Traffic Management to Enhance Quality of Service (QoS) of Multimedia over Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Bobby Vandalore, B.Tech., M.S.

* * * * *

The Ohio State University

2000

Dissertation Committee:

Professor Raj Jain, Adviser

Professor Stanley Ahalt

Professor Wu-chi Feng

Approved by

Adviser

Department of Computer and Information Science

ABSTRACT

The Internet is growing exponentially and is impacting every aspect of modern life. Voice and video-based multimedia applications are expected to become a significant portion of the World Wide Web, but support for multimedia applications in the current Internet is still in its infancy. Though Internet Engineering Task Force (IETF) is developing technologies such as integrated services, differentiated services, and multiprotocol label switching, to meet the demands of multimedia applications; these technologies are still under development and not yet widely deployed.

Asynchronous Transfer Mode (ATM) is a high speed networking technology that provides seamless support for voice, video and data applications. ATM is being widely deployed in campuses and carrier backbones. To accommodate heterogeneity and to achieve efficiency we expect that future multimedia applications will be adaptive. We propose to use the ATM's ABR (available bit rate) service, since it has minimum rate guarantees and closed loop feedback control that minimizes cell loss.

We design and analyze traffic management methods to enhance the quality-of-service for multimedia applications over ABR service. Multimedia applications require that bandwidth, delay and loss are guaranteed in a certain range. We develop a rate allocation switch algorithm and a general form of fairness to enhance the bandwidth capability. We prove by simulation and analysis that the scheme developed

converges to the fair allocation in various network topologies. We design three additional rate allocation schemes based on an overload factor. A comparison of these algorithm is given using simulation results and analysis. End-to-end delay has various components such as propagation delay, processing delay, and queuing delay. The varying component of delay is queuing delay. The queuing delay is reduced, by using an appropriate queue control function. We develop several queue control functions that can be used to dynamically control the queuing delay. Simulation and analysis is done to identify the best queue control function. We have built a software-based testbed to evaluate and demonstrate the various methods developed in this thesis.

Most of methods developed to solve the problems are general, and can be applied to design methods to support adaptable multimedia applications of the future Internet.

Traffic Management to Enhance Quality of Service (QoS) of Multimedia over Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks

By

Bobby Vandalore, Ph.D.

The Ohio State University, 2000

Professor Raj Jain, Adviser

The Internet is growing exponentially and is impacting every aspect of modern life. Voice and video-based multimedia applications are expected to become a significant portion of the World Wide Web, but support for multimedia applications in the current Internet is still in its infancy. Though Internet Engineering Task Force (IETF) is developing technologies such as integrated services, differentiated services, and multiprotocol label switching, to meet the demands of multimedia applications; these technologies are still under development and not yet widely deployed.

Asynchronous Transfer Mode (ATM) is a high speed networking technology that provides seamless support for voice, video and data applications. ATM is being widely deployed in campuses and carrier backbones. To accommodate heterogeneity and to achieve efficiency we expect that future multimedia applications will be adaptive. We

propose to use the ATM's ABR (available bit rate) service, since it has minimum rate guarantees and closed loop feedback control that minimizes cell loss.

We design and analyze traffic management methods to enhance the quality-of-service for multimedia applications over ABR service. Multimedia applications require that bandwidth, delay and loss are guaranteed in a certain range. We develop a rate allocation switch algorithm and a general form of fairness to enhance the bandwidth capability. We prove by simulation and analysis that the scheme developed converges to the fair allocation in various network topologies. We design three additional rate allocation schemes based on an overload factor. A comparison of these algorithm is given using simulation results and analysis. End-to-end delay has various components such as propagation delay, processing delay, and queuing delay. The varying component of delay is queuing delay. The queuing delay is reduced, by using an appropriate queue control function. We develop several queue control functions that can be used to dynamically control the queuing delay. Simulation and analysis is done to identify the best queue control function. We have built a software-based testbed to evaluate and demonstrate the various methods developed in this thesis.

Most of methods developed to solve the problems are general, and can be applied to design methods to support adaptable multimedia applications of the future Internet.

To my grandmother, Janaki Bai,

my parents, Shylaja and Rajaraman,

and my sister Uma

ACKNOWLEDGMENTS

At the end of my academic life, as I look back into the past five years I can count a number of people, incidents, and places that have made it possible for me to arrive at this stage. Each one of them has made an impact on me and it would have been impossible for me to accomplish this dissertation without them. In the following few paragraphs, I have tried to list them by taking a journey back through my memory lanes. I do realize that there may be omissions due to my own lapses because of the overwhelmed and dazed state I am in now.

First of all, I would like to express my heartfelt gratitude for my advisor, Professor Raj Jain. He was kind enough to take me as his student five years back. He gave me encouragement to continue my PhD after finishing my Master's degree. He has been of immense help during this period both technically and at the personal level. He has not only guided me in the research career but also influenced the choice of my future career.

I thank all the Professors who served as committee members. Specifically, I would like to thank Dr. Stanley Ahalt and Dr. Wu-chi Feng for serving my dissertation committee. I also had opportunity to work closely with both Stanley Ahalt in the video project and Wu-chi Feng in the area of multimedia. Discussions conducted with them, their technical expertise, and feedback (especially from Wu-chi Feng) has

immensely helped me in writing this dissertation. I would also like to thank Professors Anish Arora and Steve Lai for serving my candidacy exam committee.

During my stay at OSU, I had opportunities to explore not only my field of research but other areas of computer science as well. I would like to thank Professors Rephael Wenger and Ken Supowit for teaching me the aspects of theoretical computer science. Professors Roni Yagel, Rick Parent, and Kikuo Fujimura taught me the concepts of graphics which is my other area of interest.

I owe my gratitude to my colleagues at *Netlab* for making it an inspiring place to do research. Shivkumar, Rohit, and Mukul made *Netlab* an enjoyable place to work. I would also like to thank the other members of the *Netlab* family namely, Chunlei, Wei Sun, and Sohail. Murali, Arvind, and Sitaraman helped me a lot in the software switch project. Sonia deserves a special mention, her contribution and feedback towards my research is invaluable.

Colleagues in the networking community have been very helpful. I would like to thank Professor Yuan Zheng, Dr. Sudhir Dixit, Sastri Kota, for mentoring me. I also thank the ATM Forum community for providing valuable feedback on my work.

My stay at OSU was made pleasant because of various staff members of the CIS department. I thank Elley who was helpful during my initial years when I worked as a teaching assistant. I also thank Tom, Marty, James, Deanne, Sandy, Elizabeth, and Mark for taking care of University red-tape issues and various needs of a graduate student.

I am very grateful for the generous financial support that I received for funding my studies. These sources include CIS department teaching assistantship, research associateship, National Science Foundation, and Rome Air Force Laboratories.

One of the places which helped me pull through the graduate school was the 8th floor lounge, in which I have memories of discussing topics ranging from research ideas to movies, and having regular lonely late night coffee to keep me awake. Another place I will remember is Dreese Lab 274, cubicle 15, my office for the past five years. Other places that made my stay in Columbus enjoyable include Scioto River Valley, nearby Hocking Hills, and Lennox theater complex. My yearly journey to cities of Chennai and Bangalore in India to visit family and friends, were always relaxing and enjoyable. They gave me the strength I needed for pursuing graduate studies.

Graduate school was not always hard work, I was fortunate enough to enjoy other aspects of life also. I will remember the interesting pool games, my bowling night goof ups, boisterous football Saturdays, tiring racquet-ball games, the yearly CIS department's welcome party, learning violin, and social dance experiences.

Friends made it possible to forget at times that I was in graduate school. I would like to thank all of them for providing me with fond memories, for tolerating me, and accepting me among them. A non-exhaustive list of these friends include Gopal, Shaji, Viswananth, Kram, Sri, Sudhish, Sandeep, Praveen, Raja, Anil, Manpreet, Rakesh, Binoy, Bijoy, Rajesh, Mohammed, Jayaram, Rashmi, Swapna, Sujatha, Bama, Sumathi, and Gayathri. Bindu and Meena take a special place among friends. They helped me a lot by listening to me, by their humor, and by being there for me.

I express my heartfelt gratitude towards my family members who have been supportive throughout my life. My family members including my grandmother Janaki, my parents Shylaja and Rajaraman, my sister Uma, encouraged me to throughout the graduate school years. I am at this stage only because of my parents' love and their immense belief in me. I also thank the members of my extended family including,

Devaraj, Divya, Bharat, Mamtha, Sheela, Preethi, Rohit, Nisha, Swetha, and Sneha. Lastly, I would like to thank Prasanthi, my fiancee, for expediting things! She has asked me to finish it fast so that we could get married soon.

VITA

1970	Born, Chennai (Madras), India
1993	B.Tech., Computer Science, Indian Institute of Technology, Chennai, India
1993-1994	Project Associate, Distributed Systems and Optical Networks Laboratory (DONlab), Indian Institute of Technology, Chennai, India
1995	M.S., Computer and Information Science, The Ohio State University, Columbus, Ohio, USA
1994-present	Graduate Teaching Associate, Graduate Research Associate, and System Administrator, The Ohio State University, Columbus, Ohio, USA

PUBLICATIONS

Research Publications

B. Vandalore, R. Jain, R. Goyal, S. Fahmy. Dynamic Queue Control Functions for ATM ABR Switch Schemes: Design and Analysis. *Journal of Computer Networks*, August 1999, Vol. 31, Issue 18, pp. 1935-49.

B. Vandalore, S. Fahmy, R. Jain, R. Goyal and M. Goyal. QoS and Multipoint support for Multimedia Applications over ATM ABR service *IEEE Communications Magazine*, January 1999, pp. 53-57.

B. Vandalore, S. Fahmy, R. Jain, R. Goyal, and M. Goyal. General Weighted Fairness and its Support in Explicit Rate Switch Algorithms *Journal of Computer Communications*, January 2000, Vol. 23, Issue 2, pp. 149-161.

B. Vandalore, W. Feng, R. Jain, and S. Fahmy. A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia *To appear in Journal of Real Time Systems (Special Issue on Adaptive Multimedia)*, January 2000.

S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore. The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks *Accepted in IEEE/ACM Transactions on Networking*, February 2000.

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*, 1999, Vol. 22, Issue 14, pp. 1307-1320.

S. Fahmy, R. Jain, R. Goyal, B. Vandalore, and S. Kalyanaraman. Design and Evaluation of Feedback Consolidation for ABR Point-to-Multipoint Connections in ATM Networks *Journal of Computer Communications*, 1999. Vol. 22, Issue 12, pp. 1085-1103.

R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and B. Vandalore. Improving the Performance of TCP over the ATM-UBR service , Journal of Computer Communications, volume 21, number 10, pp. 898-911, July 1998.

R. Goyal, R. Jain, S. Kota, M. Goyal, S. Fahmy, and B. Vandalore. Traffic Management for TCP/IP over Satellite-ATM Networks *IEEE Communications Magazine*, March 1999.

FIELDS OF STUDY

Major Field: Computer and Information Science

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Internet is growing exponentially and is impacting every aspect of modern life. The amount of information exchanged through e-mail sent over the Internet is easily greater than the amount transmitted over any other traditional medium of communication. The Internet has grown out of its academic roots and is now strongly entrenched in our every day lives. The Internet has led to an economic boom that has resulted in the record long economic expansion in United States. It has enabled businesses to attain decreased inventory levels, increased productivity, reduced product development time, and extended their reach to global customer base. Several developing nations are aggressively building their Internet infrastructure to bridge their economic gap from developed nations. But even with the tremendous growth of the Internet and the vast amount of research focused on it, there is a more to be accomplished. Though one can compare, the Internet to other modern inventions such as the telephone and television, it has certain unique characteristics. The most important characteristic is that it is a technology that is evolving all the time to support increasingly richer set of features. One can take the example of e-mail, it has evolved from being a simple text based communication to one that supports

media rich components through its Multipurpose Internet Mail Extensions (MIME) capability [52].

1.1 Evolutionary Nature of Internet

One of areas of the Internet which is experiencing explosive growth is the World-Wide-Web (WWW). The web supports several kind of traffic including, data, rich media, and e-commerce. Voice and video based multimedia applications are expected to become a significant portion of the Web. But support for multimedia applications in the current Internet is still in its infancy. In corporate intranets where bandwidth is reasonably abundant, ranging from 380 Kbps to 100 Mbps, multimedia applications such as video conferencing can be easily supported. But in public networks such as the Internet, the quality-of-service (QoS) achieved by multimedia applications is much smaller than in intranets. With the Internet Engineering Task Force (IETF) working to solve the QoS problem, we expect the support for multimedia applications to improve considerably in coming years. We believe (video-based) interactive multimedia applications will be the next killer application after Voice over IP (Internet Protocol). With the advent of optical networks, which increases bandwidth dramatically, increasing processing speeds, and advances in compression technology, networked multimedia applications of the future may be able to support HDTV (high definition television) quality of video.

Due to the evolutionary nature of both the applications and the Internet, several issues arise including those dealing with guaranteeing quality of service, pricing, and security. Recommendations and protocols to address these problems are being currently explored by the various working groups of the IETF. Currently, only best-effort

traffic is widely deployed in the Internet. So, one of the critical problem among these is the problem of providing guaranteed quality of service. To overcome this shortcoming IETF is developing service models and network protocols such as the next generation Internet Protocol (IPv6), integrated services, Resource Reservation Protocol (RSVP), Real-Time Transport Protocol (RTP), Differentiated Services (diffserv), and Multi-Protocol Label Switching (MPLS).

Another aspect of the Internet that continues to evolve is its bandwidth capability and support of traffic at various service qualities. Access bandwidth of the Internet is expanding with the deployment of Digital Subscriber Link (DSL) and cable modems technologies. Fiber optic technology is being used to increase the backbone bandwidth. Fiber technology is expected to further increase the bandwidth available at the edges of the Internet. Broadband access of the Internet is expected grow from its current 5% to around 50% by 2004. With the advent of high speed networks and their increased bandwidth capability, user expectation has dramatically increased. Enhancing the quality of service is one way of meeting the increasing expectations of the user. Though IETF is developing technologies such as differentiated services, and MPLS, to meet these demands, these technologies are still under development and not yet widely deployed. It is therefore difficult to study methods on how to provide enhanced quality of service in the current Internet infrastructure.

Asynchronous transfer mode (ATM) is a cell switching, connection oriented, high speed technology. ATM technology has rich support of quality of service. It is already being widely deployed in the backbones of carrier and campus networks. We believe that by the developing methods for enhancing quality of service in ATM networks we can extend these methods to improve quality of service of the future Internet.

We now give a brief overview of the various service categories and components used in ATM technology before discussing the problem domains that are addressed in this thesis. The ATM Forum has defined five service categories: constant-bit-rate (CBR), real-time variable bit rate (rt-VBR), non real-time VBR (nrt-VBR), available bit rate (ABR) and unspecified bit rate (UBR). The International telecommunication union (ITU-T) defines similar service categories for ATM. CBR and rt-VBR provide delay and loss guarantees and can be used to transfer delay and loss sensitive, unadaptive, multimedia applications. Nrt-VBR provides loss guarantees such as percentage cell loss. ABR and UBR are usually used by applications to transfer time-insensitive data. UBR is a simple service that gives no guarantees. The ABR service category has the attractive features of minimum rate guarantees and closed loop feedback control mechanism that minimizes networks queues and cell loss.

1.2 Components of ATM Networks

The components of the ATM networks are shown in Figure 1.1. The main components used to support multimedia applications are the following:

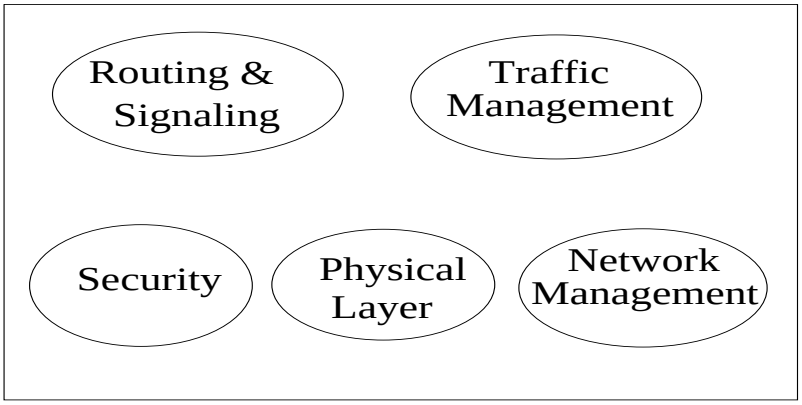


Figure 1.1: Components for supporting multimedia applications

- **Routing and Signaling:** The private network-to-network interface (PNNI) protocol provides routing and signaling capabilities between ATM switches. The current PNNI protocol (version 1.0) supports anycast (connection to one among a group) and quality of service based routing. The user-to-network interface signaling currently supports point-to-point connections, point-to-multipoint connections, and leaf initiated joins.

