start phase. Consequently, the link experiences low throughput as the connection remains idle for a long period of time and takes several round-trip times to fully utilize the network. TCP Reno implements fast retransmit and recovery to enable the connection to quickly recover from isolated segment losses [Ste97].

In fast retransmit, the arrival of three duplicate ACKs at the sender infers that a loss has occurred, and retransmission is begun without waiting for the RTO to expire. In fast recovery, the receipt of three duplicate ACKs, while indicating a loss of a segment, does not result in a full slow start. This is because obviously later segments got through, so the network congestion is not too bad.

Fast recovery involves setting ssthresh to half of cwnd; retransmitting the lost segment and setting cwnd to ssthresh plus three segments. Each time another duplicate ACK arrives, the sender increases cwnd by a segment and transmits a segment (if allowed by the new value of cwnd). Once an ACK is received for new data, it resets cwnd to the ssthresh value, and enters the congestion avoidance phase.

6.4.1.3 SACK: Selective Acknowledgments

The fast retransmit and recovery mechanism works well if no more than one segment is lost per window. However, if more than one segment is lost per window, the first lost segment will be detected by the sender from the duplicate ACKs received, and the second lost segment is not detected until the RTO timer expires at the sender. The connection is forced back to slow start phase and cwnd is set to one. This situation can be avoided by implementing selective acknowledgments (SACK) [MMFR96].

The availability of SACK is announced as an option in a SYN segment during the establishment of a connection. When the TCP receiver receives out-of-order segments, it sends duplicate ACKs with a "SACK block" option mapping out the as-yet unacknowledged segments. This SACK block specifies the out-of-order segments that have been received and are being buffered at the receiver. Thus the TCP sender optimizes its retransmission strategy, avoiding retransmitting segments identified in the SACK block. Consequently, the sender can recover from multiple lost segments in about one round-trip time. However if the sender's RTO expires, the sender must retransmit that segment regardless of whether it appeared in a SACK block.

6.4.2 TCP over UBR

This section discusses TCP/IP applications over ATM UBR class service. UBR class service category is a best-effort service, and no specific QoS must be provided. Unlike the ABR service, UBR does not provide any network feedback control nor source description. During congestion, cells are dropped at switches and sources are not expected to decrease their transmission rate. Rate adaptation and error recovery are handled by higher layers between TCP end systems.

Earlier generation drop schemes simply drop all incoming cells whenever the buffer is full without any regard to the AAL5 frame delineation. This results in low TCP throughput. Therefore, in order to achieve high throughput, intelligent buffer management policies employing a packet-based (or frame-based) discard strategy have been developed. These mechanisms aim to reduce cell losses in packets by minimizing the incomplete packet transmission rate. However, these mechanisms do not take fairness into consideration, i.e. when the buffer threshold for packet drop is reached, incoming packets from any connections are dropped irrespective of their current load in the buffer. Hence, more sophisticated mechanisms are developed to alleviate the fairness problem by fairly allocating the buffer space to each connection. Section 6.4.2.1 - 6.4.2.3 describe these schemes, some of which are used for the purpose of comparison later in the chapter. Section 6.4.2.4 provides a brief discussion of some other packet drop schemes.

6.4.2.1 Simply Drop Cell and Partial Packet Drop

Drop Tail (DT) is the simplest method to implement an ATM switch for UBR service category. DT does not selectively drop packets, it simply drops cells from the tail of the buffer whenever the buffer is full. It does not keep track of state information for each connection either. Obviously, this scheme is not the best policy as it does not provide any strategy to prevent cell loss. It also unnecessarily transmits incomplete packets (i.e. packets in which at least one cell is dropped by the switch). These incomplete packets are useless to the higher-layer protocols of the destinations. This results in a waste of bandwidth and leads to reduce effective TCP throughput. In addition, if a cell is dropped at a switch, the TCP source loses time waiting for retransmission timeout. The link is occasionally idle and the resulting throughput can be very low though TCP congestion mechanisms can effectively recover from loss later.

DT suffers from fairness and global synchronization problems in which connections sending more data will get more system resources. Moreover, it is biased against long packets as the probability of dropping cells at a congested ATM switch is higher than the shorter packets. Bursty connections will have poor performance too since the probability of filling up the buffer is higher.

Simulation studies in [RF95] show that DT performance is poor. Using 10 TCP (Tahoe) connections with different buffer sizes and packet sizes, it shows that the effective throughput for TCP connections over packet IP networks is at least 90% whereas TCP over ATM with DT can be as low as 34%. The performance is even worse if largest packet size and smallest buffer size are used.

Partial Packet Discard (PPD) is proposed [AA93] to overcome the shortcoming of DT. A packet is useless to the destination if at least one cell from the packet is dropped at an ATM switch. The remaining cells from the packet should be dropped rather than carried over the network which results in a waste of bandwidth. Hence the aim of the PPD is to improve DT by dropping all subsequent cells (except the last cell) belonging to the same packet which has already suffered a cell loss due to buffer overflow. This improves the network utilization and decreases the delay of the good packets (i.e. packets can reach the destination successfully without any errors).

PPD needs to keep track of packet boundaries and maintains a drop list so that the appropriate cell can be dropped. A drop list is a set of connections which at least one cell of a packet has been dropped. PPD is slightly more complex than DT, but can still be considered fairly simple. Similar to DT, PPD also suffers from fairness and global synchronization problems.

Romanov et la. [RF95] show that the effective throughput of TCP over ATM is improved with PPD. Nevertheless, due to the significant number of incomplete packets transmitted, the performance of PPD is still poor. Using the same simulation parameters as in DT, the performance could be as low as 50%.

6.4.2.2 Packet Drop without Fairness

Early Packet Discard (EPD) is proposed by [HWMB94] (patent filed in March 1993) and [RF95] to prevent useless data from being transmitted as is the case with PPD. EPD drops complete packets rather than partial packets. Consequently, bandwidth is not wasted in carrying incomplete packets which would have been discarded by the destination during reassembly. EPD uses a buffer occupancy threshold (R) to indicate the onset of congestion. Once the queue length reaches this threshold, as long as there is enough buffer space, EPD continues to accept cells from those packets which have already been partially admitted; whereas all cells from any new packets are dropped. Ideally, if a switch accepts the first cell of a packet from each connection before its queue length threshold value is reached, there should have enough excess buffer capacity to accommodate the entire packet for each connection. Therefore, typically, the EPD threshold value should be set to at least *buffer size* $-$ $N \times$ *the maximum packet size* where N is the expected number of active connections at any time [GJK+98].

Similar to PPD, EPD needs to maintain a drop list. To decide when to drop new incoming packets, it also needs to keep track of threshold-crossing and to identify the beginning of packets.

In [RF95], the performance of EPD is demonstrated to outperform DT and PPD. In the simulations, the EPD threshold is set to half the ATM switch buffer size and TCP window size is 64 Kbyte. EPD achieves more than 55% of effective throughput with a smaller buffer size and the biggest packet size. When the buffer size is slightly increased, EPD achieves more than 95% of effective throughput compared to the throughput produced by the PPD and DT.

Although EPD improves the efficiency of TCP over UBR, it does not significantly improve fairness. This is due to the fact that when the queue length threshold value is reached, EPD simply drops the new incoming packet from any connections irrespective of its current load in the buffer. As a result, the probability of dropping a packet from a connection which has a smaller packet is higher even though it has a lesser share of the fair share in the buffer. Thus, connections with larger packets will have a higher chance of achieving more throughput than the connections with smaller packets. The fairness problem of EPD has been demonstrated by simulation studies in [GJK+98, FL95].

6.4.2.3 Packet Drop with Fairness

Selective Drop (SD) scheme is proposed by Goyal et la. [GJK+98, GJK+97] to alleviate the fairness problem of EPD. Fairness is achieved by fair allocation of buffer space to each VC. If the buffer occupancy of a VC exceeds its fair share, its next incoming packet is dropped. Hence per-VC accounting is required to keep track of the number of cells each VC has in the buffer.

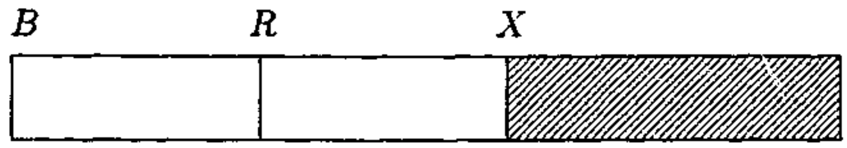
Let assume there are X cells in the buffer belonging to N_a number of active VCs (VCs with at least one cell in the buffer). Fair allocation of buffer space, F_s, for each VC would be:

$$F_s = \frac{X}{N_a} \tag{6.1}$$

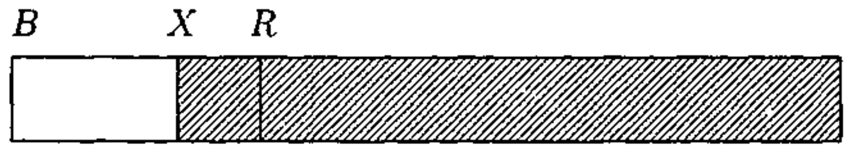
The amount a VC is overloading the buffer (i.e. by what ratio it exceeds the fair allocation) can then be measured by the ratio of the number of cells a VC has in the buffer to the fair allocation. So if a VC i has Y cells in the buffer, its load ratio, L_i is:

$$\begin{aligned} L_i &= \frac{Y_i}{F_s} \\ &= \frac{Y_i \times N_a}{X} \end{aligned} \tag{6.2}$$

Therefore, if $L_i > 1$, then VC i is consuming more than the fair share, and if $L_i < 1$, VC i has underutilized the buffer space. Figure 6.5 illustrates the drop



$X < R$: in accepting mode (no packets are dropped)



$X > R$: in discarding mode if $L_i > Z$

Figure 6.5: Drop conditions for SD.

conditions for SD. For a given buffer size B, if the queue length X is less than or equal to a minimum static threshold value, R, then SD is in accepting mode, and no cells are dropped. Otherwise, if the queue length exceeds this threshold value, R, a new incoming packet from VC i will be dropped if the load ratio of that VC is greater than a predefined parameter Z. However, if the load ratio of that VC i is not exceeding Z, the packet will not be dropped even if the queue length exceeds threshold R. Therefore the condition to drop a new incoming packet from a VC is:

$$(X > R)\ AND\ (L_i > Z) \tag{6.3}$$

The optimum value of Z is found to be close to one from the simulation studies of [GJK+98, GJK+97]. Although a higher value of Z would increase network utilization, it decreases fairness since packets from connections with more than their fair share could be accepted. SD improves fairness and alleviates the global synchronization problem since connections which overload the buffer are forced to reduce their congestion windows, preventing them from consuming more than the fair share of available resources. With these distinct advantages, the fairness concept of SD is thus adopted in the proposed selective drop scheme using a different approach.

SD is certainly more complex than EPD. Besides having some static threshold values, it needs to constantly monitor the activity of each VC and update their states. Additionally, more computational power is required to calculate fair share and load ratio.

Fair Buffer Allocation (FBA) [HK] packets dropping mechanism is similar to SD but uses a more complex form of drop threshold. When queue length exceeds a fixed threshold value, instead of comparing the load ratio of VC i with a fixed parameter Z, it is compared with a dynamic drop threshold D_t:

$$D_t = Z \times \frac{B - R}{X - R} \tag{6.4}$$

Therefore a packet is dropped from VC i if the following conditions are met:

$$(X > R) \ AND \ (L_i > Z \times \frac{B - R}{X - R}) \tag{6.5}$$

where B is the buffer size, X is the number of cells in the buffer, R is a minimum queue length threshold value, L_i is the load ratio of VC i, and Z is a linear scaling factor. FBA is a dynamic scheme since it adjusts its drop threshold according to the congestion level in the switch. As X increases (congestion increases), load ratio L_i is compared with a smaller value. Consequently, more packets from connections which have more cells in the buffer will be dropped, and only connections with fewer cells in the buffer will not incur losses.

An extensive simulation study was conducted by Goyal et la. [GJK+98] to compare the TCP performance of DT, EPD, SD and FBA. In the simulations, TCP segment size was set to 512 bytes. LAN and WAN network configurations with varying buffer sizes and number of sources were used. Simulation results show that both FBA and SD achieve better efficiency and fairness over DT regardless the latency of the network. Both FBA and SD also improve efficiency and fairness over EPD on a LAN environment; however there is no significant improvement on WAN. The study also shows that though FBA is more complex than SD, it does not improve TCP performance significantly over SD.

6.4.2.4 Other Packet Dropping Schemes

Most other schemes are the improvement and extension of earlier schemes, DT, PPD and EPD, and hence they inherit the fairness problem when working in conjunction with earlier schemes. This section briefly describes some of these schemes. A more detail survey of packet dropping schemes can be found in [LB99].

Lakshman et la. proposes the **drop from front** (DFF) [LNO96] in which cells are dropped closest to the beginning of the buffer rather than at the tail when the buffer is full. As a result, the TCP source is notified of congestion a buffer length drain time earlier than the tail drop scheme. Although DFF improves TCP throughput, it is sometimes considered an unnecessary complexity as it requires explicit queue manipulation to remove an existing entry.

Cheon and Panwar introduces the **early selective packet discard** (ESPD) [CP98]. Two queue threshold values: high and low thresholds are maintained. Fairness is achieved by assuming highly active connections have a higher probability to reach the high threshold value, and thus are marked red. Any new incoming packets of a marked connection are dropped until the mark is removed. The mark is removed when the queue length falls below the low threshold value. The trade-off between efficiency, fairness and complexity of the scheme is not clear. Moreover, tuning of the two threshold values have been shown to be a challenging task under different network conditions.

Chiou and Tsai presents the **age priority packet discarding** (APPD) [CT98]. Cells are dropped based on the *Age* of the packets competing for resources. *Age* is defined as the number of cells a considered packet has in the buffer. Whenever the residual buffer space is not enough to accommodate new incoming cells, APPD gives priority to the packet with higher *Age*. Hence a cell from a packet which has more cells already in the buffer is accepted, otherwise it is dropped. Thus APPD minimizes the waste in bandwidth by reducing the number of incomplete long packets transmitted through the network. An additional *Status_Table* is needed to keep track of connections in which cells have been discarded.