- **Traffic Management:** Traffic management includes connection admission control (CAC), usage parameter control (UPC), and congestion control. CAC is used to decide whether a connection request can be accepted or rejected, and to decide the parameters associated with that connection. UPC ensures the sources respect the traffic contract by monitoring and controlling traffic at the entrance to the network. It detects violations and takes appropriate actions (such as cell tagging and cell discard). It protects the network and the QoS of other connections from misbehaving connections. For the ABR service, feedback and flow control are used to regulate traffic and adaptively share the available bandwidth in a fair manner.

1.3 Motivation: Why Traffic Management of ABR for Multimedia?

We have chosen to develop the traffic management methods to enhance the quality of service of adaptive multimedia over the ATM ABR service category for the following reasons:

- The Internet is a heterogeneous environment connecting various networking technologies. Even with networking support through service classes, the available network resources to a multimedia applications will change dynamically over time. For example, the network conditions may change due to difference in link speeds (ranging from 28.8 Kbps modem links to 622 Mbps OC-12 links) or variability in a wireless environment caused by interference and mobility. One way of achieving the desired quality of service in such situations is by massively over-provisioning resources for applications. This solution, however, leads to inefficiency. Without over-provisioning, network resources can be used efficiently if applications are capable of adapting to changing network conditions.

  One way to support adaptability at the network is to use a feedback mechanism. Recently IETF has standardized a binary feedback mechanism known as explicit congestion notification (ECN) in the IP layer, to indicate congestion status of the network [96]. We believe that future multimedia applications and the Internet will be adaptive in their nature for efficiency and to support heterogeneity. Therefore, we chose the ABR service category because it provides a closed loop feedback mechanism.

- The ABR service provides minimum rate guarantees which can be used by the multimedia applications to achieve a minimum quality of service. Cell loss is low in ABR service since it uses closed loop feedback control to throttle sources in the event of congestion.

Traffic management is a challenging and a complex problem. We believe that the traffic management component is critical in providing QoS guarantees for multimedia. Hence, we focus on the *traffic management issues for supporting delay sensitive, loss sensitive, adaptive multimedia applications*. Though routing and signaling components are necessary we do not address them here. Other components, such as the physical layer interface, testing, and network security that are not critical in supporting multimedia applications are also not examined in this dissertation.

1.4 Key Contributions of this Research

The main focus of this dissertation is to explore ways of enhancing quality of service delivered to adaptive multimedia applications in ATM ABR connections. A survey of techniques that can be used to make multimedia adaptive is also given.

The research should be helpful in designing and implementing future adaptive multimedia applications in a heterogeneous environment such as Internet. The schemes proposed here provide general mechanisms to enhance quality of service delivered to the multimedia.

The key contributions of this dissertation include:

1. A generalized definition of fairness and modification of ERICA+ (Explicit Rate Indication for Congestion Avoidance) rate allocation scheme to provide the generalized fairness.

2. Modified ERICA+ rate allocation scheme to provide MCR guarantees and generalized fairness.

3. New rate allocation schemes that use overload factor and provide MCR guarantees and generalized fairness

4. New queue control functions to control queuing delay that is helpful in minimizing delay experienced by ABR connections.

5. Extensive simulation and performance analysis of the rate allocation schemes and queue control functions developed under various network topologies and traffic scenarios.

6. A survey of techniques that can used at the application layer for adaptive streaming of multimedia.

7. An implementation of the schemes developed in a software-based ATM switch running Linux PCs using off-the-shelf ATM cards.

1.5 Dissertation Outline

The rest of this dissertation is organized as follows:

Chapter 2 gives the necessary background material for understanding the traffic management issues for providing QoS for various traffic including multimedia traffic. The chapter defines the QoS problem and gives an overview of various QoS models proposed by different standards bodies such as IETF, ISO, ATM Forum, and IEEE. A comparison of the QoS methods and their suitability for supporting data, multimedia (voice and video) traffic is also given in the chapter.

Chapter 3 explains the ABR service and ERICA+ algorithm. Then the chapter provides a survey of fairness definitions and rate allocation schemes that support MCR guarantees.

Chapter 4 gives the problem statement and methodology and proposed approaches for providing throughput continuity and minimizing queuing delay. The methods developed to solve different parts of the problem statement are discussed in subsequent chapters.

Chapter 5 gives the generalized definition of fairness. It discusses how ERICA+ switch algorithm can be modified to provide MCR guarantees and generalized fair allocation. Comprehensive simulation is done to test the algorithm under various network topologies. An analytical proof for the convergence of the algorithm is given at the end of the chapter.

Chapter 6 proposes three new ABR rate allocation algorithms that provide generalized fair allocation and give MCR guarantees. The algorithms are based on the overload (which is a measure of the load at the link) and have similar algorithmic structure. The algorithms are tested under different network configurations to demonstrate their convergence. Lastly, the chapter gives a comparison of these algorithms based on the simulation results.

Chapter 7 discuss the design and analysis of several queue control algorithms. It discusses the step, linear, hyperbolic, and inverse hyperbolic queue control functions. A comparison of these queue control function is done using simulation and analysis.

Chapter 8 provides a survey of application layer techniques for adaptive streaming of multimedia. These techniques will provide insight into the design and implementation of future adaptive application.

Chapter 9 gives an overview of the ATM on Linux project and the Kansas State university's implementation of a software based ATM switch. It then discuss our

modifications of the software switch in order to implement the ABR rate allocation algorithms.

Finally, to conclude chapter 10 provides a summary of the results of schemes presented in this dissertation. Lastly, it discusses some open issues and future research problems in traffic management.

CHAPTER 2

BACKGROUND AND SURVEY OF QOS METHODS

The advent of high speed network technologies such has ATM, Fast Ethernet, Frame Relay and Gigabit Ethernet, and the availability of cheap computing power (such as PCs) has led to the exponential growth of the Internet and increased user expectations. Multimedia applications, such as video conferencing, interactive games, digital libraries, distance education, will use the large bandwidth and require stringent delay requirements are expected to be supported by the Internet. The current widely supported best-effort service model does not provide any guarantees of bandwidth, delay or loss. An end-to-end quality of service (QoS) framework needs to be designed and implemented to meet such guarantees. The framework should aim to provide a service in which various parameters such as bandwidth, delay and jitter meet the constraints imposed by real-time and non-real-time applications. Currently, there is considerable effort by various organizations such as IEEE, IETF, ATM Forum, ISO and ITU-T to design and implement a quality of service framework for various networking technologies. Since we mainly consider the problem of enhancing quality of service, in this chapter we will provide an overview and comparison of the quality of service mechanisms provided by the IEEE 802.1D standard for 802.3 IEEE LANs, the IETF efforts for the Internet, and the ATM Forum specifications for ATM technology.

We discuss which features of these quality of service mechanisms can be used to support traffic generated by multimedia applications comprising of voice, video and traditional applications which consists of bursty (variable) data.

2.1 Introduction

IEEE 802.1D standard is the telecommunication standard for medium access control (MAC) bridges that interconnect IEEE 802.3 LANs (local area networks). The standard specifies usage of up to eight priority levels of traffic. It provides recommendations on how to map traffic classes to these priority levels. It also provides example mappings depending number queues available for each port.

IETF quality of service is provided by the integrated services model, RSVP (resource reservation setup protocol) signaling, differentiated services and traffic engineering of MPLS (multiprotocol label switching). Integrated services and RSVP have been designed and RFCs (request for comment) are available which precisely define these services and protocol. The Differentiated service working group is currently working on approving a standard (RFC) for differentiating various services. The advantage here is that flows are aggregated avoiding per-flow state accounting. MPLS is designed for enabling fast label switching instead of routing. Recently, there has been discussion on how to provide quality of service support by traffic engineering of MPLS. In addition various working groups of IETF have designed various protocols such as RTP (real-time transport protocol), RTSP (real-time streaming protocol) to support real-time applications over the Internet.

ATM technology merges the concepts of traditional packet switching of the data communications industry and circuit switching of the telecommunication industry.

ATM is the chosen technology by for implementing the Broadband Integrated Services Digital Network (B-ISDN) envisioned by ITU-T. ATM Forum (ATMF) and ITU-T are the two organizations that standardize the ATM technology. Eventually both the standards are expected to be complaint of each other. ATM Forum has defined service categories and a QoS framework to meet the requirements of applications ranging from non-real-time to real-time applications.

The ISO (International Standards Organization) has two new QoS initiatives. The ISO QoS framework by ISO SC21 QoS Working Group and the Enhanced Communications and Facilities (ECFF) by the ISO SC6 ECFF Working Group. ISO SC21 QoS Working Group's main contribution is the development of OSI QoS framework to support quality of service for the OSI communications. This framework defines terminology, concepts and mechanism for QoS in the OSI. ISO SC6 ECFF Working Group has introduced multimedia communication requirements as ECFF guidelines for current and future OSI transport and network standards.

2.2 QoS Problem

The problem of quality of service arises due to the different needs of receivers, senders and service provider [67]. The senders desire to maximize the amount of data they send for a given cost. The service provider desires to earn as much as possible for a given quality of service. The receivers on the other hand would like to receive the best quality for the given cost. Even if one of the three agree to waive their goals they problem of QoS can be easily solved. So any QoS framework should aim to satisfy the requirements of all three to the best possible extent.

2.3 802.1D QoS methods

The 802.1D is an extended revision of the ISO/IEC 10038: 1993 standard. The earlier edition defined the concept of media access control bridging in a LAN environment. In the current edition (802.1D), in addition to extending the concept of filtering services provided earlier, the following additional objectives are aimed to be achieved:

- To support transmission of time-critical information (as in the requirements of real-time applications) by provisioning expedited traffic capabilities.
- Provide filtering services to support dynamic definition and establishment of "groups" (i.e., multicast capabilities).

In order to achieve the above goals, the standard defines the concept of traffic classes and the effect on the forwarding process in presence of multiple traffic classes. A related standard P802.1Q/D11 defines the VLAN (virtual LAN) frame format that is able to carry VLAN identification and user priority information over VLAN environment, such as CSMA/CD (carrier sense multiple access/collision detect), that do not inherently signal priority information. The additional information is carried in *Tag Header* following the destination address in the frame. The P802.1Q is an independent standard though there are many common elements and concepts between it and the 802.1D standard.

QoS mechanisms of 802.1D

The 802.1D standard aims to support transmission of expedited traffic, which can be defined as traffic that contains time-critical information. This is achieved by using the "user_priority" field. The "user_priority" field can have values ranging from 0 to

7, where 7 has the highest priority and 0 the lowest. One or more queues can be used at each port. The standard recommends how to map the user_priority level to the given number of queues.

The static conformance requirements define the functionalities of a conforming MAC bridge. The capability to control the mapping of the priority to support multiple traffic classes by the MAC Bridge is provided optionally.

The standard defines an abstraction of features common to a number specific MAC services. The transfer of data between user and destination is via the MA-UNITDATA request primitive and the corresponding MA-UNITDATA indication primitive issued at the MAC service access point. Each request or indication has four parameters, destination address, source address, MAC service data unit and priority.

The MAC service consists of two layers, the internal sublayer and the MAC relay entity. The internal sublayer can be mapped onto one of the MAC service defined by ISO/IEC 15802-1 such as 802.3 LAN (CSMA/CD), ISO/IEC 8802-4 (token-passing bus), ISO/IEC 8802-5 (token-passing ring), FDDI, ISO/IEC 8802-6 (distributed queue dual bus, DQDB), 802.11 (wireless LANs) and ISO/IEC 8802-12 (demand priority). The user_priority field of the abstract MAC service is mapped onto an access-priority that is specific to the given MAC layer. Table 2.3 summarizes the mapping of user_priority to access-priority of the different MAC services.

Forwarding

The relay operation which affects the quality of service has the following functionalities:

- Selection of traffic class, following the application of filtering information.
- Queuing of frames by traffic class

MAC layer	Standard	Mapping of user_priority
name Shared Ethernet	802.3	not mapped
Token-passing bus	8802-4	PPP bits of frame control field
Token-passing ring	8802-5	YYY bits of frame control field
FDDI	ISO 9314-2	PPP bits of frame control field
DQDB	8802-6	QOS_DELAY subfield of MCP header
Wireless LANs	802.11	not mapped

Table 2.1: Mapping of user_priority to access-priority of specific MAC layer

- Selection of queued frames for transmission
- Selection of out bound access priority

The user_priority is regenerated based on the current user_priority value and the user priority regeneration table. The ability to modify the user priority regeneration table is provided as an option. The regeneration of user priority should be consistent with the end-to-end requirements. One or more queues may be provided at each port. Up to eight traffic classes can be supported using separate queues for each traffic class. The forwarding process makes sure that the transit delay is less than the "maximum transit delay" parameter.

User priority and traffic classes

The priorities, queue mappings, and queue service disciplines can be managed to support the user requirements. The goal of the recommendation is to provide a reasonable default mapping for a typical LAN bridged environment. The default values would greatly facilitate plug and play networking. These default mapping are used to support the emerging integrated services mapping work by IETF's ISSLL (integrated services over specific link layers) working group. The standard provides a

simple and practical approach of mapping different traffic classes to the user_priority. The standard gives the following as the recognized traffic types and their mapping to user_priority :

1. *Network control* (NC): Information characterized by "must get there" requirement. It is used to maintain and support network infrastructure. Mapped to highest user_priority 7.

2. *Voice* (VO): Requires tight delay and jitter (order of 10 ms). Mapped to user_priority 6.

3. *Video* (VI): Requires delay in the order of 100 ms. Mapped to user_priority 5.

4. *Controlled Load* (CL): Business application which are subject to admission control. Mapped to user_priority 4.

5. *Excellent Effort* (EE): or "CEO's best effort", best-effort traffic of most important customers. Mapped to user_priority 3.

6. *Best Effort* (BE): Current LAN traffic. Mapped to user_priority 0. The reason it is mapped to 0, is so that the best-effort traffic is treated in the same manner as it is now in bridges confirming to ISO/IEC 10038: 1993 edition.

7. *Background* (BK): Bulk data transfers and other applications which should adversely affect the performance to other users. Mapped to user_priority 1, this implies that user_priority 1 and 2 should be given lower lower priority than 0. Currently, user_priority 2 is unused.

Number of queues	Traffic types
1	{*BE*, EE, BK, VO, CL, VI, NC}
2	{*BE*, EE, BK}, {*VO*, CL, VI, NC}
3	{*BE*, EE, BK}, {*CL*, VI}, {*VO*, NC}
4	{*BK*}, {*BE*, EE}, {*CL*,VI},{*VO*,NC}
5	{*BK*},{*BE*,EE},{*CL*},{*VI*},{*VO*,NC}
6	{*BK*},{*BE*}, {*EE*},{*CL*},{*VI*},{*VO*,NC}
7	{*BK*},{*BE*}, {*EE*},{*CL*},{*VI*},{*VO*},{*NC*}

Table 2.2: Mapping traffic types to traffic classes for given number of queues

The objectives are to meet the latency and throughput requirements of real-time applications. The standard focuses on meeting the latency requirements using segregation of few traffic classes. The default queuing service recommended by the standard is strict priority. This simple scheme can give latency bounds. A more sophisticated service discipline is required for active bandwidth management. The grouping of different traffic for a given number of traffic classes is shown in Table 2.2. Though the distinguishing traffic type in each group italicized, all traffic belonging to the group are treated in same manner. That is, all traffic of a given class are treated as equals.

The standard default mappings between the defining traffic types and number of queues is given in Table 2.3. The standard also gives justification for the recommended grouping of traffic types into traffic classes.

Number of queues	Defining traffic type							
1	BE							
2	BE				VO			
3	BE				CL		VO	
4	BK		BE		CL		VO	
5	BK		BE		CL	VI	VO	
6	BK		BE	EE	CL	VI	VO	
7	BK		BE	EE	CL	VI	VO	NC
8	BK	–	BE	EE	CL	VI	VO	NC

Table 2.3: Defining traffic types

2.4 IETF QoS methods

In this section we give an overview of the methods developed by IETF for providing QoS in the Internet.

The goals of different IETF working groups in providing a QoS framework for the Internet are as follows:

- *Intserv*: Intergrated service working group. Define a set of service models to support quality of service requirements of various applications.
- *Diffserv*: Differentiated service working group. Differentiate traffic based on application type and provide appropriate service.
- *Qosr*: QoS-based Routing working group. Provide a framework for QoS-based routing protocols.

- *RSVP*: Resource reservation setup protocol working group. Design a signaling mechanism to reserve resources along the connection to support integrated services.

- *MPLS*: Multiprotocol label switching. Provides standards for label switching which can be useful in building fast routers/switchs. Recently, there has been discussion that traffic engineering of label switching can be used to support quality of service.

2.4.1 Integrated Services

To provide quality of service in the Internet, the Integrated Services working group has specified *control load service (CLS)* and *guaranteed service (GS)* in addition to the current best-effort service. RFC 2216 [107] gives a framework which specifies the functionalities of network components which can support multiple, dynamic quality of service in a internetwork. The control load service and the integrated service are specified according to this framework. The framework defines a set of functionalities including end-to-end behavior and the setup protocol needed to define a specific service.

Control Load Service

The control load service aims to provide a quality of service to applications that closely approximates the quality the application would receive from an unloaded network. It uses admission control to provide such a service even when the network is overloaded. The requirements of a network element to support control load service is given in RFC 2211 [123].

Tspec (traffic specification) (defined in RFC 2215 [106]) which consists of a token bucket plus a peak rate (p), minimum policed unit (m) and maximum packet size (M) is used to model the traffic parameters. The token bucket is characterized by "token rate" (r) and "bucket size" (b). The policing is done using a token bucket filter function.

The control load service does accept specific target values of control parameters such as delay and loss. A setup protocol, such as RSVP, is used to reserve adequate resources along the packet flow. Appropriate scheduling mechanisms should be used in the network elements to ensure that expected quality of service is provided for the admitted flows.

Guaranteed Service

The Guaranteed Service aims to provide an assured level of band-width and bounded delay with no queuing loss for all conforming datagrams. The end-to-end behavior conforms to the fluid model [91] and the delay experienced by a datagram is not more than the fluid delay and the specified error bounds. This service does not minimize jitter.

The guaranteed service uses the TSpec parameter to model and characterize the data flow. A setup protocol such as RSVP (or Q.2931 signaling) is used to reserve adequate resources along the data flow. The RSpec is used to represent the request specification, which consists of rate R and slack term S, where R should be greater than or equal to r (token rate) and S should be non-negative. The slack term S indicates the difference between the desired delay and the expected delay by reserving rate R. The slack term can be used by network elements to reduce the reserved resources.

Each network element indicates the error terms C (rate-dependant delay) and D (worst case non-rate-based delay) for each data flow. The end-to-end delay experienced by is given by the following expression:

$$\text{End-to-end delay} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{M+\sum C}{R} + \sum D$$

Policing is done at the edge of the network. Inside the network policing is done by reshaping. Reshaping is done at all heterogeneous source branch points and at all source merge points. A branch point is the place where the multicast tree branches to multiple outgoing links. The non-conforming packets are given best-effort service. Appropriate scheduling mechanism such as weighted fair queuing (WFQ) [27] is used at each network element.

2.4.2 Resource Reservation Setup Protocol (RSVP)

RSVP is the signaling protocol to be used for resource reservation at network elements to provide support for integrated services [18]. The key features of RSVP are:

- Supports both multicast and unicast applications.
- Makes unidirectional reservations, i.e., it is simplex.
- Is receiver-oriented to support heterogeneous receivers.
- Uses "soft" state, hence it can adapt to dynamic group membership changes and changing routes.
- Provides several reservation styles (wildcard, fixed filter, shared explicit) to support the needs of a variety of applications.

- Supports both IPv4 and IPv6.

2.4.3 QoS-based Routing

The current routing in Internet is not "QoS-aware". The QoS routing working group of IETF has developed a framework for QoS-based routing in the Internet [25]. The framework distinguishes between intradomain routing and interdomain routing. The desired features of QoS-based routing protocols are:

- The protocol should be able to determine available resources along the path computed to the destination. It is desirable that the available resources are relatively static.
- Optionally, it is desirable that the protocol be able to compute multiple paths for a given destination, based on service classes.
- The routing policy should account for monetary cost, as a function of flow parameters, usage characteristics and administrative factors.

It is desirable that the routing protocol be scalable. One way to achieve is to have a hierarchical scheme similar to PNNI [44]. The open shortest path first routing protocol is being currently extended to support QoS [56, 132]

2.4.4 Differentiated Services

In IPv4 packet header, 3 bits are used for precedence and a 4 bit ToS (type of service). Currently, OSPF (open shortest path first) and IS-IS (intermediate system to intermediate system intra-domain protocol) may compute paths for each ToS. Differentiated services will standardize the usage of precedence bits. The IETF

Intserv/RSVP (Integrated services and RSVP) was initially designed to enable support of QoS guarantees for applications. The Intserv/RSVP model has the major short coming that it is not scalable since it needs to maintain states on a per flow basis. Due to this limitation though the Intserv/RSVP has been standardized it has not been be deployed widely as anticipated. There is an immediate need to provide support for QoS. To address this immediate need, the Differentiated Services (diffserv) working group was formed in February 1998 to provide support for service differentiation. Service differentiation is desired to accommodate heterogeneous applications' requirements and user expectations, and to permit differentiated pricing. The DS (differentiated services) field in the header of IPv4 (ToS octet) and IPv6 (traffic class octet) is used for service differentiation. RFC 2474 gives the definition of the DS field [87].

The following terminology is used in differentiated services:

- Behavior Aggregate (BA) Classifier : Classifier based only on DS field.
- MF classifier: a multifield classifier. Classification based on multiple fields such as source address, destination address, DS field.
- Microflow: a single instance of application-to-application flow.
- DS codepoint : Currently only the first 6 bits of the DS byte is used to specify a codepoint value. A specific value of the DSCP (DS codepoint) of DS field is known as DS codepoint. This is used to select the PHB (per-hop behavior). There are currently three subsets of codepoints: xxxxx0 is a standardized set of code points different PHBs, xxxx11 is for experimental or local use, xxxx01 is currently experimental but may be standardized in future. Class selector

codepoints have value xxx000, they should be implemented in such a manner which is consistent with the current usage of precedence bits. For example codepoint 111000 should get better treatment (e.g., higher precedence) than codepoint 000000.

- Per-hop behavior : externally observable forwarding behavior applied at a DS compliant node. For example a typical behavior can be that x% link bandwidth is utilized. A PHB group consists of related PHB which share common constraints. The default PHB maps to the common best-effort forwarding behavior.

- Traffic profile: specifies the temporal properties of the traffic stream. It can be used to deduce in-profile and out-of-profile packets. Traffic conditioning, such as marking or shaping is applied to out-of-profile packets.

- Traffic conditioning : control function applied to a traffic aggregate. It is implemented using traffic conditioners, which may contain, meters (e.g, token bucket), markers, shapers and policers.

The specification of an architecture for differentiated service is given in RFC 2475 [13]. It uses the framework for differentiated services given in the IETF draft [11]. The architecture is composed of network components which perform functions including a small set of per-hop behaviors, packet classification functions and traffic conditioning functions including metering, marking, shaping and policing. Scalability is achieved by implementing the complex functions only at the network boundary. Inside the network, the per-hop behaviors dictated by the DS field value are applied to aggregated traffic. Currently two PHB groups (assured forwarding (AF) [58] and

Drop Precedence	Class 1	Class 2	Class 3	Class 4
Low Drop Prec	001010	010010	011010	100010
Medium Drop Prec	001100	010100	011100	100100
High Drop Prec	001110	010110	011110	100110

Table 2.4: Code points for Assured Forwarding PHB group

expedited forwarding (EF) [66]) are proposed in the diffserv group. The service level agreement (SLA) between the provider and user dictates the type of PHB used and the type of traffic conditioning performed.

Assured Forwarding

The AF PHB group has four AF classes. Within each AF class there are three drop precedences: low, medium, and high. The level of forwarding assurance depends on resource allocation for the AF class, the current load, and the drop precedence. The four classes have a priority ordering which can be realized by assigning decreasing weights in a WFQ scheduling. The three drop precedences can be achieved by using token bucket with one rate and two bucket sizes. Lower delay is provided for lower AF classes. Table 2.4 shows the recommended codepoint assignments for AF classes. The service provided by the AF classes is similar to the nrt-VBR,ABR,GFR service categories in ATM.

Expedited Forwarding

Expedited forwarding provides support for services which need low loss, low latency, low jitter, and guaranteed bandwidth. Such service can be thought of as a virtual leased line by the application. Queuing gives rise to loss, latency, and jitter. Ensuring small queues can provide the desired properties. Queuing arises due

to mismatch in the output rate and current input rate. To ensure small queues, the incoming aggregate traffic should be conditioned and a minimum output rate should be guaranteed. Simulation experiments of expedited forwarding show that desired requirements can be met when end-to-end EF is enabled [66].

2.4.5 Multiprotocol Label Switching

The Multiprotocol Label Switching (MPLS) effort was started in IETF to provide label switching of IPv4 and IPv6 at the network layer (layer 3). In a connection less network each router makes an independent decision on packet forwarding based on the packet header. Currently, forwarding at the network layer is based on the "longest prefix" match found in the routing table for the particular destination address found in packet header. Choosing the next hop consists of two functions. The first one partitions the entire set of packets in "forwarding equivalence classes (FECs)" and the second maps each FEC to the next hop. As the packet enters the MPLS domain at the first router an association is formed between the FEC and a label (a short fixed length field). Subsequent hops do not have to look into the packet header and can forward based on the label. The labels need to have only local significance. A label is similar to the VPI/VCI (virtual path identifier/virtual circuit identifier) in ATM. The routing mechanism, such as OSPF, is independent of forwarding. Initially label switching was designed to build fast hardware based routers. Some of the components and features of MPLS switching architecture [100, 19] are:

- Label Distribution Protocol : Set of procedures by which a label switching router (LSR) informs another LSR of the label/FEC bindings. One possible way of distributing labels can be based on address prefixes. Usually the downstream

router decides to associate a label with a particular FEC and informs the binding to the upstream router.

- Label Attributes : The binding of label L to FEC may have associated "attributes". For example, the uniqueness attribute can be used to avoid loops in routing.

- Label Stack : In general a set of labels can be associated with the packet. A label stack defines an ordering of the labels associated with the packet. The label stack can be used to support hierarchy in MPLS.

- Label push and pop : A label is attached to a packet when it enters a MPLS domain. This is known as label push. When the packet leaves the MPLS domain the label is removed. The label is said to be popped.

- Aggregation : A set of FECs can be mapped to another FEC (label) to provide aggregation.

- Support for explicit routing : In some cases, such as in QoS based routing, a source might find an alternate route which needs to be explicitly specified. MPLS can be used to enforce that packets follow the explicit route by appropriate label association.

- MPLS should support both topology-driven and traffic/request-driven label assignment.

Recently, the MPLS working group in the IETF has been discussing traffic engineering of MPLS that can be used to provide support for quality of service. Traffic engineering is essentially an performance optimization problem. The objective of the

performance optimization can be maximization of throughput, minimization of delay and loss. Support for QoS can be achieved by achieving these optimizations. Efficient resource utilization can be achieved by minimizing effect of congestion and load balancing through path splitting.

The requirements for traffic engineering of MPLS is discussed in internet draft [9]. The authors consider the traffic engineering as a control problem. They claim that, by appropriate control policy, such as adaptive feedback and control or pro-actively predicting future traffic, the objectives of performance optimization can be achieved.

Traffic trunks is defined as an aggregation of traffic flows of the same class in a label switched path. Traffic trunk is similar to SVC (switched virtual circuit) in ATM, and they are associated with traffic and not the path of the traffic. Traffic trunk can be viewed as atomic objects, which can be routed along a different path. A set of attributes, such as priority, preemption, policing can be associated with the traffic trunks. The attributes and the characteristics of the traffic determine the FEC of the traffic trunk. Load balancing can be achieved by using multiple traffic trunks for the same egress. The preemption attribute implies that a trunk may preempt another. A traffic trunk can be preemptor, non-preemptor, preemptable or non-preemptable. Priority attribute defines relative importance of traffic trunks. In the context of constraint based routing (such as QoS routing) in MPLS, priority can be used in path selection and admission control.

2.5 ATM Forum QoS methods

This section overview the ATM Forum QoS methods.

2.5.1 Goals

The goals of ATM Forum QoS methods are the following :

- Define service categories to meet the requirements various applications.
- Signaling and routing mechanisms should support quality of service.
- Identify and provide guidelines for network functions which are needed to support end-to-end quality of service.

2.5.2 ATM Service Architecture

The ATM Forum identifies the following generic network functions in their QoS model:

Connection Admission Control defines how the network controls the connection admission during call setup.

Feedback is the mechanism by which the network informs the end system the state of the network.

Usage Parameter control (UPC) defines how the network monitors and controls of the end systems.

Cell loss Priority control. Some service categories may generate cells with cell loss priority (CLP) marking. The cell loss priority control mechanism defines how the network treats such cells. It can be either transparent or significant.

Traffic Shaping. Shaping can be done at the sources to achieve desired traffic characteristics.

Frame Discard. In the event of congestion the network may discard at the frame level rather than at the cell level.

Flow control defines how the network controls the connection. It can be either closed-loop (as in ABR) or open loop.

2.5.3 ATM Service Categories

The following service categories are defined by the ATM Forum:

- **CBR:** The constant bit rate service category provides constant bandwidth characterized by peak cell rate (PCR) value. Cells with transfer delay more than maximum cell transfer delay (maxCTD) are assumed to be of significantly reduced value. CBR is intended to provide support for real-time applications requiring constrained delay variation (e.g., voice, video, circuit emulation). The advantage of the CBR service is that it provides rigid guarantees but is expensive.

- **rt-VBR:** The real-time variable bit rate service category is intended for real-time applications, such as voice and video. Rt-VBR connections are characterized by PCR, sustainable cell rate (SCR), maximum burst size (MBS). Rt-VBR sources can be bursty, i.e., the transmission rates vary. Cells delayed more than maxCTD are of significantly reduced value to the applications. Rt-VBR can support statistical multiplexing of real-time sources.

- **nrt-VBR:** The non-real-time VBR service is intended for bursty non-real-time applications. The connection is characterized by PCR, SCR and MBS. A low

cell loss is expected for conforming cells. No delay bounds are given. nrt-VBR can be used for transferring stored video where delay is not critical.

- **UBR:** The unspecified bit rate service is intended for non-real-time applications (e.g, email, file transfer). The connection is characterized by PCR. It is easy to use, but provides no guarantees are given on cell loss ratio (CLR) and delay.

- **ABR:** The available bit rate service category is intended to support adaptive applications. A flow control mechanism that provides feedback about the network congestion status to the sources through resource management (RM) cells. End system is expected to adapt itself according to the feedback. The connection is characterized by PCR and minimum cell rate (MCR). This is the only service that provides closed look feedback which makes it suitable for adaptable applications.

- **GFR:** The GFR service provides support for non-real-time applications. The applications are assumed to benefit from minimum guarantees. The connection is characterized by PCR, MCR, MBS and maximum frame size (MFS). The service commits to accept cells up to rate of MCR, though it may send cells at rate up to PCR. GFR is better than UBR and it is to intended to transport aggregated TCP/IP traffic in the Internet backbone.

2.5.4 ATM Layer QoS Parameters

The QoS parameters at the ATM layer measure the performance of a connection. The ATM Forum identifies the following six ATM layer QoS parameters:

1. Peak-to-peak cell delay variation *(peak-to-peak CDV)* : This is a end-to-end delay parameter. It is defined as the difference between the $(1-\alpha)$ quantile of cell transfer delay and the fixed portion of end-to-end delay (due to propagation delay).

2. Maximum cell transfer delay *(maxCTD)* : This is the $(1-\alpha)$ quantile of the cell transfer delay.

3. Cell loss ratio *(CLR)* : This is defined as the ratio of number lost cells to total transmitted cells

4. Cell error ratio *(CER)* : This is defined as:

$$CER = \frac{\text{Errored cells}}{\text{Successfully transferred cells} + \text{Errored cells}}$$

5. Severely-errored cell block ratio (SECBR) : This parameter is defined as :

$$SECBR = \frac{\text{Severely errored cell blocks}}{\text{Total transmitted cells blocks}}$$

   Where a cell block is a sequence of N cells transmitted consecutively. When more than M of the N cells are corrupted the block is said to be severely errored.