By taking the advantages of TCP fast retransmit and recovery mechanism, a **balanced packet discard** (BPD) scheme [CH00] is developed by Cohen and Hamo [CH00] to minimize the TCP timeout problem. BPD maintains two queue threshold values: Lower-Threshold (LT) and Upper-Threshold (UT). Whenever the lower-threshold is reached, BPD drops the incoming cell as well as all subsequence cells from the same packet. BPD also marks this VC as a "damaged" VC. A damaged VC is given a higher priority over non-damaged VCs, that is, cells are only dropped when the upper-threshold is reached. Hence, the damaged VC is given the chance to recover from the first loss, without immediate additional packet losses. As a result, multiple packet losses within the same window are minimized. The gap between the lower-threshold and the upper-threshold is known as a *recovery period*, that is, the spare buffer capacity is reserved for a subsequence number of packets from a damaged VC. Hence the upper-threshold value is the upper bound of this recovery period. A damaged VC becomes a regular VC when this threshold is reached. Although BPD has a priority mechanism, the gap between the two threshold values has to be carefully engineered in order to avoid multiple packet losses from the same window.

6.4.3 TCP over ABR

ATM ABR service supports traditional TCP/IP-based data communications by closed-loop and rate-based congestion control. Sources adjust their transmission rates according to the feedback information from the network. Consequently, cell loss rate can be drastically reduced when congestion occurs. Earlier generation switches use a binary feedback scheme to indicate congestion. Later generation switches use the ER field of the RM cells to convey congestion information to sources.

There has been an increase in support for ABR service from ATM switches and network card interface vendors. Hence it is important to study [illegible] ABR rate control on TCP performance.

ABR service can be provided to a TCP source and destination, in either an end-to-end configuration or in an ATM backbone. In the former case, the TCP source and destination are directly connected to an ATM network, whereas in latter case, they are indirectly connected to an ATM network via edge routers. In both cases, ABR control is terminated at the ATM layer inside the TCP hosts or edge routers.

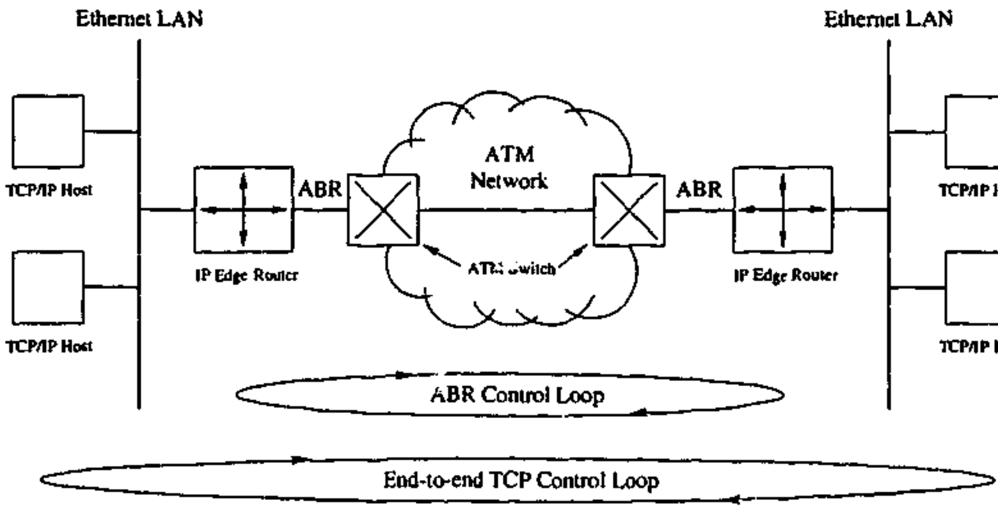


Figure 6.6: ABR service on a backbone ATM network.

When congestion occurs, TCP/IP protocols are not informed of congestion inside the ATM network. Cells are lost in the routers or TCP hosts when buffers overflow. Figure 6.6 shows the ABR service is provided on a backbone ATM network which interconnects two Ethernet LANs via edge routers [HA00]. Transmission rates are controlled by two control loops: the ABR control loop bounded by the two edge routers, and the end-to-end TCP control loop provided by the TCP source and destination. The ABR control loop provides congestion information within the ATM network to the routers. The end-to-end TCP control loop provides congestion information in the routers or in ATM network to the TCP source and destination.

Some research studies [Has97, JY96] reveal that TCP performance is degradated by the interaction of these two control loops. The ABR rate control simply pushes the congestion within the ATM network to the edge routers. Thus, to avoid cell loss, the edge routers need to provide one receiver's window worth of buffer capacity for each TCP connection.

Another essential factor to achieve good TCP performance over ABR is correct setting of ABR parameters at switches as well as at end systems. Fang and Lin [FL97] show that incorrect setting of these ABR parameters leads to degradation of TCP performance.

Lastly, ABR service can be provided by either binary or explicit-rate feedback congestion control. Binary feedback uses the EFCI bit of data cells (EFCI marking) or CI/NI bit of RM cells (Relative rate marking) to indicate congestion. Sources are informed to either increase or decrease their transmission rate based on set or reset of these bits. On the other hand, explicit-rate feedback uses the ER field of the RM cells to convey the specific allowed rate to each source. It is a more sophisticated control but provides better TCP performance than binary feedback schemes in terms of fairness and robustness. A simulation study from Saito et la. [SKK+96] shows that ABR service provided by the explicit-rate feedback control achieved better TCP performance than the binary feedback control.

6.5 The Proposed Fuzzy Logic Selective Cell Drop Scheme

Conventional cell drop schemes that implement static threshold values [GJK+98, RF95, HK] on queue and fairness controls have an inherent disadvantage. They cannot adapt well to traffic fluctuations, resulting in high cell loss, low TCP throughput and unfairness under some situations. In this section, an intelligent selective cell drop scheme which improves TCP throughput and fairness over UBR based on fuzzy logic prediction and control is proposed. Unlike conventional cell drop schemes, the proposed fuzzy logic selective cell drop controller makes intelligent decisions based on imprecise quantities with soft-computing. Moreover, with prediction, it can react to congestion sooner and thus improves TCP throughput.

Figure 6.7 shows the proposed fuzzy logic selective cell drop controller. It consists of a fuzzy logic selective cell drop factor generator and a fuzzy logic predictor.

To ensure fair buffer allocation among TCP sources and not to inherit the fairness problem of EPD, the fairness concept is adopted as in SD. The fair share of each VC is maintained by keeping track of the buffer occupancy against the number of active VCs. Then the load ratio of a VC is used to ensure that it does not exceed the fair buffer allocation. This is defined as the ratio of the number of cells a VC has in a buffer to the equal share [GJK+98]. Consequently, the global synchronization problem is also alleviated as sources which consume more than

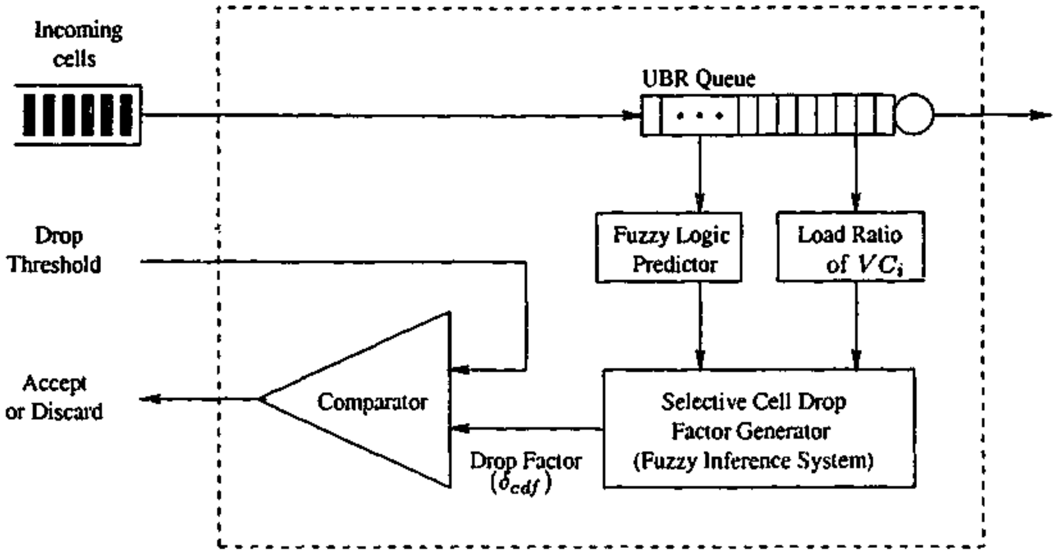


Figure 6.7: The Fuzzy Logic Selective Cell Drop Controller.

their fair share of resources are informed to decrease their congestion windows first before other connections. Therefore, congestion windows of all TCP connections are not decreased at the same time, avoiding all connections entering slow start mechanism and ensuring the network is utilized as much as possible.

The predictor predicts the UBR queue one round-trip delay in advance. This predicted queue value, together with the load ratio of VC_i are provided to the fuzzy logic factor generator which produces a selective cell drop factor (δ_{cdf}). This factor is compared with a predefined drop threshold. If it is lower than the threshold value, the cells from a packet are accepted. However, if it is higher than the threshold value, the FSCD continues to accept cells from the packet which has already been partially admitted; whereas new incoming packets are dropped.

As shown in Figure 6.7, the fuzzy logic selective cell drop factor generator is integrated with the fuzzy logic predictor. The output variable, selective cell drop factor, is governed by two input variables, i.e. the predicted UBR queue length, the load ratio of VC_i; and the linguistic information stored in the rule base.

Triangular and trapezium shapes are adopted for both input and output membership functions. Figure 6.8 shows the membership functions of the input and output linguistic variables when the buffer size is 1000 cells. The design of the

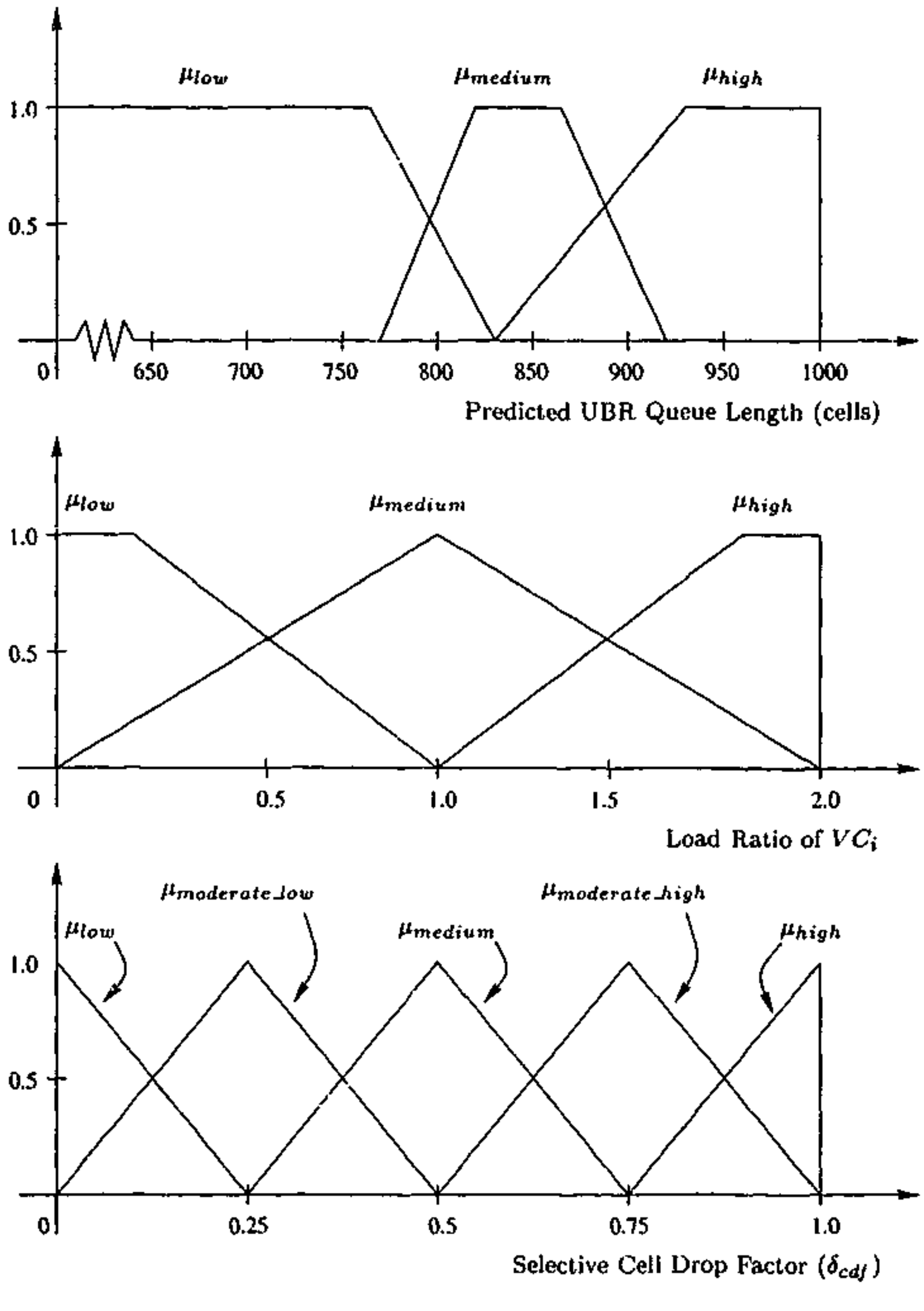


Figure 6.8: Membership functions of the linguistic values for representing the linguistic variables "*predicted UBR queue length*", "*load ratio of VC_i*" and selective cell drop factor (δ_{cdf}). Note: The linguistic values shown here are used in Section 6.7.2.2 simulation when buffer size is 1000 cells.

if pq is	and load ratio is	then δ_{cdf} is
high	high	high
high	medium or low	moderate_high
medium	high	moderate_high
medium	medium	medium
medium	low	moderate_low
low	high	moderate_low
low	medium or low	low

pq: predicted UBR queue

Table 6.1: Linguistic Rules for the Fuzzy Logic Selective Cell Drop Controller.

fuzzy linguistic rules for the inference of δ_{cdf} takes into consideration of the following conditions:

- Under severe congestion: when the predicted queue length is increasing rapidly and cells are about to overflow, then the selective cell drop factor should be high or moderate_high to drop new incoming packets regardless the load ratio of that particular VC.

- Under moderate load condition: the predicted queue length is increasing moderately; if the load ratio of VC_i is high, then the selective cell drop factor should be moderate_high to drop new packets from that VC in preference to packets of a VC which has a low load ratio.

- Under a light load condition, the predicted queue length is small, therefore the selective cell drop factor should be low, so that no cells are dropped.

Since the fuzzy system has two inputs and each has three membership functions, it has a total of 9 linguistic rules. After some trial optimization, 7 independent rules are adopted. Table 6.1 shows the linguistic rules for the fuzzy logic selective cell drop controller.

6.6 The Proposed Fuzzy Logic Congestion Control Scheme

A fuzzy logic ABR service congestion control scheme was proposed in Chapter 4. A key feature of this scheme is its ability to target link utilization dynamically based on the predicted future buffer condition in the switch. This is achieved by using fuzzy logic target utilization and fuzzy logic prediction. The predictor estimates the ABR queue length one round-trip delay in advance. The predicted queue length, together with the queue growth rate and current queue length are used to produce a target utilization factor. By using ABR and VBR class services, simulation results show that the fuzzy logic control scheme outperforms the conventional control scheme in terms of QoS measurements.

To investigate the effects of fuzzy logic prediction and control on TCP over ATM ABR congestion control, this proposed scheme is applied to the two switch algorithms later in the chapter. TCP performance and QoS measurements are then evaluated by a comparison with the conventional control scheme.

6.7 Performance Evaluation

In this section, simulation results are presented to illustrate the TCP performance over UBR and ABR on the fuzzy logic control schemes and their advantages over the conventional control schemes.

The simulation model and settings are described in the following section. Then, TCP performance over UBR is studied by comparing the proposed fuzzy logic selective cell drop scheme (Section 6.5) with the well known conventional schemes described in Section 6.4.2. Subsequently, TCP performance over ABR is investigated by comparing the proposed fuzzy logic congestion control scheme (Chapter 4) with the conventional control scheme on an approximate fair rate switch algorithm and an exact fair rate switch algorithm. Finally, a comparison of performance and cost between various TCP over ABR and UBR control mechanisms is presented in Section 6.7.4.

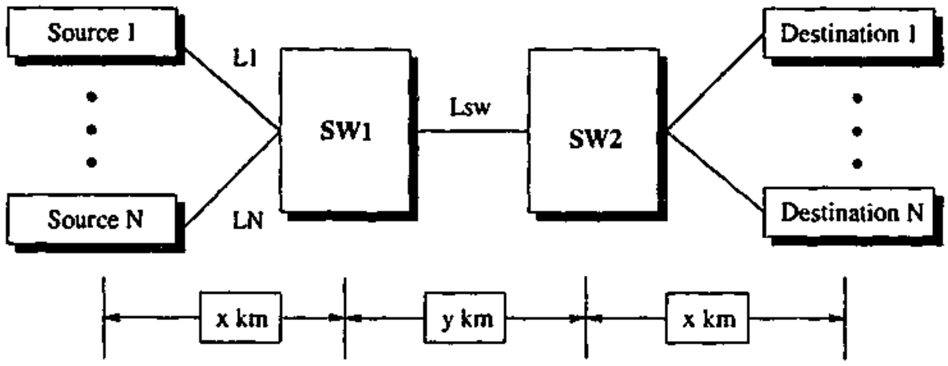


Figure 6.9: Simulation Model.

6.7.1 Simulation Model and Parameters

Figure 6.9 shows the simulation model used in the simulation. It is an N peer-to-peer connection network which shares a single bottleneck link between switches. The switches implement explicit rate allocation algorithms for ABR service and drop schemes, such as, EPD, SD, FBA and FSCD, for UBR service.

All traffic is unidirectional. The TCP sources implement basic TCP window-based flow control protocol. This includes slow starts, congestion avoidance and retransmission [Jac88, Ste97]. For fast recovery from isolated segment losses, the TCP sources also implement the fast retransmit and recovery mechanism [Jac90, Ste97]. A large, infinite file transfer application runs on top of TCP for the TCP sources. The TCP sources start sending data at the same time and keep sending a segment as long as it is allowed by the TCP window. All simulation parameter settings are tabulated in Table 6.2. Simulation duration is 5 seconds.

Performance Metrics

Two major performance metrics, efficiency and fairness [GJK+98, Kal97, RF95], for TCP/IP over ATM are used for the discussion in this chapter. They also conform to the recommendations of the test working group of the ATM Forum [JB97].

Definition 1 : Efficiency (E): *Efficiency is the ratio of the sum of all TCP throughput to the maximum possible throughput on a bottleneck link.* Thus it can

Parameter	**Value**
TCP Source:	
Maximum segment size (byte)	9140 or 1460
Timer granularity (*msec*)	100
Receiver winder size (Kbyte)	64
ACK delay timer	Not set
Link:	
Speed (Mbps)	155.52
Propagation delay (μs/Km)	5
Length (Km) x:y:x	1:1:1, 10:100:10
ABR End System:	
MCR (Mbps)	0
PCR (Mbps)	155.52
ICR (Mbps)	15.52
Buffer size	infinity
ABR-Switch:	
Output buffer size, (Cells)	200 - 1400, and infinity
UBR-End System:	
Buffer size	infinity
UBR-Switch:	
Buffer size, K (Cells)	1000, 2000, 3000, and infinity
Minimum drop threshold, R	0.9
Linear scale factor, Z	0.8

Table 6.2: Simulation Parameters.

be expressed as:

$$Efficiency\ (E) = \frac{\sum_{i=N}^{i=1} x_i}{C} \tag{6.6}$$

where N is the total TCP source-destination pairs, x_i is the throughput of the ith TCP connection ($0 < i \leq N$), and C is the maximum possible TCP throughput achievable over a bottleneck ATM link.

TCP throughput is measured at the destination TCP layer. It is defined as the total number of bytes (excluding retransmission or losses) delivered to the destination application divided by the total connection time. This is also known as *goodput* by some researchers [CH00, AA93, FJ93].

The maximum possible TCP throughput over UBR can be calculated as follows: to deliver a 9140 byte TCP segment over a 155.52 Mbps link, the ATM layer receives

8 Bytes	20 Bytes	20 Bytes	9140 Bytes	8 Bytes
LLC/SNAP Header	IP Header	TCP Header	TCP Payload	AAL5 Trailer

Figure 6.10: Payload receives at the ATM layer for a TCP segment size of 9140 bytes.

the payload which is illustrated in Figure 6.10. This payload is fragmented and padded to form 192 ATM cells. Hence 9140 bytes at the TCP layer results in 10176 bytes at the ATM layer. Thus, the maximum possible throughput is 9140/10176 = 89.82 % = 139.69 Mbps on a 155.52 Mbps link.

On the other hand, the maximum possible TCP throughput over ABR has to account for the overhead of RM cells (every 32 cells have an RM cell). Thus, it can be approximated as: $9140/10176 \times 31/32 \approx 87.01$ % which is 135.32 Mbps over a 155.52 Mbps link.

Besides providing high overall throughput, the network must also distribute throughput fairly among competing connections. The definition of fairness depends on the type of services. For UBR service, fairness can be defined as its ability to allocate throughput fairly among competing TCP connections. As for ABR service, fairness is determined by the ability to meet the MCR guarantee and to share the remaining bandwidth fairly. In general, fairness is measured by the fairness Index (F) and is defined by Equation (6.7).

Definition 2 : Fairness Index (F): *Fairness Index is a measurement of the dispersion of the individual throughputs.* It can be expressed as:

$$\textit{Fairness Index } (F) = \frac{\left(\sum_{i=N}^{i=1} \frac{x_i}{e_i}\right)^2}{N \times \sum_{i=N}^{i=1} \left(\frac{x_i}{e_i}\right)^2} \tag{6.7}$$

where N is the total TCP source-destination pairs, x_i is the measured throughput at the destination TCP layer of the ith TCP connection $(0 < i \leq N)$, and e_i is the fair share of the available bandwidth of the ith TCP connection.

For a symmetrical N-source network configuration, e_i is simply an equal share of the bottleneck link bandwidth ($e_i = C/N$). Whereas for a more complex configuration, e_i can be derived from the max-min fairness definition (Section 3.5). A low fairness index value indicates high unfairness among the connections, and vice versa. The desired fairness index value must be close to 1. Hence, a fairness index of 0.99 is considered to be near perfect. However, a fairness index of 0.90 may or may not acceptable depending on the application and the number of sources involved [GJK+98].

With a finite buffer size and variable ABR capacity (network which consists of VBR and ABR sources), cells may be dropped when the buffer is overflown. These lead to timeouts and retransmissions of TCP segments. As a result, throughput is decreased. Hence cell loss is also an important factor on TCP over ATM. Cell loss (in percentage) is defined as the ratio of a number of cells dropped to the number of cells sent during the simulation.

6.7.2 TCP Performance Improvement over ATM UBR

This section presents the simulation results to illustrate the TCP performance over UBR service on the proposed fuzzy logic selective cell drop scheme (Section 6.5) and its advantages over the conventional schemes, such as EPD, SD, and FBA.

Firstly zero TCP packet loss is considered in which an infinite buffer size is assumed. Then the TCP performance on various drop schemes with different TCP congestion control options is investigated. Different buffer sizes, link distances and TCP segment sizes are used in the simulation. Hence, in Section 6.7.2.2, the basic TCP window-based flow and congestion control mechanism on various drop schemes is first examined. Then in Section 6.7.2.3 the effects of the fast retransmit and recovery mechanism is established.

For all cases simulated, five TCP sources and destinations on the network configuration are used as shown in Figure 6.9.

Link (Km)	Maximum Queue (Cells)	Efficiency	Fairness
1	6288	1	1
100	5860	1	1

Table 6.3: TCP over UBR: Buffer requirements for zero loss (TCP segment size 9140 bytes).

6.7.2.1 Zero TCP Packet Loss

To achieve zero cell loss, infinite UBR service buffering is assumed at the switches. Table 6.3 shows the buffer requirements for TCP running over UBR using a segment size of 9140 bytes. All connections achieve 100% possible throughput and perfect fairness. For a 1 Km link distance, the maximum queue length is 6288 cells = 6288 $\times$ 48 bytes $\approx$ 301,824 bytes. This is approximately equal to the sum of the window size of five TCP connections, i.e. 5 $\times$ 64 Kbytes = 320,000 bytes. As the link distance increases, the maximum queue length decreases. This is because the switch needs a larger round-trip time to clear the buffer before it receives the next window size of data.

6.7.2.2 Basic TCP Flow and Congestion Control Mechanism

In this section, different link distances, buffer sizes and TCP segment sizes are used to illustrate TCP performance over UBR on both the conventional and the fuzzy logic selective cell drop schemes.

Tables 6.4 and 6.5 list the results of TCP performance on various cell drop schemes with the basic TCP flow and congestion control mechanism. The consolidated results are presented in Figure 6.11. In the figure, the average efficiency and fairness values are used to compare the performance of various cell drop schemes with different link distances and segment sizes.

Tables 6.4 and 6.5, and Figure 6.11 show that the fuzzy logic selective cell drop scheme consistently improves both efficiency and fairness. This is because the fuzzy logic predictor estimates the UBR queue length one round-trip delay in advance, so the buffer is less likely to be overflown or empty. In addition, soft-computing provides a more accurate estimation of parameters (e.g. buffer occupancy and load ratio) as opposed to the "hard" threshold values used in the conventional schemes. From the tables and figure, the fuzzy logic selective cell drop scheme achieves higher efficiency without a trade-off in fairness. Moreover, regardless of buffer size, the fairness index is maintained at above 0.97 (best shown in Figure 6.11(b)) in most cases. On the other hand, the conventional cell drop schemes do not perform consistently. For example, the FBA scheme performs better than the SD scheme with smaller TCP segment size (Table 6.4), but not in most cases when larger segment size is used (Table 6.5).

It can also be observed that fairness and efficiency improve with a larger buffer size. Since the buffer size is dependent on the sum of the window size of TCP connections, a larger buffer size results in more cells being accepted before the buffer is overflown. Hence efficiency improves with increasing buffer sizes. For example, from the tables, on average, a buffer size of 3000 cells improves efficiency by about 40% compared to a buffer size of 1000 cells.

In addition, efficiency is also improved by using larger segment size, especially when the buffer size is larger. When the segment size is larger, more data (cells) can be delivered to the destination. This also implies that a larger buffer is required to queue the data if the link is fully utilized. Thus the buffer is better utilized with a larger segment size. Therefore there is a trend of efficiency increases when the buffer size is increased. However, fairness is lower for larger segment size. This may be due to TCP synchronization effects in which some connections are able to send their entire windows of segments when the buffer is about to overflow, while other connections have to drop their segments.

Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.3484	0.3491	0.2937	0.4003
	2000	0.4062	0.4961	0.5638	0.5712
	3000	0.4969	0.5861	0.7601	0.7773
Column Average		0.4172	0.4771	0.5392	0.5829
100	1000	0.3247	0.3898	0.3755	0.4060
	2000	0.4596	0.4997	0.6178	0.6268
	3000	0.5742	0.6772	0.8478	0.8539
Column Average		0.4528	0.5222	0.6137	0.6289

(a) TCP Efficiency

Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.9406	0.8767	0.8920	0.9788
	2000	0.9875	0.9856	0.9922	0.9952
	3000	0.9376	0.9953	0.9986	0.9989
Column Average		0.9552	0.9525	0.9609	0.9910
100	1000	0.9569	0.9848	0.9765	0.9944
	2000	0.9684	0.9909	0.9994	0.9978
	3000	0.9613	0.9921	0.9958	0.9985
Column Average		0.9622	0.9893	0.9906	0.9969

(b) Fairness

Table 6.4: TCP over UBR: Performance Comparison of the Conventional and the Fuzzy Logic Selective Cell Drop Schemes (TCP segment size 1460 bytes).