6. Cell misinsertion ratio *(CMR)*: This is defined as:

$$CMR = \frac{\text{Misinserted cells}}{\text{Time interval}}$$

   This parameter is a rate rather than a ratio since the mechanism which gives rise to misinsertion (such as undetected errors in cell header) is independent of the number cells transmitted.

The first three parameters are negotiated during connection setup for certain service categories. The last three are non-negotiated parameters.

2.5.5 Specification of service categories

The cells conforming to a connection are described by the connection descriptor. The connection descriptor consists of the source traffic descriptor, the specification of CDVT (CDV tolerance), and the conformance definition. The source traffic descriptor consists of a set of traffic parameters of the ATM source, which describes the traffic characteristics of the source. An overview of the conformance definitions is given in the next subsection.

The generic cell rate algorithm (GCRA) and its variation is used to describe the source. The GCRA is a virtual scheduling algorithm. GCRA is defined by two parameters, increment (I) and limit (L). Increment is the interval between the expected cell arrivals, and L is limit on number of cells.

Table 2.5 summarizes the attributes such as the traffic parameters, the QoS parameters negotiated at the source, and the feedback option for each of the ATM service category.

CDVT is the cell delay variation tolerance. Different values of CDVT may apply along the path. CLR is expected to be low for ABR and eligible frames of GFR. ABR is only service category with feedback control. The virtual source virtual destination (VS/VD) option can be used in ABR to segment the control loop. In such a situation the intermediate switches act as virtual destination turning around RM cells, and as virtual source generating RM cells.

	ATM Layer Service Category					
Attribute	CBR	rt-VBR	nrt-VBR	UBR	ABR	GFR
Traffic Parameter:						
PCR and CDVT	Specified					
SCR, MBS, CDVT	NA	Specified		NA	NA	NA
MCR	NA				Spec.	NA
MCR, MBS, CDVT, MFS	NA					Spec.
QoS Parameter:						
Peak-to-peak CDV	Specified		Unspecified			
maxCTD	Specified		Unspecified			
CLR	Spec.			Unspecified		
Feedback	Unspecified				Spec.	Unspec.

Table 2.5: Summary of specifications of ATM service categories

2.5.6 ATM Conformance Definitions

The conformance algorithm and the traffic parameters of a connection dictate which cells conform. When a cell has its cell loss priority (CLP) bit set, the network provides a best-effort delivery to that cell. The ATM service categories are further classified based on how they treat the cells whose CLP bit is set. The CLP=0 flow defines the flow of cells in the connection where the cells have CLP bit as 0. The CLP=1 flow is for those cells with CLP bit set. The CLP=0+1 defines the entire flow of the cells of the connection. When the CLP bit is set at the end system the cell it is said to be a marked cell. When the CLP bit is set by a network element, such as a switch, the cell said to be tagged. The distinction is necessary in order to know where CLP bit was net. For CLP=1 flow, the network can take two approaches for handling these cells. In the *CLP-transparent* approach, the CLP bit is ignored and the flow's connection guarantees are always with respect to the CLP=0+1 flow. In

Conformance Definition	PCR flow	SCR flow	Tagging option enabled	MCR MCR	CLR specified
CBR.1	0 + 1	NS	NA	NS	0+1
VBR.1	0 + 1	0 + 1	NA	NS	0+1
VBR.2	0 + 1	0	No	NS	0
VBR.3	0 + 1	0	Yes	NS	0
ABR	0	NS	NA	Yes	IS
GFR.1	0+1	NS	NA	Yes	IS
GFR.2	0+1	NS	Yes	Yes	IS
UBR.1	0+1	NS	NA	NS	US
UBR.2	0+1	NS	Yes	NS	US

IS - implementation specific, US - unspecified, NA - not applicable, NS - not specified

Table 2.6: Summary of conformance definitions of CBR, VBR, ABR, GFR and UBR service categories

CLP-significant approach, the network element makes a best-effort delivery of CLP=1 cells. The network element may choose tagging as option. Table 2.6 summarizes the conformance definitions of the ATM service categories.

The ATM service categories provide a rich functionality in terms of support for quality of service. The applications that can be supported range from voice with stringent delay and jitter requirement to non-real-time applications. Though ATM was envisioned to implement the B-ISDN, currently ATM is only prevalent in ISP (Internet Service Provider) and campus backbones.

2.5.7 Signaling and Routing in ATM

The private network-network interface (PNNI) version 1.0 [44], is used for signaling and routing. The UNI version 4.0 [43] is the signaling protocol at the user to network interface. The signaling protocols handle connection setup and parameter negotiation

for the service categories. The PNNI specification specifies how the routing protocol handles the QoS parameters during connection setup. The PNNI routing protocol supports end-to-end QoS.

2.6 Comparison of QoS methods

In this section we compare the QoS methods from the IEEE (802.1D standard), the IETF and the ATM Forum.

QoS methods can be compared along a number dimensions, where each dimension can be an attribute of the provided QoS, cost or complexity. In the following subsections we compare the QoS methods along a number of such dimensions. Table 2.7 gives a summary of these comparisons.

Absolute versus Relative Specification of Parameters

In absolute (also known as quantitative) QoS the parameters are quantified by specific values. For example the delay is bounded by 100 ms is an absolute. In relative (also known as qualitative) QoS the QoS provided is qualitative. For example, class 1 is assured that it will receive twice the bandwidth of class 2. Relative QoS does not necessarily imply that QoS is guaranteed.

The QoS provided by 802.1D standard is relative. The user_priority is used to indicate the relative priority received by a packet.

The Intserv/RSVP QoS method providing guaranteed service is absolute in nature. The control load service is semi-absolute since deterministic guarantees are not provided. The diffserv QoS is relative since the service differentiation in conjunction with traffic conditioning is essentially prioritizing traffic. The MPLS can provide both relative and absolute QoS depending on the traffic engineering mechanisms used.

ATM Forum QoS methods are absolute in nature, since the QoS parameters such as bandwidth, delay and jitter can be specified.

Granularity

The quality of service objectives can be specified on a per-flow basis or for an aggregate of flows.

802.1D QoS is on a per-flow, per-hop basis. The Insterv/RSVP QoS maintains per-flow reservation state and can provide per-flow QoS guarantees. The Diffserv QoS is essentially provided for traffic aggregates. In a small network service, however, differentiation can be done on a per-flow basis. MPLS has the capability of providing both per-flow and aggregate QoS depending on the granularity of the traffic which is being switched.

QoS Metrics

A QoS method may provide a number of metrics, such as throughput, delay or jitter.

802.1D only ensures that traffic of different traffic classes receive service based on their priorities. It does not have any guarantees of specific QoS metrics.

Insterv/RSVP QoS method's guaranteed service can provide guarantees on bandwidth and delay. Packet loss depends heavily on the link layer technology used and is therefore missing in the specification. Control load service may provide probabilistically guaranteed bandwidth, delay and loss metrics. Diffserv does not provide any explicit QoS metrics, since it is essentially used to differentiate traffic aggregates. It relies on admission control and reservation policies at network boundaries to provide the desired QoS. MPLS does not provide any QoS metrics.

All ATM Forum service categories provide the peak cell rate QoS metric. Table 2.5 gives the various QoS metrics provided by different ATM service categories.

QoS level

QoS can be provided at the session layer, transport layer, datalink layer (per-hop) or in back-bones.

802.1D QoS is on a per-hop basis since packet gets prioritized treatment at each bridge on their way to the destination. The Insterv/RSVP QoS method is of an end-to-end nature. Diffserv provides support for QoS in backbone level. MPLS is envisaged to support QoS at the backbones. ATMF QoS provides end-to-end QoS guarantees in a end-to-end ATM network. ATMF QoS methods can also provide QoS at backbones in networks such as today's Internet, when IP over ATM is used at backbones.

Admission Control and Policing

Admission control is used to decide whether the service request can be admitted without hindering existing services.

802.1D does not have explicit admission control. A subnet bandwidth manager [105] (SBM) or a bandwidth broker [88] can be used to manage the bandwidth and policing in 802 LAN environments. Currently the Integrated Services over Specific Link Layers (issll) working group is chartered to define a mapping of upper layer QoS services over link layers. Specifically, internet draft by Seaman, Smit, Crawley and Wroclawski [105] discusses the mapping of integrated services over IEEE 802 networks.

IETF QoS methods rely on RSVP for signaling and admission control. Bandwidth brokers can be used at network boundaries (LANs) for managing resources and

policing. The RSVP admission policy (RAP) working group of IETF is currently developing the common open policy service (COPS) [17] protocol for deploying policing in the Internet.

ATMF uses connection admission control (CAC) for admission control and UPC (usage parameter control) for policing.

QoS Routing

802.1D does not provide support for QoS-aware routing. The routing of packets is topology driven and depends on the particular spanning tree of bridges realized. IETF has provided a framework for QoS routing. Currently OSPF is being extend to support QoS [56, 132]. ATMF PNNI provides support for QoS-aware routing.

Availability

802.1D QoS methods are currently available in IEEE 802 LANs. Of the several underlying 802 LAN protocols mentioned in 802.1D, shared Ethernet and wireless LANs cannot support QoS. Though the Intserv/RSVP has been standardized and available it is not widely deployed due to several limitations, which are discussed in section 2.6. ATMF service categories such as CBR, VBR, and UBR are currently available. Though ABR is available it is not widely deployed. GFR service is not yet standardized and hence not available.

Cost, Accounting and Billing

Cost-wise, the IEEE 802 LAN technology is the cheapest for providing QoS support in LAN environments. The cost of IETF QoS methods such as Diffserv and MPLS cannot be estimated since they are currently not available. Recently, the Authentication, Authorization and Accounting (aaa) working group of IETF has been chartered to address the accounting issues in the Internet [6, 2]. ATM technology is

expensive compared to other technologies because it is more complex and provides better quality-of-service.

The policing mechanisms such as COPS [17] can be used for accounting and billing. The diffserv model of IETF is more amenable for accounting and billing due to service differentiation.

Complexity

802.1D is simple, needing only priority and up to eight queues at bridges to support different traffic classes. IETF QoS methods are still evolving and it is difficult to estimate the eventual complexity of the methods. ATMF QoS methods are well developed and mature, but they pay the price of being complex. The complexity arises due elaborate traffic management.

Scope

802.1D is applicable only in LAN environments. The combination of IETF QoS methods, where Intserv/RSVP is used at stub networks and diffserv and MPLS in the backbones is envisaged to provide truly scalable WAN (wide area network) internetwork. Due to PNNI, and merging capability at VC and VP level, ATMF QoS methods can be used in both LAN and WAN situations.

Security

All QoS methods are vulnerable to denial-of-service and other such attacks. Security issues can be solved in conjunction with policy management. AAA working group is chartered to address the security issues of IETF QoS methods. The security consideration for ATM is discussed in ATMF specifications of security framework [41].

Limitations

In this subsection we discuss the limitation of the various QoS methods.

The limitations of 802.1D include the following:

- Provides only relative QoS, hence cannot guarantee bounded values of QoS parameters.
- Applicable to LAN environments, hence cannot support large number of nodes.

The limitations of Intserv/RSVP QoS methods include the following:

- Scalability: RSVP needs to maintain per-flow state, which is not scalable as the number of flows increase. It is only suitable for private networks.
- Receiver based: In some situations the sender needs to send notification and control information.
- Soft State: For stability route/path pinning is necessary.
- Negotiation: RSVP does not provide support backtracking and renegotiation.
- RSVP capable routers are need at nodes in the end-to-end path.

Limitations of diffserv include the following:

- Provides service differentiation on a per-hop basis so it is difficult to provide end-to-end guarantees.
- Provides static service level agreements which is not suitable to network traffic and topology which can by highly dynamic.
- Provides only limited support dynamic membership changes in multicast group.
- Provides only unidirectional support so the receiver does not have control.

- Has security considerations due to possibility of modifying DS field at intermediate nodes.

- Provides QoS guarantees for aggregates only. Difficult to support QoS for real-time and high-bandwidth applications that need per-flow QoS guarantees.

- Does not provide feedback mechanism to mitigate congestion.

Limitations of ATMF QoS methods include:

- Complexity: Though ATMF service categories provide rich QoS capabilities they are complex to implement.

- End-to-end QoS of ATMF methods can be realized only in a native ATM environment. Applications need to be modified to run directly over ATM.

- Cost: ATM technology is expensive since it is a complex and new technology.

2.7 Suitability of QoS methods to Applications

In this section we discuss the suitability of the 802.1D, IETF and ATM Forum QoS methods for voice, video, and bursty data applications. Table 2.8 summarizes the QoS requirements of voice, video, and bursty data applications. The requirements of each of these applications might differ from the characterized ones in the table depending on the applications interaction requirements and desired quality.

QoS features to support voice

Voice applications need tight delay and jitter bounds. 802.1D provides only relative QoS based on user_priority, which can be used to meet the bandwidth requirements of voice traffic. Delay and jitter are not guaranteed in 802.1D QoS. Delay

Dimension	802.1D	Intserv/RSVP	Diffserv	MPLS	ATMF
QoS nature	Relative	Absolute	Relative	Imp.Sp	Absolute
Granularity	Aggregate	Per-flow	Aggregate	Aggregate	Per-flow
QoS Metrics	None	bwd, delay, loss	None	None	bwd, delay loss, jitter
QoS level	per-hop	end-to-end	backbone	backbone	end-to-end
Adm. Control	SBM	BB	BB	BB	CAC
QoS-Routing	No	Imp.Sp	Imp.Sp	Imp.Sp	PNNI
Availability	Yes	Yes	No	No	Yes
Cost	Low	Medium	-	-	High
Complexity	Low	High	Low	Low	High
Scalability	Low	Low	High	High	High

bwd - bandwidth, Imp.Sp - implementation specific, SBM - Subnet bandwidth manager, BB - Bandwidth broker

Table 2.7: Summary of comparison of QoS methods along different dimensions

Application	Bandwidth	Delay	Jitter	Loss
Voice	8 - 64 Kbps	10 ms	10 ms	$10^{-4} - 10^{-5}$
Video	384 Kbps - 15 Mbps	100 ms	100 ms	$10^{-3} - 10^{-4}$
Data	Highly variable	-	-	No loss

Table 2.8: Characteristics of voice, video, and bursty data applications

bounds may be provided by implementing intelligent scheduling policies and by providing support for admission control using a subnet bandwidth manager.

The guaranteed service of Insterv/RSVP QoS method can provide the bandwidth and delay requirements, but it does not provide guarantees for jitter.

The control load service can be used for adaptive voice applications. Adaptive voice applications can adapt to the varying bandwidth availability by using different parameter values in voice compression methods. The Audio/Video transport (avt) working group of IETF has developed the real-time transport protocol (RTP) to support real-time audio and video applications over UDP (user datagram protocol) and IP multicast [103]. The IP telephony working group (iptel) of IETF is chartered to develop methods to support voice and telephony applications in the Internet [113].

The expedited forwarding PHB of diffserv can be used to support voice applications. Simulation results of EF show that it can provide the required bandwidth, delay and jitter bounds for voice [66]. Use of MPLS for voice depends heavily of the specific traffic engineering methods used.

The CBR and rt-VBR service category provide excellent support for voice applications. Both services provide guaranteed bandwidth (PCR, SCR), delay (max-CTD), jitter (peak-to-peak CDV) and loss (CLR) bounds. ATM Adaptation Layer-1 (AAL1) has been specifically designed for transporting voice applications. The voice and telephony over ATM (VTOA) group of ATMF has defined a number specifications including circuit emulation service inter-operability and voice and telephony over ATM to desktop [46, 48, 47, 45].

QoS features to support Video Video applications need bandwidth in the range of 384 Kbps (low quality video is available at 28.8 Kbps, e.g., realvideo) to 15 Mbps

(high definition TV). Video applications, especially the non-interactive ones, are more tolerant to jitter because buffering can be used at receiver before playback to absorb packet jitter.

802.1D QoS methods can meet the bandwidth requirements of video. To meet the delay requirement, admission control using SBM and fair scheduling, such as WFQ, need to be used.

The guaranteed service of Insterv/RSVP and EF PHB of diffserv can be used for interactive video applications. Adaptive video applications can be supported by the control load service. The Multiparty Multimedia Session Control (mmusic) working group has developed a real-time streaming protocol and session description protocol [57] to support multimedia streaming applications.

For ATM, the CBR and rt-VBR service categories can be used for video applications because both services provide guaranteed bandwidth (PCR, SCR), delay (maxCTD), jitter (peak-to-peak CDV) bounds (CLR). The service aspects and applications group of the ATMF has developed specifications for transporting MPEG-2 applications over ATM [49].

QoS features to support bursty data

Data applications are bursty in nature, with varying bandwidth requirements and are delay tolerant. Bursty data traffic when transported in conjunction with other delay sensitive multimedia traffic, is best handled by best-effort service.

802.1D standard recommends that user_priority 0, be mapped to best-effort service. When more than one queue is available at bridges, best-effort service can get higher priority than background traffic.

The best-effort service of the Integrated Service model is used for bursty data applications. The Internet protocol suite consisting of the Transport Control Protocol/Internet Protocol (TCP/IP) can be used to support bursty data applications. TCP is a connection-oriented, reliable transport protocol that can run over any link layer. TCP uses congestion avoidance mechanism to adapt to changing network conditions.

High-bandwidth data applications may use the control load service to achieve higher throughput, since control load service provides better bandwidth than best-effort service.

ABR, GFR and UBR service categories of ATMF are designed to provide best-effort service. nrt-VBR may also be used for best effort service. UBR does not provide any guarantees. Hence, it is least suited to carry bursty data. ABR provides MCR guarantees, low cell loss, and feedback mechanism, which are useful to bursty data applications. GFR guarantees MCR that enables bursty data applications a minimum throughput.

Summary

The ATMF QoS methods are the most comprehensive and provide features to support the requirements of voice, video, and bursty data applications. Table 2.9 summarizes the features of various QoS methods which are useful in supporting voice, video, and bursty data applications.

Application	802.1D	IETF		ATMF
		Insterv/RSVP	Diffserv	
Voice	None	GS (bwd,delay, loss)	EF (bwd,delay, loss)	CBR, rt-VBR (AAL1, PCR, SCR, pp-CDV, maxCTD and CLR)
Video	None	GS (bwd,delay, loss)	EF (bwd,delay, loss)	CBR, rt-VBR (PCR, SCR, maxCTD and CLR)
Bursty Data	Priority	TCP/IP	-	ABR, GFR (MCR)

pp-CDV - peak-to-peak CDV, bwd - bandwidth

Table 2.9: Summary of features of QoS methods which are useful to support voice, video, and bursty data applications

2.8 Chapter Summary

In this chapter we have provided an overview of 802.1D, IETF and ATMF QoS methods. 802.1D standard uses the user_priority to provide qualitative QoS. Integrated service model has defined the guaranteed service and control load service in addition to the currently available best-effort service in the Internet. Guaranteed service provides bandwidth, delay and loss guarantees. Control load service aims to provide a QoS to applications which closely approximates the quality the application would receive from an unloaded network. RSVP is the signaling protocol to provide the resource reservation and admission control to support integrated services. QoS-aware routing protocols are being developed by IETF to support routing with QoS constraints. The differentiated services model provides service differentiation based on the DS field. This model is intended to be used on the backbones and Intserv/RSVP

is intended for edge networks. MPLS is designed to provide fast routing at the network layer by attaching a label to each packet that enters the MPLS domain. Inside the MPLS domain the packet is switched based on the label. Traffic engineering of MPLS can be used to optimize network resources and utilization that can lead to better support for QoS. ATM Forum has defined CBR, rt-VBR, nrt-VBR, ABR, GFR and UBR service categories to provide QoS support for a range of application from non-realtime to real-time.

We compared the above QoS methods along several dimensions including cost, availability, QoS nature, and granularity of QoS. Finally, we identified the features of the above QoS methods that can be useful to support voice, video, and bursty data applications.

In conclusion,

- The 802.1D QoS methods is most applicable to LAN environments.
- Intserv/RSVP methods have not been widely deployed due to scalability and complexity limitations. Diffserv and MPLS are currently being developed by IETF.
- The ATMF QoS methods provide rich support for QoS, but due to high cost and need for change in applications, native ATM environments have not widely deployed.

CHAPTER 3

BACKGROUND AND SURVEY OF WORK ON MULTIMEDIA OVER ABR

In this chapter we provide background material for understanding the ABR service and give an overview of the ERICA+ rate allocation algorithm. Then we discuss different fairness conditions and provide a survey of state of the art in ABR rate allocation schemes that provide rate guarantees in the presence of MCR.

3.1 Overview of ABR Service

The ABR service category is the only service category which uses closed-loop feedback for flow control. All other service categories have open loop flow control. In ABR, one resource management (RM) cell is sent for every $Nrm - 1$ (value of Nrm parameter is usually 32) data cells by the source. The source indicates its current source rate in the RM cell. The RM cell is turned around at the destination and sent back to the source (Figure 3.1). The switches along the RM cell path indicate the current maximum rate that they can support in the explicit rate field of the RM cell. The sources, in turn, adjust their rates accordingly.

To get a guarantee for the minimum amount of service the user can specify a minimum cell rate (MCR) in ATM ABR service that is used. The ABR service guarantees

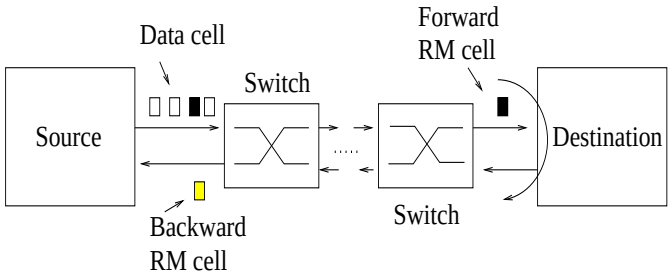


Figure 3.1: ABR flow control. RM cells are sent periodically by the source. The RM cell is turned around at the destination. The RM cells in the forward direction are called FRM cells and those in the backward direction are called BRM cells. The switches along the RM cell path indicate the rate that they can currently support.

that the allowed cell rate (ACR) is never less than MCR. When the MCR is zero for all sources, the available bandwidth can be allocated equally among the competing sources. This allocation achieves max-min (explained in section 3.2 fairness. When MCRs are non-zero, the ATM Forum TM 4.0 specification [42] recommends other definitions of fairness that allocate the excess bandwidth (which is the available ABR capacity minus the sum of MCRs) equally among sources, or proportional to MCRs. These definitions are discussed in detail in section 3.2.

3.1.1 Overview of ERICA+

ERICA+ algorithm (pseudo code given below) operates at the output port of a switch. It periodically monitors the load, active number of VCs and provides feedback in the BRM (backward RM) cells. The measurement period is called the "averaging interval." The measurements are done in the forward direction and feedback is given in the reverse direction. The complete description of ERICA+ algorithm can be obtained in reference [71]. In overload conditions, the algorithm calculates the maximum of the *FairShare* (link capacity divided by number of VCs) and *VCShare*

(current cell rate divided by overload) as the feedback rate. In underload conditions, the maximum of *MaxAllocPrev* (which is the maximum allocation given in the previous averaging interval) and the previous two terms is the feedback rate. Part of the available ABR capacity is used for draining queues. The Target ABR capacity is obtained by multiplying the total available ABR capacity by a $Fraction$ term. $1 - Fraction$ amount of the link capacity is used to drain the queues. $Fraction$ can be either a constant less than one (e.g., $Fraction = 0.9$ implies 90% link utilization), or dynamic function of switch queue length. ERICA+ uses two hyperbolic curves that control the queue in underload and overload regions. To ensure that a minimum amount of link capacity is used for data, the $Fraction$ value is limited to a minimum of $QDLF$ (queue drain limit factor). Typically a value $QDLF = 0.5$ is used. The dynamic queue control function given above can be expressed as follows

$$f(Q) = \begin{cases} \frac{bQ_0}{(b-1)Q+Q_0} & 0 \le Q \le Q_0 \\ min(\text{QDLF}, \frac{aQ_0}{(a-1)Q+Q_0}) & Q_0 < Q \le \infty \end{cases}$$

ERICA+ Algorithm

At the end of the Averaging Interval:

$$\begin{aligned}
\text{Total ABR Capacity} &\leftarrow \text{Link Capacity} - \text{VBR Capacity} \\
\text{Target ABR Capacity} &\leftarrow Fraction \times \text{Total ABR Capacity} \\
z &\leftarrow \frac{\text{ABR Input Rate}}{\text{Target ABR Capacity}} \\
\text{FairShare} &\leftarrow \frac{\text{Target ABR Capacity}}{\text{Number of Active VCs}} \\
\text{MaxAllocPrev} &\leftarrow \text{MaxAllocCur} \\
\text{MaxAllocCur} &\leftarrow \text{FairShare}
\end{aligned}$$

When an FRM is received:

$$\text{CCR[VC]} \quad \leftarrow \quad \text{CCR\_in\_RM\_Cell}$$

When a BRM is received:

$$\begin{aligned}
\text{VCShare} \quad &\leftarrow \quad \frac{CCR[VC]}{z} \\
\text{IF } (z > 1 + \delta) & \\
\text{THEN ER} \quad &\leftarrow \quad \max(\text{FairShare, VCShare}) \\
\text{ELSE ER} \quad &\leftarrow \quad \max(\text{MaxAllocPrev, VCShare}) \\
\text{MaxAllocCur} \quad &\leftarrow \quad \max(\text{MaxAllocCur,ER}) \\
\text{IF (ER} > \text{FairShare} \quad &\text{AND} \quad \text{CCR[VC]} < \text{FairShare)} \\
\text{THEN ER} \quad &\leftarrow \quad \text{FairShare} \\
\text{ER_RM_Cell} \quad &\leftarrow \quad \text{Min (ER_RM_Cell, ER, Target ABR Capacity)}
\end{aligned}$$

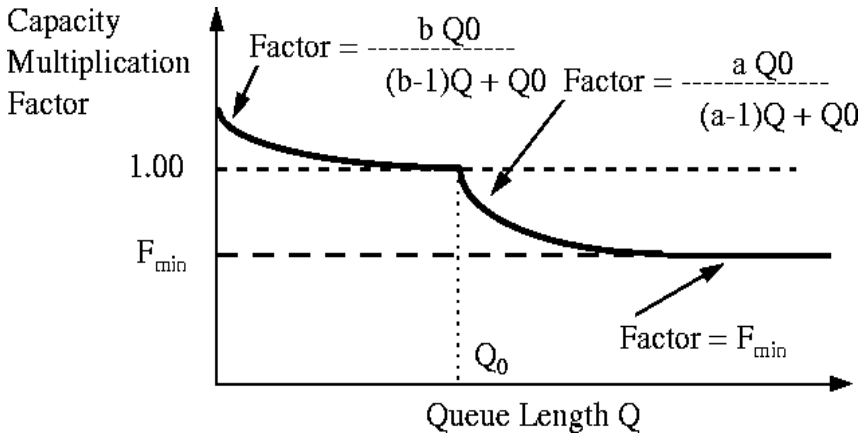


Figure 3.2: The dynamic queue control function used in ERICA+. F_{min} (QDLF) thresholds the amount of capacity used for queue draining. Q_0 is the target queue length, its value is dependent on the "Target delay" parameter and the link capacity.

The "Target Delay" parameter specifies the desired queue length (Q_0) at steady state. The 'a-curve' (hyperbola given by $\frac{aQ_0}{(a-1)Q+Q_0}$) is used as long as it evaluates to value greater than QDLF. The design of queue control functions which enable the switch algorithms to reduce rate oscillations, achieve constant delay and high link utilization (100%) is discussed in detail in chapter 7.

3.2 Fairness

One commonly used fairness condition is *max-min* allocation. Max-min allocation, calls for maximizing the allocation of the minimum rate source, i.e., to give the competing sources a "maximum possible equal share". Assume x_i is the rate of the ith source of the n contenting sources. An allocation vector $\{x_1, x_2, \ldots, x_n\}$ is said to be feasible if all link load levels are less than or equal to 100%. In max-min allocation the allocation vector is the largest vector among the feasible vectors in the lexicographic ordering. Max-min fairness assumes that the MCRs are zero.

In this section we give the various fairness definitions given in the ATM Forum traffic management (TM) 4.0 specifications and discuss an ATM Forum contribution which considers possible fair share policies in the context of MCR.

3.2.1 ATM Forum TM 4.0 Fairness definitions

The ATM Forum traffic management specification version 4.0 [42] defines several fairness definitions in section I.3 of its appendix.

Assume the following notation:

B The available bandwidth to be shared at the link

n Competing sources bottlenecked at this link

MCR_i Indicates the minimum cell rate for connection i

$M = \sum MCR_i$ Total amount for bandwidth used by MCRs of all connections

w_i Weight assigned to connection i

B_i The allocation to be calculated for connection i

The following are the possible fairness criterias:

Max-Min: The available bandwidth B is equally shared among the competing n connections

$$B_i = \frac{B}{n}$$

MCR plus equal share: The allocation in this case is the MCR plus equal share of the excess bandwidth that is obtained by removing the sum of the MCRs from bandwidth B

$$B_i = MCR_i + \frac{B - M}{n}$$

The definition converges to the Max-Min criteria as all MCRs approach zero.

Maximum of MCR or Max-Min share: The allocation in this case is the maximum among the values of MCR or Max-Min share

$$B_i = max(MCR_i, \text{Max-Min share})$$

This definition also converges to the Max-Min criterion as all MCRs approach zero.

Proportional to MCR: In this case, the allocation the is weighted proportional to its MCR.

$$B_i = \frac{MCR_i B}{M}$$

This criterion does not apply if there are connections with zero MCR.

Weighted allocation: The allocation is proportional to its pre-determined weight

$$B_i = \frac{w_i B}{\sum_j w_j}$$

The weight may be defined independent of MCR or dependent on MCR.

3.2.2 Fair Share in Context of MCR

The AF-TM/94-0977 ATM Forum contribution [62] is the earliest work to discuss various policies of fair share in the presence of MCR. The author defines four fair share policies namely, MCRadd, MCRmin, MCRprop and MCRlinear that led to various fairness definitions of the TM specifications.

MCRadd is the same as the "MCR plus equal share" definition: MCRmin is the same as the "Maximum of MCR or Max-Min share"; and MCRprop is the same as the "Proportional to MCR" definitions of the TM specifications.

MCRlinear is defined as follows:

$$B_i = MCR_i + w_i(B - M)$$

Here the fair share is MCR plus a share of the excess bandwidth according to a predetermined weight w_i. For example by having $w_i = b/n + (1-b)\frac{MCR_i}{M}$ a fraction b of the excess bandwidth $B-M$ is divided fairly, and the remaining portion $(1-b)(B-M)$ is divided in an MCR proportional manner.

The author gives several numerical examples to show the different allocation attained in each of the proposed policies. He argues that there is no "correct" definition of fair share and says it should be dictated by charging policy.

3.3 ABR Rate Control Algorithms with MCR Guarantees

There has been considerable research done in developing distributed rate control algorithms that achieved max-min fairness [71, 53, 69, 109, 4, 99]. An excellent survey of these is given in references [7, 70] and multipoint issues of these algorithms are discussed in reference [31]. In this section we survey the ABR rate control algorithms that provide MCR guarantees.

3.3.1 Stochastic Approximation Approach for Max-Min Fair

Abraham and Kumar have proposed an adaptive rate control algorithm with MCR guarantees [3]. The max-min fairness definition is extended to the case where MCRs are non-zero. The authors show that the lexicographically largest rate vector is a generalization of the max-min fairness definition.

The rate allocation problem is posed as a problem of finding roots of a non-linear equation. A definition of the link parameter is given and the rate allocation problem is expressed as an equation using this parameter. A stochastic approximation technique is used to find the root of this equation. This algorithm finds successively better approximations to the root (similar to the Newton-Ralphson root finding method) until the lexicographically largest allocation vector is found.

The variability of the link capacity (due the presence of CBR and VBR) traffic is modeled as a zero mean random process and the available capacity is viewed as the sum of a constant and a value obtained from this random process. Simulation results

of the algorithm show that the max-min fair allocation are achieved in the presence of both MCR and capacity variation. In the end of the chapter an analytical proof for the convergence of the rate control algorithm is given.