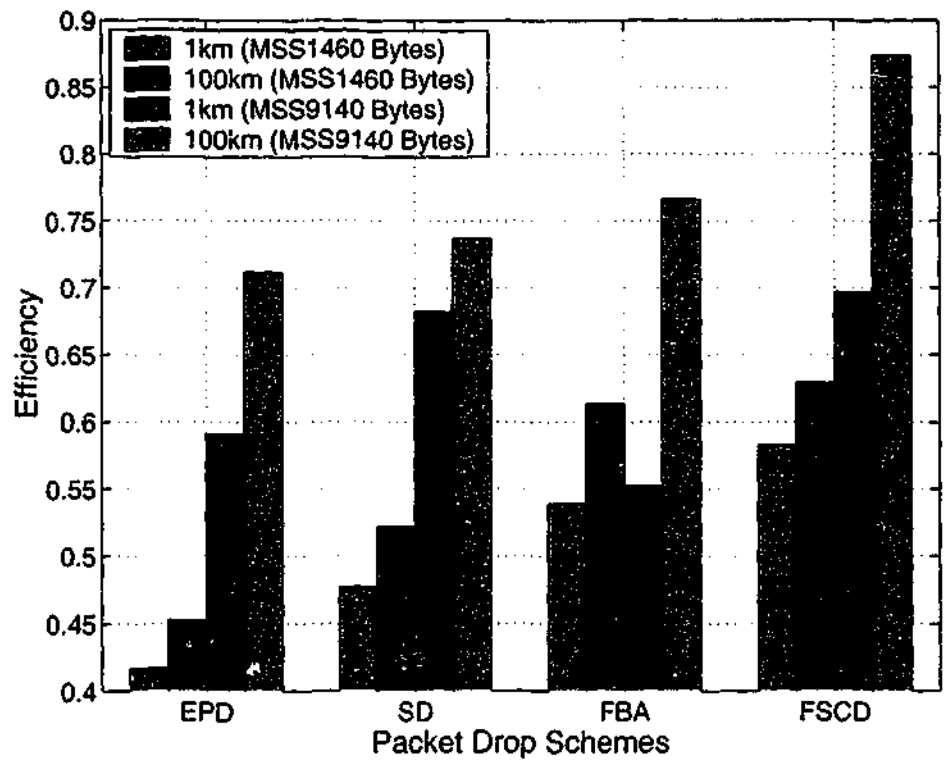
Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.2227	0.2949	0.2436	0.3841
	2000	0.6834	0.7964	0.6894	0.8345
	3000	0.8676	0.9552	0.7243	0.8718
Column Average		0.5912	0.6822	0.5524	0.6968
100	1000	0.4120	0.3701	0.5186	0.7090
	2000	0.8729	0.9197	0.8598	0.9531
	3000	0.8504	0.9225	0.9199	0.9596
Column Average		0.7118	0.7374	0.7661	0.8739

(a) TCP Efficiency

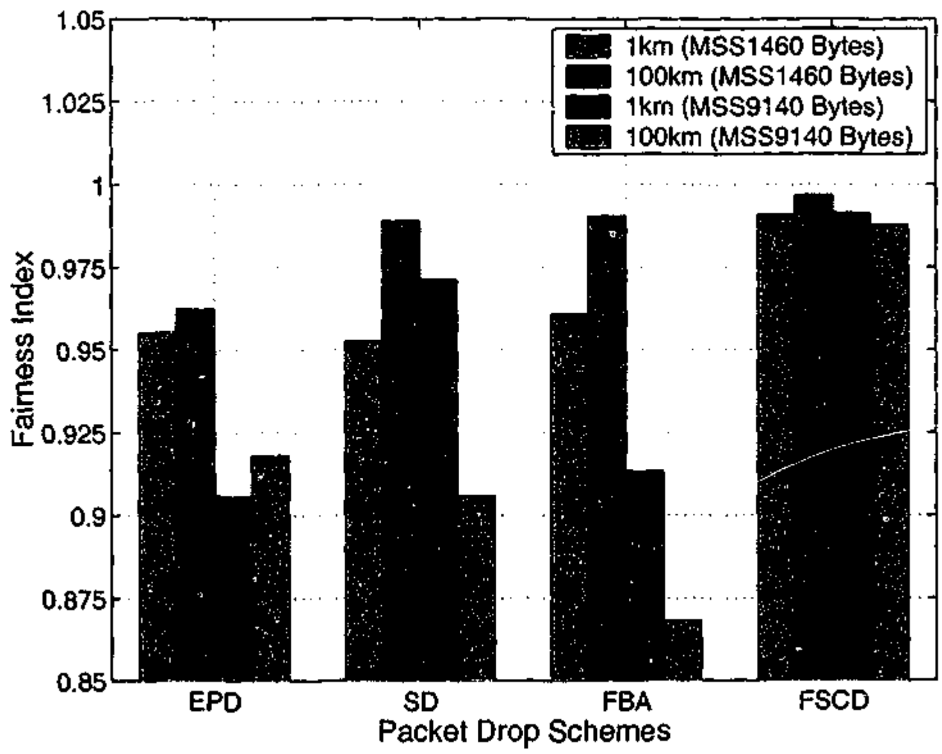
Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.7968	0.9562	0.8056	0.9804
	2000	0.9650	0.9882	0.9540	0.9954
	3000	0.9556	0.9699	0.9805	0.9985
Column Average		0.9058	0.9714	0.9134	0.9914
100	1000	0.8048	0.7310	0.8255	0.9799
	2000	0.9592	0.9951	0.8549	0.9958
	3000	0.9903	0.9916	0.9242	0.9881
Column Average		0.9181	0.9059	0.8682	0.9879

(b) Fairness

Table 6.5: TCP over UBR: Performance Comparison of the Conventional and the Fuzzy Logic Selective Cell Drop Schemes (TCP segment size 9140 bytes).



(a) Average TCP Efficiency



(b) Average TCP Fairness

Figure 6.11: TCP over UBR: Performance Comparison of the Conventional and the Fuzzy Logic Selective Cell Drop Schemes.

6.7.2.3 TCP Reno: Fast Retransmit and Recovery

Tables 6.6 and 6.7 show the simulation results of TCP Reno on various cell drop schemes with a segment size of 1460 and 9140 bytes respectively. Figure 6.12 presents the consolidated results and enables comparisons of the average efficiency and fairness values for different link distances and segment sizes to be made.

The tables and figure indicate that fast retransmit and recovery mechanism significantly improves efficiency compared with the corresponding drop schemes in basic TCP congestion control protocol, especially with EPD and SD. For example, Table 6.6 shows that with a segment size of 1460 bytes, efficiency is improved by about 50% in the EPD and SD. Overall, all schemes achieve a similar amount of efficiency with TCP Reno. However the FSCD scheme achieves slightly higher efficiency and fairness compared to various conventional schemes.

Another observation is that fairness and efficiency improve with a larger buffer size. This is consistent with the earlier results obtained with basic TCP congestion control mechanism (Section 6.7.2.2). A larger buffer size results in more cells being accepted before the buffer is overflown.

However, fairness and efficiency degrade with larger segments on all drop schemes. Fast retransmit and recovery mechanism retransmits a lost segment without waiting for timeout. With a larger segment, dropping one cell of a segment causes a larger number of cells to be retransmitted by the TCP source. If there is already not enough buffer space in the switch, this will deepen the congestion since much of the data will be retransmitted again, clogging the already congested link. This results in a waste of bandwidth and thus lower throughput.

Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.9330	0.9363	0.8285	0.9456
	2000	0.9545	0.9857	0.9955	1.0000
	3000	0.9963	1.0000	1.0000	0.9900
Column Average		0.9613	0.9740	0.9413	0.9785
100	1000	0.9251	0.9588	0.9077	0.9703
	2000	0.9943	1.0000	1.0000	1.0000
	3000	1.0000	1.0000	1.0000	1.0000
Column Average		0.9731	0.9863	0.9692	0.9901

(a) TCP Efficiency

Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.9939	0.9765	0.9738	0.9970
	2000	0.9794	0.9809	0.9890	0.9964
	3000	0.9980	0.9918	0.9790	0.9991
Column Average		0.9904	0.9831	0.9806	0.9975
100	1000	0.9759	0.9958	0.9900	0.9917
	2000	0.9854	0.9913	0.9921	0.9960
	3000	0.9932	0.9953	0.9916	0.9955
Column Average		0.9848	0.9941	0.9912	0.9944

(b) Fairness

Table 6.6: TCP Reno over UBR: Performance Comparison of the Conventional and the Fuzzy Logic Selective Cell Drop Schemes (TCP segment size 1460 bytes).

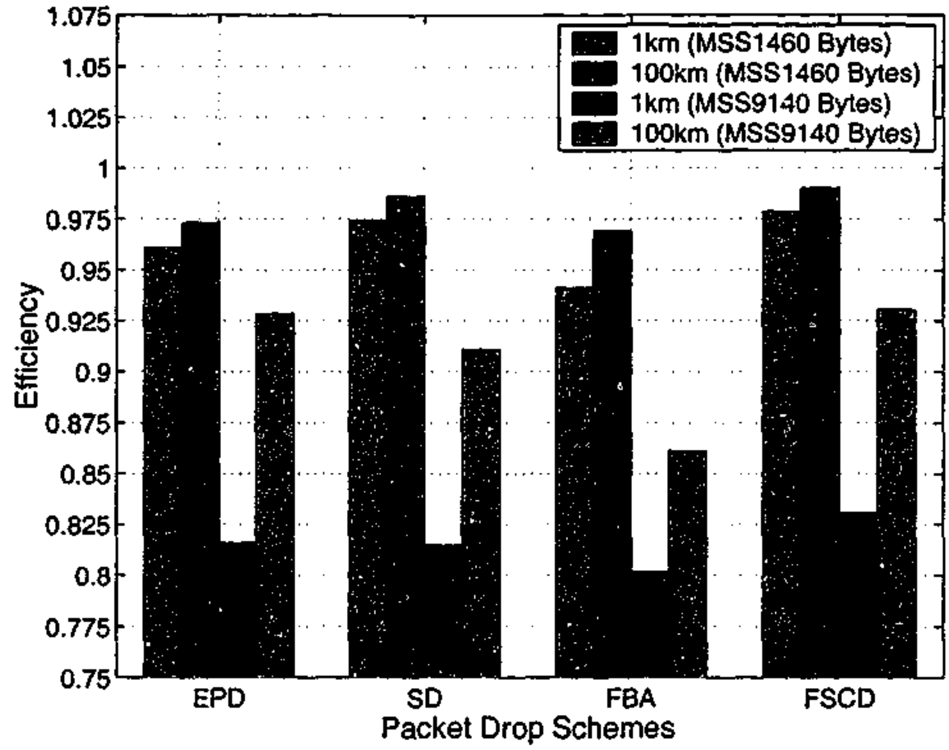
Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.5393	0.5551	0.5993	0.5822
	2000	0.9446	0.9270	0.8510	0.9478
	3000	0.9639	0.9622	0.9538	0.9616
Column Average		0.8159	0.8148	0.8014	0.8305
100	1000	0.8601	0.8090	0.8039	0.8774
	2000	0.9605	0.9596	0.8552	0.9533
	3000	0.9650	0.9644	0.9248	0.9616
Column Average		0.9285	0.9110	0.8613	0.9308

(a) TCP Efficiency

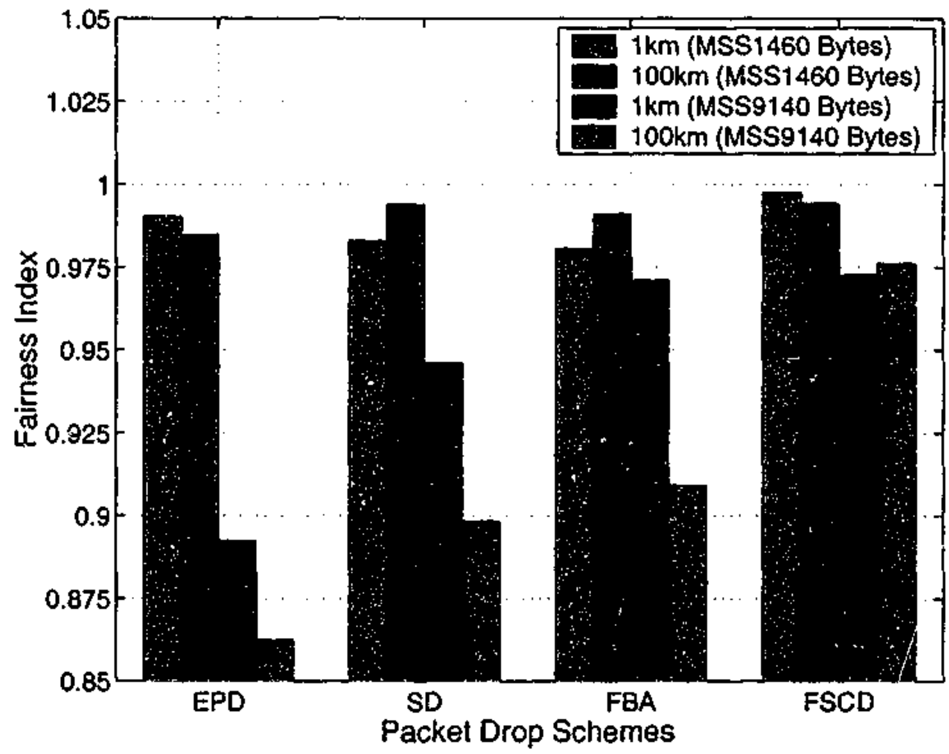
Link (Km)	Buffer Size (Cells)	Packet Drop Scheme			
		EPD	SD	FBA	FSCD
1	1000	0.8858	0.9566	0.9572	0.9694
	2000	0.8543	0.9731	0.9714	0.9721
	3000	0.9368	0.9088	0.9844	0.9766
Column Average		0.8923	0.9462	0.9710	0.9727
100	1000	0.6803	0.7206	0.8830	0.9351
	2000	0.9727	0.9826	0.9126	0.9992
	3000	0.9347	0.9918	0.9318	0.9941
Column Average		0.8626	0.8983	0.9091	0.9761

(b) Fairness

Table 6.7: TCP Reno over UBR: Performance Comparison of the Conventional and the Fuzzy Logic Selective Cell Drop Schemes (TCP segment size 9140 bytes).



(a) Average TCP Efficiency



(b) Average TCP Fairness

Figure 6.12: TCP Reno over UBR: Performance Comparison of the Conventional and the Fuzzy Logic Selective Cell Drop Schemes.

6.7.3 TCP Performance Improvement over ATM ABR

This section presents the simulation results to illustrate the TCP performance over ABR service with the proposed fuzzy logic congestion control scheme (Chapter 4) and its advantages over the conventional scheme. The proposed fuzzy logic control scheme is first applied to the ERICA+ and BEMO ABR switch algorithms. The performance of the proposed scheme is then compared with the conventional congestion control scheme on the two switch algorithms.

Firstly only TCP traffic is used. Then in Section 6.7.3.2 the interaction of TCP traffic with VBR traffic is considered.

For all cases simulated, the distance between switches is assumed to be 100 Km, and for the links connecting sources or destinations to switches, the distance is 10 Km. It is a WAN configuration. The initial cell rate (ICR) at the ATM-end systems is set to 10% of the link rate. TCP segment size is 9140 bytes.

6.7.3.1 Zero TCP Packet Loss

Five TCP sources and destinations on the network configuration (Figure 6.9) is used. To study the buffer requirements on zero cell loss ratio, an infinite ABR queue length is assumed at the switches.