3.3.2 Generalized Max-Min Fair Flow Control Mechanism

Hou, Tzeng and Panwar define generalized max-min fairness criterion for ABR rate allocation and give a distributed algorithm to achieve that allocation [61]. The generalized max-min fairness definition is similar to one given in the paper discussed in the previous subsection.

The paper first gives a simple centralized algorithm for doing rate allocation. It works as follows: first find the most bottlenecked link and divide the rate fairly among the connections that share that link. Then remove that link from the problem set. Now the problem is of a reduced size and we apply the same approach recursively until the rate of all sessions are found.

The authors argue that the centralized approach is inefficient and also requires global knowledge. They propose a distributed algorithm for doing rate allocation by generalizing Charny's consistent marking technique. They claim that if the consistent marking technique is directly applied the rates will oscillate. They observe that this problem could be avoided by making sure a session is not marked at its bottleneck link. A distributed algorithm that implements this generalized consistent marking technique is given. Simulation results show that the algorithm converges to the generalized max-min fair rates with MCR guarantees. An analytical proof for the convergence of the algorithm is also given.

3.4 Chapter Summary

In this chapter, we gave an overview of the ABR feedback control mechanism followed by an description of the ERICA+ switch scheme. Then the various fairness definitions with MCR were discussed. Two distributed rate algorithms that guarantee generalized max-min fair rates in the presence of MCR were discussed in the end.

CHAPTER 4

PROBLEM STATEMENT AND METHODOLOGY

Providing quality of service guarantees to support multimedia applications is a challenging problem because several components including traffic management, routing and signaling, multicast, and operating systems support are necessary to provide QoS for multimedia applications. The main issue we propose to address in this thesis is the "development of traffic management methods that enhance quality of service delivered to multimedia applications that are transported over the ATM ABR service".

4.1 Problem Statement

Multimedia traffic requires that bandwidth, delay and loss be guaranteed within a range. A bandwidth bound is necessary to provide acceptable quality of service. Delay and loss bounds ensure timely delivery and prevent degradation of quality.

Formally, we state the requirement of multimedia traffic as follows:

1. **Bandwidth:** The quality of service is directly dependent on the bandwidth available to the application. One way to express the bandwidth requirement of a multimedia application is to specify a 3-tuple (β_{min},β_{avg},β_{max}), where β_{min}

is the minimum bandwidth, β_{avg} is average bandwidth and β_{max} is maximum bandwidth.

2. **Delay and delay variation:** The end-to-end delay should be bounded by the delay d_{max} for each packet. For delay sensitive (interactive) multimedia applications, this delay bound is strict so it cannot adapt. For non-delay sensitive applications, a probability bound is sufficient. A typical multimedia application, like video conferencing, sends frames containing video information periodically. Usually a clock is used at the source to encode timing information in the frame. The receiver should synchronize its clock to the timing information stored in the video. This imposes the restriction that the delay between consecutive frames should be bounded. The requirements of delay and delay variation can be expressed as a 3-tuple $(d_{min}, d_{avg}, d_{max})$.

3. **Loss:** The loss rate of packets has to be bounded to provide an acceptable quality of service. The loss rate is specified as a percentage of packets sent, and the loss requirement is expressed as a bound on that percentage.

Usually the CBR or VBR service is used to transport multimedia traffic [49]. We believe that a network feedback mechanism will be used by future adaptive multimedia applications to utilize resources efficiently. Therefore, we propose to use the ABR service, which has the closed loop feedback mechanism, and study the traffic management methods to enhance quality of service delivered to multimedia. The minimum rate guarantees provided by ABR service will provide a minimal acceptable quality of service. The low cell loss and feedback mechanism of ABR service

are additional features which enable transporting adaptable multimedia applications feasible [76, 73].

Specifically, we propose to address the following issues:

1. *Provide throughput continuity.* Multimedia applications such as video cannot sustain long durations over which no packets are transferred. Some guarantees against starvation are required. Currently, ABR provides a discontinuous service because cells may be queued at the source or the network for indefinite periods. Study of issues help to provide throughput continuity.

2. *Fairness.* In ABR service the excess bandwidth can be divided according to various fairness definitions. A study of the issues that use fairness criteria to give a bias towards multimedia applications when sharing excess bandwidth is necessary.

3. *Control delay by Queue Control.* One of the requirements of multimedia is that end-to-end delay should be bounded. Delay consists of various components such as processing delay, propagation delay and queuing delay. The variability portion of delay is due the queuing delay. We will study techniques for minimizing queuing delay by designing and implementing queue management policies to control the queue length.

4. *Performance analysis.* It is essential to perform comprehensive analysis of the mechanisms developed under various network conditions and traffic scenarios. The analysis should use metrics such as convergence time to fairness values, complexity, delay experienced, and buffer requirements to compare the various methods developed.

5. *Adaptive techniques available to multimedia applications.* Survey the state of the art of techniques that can be used by multimedia applications for adapting to network congestion. Explore how these techniques can be enhanced by using information from network feedback. This study is essential for designing future adaptive multimedia applications.

6. *Development of a testbed.* A flexible testbed is essential for testing and doing experimental analysis of methods we develop. The testbed should be software based to provide flexibility.

4.2 Proposed Approaches

We discuss several methods to solve each problem. We have used modeling and simulation techniques (which are explained further in next section) to implement and evaluate the different methods.

4.2.1 Throughput Continuity

The ABR service needs to provide throughput continuity requested by the multimedia applications. ABR can support a minimum cell rate (MCR) guarantee. The bandwidth requirement β_{min} can be mapped to MCR. β_{max} can be mapped to the peak cell rate (PCR) of a leaky bucket which is used at the source. β_{avg} cannot be guaranteed, except by reserving an MCR more than β_{avg}. It is preferable to use an MCR of at least β_{avg} since this guarantees an acceptable level of service. Reserving part of the link capacity to ABR service will enable it to provide bandwidth guarantees even in the presence of higher priority traffic such CBR and VBR.

We plan to develop switch rate allocation algorithms suitable for multimedia applications. We will begin by extending our ERICA+ algorithm [71]. ERICA+ monitors the load on each link and determines a load factor z, the ABR capacity, and the number of currently active virtual connections or VCs. It calculates the explicit rate it can support and indicates that rate in the backward RM cells.

Many switch algorithms including ERICA+ do not appropriately support sources with non-zero MCR requirements. We propose a general modification to support MCR guarantees as explained below:

1. We reduce the problem with non-zero MCRs to one with zero MCRs. We subtract MCR from each connection's current source rate to obtain the excess rate over MCR for each source.

2. The switch algorithm for zero MCRs is applied to these excess rates to obtain the feedback rate.

3. The MCR for each source is added to the explicit feedback rate calculated in the previous step. The resulting rate is indicated in the explicit rate field of the RM cell.

4.2.2 Fairness

When multiple sources are sharing the link, the bandwidth should be allocated in a fair manner to prevent one source from starving the other sources. In the ABR service the excess bandwidth should be divided in a fair manner on the fairness criteria. By using a different weight for multimedia ABR and other ABR connections we propose to bias the extra bandwidth towards the multimedia connections. We assume that

the multimedia application users are willing to pay more to get this excess amount. We propose to show how such a pricing policy can be achieved by using appropriate weight functions.

Assume the following notation:

B The available bandwidth to be shared at the link

n Competing sources bottlenecked at this link

MCR_i Indicates the minimum cell rate for connection i

$M = \sum_{i=1}^{n} MCR_i$ Total amount for bandwidth used by MCRs of all connections

w_i Weight assigned to connection i

B_i The allocation to be calculated for connection i

We propose to use a new generalized fairness definition (which was added due to our efforts in version 4.1 of the traffic management specifications under the name *MCR plus weighted share*)

The bandwidth allocation for a connection is its MCR plus a weighted share of the bandwidth B with the allocated MCRs removed.

$$B_i = MCR_i + \frac{w_i(B - M)}{\sum_j w_j}$$

Later, in chapter 5 we give a formal definition of fairness and show that the ATM Forurm TM specifications fairness definitions are a special case of this generalized definition. This definition has the advantage of inherently guaranteeing MCR allocation to connections.

4.2.3 Controlling Delay by Queue Control

The end-to-end delay can be controlled by properly managing the queues. We propose to develop a queue control function appropriate for multimedia applications.

An example of a queue control function is the hyperbolic function used in the ERICA+ switch algorithm. It has two hyperbolic curves: an 'a-curve' and 'b-curve'. A target delay parameter is used to control the delay. We propose to investigate several queue control functions as shown in figure 4.1. When these functions are used, the rates oscillate in the regions corresponding to curves 'a-curve' (overloaded) and 'b-curve' (under loaded) and stabilize it with a queue length within the steady state region. In the steady state, the rates are constant and the queue lengths, and hence the delays, are constant. Other possible queue control functions are simple threshold, linear, and hysteresis functions. The delay variation (network jitter) can be absorbed, by using buffering at the receiver end.

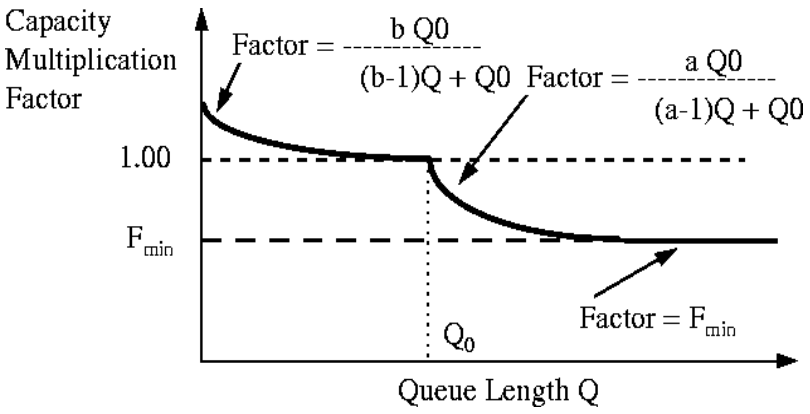


Figure 4.1: Hyperbolic Queue control function used in ERICA+. The 'a-curve' is used in the overloaded ($Q > Q_0$) region. The 'b-curve' is used in underloaded ($Q < Q_0$) region. They meet at a point where *Factor* value is one.

4.3 Performance Analysis

To compare the performance and study the tradeoffs of the various methods, it is essential to perform comprehensive performance analysis. The analysis will also be helpful in finding the appropriate values of the various parameters, such as the amount of buffer space to allocate, the MCR value, and the queue control function. We will compare different methods using convergence time to steady state and fairness as metrics.

The proposed schemes will be analyzed using simulation for various parameter values and different network configurations. This sensitivity analysis of parameters will show how the proposed schemes are affected. It will also enable us to find the correct value of parameters to be used for each scheme.

The configurations will be chosen to test the performance of the schemes in the presence of link bottlenecks, source bottlenecks, transient sources, and multiple round trip times. We also use bursty background traffic to approximate real-world environments when testing the rate allocation algorithms.

4.4 Development Of Testbed

In order to study and evaluate the methods developed it is essential to implemented these in a flexible testbed environment. It is difficult to use hardware ATM switches to develop a flexible testbed because they are costly and their code is proprietary. Therefore, we plan to build a software based testbed. Following will be main features of the testbed:

1. We propose to use cost effective components such as PCs and off-the shelf ATM cards. We plan to use the Linux operating system since its source code is freely available and the device driver for ATM cards is available.

2. *Software Switch:* Figure 4.2 shows an n-port software switch. It consists of a $n \times n$ commercial switch whose output ports are connected to personal computers (PCs). Each PC has two ATM network interface cards. The link speed of one card matches that of the switch, while the other card runs at a lower speed. As shown in the figure, the input speed is 155 Mbps and the output speed is 45 Mbps. The PCs have rate allocation software (e.g., ERICA+) and have queuing and scheduling techniques that we want to experiment with. Note that the PCs act like a 1×1 switch. The input and output rates of the commercial switch match and the traffic is arranged so that there is no queuing in the commercial switch. The output rate of the PCs is lower than the input rate and so all queuing, if any, takes place in the PCs. That is, the PCs are the bottlenecks. The rate allocation in PCs and queuing and scheduling in PCs have a significant impact on the traffic while those in the commercial switch have negligible impact. Thus, we can easily vary the queuing and rate allocation strategies and see their impact on real traffic.

4.5 Research Methodology

As mentioned in the computing research associates report [110] we believe that networking research and development is an iterative process. It is a "spiral design" requiring continuous feedback from network researchers, industry people, standards bodies and researchers developing network applications. We have conducted our

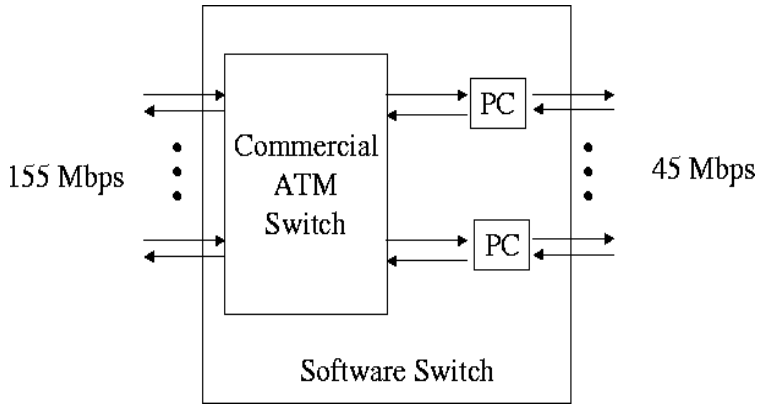


Figure 4.2: Software switch

research in the above manner by using a combination of simulation modeling, analysis and experimental design and closely working with standards bodies such as the ATM Forum.

Simulation Modeling

We use discrete event-driven simulation to model the various components of the ATM network. The simulation tool we use is an extensively modified version of the *netsim* simulator [54] developed at NIST. The simulation modeling is done at the cell level granularity. This model closely reflects the operation of a real ATM network. Another benefit of doing the simulation at the cell level was that we were able to reuse the code in developing our software switch. The simulation tool, written in the C programming language, models the various components of ATM networks such as sources, destinations, links, and switchs. We implement our methods by modifying these components appropriately. Specifically, for this project we have written:

- Source components that generate infinite, self-similar, poisson and on-off traffic patterns.

- Switch component to implement the generalized fair rate allocation algorithms and the proposed queue control functions.

Analytical Explanation

For each problem and scheme developed, we first do a preliminary analysis before performing simulation. The initial results from the simulation are then used to refine the methods developed. This iterative process is helpful in validating both simulation model and algorithms developed. Throughout the research process we strive to provide adequate analytical explanations for the algorithm behavior and, when appropriate, also provide formal proofs of properties of the algorithm such as convergence time.

Dissemination of Research

An important and distinguishing feature of our research is that we work closely with standards bodies, such as the ATM Forum, to ensure that our research is disseminated to industry in a timely manner. All of our work was initially presented as contributions at ATM Forum meetings. We benefited from the feedback provided, which improved our schemes and made sure that our methods are complaint to the latest standards. Our contributions have also impacted the ATM Forum traffic management standard specifications. For example, our new definition of general fairness used in the rate allocation scheme has been included in the current ATM Forum specifications.

4.6 Chapter Summary

We have defined the problem of designing traffic management methods to enhance the quality of service delivered to multimedia traffic that are transported over the

ATM ABR service. The problem domain can be broadly grouped as follows, and each of these is discussed in detail in subsequent chapters of this dissertation.

1. Extending current ABR schemes to provide throughput continuity and fairly dividing the excess bandwidth.

2. Designing new ABR schemes to provide throughput continuity.

3. Design and analysis of queue control algorithms to reduce queuing delay.

4. Study of state of the art techniques that can be used by multimedia applications to adapt to network changes.

5. Implementation of a software-based testbed for experimental analysis of the methods developed.

CHAPTER 5

GENERALIZED FAIRNESS AND MCR GUARANTEES

This chapter gives a new definition of general weighted (GW) fairness and shows how it can achieve various fairness definitions, such as those mentioned in the ATM Forum TM 4.0 specifications. The GW fairness can be achieved by calculating the $ExcessFairshare$ (weighted fairshare of the left over bandwidth) for each VC. We show how a switch algorithm can be modified to support the GW fairness by using the $ExcessFairshare$ term. The MCR guarantee and the generalized fairness enhance the available bandwidth and can be used for achieving the throughput continuity required by multimedia ABR connections. We use ERICA+ as an example switch algorithm and show how it can be modified to achieve the GW fairness. For simulations, the weight parameters of the GW fairness are chosen to map a typical pricing policy. Simulation results are presented to demonstrate that the modified switch algorithm achieves GW fairness. An analytical proof for convergence of the modified ERICA+ algorithm is given in the end.