Table 6.8 shows the buffer requirements on the conventional and the fuzzy logic control schemes in which 100% possible throughput and perfect fairness are achieved. The maximum queue length is very much affected by the ICR at the ABR end systems, the distance of the feedback path, and the queue control function. From the table, it shows that the fuzzy logic control scheme has a lower maximum and average queue length compared to the conventional scheme. This means that the fuzzy logic control scheme only requires a smaller buffer in order to achieve maximum possible efficiency and fairness.

Queue Length (Cells)	ERICA+		BEMO	
	Conventional	Fuzzy Logic	Conventional	Fuzzy Logic
Maximum	428	326	1328	316
Average	387	306	377	309

Table 6.8: TCP over ABR: Buffer requirements for zero loss.

6.7.3.2 With VBR Traffic

Five TCP sources and destinations, and one VBR background traffic are applied to the network configuration (Figure 6.9). The VBR source is modeled as a Poisson distributed on-off source with 10 *msecs* ON and 10 *msecs* OFF. It starts transmitting cells at $t = 300$ *msecs* at a variable rate with a mean bit rate of 124.42 Mbps (i.e. 80% of the link rate). Note that this high volume of VBR traffic is deliberately added to observe cell loss under different conditions.

At the switch, since VBR traffic has a higher priority than ABR traffic, and this simulation study is to investigate the buffer requirements and TCP performance over ABR service, the VBR queue is thus assumed to have an infinite buffer size.

Infinite ABR service buffering is assumed first in order to study the buffer requirements for a zero cell loss ratio. Table 6.9 shows the buffer requirements for the conventional and the fuzzy logic control schemes. From the table, the fuzzy logic control scheme has the least buffer requirements in order to achieve maximum efficiency and fairness without cell loss. It has a lower maximum and average queue as well as a lower cell delay variation.

To further study the efficiency, fairness and cell loss ratio at the switch with limited buffer size, the ABR service output buffer is varied from 200 to 1400 cells. Figures 6.13 - 6.16 show the resulting TCP performance over ABR service on the conventional and the fuzzy logic control schemes with various buffer sizes. Different line types represent simulations with different control schemes. The dashed lines are the conventional schemes, and the solid lines are the fuzzy logic control schemes.

Queue Length (Cells)	ERICA+		BEMO	
	Conventional	Fuzzy Logic	Conventional	Fuzzy Logic
Maximum	716	620	1326	612
Average	266	225	246	234
Cell Delay Variation	154	140	124	123

Table 6.9: TCP over ABR with VBR background traffic: Buffer requirements for zero loss.

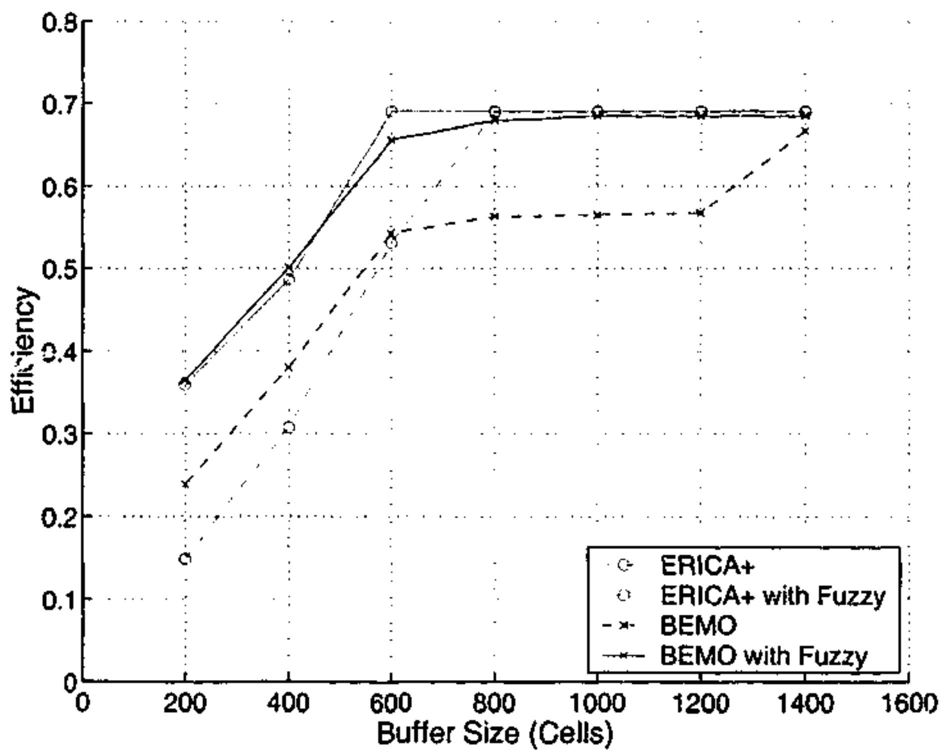


Figure 6.13: TCP over ABR: TCP Efficiency Comparison.

The efficiency, fairness, cell delay variation and cell loss ratio are shown in Figures 6.13, 6.14, 6.15 and 6.16 respectively. Figure 6.13 shows that before both schemes achieved zero cell loss, the fuzzy logic control scheme achieves a higher efficiency than the conventional control scheme. This is due to lower cell delay variation and thus lower cell loss which is illustrated in Figures 6.15 and 6.16 respectively. Figure 6.14 shows that the ERICA+ switch algorithm with fuzzy logic control improves fairness, whereas the BEMO switch algorithm with fuzzy logic control has a slight decreased of fairness for better efficiency before achieving perfect fairness with zero cell loss.

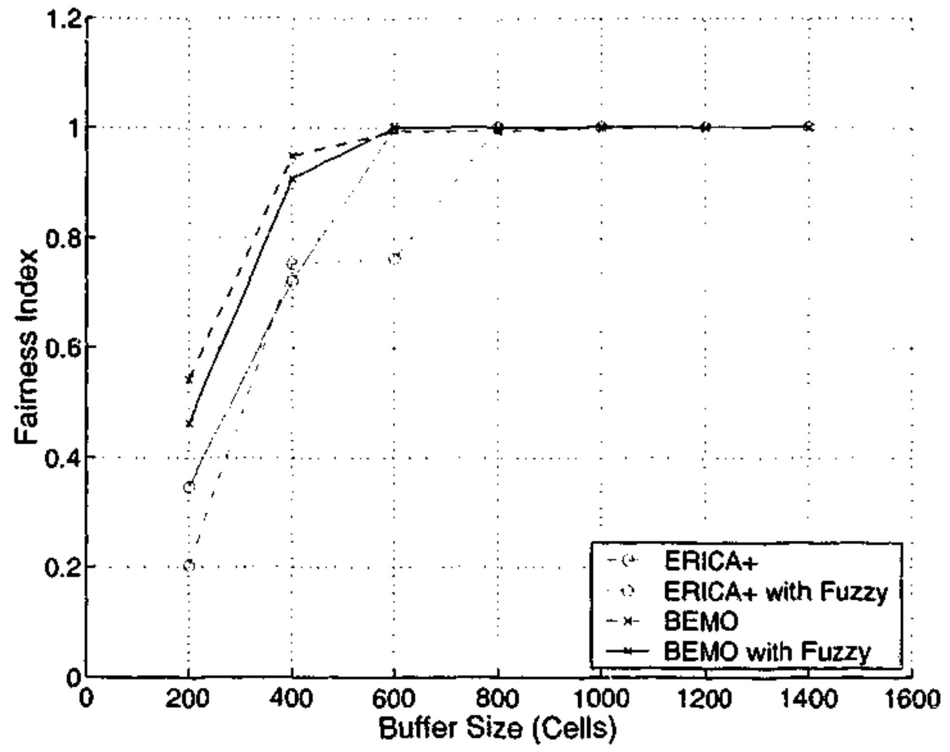


Figure 6.14: TCP over ABR: Fairness Comparison.

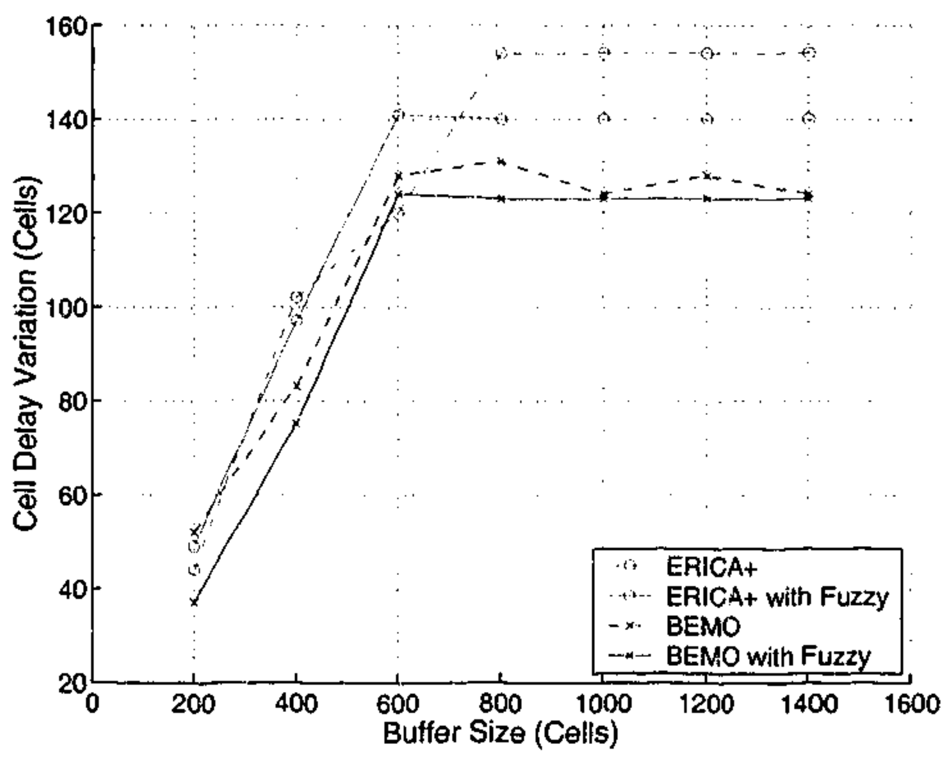


Figure 6.15: TCP over ABR: Cell Delay Variation Comparison.

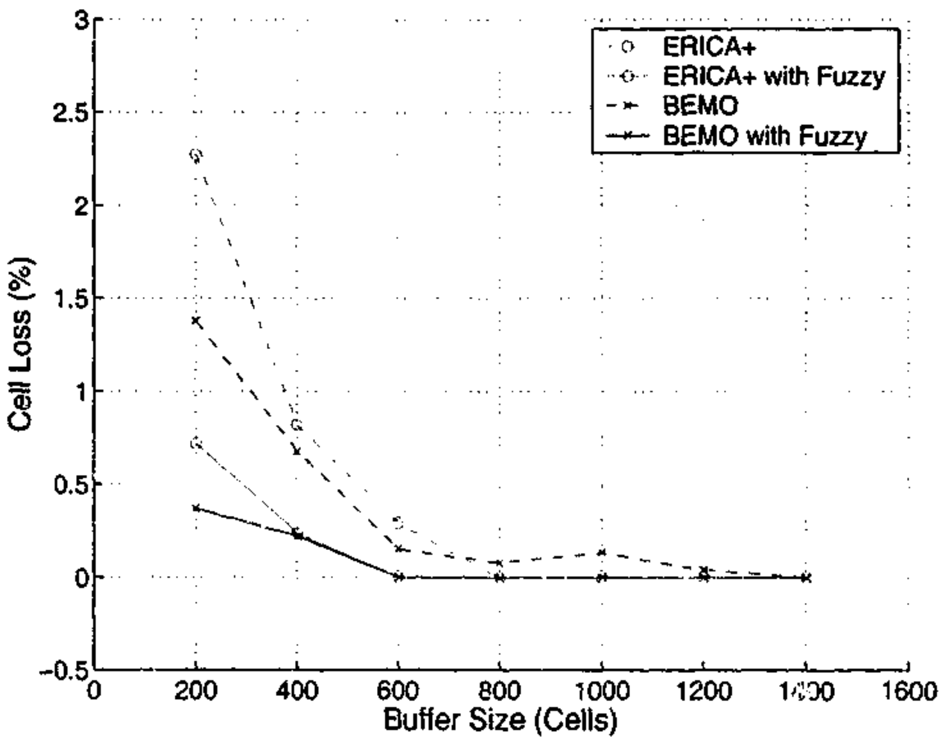


Figure 6.16: TCP over ABR: Cell Loss Comparison.

6.7.4 Comparison of TCP over ABR and UBR Mechanisms

Due to the end-to-end explicit rate congestion feedback mechanism, ABR is an expensive solution to reduce cell loss and improve TCP performance. In contrast, UBR mechanisms are less expensive and simple to implement. This section compares the performance between these two mechanisms.

Table 6.10 shows the simulation results of the buffer requirements for zero loss and finite buffer size on ABR-ER and UBR mechanisms. Under the cases of no cell loss, both TCP over UBR and ABR-ER achieve maximum efficiency and perfect fairness. However, TCP over UBR requires larger buffer sizes at switches in order to have the same level of performance. If buffer size is limited to, for example, 1000 cells, FSCD can only achieve 71% efficiency and a 0.98 fairness value. Hence, although the ABR-ER mechanism is expensive to implement, it does not require a large buffer size within the network in order to achieve better TCP performance.

A similar study has been conducted by Li et la. [LST96], with the comparison being between ABR-ER and UBR-EPD in a LAN environment. It demonstrates that when buffer size is limited, TCP performance of UBR-EPD degrades in terms

Service Category	**Maximum Queue (Cells)**	**Efficiency**	**Fairness**
ABR-ER	326†	1	1
UBR	5860	1	1
UBR	1000‡	0.7090	0.9799

†ERICA+ with Fuzzy Logic control
‡Fuzzy Logic Selective Cell drop scheme

Table 6.10: TCP over ABR-ER versus UBR: Buffer requirements (TCP segment size 9140 bytes, 100 Km link distance).

Scale†	**Performance**	**Cost**
1	ABR-ER	ABR-ER
2	UBR-FSCD	ABR-EFCI
3	UBR-FBA, UBR-SD	UBR-FSCD
4	UBR-EPD	UBR-FBA
5	UBR-PPD	UBR-SD
6	UBR-EFCI	UBR-EPD
7	UBR-DT	UBR-PPD
8	—	UBR-DT

†Scale 1 has the best performance and yet the most expensive to implement

Table 6.11: Performance and Cost Comparison among various TCP over ABR and UBR control mechanisms.

of fairness. TCP over UBR-EPD also suffers from the "beat-down" problem while TCP connections traverse multiple congestion links. When the buffer is not overflown, UBR-EPD has similar performance as ABR but requires a larger buffer size.

Fang and Lin [FL97] compared the TCP performance of ABR-EFCI over UBR-EPD. A different buffer size and LAN and WAN distances were used in the studies. It shows that ABR-EFCI does not have a significant performance gain over the UBR-EPD. In some situations, the ABR-EFCI is sensitive to ABR parameter tuning. As a result, its performance is poorer than the UBR-EPD.

Table 6.11 compares the performance and cost for various TCP over ABR and UBR control mechanisms. ABR-ER has the best performance but it is the most expensive service to implement, whereas the UBR-DT is the least expensive service despite its poor performance. Among UBR mechanisms, based on simulation results, it can be concluded that the FSCD has the best performance with the expense of slightly more computation due to prediction. The FBA is more complex than SD, but it does not improve TCP performance consistently and significantly when compared to SD. However, both the FBA and SD perform better than the EPD especially in terms of fairness.

6.8 Conclusions

This chapter has presented an examination of the Internet TCP/IP traffic over ATM UBR and ABR classes of service. In both cases, fuzzy logic prediction and control have been used to improve the efficiency and fairness of traffic throughput.

For TCP over UBR, a fuzzy logic selective cell drop (FSCD) scheme was proposed. A key feature of the scheme is its ability to accept or drop a new incoming packet dynamically based on the predicted future buffer condition in the switch. This is achieved by using a fuzzy logic predictor and a fuzzy logic selective cell drop factor generator. The fuzzy logic predictor estimates the UBR output switch queue length one round-trip delay in advance. Thereby it tackles the problem of long round-trip delays in the network. In addition, a fairness concept is adopted as in SD by keeping track of the buffer load ratio of each VC, which prevents any VC from overloading the buffer. The predicted queue length and the load ratio of the VC are then fed into the fuzzy logic selective cell drop factor generator which produces a drop factor. The decision to drop a packet is then based on this drop factor and a predefined threshold value.

The performance of this proposed cell drop scheme was compared with other conventional schemes such as EPD, SD and FBA, by simulations. Persistent TCP traffic was applied to a two nodes network with five peer-to-peer connections. Different buffer sizes, link distances and segment sizes with different TCP congestion control options were tested with various cell drop schemes. Simulation results show that the proposed scheme achieves higher TCP efficiency and better fairness compared with various conventional schemes on the basic TCP congestion control mechanism. With TCP Reno, efficiency is further improved for all UBR cell drop schemes, in particular the EPD and SD schemes. However, the proposed scheme achieves higher performance over all other schemes. Thus it can be concluded that although TCP performance can be improved by efficient TCP end-to-end congestion control mechanisms, with the application of fuzzy logic prediction and control, the performance is further improved.

To study the effect of fuzzy logic prediction and control on TCP over ABR, the fuzzy logic ABR service congestion control scheme described in Chapter 4 was applied to both approximate and exact fair rate computation ER switch algorithms. The performance of the fuzzy logic control with the conventional schemes was then compared. When the same two nodes network with five TCP connections were used, simulation results show that on zero TCP packet loss, the fuzzy logic control scheme achieves maximum efficiency and perfect fairness with a smaller buffer size. With the addition of a high volume VBR connection, the fuzzy logic control scheme achieves higher efficiency with lower cell loss and cell delay variation.

A comparative study of the performance and cost of various TCP over ABR and UBR control mechanisms was also conducted. Under the cases of no cell loss, both TCP over ABR and UBR achieve maximum efficiency and perfect fairness. However, TCP over UBR requires larger buffer sizes at switches, whereas TCP over ABR compensates the complexity of feedback mechanism with smaller buffer size. Hence ABR-ER is the most expensive service to implement though it has the best TCP performance. Among UBR mechanisms, based on simulation results, it can be concluded that FSCD has the best performance although it requires slightly more computation due to prediction.

Chapter 7

Conclusions and Future Work

7.1 Concluding Remarks

This thesis presents the application of fuzzy logic prediction and control on traffic management in high speed networks. Three major aspects of traffic management have been investigated: (1) flow and congestion control for ABR service (2) connection admission control for VBR service and (3) congestion control for TCP/IP over ATM. The primary focus has been on the development of intelligent traffic controllers in these areas. The main goal is to overcome the problems of long round-trip delays, uncertainty of traffic parameters in high speed networks, and the limitations in conventional control schemes. This is achieved by incorporating fuzzy logic prediction and control in the proposed traffic control schemes. The key feature of these proposed schemes is their ability to predict traffic or network resources far enough in advance so that traffic flow or resource usage can be controlled in time to avoid congestion and achieve high network utilization. In addition, with the robust framework of fuzzy set theory, the problems of traffic uncertainty and fluctuations in high speed networks are better dealt with. Performance of the proposed schemes is evaluated by simulation with different network conditions and configurations.

The following presents the conclusions of this study.

7.1.1 Flow and Congestion Control for ABR Service

This part of the research has been concentrated on the development of a novel fuzzy logic closed-loop rate-based congestion control scheme for ATM ABR service. Various congestion detection techniques and control mechanisms employed by existing switch schemes were investigated. Approaches such as using static threshold values or mathematical equations on queue control, incorporating traffic load and queue length information in feedback control mechanisms, application of fuzzy logic and neural networks were examined.

To tackle the problem of long round-trip delays in the feedback control loop, a novel fuzzy logic congestion control scheme which does not use static threshold values or complex queue control functions was proposed. It is based on prediction in which link utilization is targeted dynamically one round-trip delay in advance. This is achieved by using a fuzzy logic predictor and a fuzzy logic target utilization factor generator. The predictor estimates the ABR queue length one round-trip delay ahead. The predicted queue length, together with the queue growth rate and current queue length is provided as input to a fuzzy logic inference system which produces a target utilization factor. This factor is then incorporated in the feedback control message in which control information is conveyed to the source.

The proposed fuzzy logic control scheme was applied to an approximate fair rate switch algorithm (ERICA+ [JKG+96]) and an exact fair rate switch algorithm (BEMO [TQ98]). The performance of the fuzzy logic control scheme was then compared with the conventional scheme on the two switch algorithms. With network models consisting of only ABR traffic, the fuzzy logic congestion control scheme is able to optimize both transient and steady state responses. It exhibits a faster convergence time, lower buffer requirements, and smaller rate and queue oscillation during transient periods, and is much more stable in rate and queue length when in steady state. With the presence of VBR/CBR and ABR traffic, the fuzzy logic congestion control scheme exhibits lower buffer requirements and low cell loss for large round-trip delays, while still being able to maintain high link utilization. Hence, the overall requirements of QoS are improved. The proposed fuzzy logic congestion control scheme which adapts well to complex network configurations and traffic fluctuations is thus proved to be a more effective mechanism than the conventional scheme.

For the purpose of comparison to the proposed fuzzy logic control scheme, the BEMO switch algorithm was enhanced by integrating the algorithm with the conventional congestion control strategy of the ERICA+. Congestion is thus controlled by targeting the link utilization using a queue control function and a fixed non-zero queuing delay. In contrast to the ERICA+ algorithm, due to the exact fair rate computation, the enhanced BEMO algorithm does not need to keep track of the maximum previous and current allocation of ER values in order to achieve max-min fairness. It also has the advantage of fast convergence time, and is able to interoperate with other switches schemes which do not mark the ER field of RM cells.

In the ERICA+, step, linear, hyperbolic and inverse hyperbolic queue control functions were proposed to control queue and delay [VJGF98]. From the simulation results, the authors concluded that the inverse hyperbolic queue control function has the best performance as opposed to the hyperbolic function. However, after the discovery of a minor error, an analytical study conducted by this research shows that the hyperbolic queue control function is preferred as it has a decreasing function when a switch is lightly overloaded. Further simulation studies show that the hyperbolic queue control function has a better performance in terms of rate convergence time and buffer requirements.

7.1.2 Connection Admission Control for VBR Service

This part of the research has been focused on the investigation of the benefits of fuzzy logic prediction on CACs, and the design of a novel fuzzy logic measurement-based CAC. CAC approaches such as traffic and queue modeling for the conventional schemes, traffic and system performance measurements for the measurement-based schemes, fuzzy logic controls and/or neural networks for the unconventional control schemes, and the effective bandwidth approach verses statistical allocation approach were also investigated.

To overcome the inaccuracies and errors involved in the conventional measurement-based CAC, a novel measurement-based CAC using a fuzzy logic approach was proposed. It adopts the concepts of free bandwidth and the adaptive weight factor in [ZL98b, ZL98a]. In contrast to the model-based CAC, the proposed CAC does not use complicated analytical models or *a priori* traffic descriptors. Instead, traffic parameters are predicted by an on-line fuzzy logic predictor [Qiu01, QSW99]. This has the advantage of being able to forecast the workload, and also solves the difficulties involved for a user to accurately describe the traffic characteristics of the new connections. Another key feature of the proposed scheme is its ability to target QoS requirements indirectly by an adaptive weight factor. This weight factor is generated by a fuzzy logic inference system which is based on arrival traffic, queue occupancy and link load. Admission decisions are then based on a real-time measurement of the aggregate free bandwidth with the fuzzy logic adaptive weight factor as well as the predicted traffic parameters.

The effect of the measurement interval was first investigated by comparing the link utilization and cell loss ratio against the measurement periods. Both homogeneous and heterogeneous traffic were used in the simulations. Simulation results reveal that the measurement interval has to be short enough to be practical, and also be able to capture the significant statistical information of traffic streams.

The benefits of fuzzy logic prediction on CAC were then evaluated by applying the prediction technique to the conventional CAC and the proposed fuzzy logic measurement-based CAC. Simulation results show that fuzzy logic prediction significantly improves link efficiency while maintaining QoS requirements on both CAC schemes. In addition, the measurement-based approach incorporating fuzzy logic inference and using fuzzy logic prediction is shown to achieve higher network utilization while maintaining QoS objectives.

7.1.3 Congestion Control for TCP/IP over ATM

The major part of this work has been directed towards designing an intelligent and robust cell drop scheme to support TCP traffic over ATM UBR category of service. The effect of fuzzy logic prediction and control was also investigated for TCP over ABR service.

To address the problem of long round-trip delays in the network, an intelligent fuzzy logic selective cell drop (FSCD) scheme for TCP/IP over UBR was proposed. This is achieved by using a fuzzy logic predictor which estimates the UBR output switch queue length one round-trip delay in advance. The proposed scheme is thus able to accept or drop a new incoming packet dynamically based on the predicted future buffer condition in the switch. In addition, to ensure fairness among connections, a fairness concept is adopted as in SD by keeping track of the buffer ratio of each connection, preventing any connection from overloading the buffer. The predicted queue length and the load ratio of the VC are then supplied as inputs to the fuzzy logic selective cell drop factor generator which produces a drop factor. The packet dropping decision is then based on this drop factor and a predefined threshold value.

Simulation results have demonstrated that the proposed scheme consistently achieves higher TCP efficiency and better fairness over the conventional schemes with the basic TCP congestion control mechanism. With TCP Reno, efficiency and fairness have been further improved on all UBR cell drop schemes, and the proposed scheme again achieves higher performance over all other schemes. Thus it can be concluded that performance can be improved by efficient TCP end-to-end congestion control mechanisms, however, with the application of fuzzy logic prediction and control, performance is improved even further.

The effect of fuzzy logic prediction and control on the performance of TCP over ABR service was also investigated. The fuzzy logic ABR service congestion control scheme described in Chapter 4 has been applied to the conventional switch algorithms. Simulation results have shown that on zero TCP packet loss, the fuzzy logic control scheme achieves maximum efficiency and perfect fairness with a smaller buffer size. With the addition of a high volume VBR connection, the fuzzy logic control scheme achieves higher efficiency with lower cell loss and cell delay variation.

Finally, the performance and cost of various TCP over ABR and UBR control mechanisms were compared. Under the case of no cell loss, both TCP over ABR and UBR achieve maximum efficiency and perfect fairness. However, TCP over UBR requires larger buffer sizes at switches. Although TCP over ABR has smaller buffer requirements in the network, it has a complex feedback mechanism. Hence ABR-ER is the most expensive service to implement though it has the best TCP performance. Among UBR mechanisms, based on simulation results, it can be concluded that FSCD has the best performance although it requires slightly more computation due to prediction.

7.2 Recommendations for Future Work

The following outlines some extensions of the research presented in this thesis which are worth pursuing in traffic control, especially with the applications of fuzzy sets and fuzzy logic theory.

Due to the advances in hardware and switching technologies, ER switches are becoming popular since they perform better than the earlier generation EFCI switches. In the transitional period, EFCI and ER switches may coexist in the same network. Thus a switch scheme should operate efficiently in a multi-vendor and mixed EFCI-ER environment. In this research study, the performance of the proposed fuzzy logic congestion control scheme was only examined using networks with single type switches. Therefore, it would be worth investigating the interoperability of the proposed fuzzy logic ER scheme combined with EFCI switches in the same network.

In the simulation studies, the performance of the proposed fuzzy logic CAC was compared to the conventional model-based CAC. It would be worth comparing the performance with some conventional measurement-based CACs. In addition, real VBR traffic traces such as MPEG video sources could be used in the simulation. The performance of the proposed CAC could also be further examined on multi-hop network configurations.

In this research, the performance investigation of the proposed congestion control schemes for TCP over UBR and ABR were based on a simple five TCP connections with a two switch network configuration. It would be interesting to study their behavior in an environment of a large number of VCs and multiple switches.

The proposed scheme for TCP over ABR provides better performance over the conventional schemes in terms of effective throughput and fairness. A further improvement to TCP over ABR, especially under long round-trip delays, is to incorporate the feedback capabilities of ATM ABR into end-to-end TCP congestion protocol. For example, the allowed cell rate generated by the proposed ABR congestion control scheme can be integrated with the TCP explicit congestion notification protocol (ECN) proposed in [RF99]. The current TCP ECN allows routers at the ATM network boundary to modify the IP packet header as an indication of congestion to the end systems instead of dropping the packet. With the integration of an ABR congestion control algorithm, TCP end systems would benefit from a more intelligent and fast ECN response over ATM networks.

Future broadband satellite communications systems will offer high-speed Internet access and multimedia services. Several Ka-band satellite systems are in the planning stages to provide these services by IP, ATM or a combination of both [FP00].