5.1 Introduction

We present a more generalized definition of sharing the excess bandwidth using predetermined weighted than the one recommended in ATM Forum specifications [42].

In the real world, the users prefer to get a service that reflects the amount they are paying. The pricing policy requirements can be realized by appropriately mapping the policy to the weights associated with the sources.

The specification of the ABR feedback control algorithm (switch algorithm) is not yet standardized. The earliest algorithms used binary feedback techniques [129]. Distributed algorithms [82] that emulated a centralized algorithm were proposed in [20, 116]. Other simpler distributed algorithms which achieved max-min fairness were proposed in [71, 53, 69, 109, 4, 99].

Recently, a discussion on a generalized definition of max-min fairness and its distributed implementation was given in [3, 61]. A weight-based max-min fairness policy and its implementation in ABR service is given in [60]. The fairness in the presence of MCR guarantees is discussed in [62, 128]. All these works have been discussed earlier in chapter 3.

We show how the generalized fairness can be achieved by allocating the excess bandwidth proportional to weights associated with each source. We also show how a switch scheme can support non-zero MCRs. As an example, we explain how the ERICA+ [68] switch scheme can be modified to support the GW fairness and support MCR guarantees.

The modified scheme is tested using simulations with various network configurations. The simulations test the performance of the modified algorithm with different weights using a simple configuration, a transient source configuration, a link bottleneck configuration, and a source bottlenecked configuration. Scalability and robustness are tested using a configuration with one hundred TCP sources and a background VBR connection carrying long range dependent traffic. These simulations show that

the scheme realizes various fairness definitions in ATM TM 4.0 specification that are special cases of the generalized fairness.

5.2 General Weighted Fairness: Definition

We first define the following parameters:

$A_l =$ Total available bandwidth for all ABR connections on a given link l.

$A_b =$ Sum of bandwidth of under-loaded connections that are bottlenecked elsewhere.

$A = A_l - A_b$, excess bandwidth, to be shared by connections bottlenecked on this link.

$N_a =$ Number of active connections.

$N_b =$ Number of active connections bottlenecked elsewhere.

$n = N_a - N_b$, number of active connections bottlenecked on this link.

$\mu_i =$ MCR of connection i.

$\mu = \sum_{i=1}^{n} \mu_i$ Sum of MCRs of active connections bottlenecked at this link.

$w_i =$ preassigned weight associated with the connection i.

$g_i =$ GW fair allocation for connection i.

The general weighted fair allocation is defined as follows:

$$g_i = \mu_i + \frac{w_i(A - \mu)}{\sum_{j=1}^{n} w_j}$$

Note that, this definition of fairness is different from the weighted allocation given as an example fairness criterion in ATM TM 4.0 specifications. In the above definition,

only the excess bandwidth is allocated proportional to weights. The above definition ensures the allocation is at least MCR.

5.2.1 Mapping TM 4.0 Fairness to General Weighted Fairness

Here we show how the different fairness criteria mentioned in ATM TM 4.0 specification (discussed earlier in section 3.2) can be realized using the above fairness definition.

1. **Max-Min:** In this case, MCRs are zero and the bandwidth is shared equally.

$$g_i = A/n$$

   This is a special case of general weighted fairness with $\mu_i = 0$, and $w_i = c$, where c is a constant.

2. **MCR plus equal share:** The excess bandwidth is shared equally.

$$g_i = \mu_i + (A - \mu)/n$$

   By assigning equal weights we achieve the above fairness.

3. **Proportional to MCR:** The allocation is proportional to its MCR.

$$g_i = \frac{A \times \mu_i}{\mu} = \frac{(\mu + A - \mu)\mu_i}{\mu} = \mu_i + \frac{(A - \mu)\mu_i}{\mu}$$

   By assigning $w_i = \mu_i$ to all flows we can achieve the above fairness.

5.3 Relationship to Pricing/Charging Policies

In the real world, users expect a service proportional to the price they are paying for the service. In this section we discuss a simple pricing policy and arrive at a weight function to support such a policy.

Consider a very small interval T of time. The charge C that a customer pays for using a network during this interval is a function of the number of bits W that the network transported successfully:

$$C = f(W, R)$$

Where, $R = W/T$ is the average rate.

It is reasonable to assume that $f()$ is a non-decreasing function of W. That is, those sending more bits do not pay less. The function $f()$ should also, be a non-increasing function of time T or equivalently a non-decreasing function of rate R.

For economy of scale, it is important that the cost per bit does not increase as the number of bits goes up. That is, C/W is a non-decreasing function of W.

Mathematically, we have three requirements:

$$\partial C/\partial W \geq 0$$

$$\partial C/\partial R \geq 0$$

$$\partial (C/W)/\partial W \leq 0$$

One simple function that satisfies all these requirements is:

$$C = c + wW + rR$$

Here, c is the fixed cost per connection; w is the cost per bit; and r is the cost per Mbps. In general, c, w, and r can take any non-negative value.

In the presence of MCR, the above discussion can be generalized to:

$$C = f(W, R, M)$$

Where, M is the MCR. All arguments given above for R apply to M also except that the customers requesting larger M possibly pay more. One possible function is:

$$C = c + wW + rR + mM$$

where, m is dollars per Mbps of MCR. In effect, the customer pays $r + m$ dollars per Mbps up to M and then pays only r dollars per Mbps for all the extra bandwidth he/she gets over and above M.

Consider two users with MCRs M_1 and M_2. Suppose their allocated rates are R_1 and R_2 and, thus, they transmit W_1 and W_2 bits, respectively. Their costs are:

$$C_1 = c + wW_1 + rR_1 + mM_1$$

$$C_2 = c + wW_2 + rR_2 + mM_2$$

Cost per bit (C/W) should be a decreasing function of bits W. Thus, if $W_1 \geq W_2$:

$$C_1/W_1 \leq C_2/W_2$$

$$c/W_1 + w + rR_1/W_1 + mM_1/W_1 \leq c/W_2 + w + rR_2/W_2 + mM_2/W_2$$

Since $R_i = W_i/T$, we have:

$$c/(R_1T) + w + r/T + mM_1/(R_1T) \leq c/(R_2T) + w + r/T + mM_2/(R_2T)$$

$$c/R_1 + mM_1/R_1 \leq c/R_2 + mM_2/R_2$$

$$(c + mM_1)/(c + mM_2) \leq R_1/R_2$$

$$(a + M_1)/(a + M_2) \leq R_1/R_2$$

Where a ($=c/m$) is the ratio of the fixed cost and cost per unit of MCR.

Note that the allocated rates should either be proportional to a+MCR or be a non-decreasing function of MCR. We have chosen to use a+MCR as the weight function in our simulations.

5.4 General Weighted Fair Allocation Problem

In this section we give the formal specification of the general weighted fair allocation problem and give a motivation for the need of a distributed algorithm.

The following additional notation is necessary:

$\mathcal{L} =$ Set of links, $\mathcal{L}_s$ set of links that session s goes through.

$\mathcal{S} =$ Set of sessions, $\mathcal{S}_l$ set of sessions that go through link l. $N =\mid S \mid$.

$\mathcal{A} = (\mathcal{A}_l, l \in \mathcal{L})$ set of of available capacity.

$\mathcal{M} = (\mu_s, s \in S)$, where μ_s is the minimum cell rate (MCR) for session s.

$\mathcal{W} = (w_1, w_2, \ldots, w_N)$ denotes the weight vector.

$\mathcal{R} = (r_1, r_2, \ldots, r_N)$ the current allocation vector (or rate vector).

$\mathcal{G} = (g_1, g_2, \ldots, g_N)$ the general fair allocation. $\mathcal{G}_{\mathcal{S}_l}$ denotes the set of allocations of sessions going over link l

Definition 1 *General Weighted Fair Allocation Problem*

The GW fair problem is to find the rate vector equal to the GW fair allocation, i.e., $\mathcal{R} = \mathcal{G}$. *Where* $g_i \in \mathcal{G}_{\mathcal{S}_l}$ *is calculated for each link* l *as defined in the section 5.2.*

Note the 5-tuple $(\mathcal{S}, \mathcal{L}, \mathcal{C}, \mathcal{W}, \mathcal{R})$ represents an instant of the bandwidth sharing problem. When all weights are equal the allocation is equivalent to the general max-min fair allocation as defined in reference [3]. A simple centralized algorithm for solving the above problem would be to first find the correct allocation vector for the bottleneck links. Then, solve the same problem of smaller size after deleting

bottleneck links. A similar kind of centralized, recursive algorithm is discussed in reference [61]. The centralized algorithm implies that all information is known at each switch, which is not feasible, hence a distributed algorithm is necessary.

5.5 Achieving General Fairness

A typical ABR switch scheme calculates the excess bandwidth capacity available for best effort ABR after reserving bandwidth for providing MCR guarantee and higher priority classes such as VBR and CBR. The switch fairly divides the excess bandwidth among the connections bottlenecked at that link. Therefore, the ACR can be represented by the following equation.

$$ACR(i) = \mu_i + ExcessFairshare(i)$$

$ExcessFairshare$ is the amount of bandwidth allocated over the MCR in a fair manner.

In the case of GW fairness, the $ExcessFairshare$ term is given by:

$$ExcessFairshare(i) = \frac{w_i(A - \mu)}{\sum_{j=1}^{n} w_j}$$

If the network is near steady state (input rate = available capacity), then the above allocation enables the sources to attain the GW fairness. The ATM TM 4.0 specification mentions that the value of $(ACR - MCR)$ can be used in the switch algorithms. We use this term to achieve GW fairness. We have to ensure the $(ACR - MCR)$ term converges to $ExcessFairshare$ value. We use the notion of *activity level* to achieve the above objective [32]. A connection's *excess activity level* $(EAL(i))$ is defined as follows:

$$EAL(i) = minimum\left(1, \frac{max(0, SourceRate(i) - \mu_i)}{ExcessFairshare(i)}\right)$$

$SourceRate(i)$ is the rate at which the source is currently transmitting data. Note that $SourceRate(i)$ is the $ACR(i)$ given as the feedback rate earlier by the switch. The excess activity level indicates how much of the $ExcessFairshare$ is actually being used by the connection. The excess activity level is zero if $SourceRate(i)$ is less than μ_i. The activity level attains the value of one when the $ExcessFairshare$ is used by the connection. It is interesting to note that using the activity level for calculating is similar to the Charny's [20] *consistent marking* technique, where the switch marks connections which have a lower rate than their *advertised rate*. The new advertised rate is calculated using the equation:

$$\text{Advertised Rate} = \frac{\mathcal{A}_l - \sum \text{Rates of marked connections}}{\mid S_l \mid - \sum \text{Marked connections}}$$

The activity level inherently captures the notion of marking, i.e., when a source is bottlenecked elsewhere, then activity level times the fairshare (based on available left over capacity) is the actual fairshare of the bottleneck source. The computation of activity level can be done locally and is an $O(1)$ operation, compared to $O(n)$ computations required in consistent marking [21].

We expect that the links use their $ExcessFairshare$, but this might not be case. By multiplying the weights by the activity level and using these as the weights in calculating the $ExcessFairshare$, we can make sure that the rates converge to the GW fairness allocation. Therefore, the $ExcessFairshare$ share term is defined as:

$$ExcessFairshare(i) = \frac{w_i(A - \mu)}{\sum_{j=1}^{n} w_j EAL(j)}$$

Note that w_i is not multiplied by $EAL(i)$ in the numerator, since we desire to attain a value of $EAL(i) = 1$ for all sources and give excess bandwidth in proportion

to the weights. Due to this, sources which have not yet achieved their fairshare are asked to increase their rate to $ExcessFairShare$.

The rate of sources which are bottlenecked elsewhere are not affected. The rate of such a source, depends only on the explicit feedback rate that it receives from switches at which it is bottlenecked. Connections which are bottlenecked at sources also receive the correct amount of $ExcessFairShare$.

There is a possibility that the denominator becomes zero in the above expression, when all the sources are inactive ($SourceRate(i) < \mu_i$). In this case, the $ExcessFairshare$ evaluates to an infinite value. This means that since there is no load on the link, each source can have the whole available capacity as the fairshare. So, in our simulations we handle this special case by taking the minimum of available bandwidth ($A - \mu$) and value calculated by the above expression.

A switch algorithm can use the above $ExcessFairshare$ term to achieve the GW fairness. In the next section we show how the ERICA+ switching algorithm is modified to achieve the GW fairness.

5.6 Example Modifications to a Switch Algorithm

The ERICA+ algorithm operates at each output port of a switch. The switch periodically monitors the load on each link and determines a load factor (z), the available ABR capacity, and number of currently active sources or VCs. The measurement period is the "Averaging Interval". These measurements are used to calculate the feedback rate which is indicated in the backward RM (BRM) cells. The measurements are done in the forward direction and the feedback is given in the

Algorithm GWFairERICA+
At the end of the Averaging Interval:

$$\begin{aligned}
\text{Total ABR Capacity} &\leftarrow \text{Link Cap} - \text{VBR Cap} - \sum_{i=0}^{n} \min(SourceRate(i), \mu_i) \\
\text{Target ABR Capacity} &\leftarrow Factor \times \text{Total ABR Capacity} \\
\text{Input Rate} &\leftarrow \text{ABR Input Rate} - \sum_{i=0}^{n} \min(SourceRate(i), \mu_i) \\
z &\leftarrow \frac{\text{Input Rate}}{\text{Target ABR Capacity}} \\
&\textbf{foreach VC i} \\
\text{EAL(i)} &\leftarrow \frac{\min(1, \max(0, SourceRate(i) - \mu_i))}{ExcessFairshare(i))} \\
\text{SumOfWts} &\leftarrow \text{SumOfWts} + w_i EAL(i) \\
&\textbf{endfor} \\
&\textbf{foreach VC i} \\
\text{ExcessFairshare(i)} &\leftarrow \frac{(\text{Target ABR Capacity})w_i}{\text{SumOfWts}} \\
&\textbf{endfor}
\end{aligned}$$

Figure 5.1: GWFairERICA+ algorithm pseudo code for end of averaging interval computations

backward direction. The complete description of the ERICA+ algorithm is given in reference [68].

The ERICA+ algorithm uses the term $FairShare$, which is the bottleneck link capacity divided by the active number of VCs. It also uses a $MaxAllocPrevious$ term, which is the maximum allocation in the previous "Averaging Interval". This term is used to achieve max-min fairness. We modify the algorithm by replacing the $FairShare$ term by $ExcessFairshare(i)$ and adding the MCR (μ_i). The keys steps in ERICA+ which are modified to achieve GW fairness are shown below:

When a BRM is received:

$$\begin{aligned} \text{VCShare} &\leftarrow \frac{\max(0, SourceRate(i) - \mu_i)}{z} \\ \text{ER} &\leftarrow \mu_i + \max(\text{ExcessFairshare(i)}, \text{VCShare}) \\ \text{ER}_{RM\_Cell} &\leftarrow \min(\text{ER}_{RM\_Cell}, \text{ER}, \text{Target ABR Capacity}) \end{aligned}$$

Figure 5.2: Pseudo code for computations done when the first BRM cell in the averaging interval is received

The $Factor$ term is dependent on the queue length [118]. When the $Factor$ is less than one, $(1 - Factor) \times$ Total ABR Capacity is used to drain the queues. A simple choice is to use a constant queue control function (CQF), where the $Factor$ is set to a value less than 1, say 0.95. The remaining 5% of the link capacity is used for queue draining. Another option is to use a dynamic queue control function (DQF). In DQF, the $Factor$ value is one for small queue lengths and drops sharply as queue length increases. ERICA+ uses a hyperbolic function for calculating the value of the $Factor$ (Figure 3.2).

Figure 5.2 gives the pseudo code for computations done when a BRM cell is received. The $VCShare$ is used to achieve an unit overload. When the network reaches steady state the $VCShare$ term converges to $ExcessFairshare(i)$, achieving the generalized fairness criterion. The complexity of the computations done at the switching interval is O(number of VCs). The update operation when the BRM cell arrives is an $O(1)$ operation. Proof of convergence of algorithm GWFairERICA+ is given in the end this chapter.