Long latency delay path is one of the design issues on satellite networks. Hence it is useful to investigate the performance of fuzzy logic prediction and control on these emerging technologies. Therefore the proposed fuzzy logic traffic control schemes for CAC, TCP over UBR and ABR can be applied to satellite networks to further investigate their performances.

The design of fuzzy logic membership functions and linguistic rules for the traffic controllers proposed in this thesis is by experiments using simulations. They can be further optimized and automated by genetic algorithms [Vos99, SSZ97] to provide a more optimal performance.

Appendix A

Table of Abbreviations

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACR	Allowed Cell Rate
ADTF	ACR Decrease Time Factor
ANN	Artificial Neural Network
APPD	Age Priority Packet Discarding
AR	Autoregression
ATM	Asynchronous Transfer Mode
BEMO	Bi-directional Explicit Rate Marking Optimizer
B-ISDN	Broadband-ISDN
BN	Backward Notification
BPD	Balanced Packet Discard
BRM	Backward Resource Management
CAC	Connection Admission Control
CBR	Constant Bit Rate
CCR	Current Cell Rate
CDF	Cutoff Decrease Factor
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CI	Congestion Indication
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CRC	Cyclic Redundancy Check
CRM	Missing RM cell count
CS	Convergence Sublayer
CTD	Cell Transfer Delay
CTDV	Cell Transfer Delay Variation
DES	Destination End System
DFF	Drop From Front

DIR	Direction bit
DMRCA	Dynamic Max Rate Control Algorithm
DT	Drop Tail
EFCI	Explicit Forward Congestion Indication
EOM	End-Of-Message
EPD	Early Packet Discard
EPRCA	Enhanced Proportional Rate-Control Algorithm
ER	Explicit Rate
ERICA	Explicit Rate Indication for Congestion Avoidance
ESPD	Early Selective Packet Discard
FBA	Fair Buffer Allocation
FIFO	First In First Out
FSCD	Fuzzy Logic Selective Cell Drop
FMMRA	Fast Max-Min Rate Allocation
FRM	Forward RM
FRTT	Fixed Round Trip Time
FTP	File Transfer Protocol
GFC-2	Generic Fairness Configuration-2
IBT	Intrinsic Burst Tolerance
ICR	Initial Cell Rate
ID	Protocol ID byte
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU-T	International Telecommunication Union-Telecommunication
LAN	Local Area Network
LLC	Logical Link Control
M_{rm}	Minimum number of cells between RM-cell generation
MACR	Mean Allowed Cell Rate
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MIT	Massachusetts Institute of Technology
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
nrt-VBR	non-real time Variable Bit Rate
N_{rm}	Maximum number of cells between RM-cell generation
NI	No Increase
NIST	National Institute of Standards and Technology
PCR	Peak Cell Rate
PDU	Protocol Data Units
PPD	Partial Packet Discard

PTI	Payload Type Indicator
PVC	Permanent Virtual Circuit
QDLF	Queue Drain Limit Factor
QL	Queue Length bytes
QoS	Quality of Service
rt-VBR	real time Variable Bit Rate
RA	Request/Acknowledge
REM	Rate Envelope Multiplexing
RDF	Rate Decrease Factor
RIF	Rate Increase Factor
RM	Resource Management
RS	Rate Sharing
RTO	Retransmission Timeout
RTT	Round Trip Time
SAR	Segmentation and Reassembly Sublayer
SACK	Selective Acknowledgments
SCR	Sustainable Cell Rate
SD	Selective Drop
SES	Source End System
SN	Sequence Number bytes
SNAP	Subnetwork Attachment Point
SVC	Switched Virtual Circuit
T_{rm}	Upper bound on the time between RM-cell generation
TBE	Transient Buffer Exposure
TCP	Transmission Control Protocol
TCR	Tagged Cell Rate
UBR	Unspecified Bit Rate
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Identifier
VLSI	Very Large Scale Integration
VP	Virtual Path
VPI	Virtual Path Identifier
VS/VD	Virtual Source and Virtual Destination
WAN	Wide Area Network
WWW	World Wide Web

Conferences and Journal Abbreviations

APCC	IEEE Asia-Pacific Conference on Communications
ATRS	Australian Teletraffic Research Seminar
ATNAC	Australia Telecommunication Networks & Applications Conference
GLOBECOM	IEEE Global Telecommunication Conference
ICC	IEEE International Conference on Communications
ICCS	International Conference on Communication Systems
ICICS	International Conference on Information, Communications & Signal Processing
ICT	International Conference on Telecommunications
IEICE	The Institute of Electronics, Information and Communication Engineers Transactions on Communications
IEEE	Institute of the Electrical and Electronic Engineering
INFOCOM	The IEEE Conference on Computer Communications
ISPACS	International Symposium on Intelligent Signal Processing and Communication Systems
ITC	International Teletraffic Congress
JSAC	IEEE Journal on Selected Areas in Communications

References

[AA93] G. Armitage and K. Adams. Packet Reassembly during Cell Loss. *IEEE Network*, vol. 7(no. 5):pp. 26–34, Sep 1993.

[ACA96] A. Arulambalam, X. Chen, and N. Ansari. Allocating Fair Rates for Available Bit Rate Service in ATM Networks. *IEEE Communications Magazine*, vol. 34(no. 11):pp. 92–100, Nov 1996.

[ACA97] A. Arulambalam, X. Chen, and N. Ansari. An Intelligent Explicit Rate Control Algorithm for ABR Service in ATM Networks. *Proc. IEEE ICC'97, Montreal, Canada*, vol. 1:pp. 200–204, Jun 1997.

[Ada97] A. Adas. Traffic Models in Broadband Networks. *IEEE Communications Magazine*, vol. 35(no. 7):pp. 82–89, Jul 1997.

[ADG+00] M. Allman, S. Dawkins, D. Glover, J. Griner, D. Tran, and et la. Ongoing TCP Research Related to Satellites. *Internet RFC 2760*, Feb 2000.

[AK98] L. L.H. Andrew and S.-H. Kuan. Improving the Robustness of Fuzzy Logic ATM ABR Rate Control for Large Round Trip Times. *Proc. IEEE ISPACS'98*, vol. 1:pp. 188–192, Nov 1998.

[ATM96] The ATM Forum Technical Committee. *Traffic Management Specification Version 4.0, ATM Forum contribution AF-TM-0056.000*, 1996.

[Bar94] A. W. Barnhart. Use of the Extended PRCA with Various Switch Mechanisms. *ATM Forum contribution 94-0898*, 1994.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *Internet RFC 2475*, Dec 1998.

[BF95] F. Bonomi and K.W. Fendick. The Rate-based Flow Control Framework for the Available Bit Rate ATM Service. *IEEE Network*, vol. 9(no. 2):pp. 25–39, Mar-Apr 1995.

[BG87] D. Bertsekas and R. Gallager. *Data Networks, Second edition.* Englewood Cliffs, N.J. Prentice Hall, 1987.

[BJ94] J. Bennett and G. Tom Des Jardins. Comments on the July PRCA Rate Control Baseline. *ATM Forum contribution 94-0682*, Jul 1994.

[BLCT97] B. Bensaou, S. T. C. Lam, H.-W. Chu, and D. H. K. Tsang. Estimation of the Cell Loss Ratio in ATM Networks with a Fuzzy System and Application to Measurement-based Call Admission Control. *IEEE/ACM Transactions on Networking*, vol. 5(no. 4):pp. 572–584, Aug 1997.

[Bra89] R. Braden. Requirements for Internet Hosts – Communication Layers. *Internet RFC 1122*, Oct 1989.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP). *Internet RFC 2205*, Sep 1997.

[CAXC97] F. M. Chiussi, A. Arulambalam, Y. Xia, and X-Q. Chen. Explicit Rate ABR Schemes Using Traffic Load as Congestion Indicator. *Computer Communications and Networks. Proc. Sixth International Conference*, pages 76–84, 1997.

[CC96] R.-G. Cheng and C.-J. Chang. Design of a Fuzzy Traffic Controller for ATM Network. *Proc. IEEE/ACM Transactions on Networking*, vol. 4(no. 3):pp. 460–469, Jun 1996.

[CD95] G. Develekos C. Douligeris. A Fuzzy Logic Approach to Congestion Control in ATM Networks. *Proc. IEEE ICC'95*, vol. 3:pp. 1969–1973, Jun 1995.

[CD97] G. Develekos C. Douligeris. Neuro-fuzzy Control in ATM Networks. *IEEE Communications Magazine*, vol. 35(no. 5):pp. 154–162, May 1997.

[CFP96] V. Catania, G. Ficili, and D. Panno. Using Fuzzy Logic in ATM Source Traffic Control: Lessons and Perspectives. *IEEE Communications Magazine*, vol. 34(no. 11):pp. 70–81, Nov 1996.

[CH00] R. Cohen and Y. Hamo. Balanced Packet Discard for Improving TCP Performance in ATM Networks. *Proc. IEEE INFOCOM 2000*, vol. 3:pp. 1556–1565, 2000.

[Cha94] A. Charny. An Algorithm for Rate Allocation in a Cell-Switching Network with Feedback, Masters thesis. *Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, MIT TR-601, Cambridge, MA*, May 1994.

[CP98] K. Cheon and S. Panwar. On the Performance of ATM-UBR with Early Selective packet Discard . *Proc. IEEE ICC'98*, vol. 1:pp. 221–227, 1998.

[CSZ92] D. D. Clark, S. J. Shenker, and L. Zhang. Supporting Real-time Applications in an Integrated Service Packet Networks: Architecture and Mechanism. *Proc. ACM SIGCOMM'92*, pages 14–26, 1992.

[CT98] H. Chiou and Z. Tsai. Performance of ATM Switches with Age Priority Packet Discarding under the ON-OFF Source Model. *Proc. IEEE INFOCOM'98*, vol. 3:pp. 931–938, 1998.

[CXK96] F. M. Chiussi, Y. Xia, and V. P. Kumar. Dynamic Max Rate Control Algorithm for Available Bit Rate Service in ATM Networks. *Proc. IEEE GLOBECOM'96*, vol. 3:pp. 2108–2117, Nov 1996.

[DJM97] Z. Dziong, M. Juda, and L. Mason. A Framework for Bandwidth Management in ATM Networks — Aggregate Equivalent Bandwidth Estimation Approach. *Proc. IEEE/ACM Transactions on Networking*, vol. 5(no. 1):pp. 134–147, Feb 1997.

[dlCAM97] L. de la Cruz, J. J. Alins, and J. Mata. Prediction Techniques for VBR MPEG Traffic Shaping. *Proc. IEEE GLOBECOM'97*, vol. 3:pp. 1434–1439, Nov 1997.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, vol. 1(no. 4):pp. 397–413, Aug 1993.

[FJR+99] S. Fahmy, R. Jain, S. Rabie, R. Goyal, and B. Vandalore. Quality of Service for Internet Traffic over ATM Service Categories. *Journal of Computer Communications*, vol. 22(no. 14):pp. 1307–1320, Sep 1999.

[FL95] C. Fang and A. Lin. On TCP Performance of UBR with EPD and UBR-EPD with Fair Buffer Allocation Scheme. *ATM Forum Contribution 95-1645*, Dec 1995.

[FL97] C. Fang and A. Lin. TCP Performance in ATM Networks: ABR Parameter Tuning and ABR/UBR Comparisons. *Proc. IEEE Singapore International Conference on Networks*, pages 117–139, 1997.

[Flo01] S. Floyd. A Report on Recent Developments in TCP Congestion Control. *IEEE Communications Magazine*, vol. 39(no. 4):pp. 84–90, Apr 2001.

[FP00] J. Farserotu and R. Prasad. A Survey of Future Broadband Multimedia Satellite Systems, Issues and Trends . *IEEE Communications Magazine*, vol. 38(no. 6):pp. 128–133, Jun 2000.

[FR94] S. Floyd and A. Romanow. Dynamics of TCP traffic over ATM Networks. *Proc. ACM SIGCOMM'94*, pages 79–88, Sep 1994.

[GAN91] R. Guérin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas in Communications*, vol. 9(no. 7):pp. 968–981, Sep 1991.

[GH91] R. J. Gibbens and P. J. Hunt. Effective Bandwidths for the Multi-type UAS Channel. *Queuing Systems*, vol. 9:pp. 17–27, 1991.

[GJK+97] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and S.-C. Kim. UBR+: Improving Performance of TCP over ATM-UBR Service. *Proc. IEEE ICC'97*, vol. 2:pp. 1042–1048, Jun 1997.

[GJK+98] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and B. Vandalore. Improving the Performance of TCP over the ATM-UBR service. *Computer Communications*, vol. 21(no. 10):pp. 898–911, Jul 1998.

[GKK95] R. J. Gibbens, F. P. Kelly, and P. B. Key. A Decision-theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, vol. 13(no. 6):pp. 1101–1114, Aug 1995.

[GMH+98] N. Golmie, F. Mouveaux, L. Hester, Y. Saintillan, A. Koenig, and D. Su. The NIST ATM/HFC Network Simulator Version 4.0. *URL "http://w3.antd.nist.gov/Hsntg/prd_atm-sim.html"*, Dec 1998.

[Goy99] R. Goyal. Traffic Management for TCP/IP over Asynchronous Transfer Mode (ATM) Networks. *PhD Thesis*, The Ohio State University, 1999.

[GT99] M. Grossglauser and D. N. C. Tse. A Framework for Robust Measurement-based Admission Control. *IEEE/ACM Transactions on Networking*, vol. 7(no. 3):pp. 293–309, Jun 1999.

[HA00] M. Hassan and M. Atiquzzaman. *Performance of TCP/IP over ATM networks.* Boston, MA : Artech House, 2000.

[HAOS01] D. P. Hong, C. Albuquerque, C. Oliveira, and T. Suda. Evaluating the Impact of Emerging Streaming Media Applications on TCP/IP Performance. *IEEE Communications Magazine*, vol. 39(no. 4):pp. 76–82, Apr 2001.

[Has97] M. Hassan. Impact of Variable Bandwidth on the Performance of Data Communications Over ATM-ABR-based LAN Interconnection. *Proc. 20th Australasian Computer Science Conference, Sydney*, pages 422–429, 1997.

[Hei93] J. Heinanen. Multiprotocol Encapsulation over ATM Adaptation Layer 5. *Internet RFC 1483*, Jul 1993.

[Hir90] A. Hiramatsu. ATM Communications Network Control by Neural Networks. *IEEE Transactions on Neural Networks*, vol. 1(no. 1):pp. 122–130, Mar 1990.

[HK] J. Heinanen and K. Kilkki. A Fair Buffer allocation Scheme. *Unpublished Manuscript.*

[HLP95] X.Y. Hu, J. F. Lambert, and A. Pitsillides. Fast Backward Predictive Congestion Notification for ATM Networks with Significant Propagation Delays. *Proc. IEEE GLOBECOM'95*, vol. 1:pp. 275–279, Nov 1995.

[HR94] A. Heybey and N. Robertson. The Network Simulator Version 3.1. 1994.

[HWMB94] J. Hullett, T. Wong, G. Mercankosk, and Z. Budrikis. Packet Discard Strategy for Congestion Control of ATM Data Networks. *Australian Telecommunication Networks and Applications Conference*, vol. 2:pp. 633–638, Dec 1994.

[ITU93] ITU-T Recommendation I.363. *B-ISDN ATM Adaptation Layer (AAL) Specification*, 1993.

[ITU96] ITU-T Recommendation I.371. *Traffic Control and Congestion Control in B-ISDN*, 1996.

[Jac88] V. Jacobson. Congestion Avoidance and Control. *Proc. ACM SIGCOMM'88*, pages 314–329, Aug 1988.

[Jac90] V. Jacobson. Modified TCP Congestion Avoidance Algorithm. *Message to end2end-interest mailing list*, URL "ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt", Apr 1990.

[Jaf81] J. M. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, vol. 29:pp. 954–962, Jul 1981.

[Jai96] R. Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey. *Computer Networks and ISDN Systems*, Oct 1996.

[Jan98] J.-S. R. Jang. *Fuzzy Logic Toolbox, URL "http://www.mathworks.com/products/fuzzylogic/".* The MathWorks, Inc., 1998.

[JB97] R. Jain and G. Babic. Performance Testing Effort at the ATM Forum: an Overview. *IEEE Communications Magazine*, vol. 35(no. 8):pp. 110–116, Aug 1997.

[JDSZ97] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang. A Measurement-based Admission Control Algorithm for Integrated Service Packet Networks. *IEEE/ACM Transactions on Networking*, vol. 5(no. 1):pp. 56–69, Feb 1997.

[JKG+96] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan. ERICA Switch Algorithm: A Complete Description. *ATM Forum contribution 96-1172*, Aug 1996.

[JM95] S. Jung and J. S. Meditch. Adaptive Prediction and Smoothing of MPEG Video in ATM Networks. *Proc. IEEE GLOBECOM'95*, vol. 2:pp. 832–836, Nov 1995.

[JT00] S. Jagannathan and J. Talluri. Predictive Congestion Control of ATM Networks: Multiple Sources/Single Buffer Scenario. *Proc. IEEE Conference on Decision and Control*, vol. 1:pp. 47–52, 2000.

[JY96] S. Jagannath and N. Yin. End-to-end TCP Performance in IP/ATM Internetworks. *ATM Forum Contribution 96-1711*, Dec 1996.

[KA98] S.-H. Kuan and L. L.H. Andrew. Performance of Fuzzy Logic ABR Rate Control with Large Round Trip Times. *Proc. IEEE GLOBECOM'98*, vol. 4:pp. 2464–2468, Nov 1998.

[Kal97] S. Kalyanaraman. Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks. *PhD Thesis*, The Ohio State University 1997.

[Kea96] S. Kalyanaraman and R. Jain et al. Virtual Source/Virtual Destination (VS/VD): Design Considerations. *ATM Forum contribution 96-1759*, Dec 1996.

[Kel91] F. P. Kelly. Effective Bandwidths at Multi-class Queues. *Queuing Systems*, vol. 9:pp. 5–15, 1991.

[KK92] P. S. Khedkar and S. Keshav. Fuzzy Prediction of Timeseries. *Proc. International Conference on Fuzzy Systems, San Diego, USA*, pages 281–288, Mar 1992.

[KS94] S. Keung and K.-Y. Siu. Degradation in TCP Performance under Cell Loss. *ATM Forum Contribution 94-0490*, Apr 1994.

[Lau94] M. Laubach. Classical IP and ARP over ATM. *Internet RFC 1577*, Jan 1994.

[LB99] M. A. Labrador and S. Banerjee. Packet Dropping Policies for ATM and IP Networks. *IEEE Communications Surveys (URL http://www.comsoc.org/surveys)*, vol. 2(no. 3):pp. 2–14, third quarter 1999.

[LD97] Y.-C. Liu and C. Douligeris. Rate Regulation with Feedback Controller in ATM Networks — a Neural Network Approach. *Proc. IEEE Journal on Selected Areas in Communications*, vol. 1(no. 2):pp. 832–836, Feb 1997.

[LF95] A. Lin and C. Fang. A Simulation Study of ABR Robustness under Binary Switch Modes. *ATM Forum contribution 95-1019*, Aug 1995.

[LH98] M. Laubach and J. Halpern. Classical IP and ARP over ATM. *Internet RFC 2225*, Apr 1998.

[LH00] S.-J. Lee and C.-L. Hou. A Neural-fuzzy System for Congestion Control in ATM Networks . *IEEE Transactions on Systems, Man and Cybernetics*, vol. 30(no. 1):pp. 2–9, Feb 2000.

[LMR97] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan. Transporting Compressed Video over ATM Networks with ABR Feedback Control. *ATM Forum contribution 97-0652*, Jul 1997.

[LNO96] T. V. Lakshman, A. Neidhardt, and T. Ott. The Drop from Front Strategy in TCP and in TCP over ATM. *Proc. IEEE INFOCOM'96*, vol. 3:pp. 1242–1250, 1996.

[LO97] H.-P. Lin and Y.-C. Ouyang. Neural Networks Based Traffic Prediction for Cell Discarding Policy. *International Conference on Neural Networks*, vol. 4:pp. 2051–2056, 1997.

[LST96] H. Li, K.-Y. Siu, and H.-Y. Tzeng. A Simulation Study of TCP Performance in ATM Networks with ABR and UBR Service. *Proc. IEEE INFOCOM' 96*, vol. 3:pp. 1269–1276, 1996.

[MA75] E. H. Mamdani and S. Assilian. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, vol. 7:pp. 1–13, 1975.

[MAS+88] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins. Performance Models of Statistical Multiplexing in Packet Video Communications. *IEEE Transactions on Communications*, vol. 36(no. 7):pp. 834–844, Jul 1988.

[MLN97] H. R. Mehrvar and T. Le-Ngoc. Fuzzy Logic in Estimation of Traffic Burstiness for Admission Control in Broadband Networks. *Proc. IEEE ICC'97*, vol. 2:pp. 1090–1094, 1997.

[MLNH96] H. R. Mehrvar, T. Le-Ngoc, and J. Huang. Performance Evaluation of Bursty Traffic using Neural Networks. *Canadian Conference on Electrical and Computer Engineering*, vol. 2:pp. 955–958, 1996.

[MMFR96] M. Mathis, J. Madhavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. *Internet RFC 2018*, Oct 1996.

[MN68] B. B. Mandelbrot and J. W. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, vol. 10:pp. 422–437, 1968.

[MPS96] J. Mata, G. Pagan, and S. Sallent. Multiplexing and Resource Allocation of VBR MPEG Video Traffic on ATM Networks. *Proc. IEEE ICC'96*, vol. 3:pp. 1401–1405, 1996.

[NAZH95] T. D. Neame, R. G. Addie, M. Zukerman, and F. Huebner. Investigation of Traffic Models for High Speed Data Networks. *Proc. ATNAC'95*, Dec 1995.

[PcR97] A. Pitsillides, Y. A. Şekercioğlu, and G. Ramamurthy. Effective Control of Traffic Flow in ATM Networks Using Fuzzy Explicit Rate Marking (FERM). *IEEE Journal on Selected Areas in Communications*, vol. 15(no. 2):pp. 209–225, Feb 1997.

[PE96] H. G. Perros and K. M. Elsayed. Call Admission Control Schemes: A Review. *IEEE Communications Magazine*, vol. 34(no. 11):pp. 82–91, Nov 1996.

[Pos81] J. Postel. Transmission Control Protocol — DARPA Internet Program Protocol Specification. *Internet RFC 793*, Sep 1981.

[Qiu97] B. Qiu. A Predictive Fuzzy Logic Congestion Avoidance Scheme. *Proc. IEEE GLOBECOM'97, Arizona, USA*, vol. 2:pp. 967–971, Nov 1997.

[Qiu98] B. Qiu. The Application of Fuzzy Prediction for the Improvement of QoS Performance. *Proc. IEEE ICC'98, Atlanta, Georgia, USA*, vol. 3:pp. 1769–1773, Jun 1998.

[Qiu01] B. Qiu. Modelling and Prediction of Hybrid-coded Variable Bit Rate Video Sources in Fuzzy Logic Perspectives. *Journal of Circuits, Systems Signal Processing*, vol. 20(no. 3):pp. 403–414, Jun 2001.

[QSW99] B. Qiu, S. Suthaharan, and H. R. Wu. Neuro-fuzzy Modeling and Prediction of VBR MPEG Video Sources. *Journal of Real-Time Imaging*, vol. 5:pp. 359–363, 1999.

[RF95] A. Romanov and S. Floyd. Dynamics of TCP Traffic over ATM Networks. *IEEE Journal on Selected Area in Communications*, vol. 13(no. 4):pp. 633–641, May 1995.

[RF99] K. Ramakrishnan and S. Floyd. A Proposal to Add Explicit Congestion Notification (ECN) to IP. *Internet RFC 2481*, Jan 1999.

[RMV96] J. Roberts, U. Mocci, and J. Virtamo. *Broadband Network Teletraffic.* Berlin, Heidelberg, Springer, 1996.

[Rob94] L. Roberts. Enhanced PRCA (Proportional Rate-Control Algorithm). *ATM Forum contribution 94-0735R1*, Aug 1994.

[RR95] G. Ramamurthy and Q. Ren. Analysis of the Adaptive Rate Control for ABR Service in ATM Networks. *Proc. IEEE GLOBECOM'95*, vol. 2:pp. 1083–1088, Nov 1995.

[RR00] Q. Ren and G. Ramamurthy. A Real-Time Dynamic Connection Admission Controller Based On Traffic Modeling, Measurement, and Fuzzy Logic Control. *IEEE Journal on Selected Areas in Communications*, vol. 18(no. 2):pp. 184–196, Feb 2000.

[RVC01] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. *Internet RFC 3031*, Jan 2001.

[SCZ93] S. J. Shenker, D. D. Clark, and L. Zhang. A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network. *URL ftp://parcftp.parc.xerox.com/pub/net-research/archfin.ps*, 1993.

[SHL98] H. Sun, K. Huang, and L. Li. Supporting IP on the ATM Networks: An Overview. *Computer Communications*, vol. 21(no. 11):pp. 1020–1029, Aug 1998.

[Sim94] R. J. Simcoe. Test Configurations for Fairness and Other Tests. *ATM Forum contribution 94-0557*, Jul 1994.

[SKK+96] H. Saito, K. Kawashima, H. Kitazume, A. Koike, M. Ishizuka, and A. Abe. Performance Issues in Public ABR Service. *IEEE Communications Magazine*, vol. 34(no. 11):pp. 40–48, Nov 1996.

[SS91] H. Saito and K. Shiomoto. Dynamic Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, vol. 9(no. 7):pp. 982–989, Sep 1991.

[SSZ97] E. Sanchez, T. Shibata, and L. A. Zadeh. *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives.* River Edge, NJ: World Scientific Pub., Singapore, 1997.

[Ste94] W. Stevens. *TCP/IP Illustrated*, volume 1. Addison-Wesley, 1994.

[Ste97] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. *Internet RFC 2001*, Jan 1997.

[Sug85] M. Sugeno. *Industrial Applications of Fuzzy Control.* Elsevier Science Pub. Co., 1985.

[SYT99] K. Shiomoto, N. Yamanaka, and T. Takahashi. Overview of Measurement-based Connection Admission Control Methods in ATM Networks. *IEEE Communications Surveys*, pages 2–13, First Quarter 1999.

[TQ98] J. J. Tirtowidjojo and B. Qiu. An Improved Fair Share Calculation and Updating Scheme for ABR Service. *Proc. IEEE ISPACS'98*, vol. 2:pp. 933–938, Nov 1998.

[UH97] K. Uehara and K. Hirota. Fuzzy Connection Admission Control for ATM Networks Based on Possibility Distribution of Cell Loss Ratio. *IEEE Journal on Selected Areas in Communications*, vol. 15(no. 2):pp. 179–190, Feb 1997.

[Van99] B. Vandalore. Inverse Hyperbolic Queue Control Function. *Private Email Communication*, May 1999.

[VJGF98] B. Vandalore, R. Jain, R. Goyal, and S. Fahmy. Dynamic Queue Control Functions for ATM ABR Switch Schemes: Design and Analysis. *Computer Networks and ISDN Systems*, Nov 1998.

[Vos99] M. D. Vose. *The Simple Genetic Algorithm: Foundations and Theory.* The MIT Press, Cambridge, 1999.

[WCS95] J.-L. C. Wu, Y.-W. Chen, and C.-C. Shiu. Traffic Modeling and Bandwidth Allocation for MPEG Video Sources in ATM Networks. *Proc. IEEE GLOBECOM'95*, vol. 3:pp. 2237–2241, Nov 1995.

[WK90] G. M. Woodruff and R. Kositpaiboon. Multimedia Traffic Management Principles for Guaranteed ATM Network Performance . *IEEE Journal on Selected Areas in Communications*, vol. 8(no. 3):pp. 437–446, Apr 1990.

[Yin94] N. Yin. Fairness Definition in ABR Service Model. *ATM Forum contribution 94-0928R2*, Sep 1994.

[Yin95] N. Yin. Analysis of a Rate-based Traffic Management Mechanism for ABR Service. *Proc. IEEE GLOBECOM'95*, vol. 2:pp. 1076–1082, Nov 1995.

[Zad65] L. A. Zadeh. Fuzzy Sets. *Information and Control 8*, pages 338–353, 1965.

[Zad73] L. A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-3:pp. 28–44, Jan 1973.

[Zad75] L. A. Zadeh. The Concept of a Linguistic Variable and its Application to Approximate Reasoning. *Information Sciences 8*, 1975.

[ZL98a] M. Zukerman and T. K. Lee. A Framework for Real-time Measurement-based Connection Admission Control in Multi-service Networks. *Proc. IEEE GLOBECOM'98*, vol. 5:pp. 2983–2988, 1998.

[ZL98b] M. Zukerman and T. K. Lee. A Measurement Based Connection Admission Control for ATM Networks. *Proc. IEEE ICATM'98*, pages 140–144, 1998.