5.7 Simulation Configurations

We use different configurations to test the performance of the modified algorithm. We assume, unless specified otherwise, that the sources are greedy, i.e., they have infinite amount of data to send and always send data at ACR. In all configurations the data traffic is unidirectional, from source to destination. If bidirectional traffic is used, similar results will be achieved, except that the convergence time will be longer since the RM cells in the backward direction will travel along with the data traffic from destination to source. All the link bandwidths are 149.76 (155.52 less the SONET overhead), expect in the GFC-2 configuration.

5.7.1 Three Sources

This is a simple configuration in which three sources send data to three destinations over two switches and a bottleneck link (Figure 5.3). This configuration is used to demonstrate that the modified switch algorithm can achieve the general fairness for different set of weight functions.

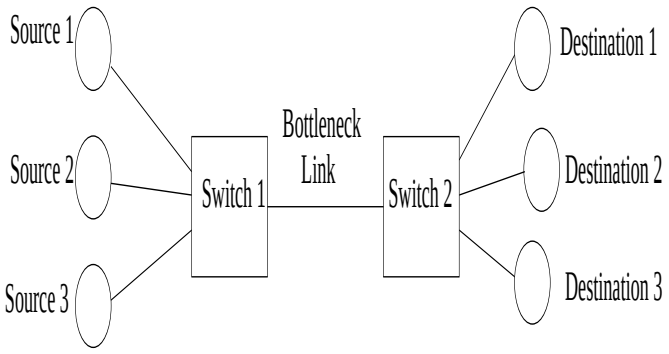


Figure 5.3: N Sources - N Destinations Configuration

5.7.2 Source Bottleneck

In this configuration (Figure 5.4), the fairshare for the sources is 50 Mbps since *Link*2 (150 Mbps) is shared by the three sources. But the source S1, is bottlenecked (at source) at 10 Mbps, which is below its fairshare (50 Mbps). That is, it cannot send data at rate more than 10 Mbps. This configuration tests whether the GW fairness can be achieved in the presence of source bottleneck.

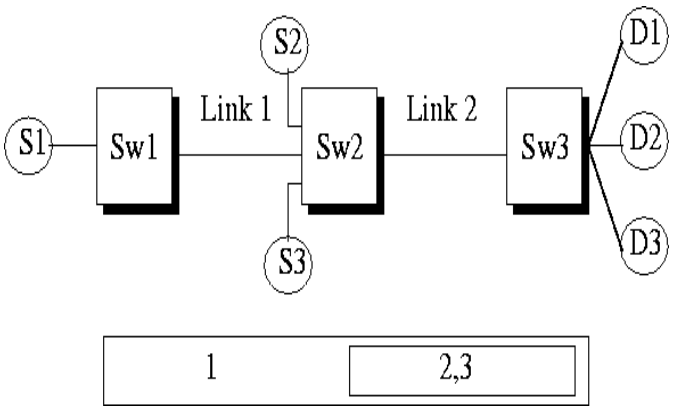
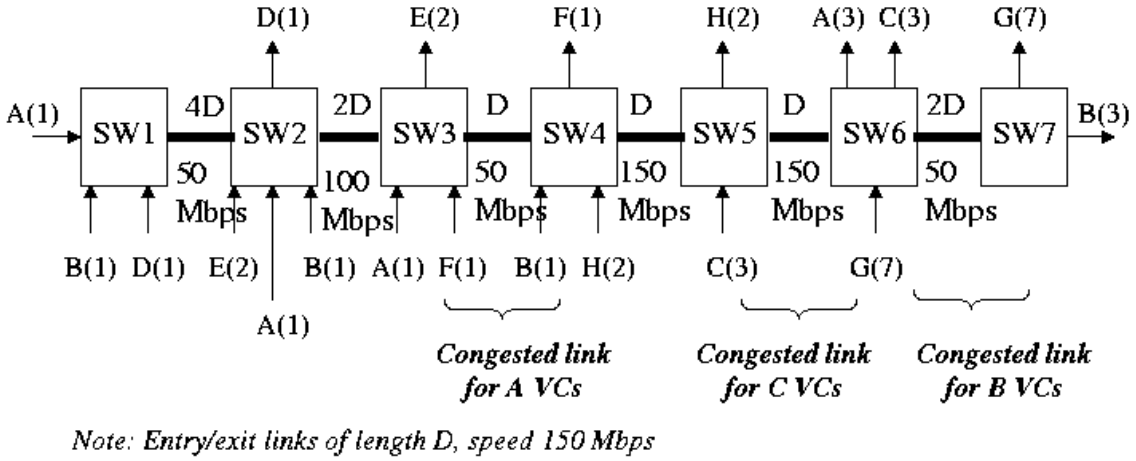


Figure 5.4: 3 Sources - Bottleneck Configuration. S1 is bottlenecked at 10 Mbps at source.

5.7.3 Generic Fairness Configuration - 2 (GFC-2)

This configuration (explained in detail in reference [108]) is a combination of the upstream and the parking lot configuration (Figure 5.5). In this configuration, all the links are bottlenecked links and round trip times are different for different type of VCs.



X(n) indicates that there 'n' number of VCs of type X.

Figure 5.5: Generic Fairness Configuration - 2.

5.7.4 TCP Sources with VBR Background

This configuration is used to test the robustness and scalability of the algorithm (Figure 5.6). In this configuration, one hundred infinite TCP sources (large file transfers) transmit data continuously through a bottleneck link to one hundred destinations. One VBR connection carrying generated self-similar traffic which models multiplexed MPEG traffic, is used as background traffic [72]. The mean bandwidth of the VBR traffic is 45 Mbps. The VBR traffic is generated with hurst parameter (H) value of 0.9, hence it has high degree of self-similarity.

5.7.5 Simulation Parameters

The simulations were done on an extensively modified version of the NIST simulator [54]. The following parameter values were used in all our simulations: Link

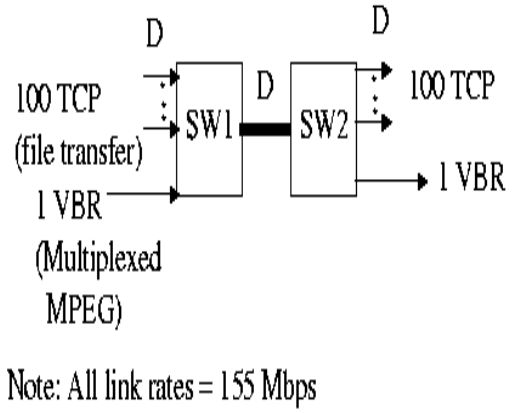


Figure 5.6: 100 TCP sources + VBR background configuration. All TCP sources are infinite sources. VBR connection carries multiplexed MPEG traffic that exhibits long range dependency.

distance = 1000 Km; Averaging interval = 5 ms; Target delay = 1.5 ms; Exponential decay factor = 0.1 (when using dynamic queue control function).

The "Averaging Interval" is the period for which the switch monitors various parameters. Feedback is given based on these monitored values. The ERICA+ algorithm uses dynamic queue control to vary the available ABR capacity dependent on queue size. At steady state the queue length of constant value can be obtained. The "Target Delay" parameter specifies the desired delay due to this constant queue length at steady state. When using dynamic queue control function we exponentially average $ExcessFairshare$ term. This is done so that effectively only one feedback is given in each interval and to absorb the variation in "Target ABR Capacity" value,

due to the queue control function. For convergence, the feedback delay, averaging interval and exponential averaging decay factor should obey the following equation:

$$\frac{\text{Averaging Interval}}{\text{Exponential Decay factor}} \geq \text{Feedback delay}$$

This ensures that at least one feedback is given in each feedback delay period.

5.8 Simulation Results

In this section we present the simulation results for the different configurations. The simulation results using both a constant queue control function (shown in graphs as configuration name and CQF) and a dynamic queue control function (shown in graphs as configuration and DQF) are given. For the CQF the value of $Factor$ used is 0.9. The tabular results are those obtained from simulations using the dynamic queue control function.

5.8.1 Three Sources

Simulations using a number of weight functions were done using the simple three sources configuration to demonstrate that GW fairness is achieved in all these cases. Let (r_1, r_2, r_3) indicate the rates of the sources. The ICRs (initial cell rate) of the sources were set to (50,40,55) Mbps in all the simulations.

The allocations of these cases using DQF are given in Table 5.1. The following can be observed from Table 5.1.

- Case 1: a = ∞, MCRs = 0. All weights are equal so the allocation (149.76/3) = 49.92 Mbps for each connection. This allocation is the same as max-min fair allocation.

Case #	Src #	mcr	a	weight function	Expected fair share	Actual share
1	1	0	∞	1	49.92	49.92
	2	0	∞	1	49.92	49.92
	3	0	∞	1	49.92	49.92
2	1	10	∞	1	29.92	29.92
	2	30	∞	1	49.92	49.92
	3	50	∞	1	69.92	69.92
3	1	10	5	15	18.54	18.53
	2	30	5	35	49.92	49.92
	3	50	5	55	81.31	81.30

Table 5.1: Three sources configuration simulation results

- Case 2: a = ∞, MCRs $\neq$ 0. The left over capacity 149.76 - (10 + 30 + 50) = 59.76 Mbps is divided equally among the three sources. So the allocation is (10 + 19.92, 30 + 19.92, 50 + 19.92) = (29.92,49.92,69.92) Mbps.

- Case 3: a = 5, MCRs $\neq$ 1. Hence, the weight function is 5 + MCR. The left over capacity, 59.76 Mbps, is divided proportional to (15,35,55). Hence the allocation is (10 + 15/105 $\times$ 59.76, 30 + 35/105 $\times$ 59.76, 50 + 55/105 $\times$ 59.76) = (18.54, 49.92, 81.31) Mbps.

Figure 5.7 shows the ACRs, queue length and utilization graphs of the three sources for case 3 using a constant queue control function. Figure 5.8 shows the corresponding graphs using a dynamic queue control function. From the figures one can observe that the sources achieve the GW fairness rate and that the queues are controlled in steady state. When using DQF, queue length values oscillate before

reaching steady state values. The utilization achieved at steady state is 100% when using DQF and 90% (same as *Factor* value) when using CQF.

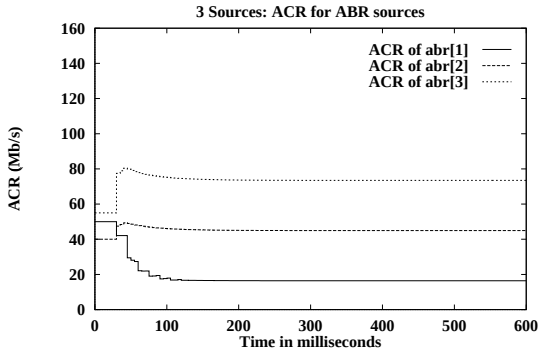
5.8.2 Three Sources: Transient

In these simulations the same simple three source configuration is used. Source-1 and source-3 transmit data throughout the simulation period. Source-2 is a transient source, which starts transmitting at 400 ms and stops at 800 ms. The total simulation time is 1200 ms. The same parameters values from the cases 1, 2 and 3 of the previous section were used in these simulations. The results of these simulations are given in Table 5.2. The non-transient (ntr) columns give the allocation when transient source-2 is not present, i.e., between 0ms to 400ms and between 800 ms to 1200 ms. The transient (tr) column gives the allocation when transient source-2 is present (i.e., between 400 ms to 800 ms).

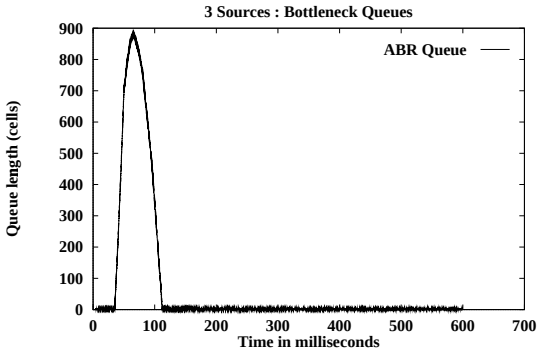
The ACR values of the sources and the utilization of the bottleneck link for case 2 are shown in figure 5.9. It can be seen both from the Table 5.2 and the graphs that the switch algorithm does converge to the general fairness allocation even in the presence of transient sources. The algorithm has a good response time, since there is only a small dip in the utilization graph when the transient source stops sending traffic (at 800 ms).

5.8.3 Source Bottleneck

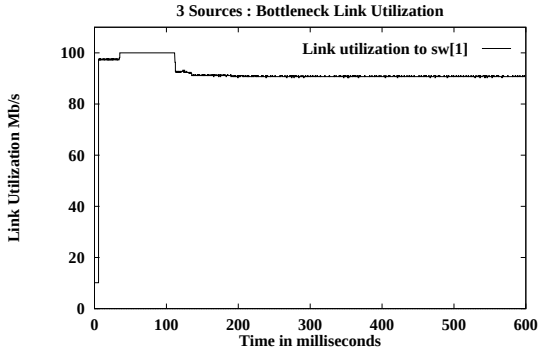
Cases 1, 2 and 3 of section 5.8.1 were simulated using the three sources bottleneck configuration. The total simulation time was 800 ms. In these simulations the source S1 is bottlenecked at 10 Mbps for the first 400 ms, i.e., it always transmits data at a



(a) ACRs



(b) Queue length

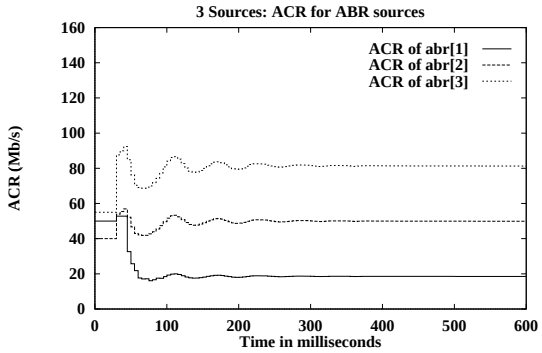


(c) Utilization

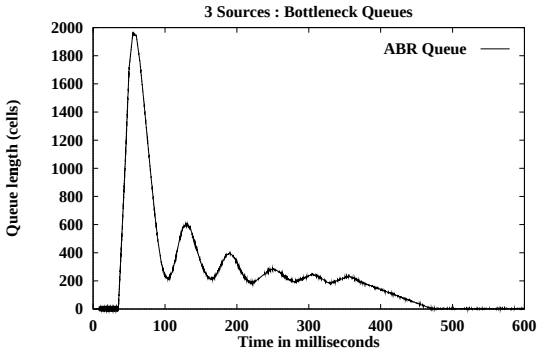
Figure 5.7: Three sources: Case 3 + CQF simulation results

rate of at most 10 Mbps, irrespective of its ACR (and ICR). After 400 ms, source S1 behaves like an infinite source and sends data at its ACR.

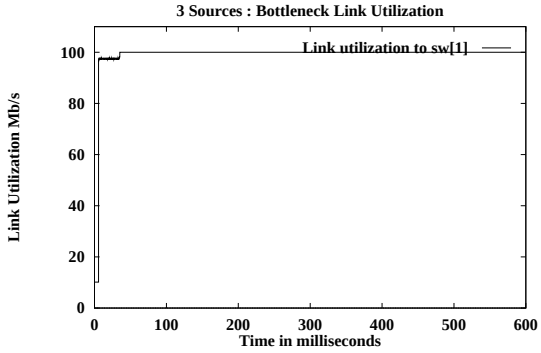
The initial ICRs were set to 50, 30, and 110 Mbps. The load on the bottleneck link is near unity. If the switch algorithm uses the CCR (current cell rate) value indicated in the RM cell as the source rate the switch cannot estimate the correct value of source rate, of the bottleneck source. But if the switch uses measured source rate then it can correctly estimate the bottlenecked source's rate. Table 5.3 shows the results both when the switch uses the CCR field and when it measures the source rate,



(a) ACRs

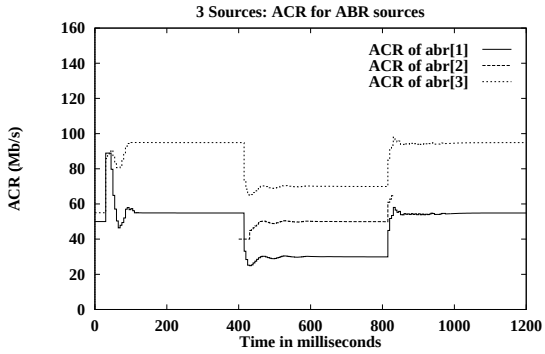


(b) Queue length

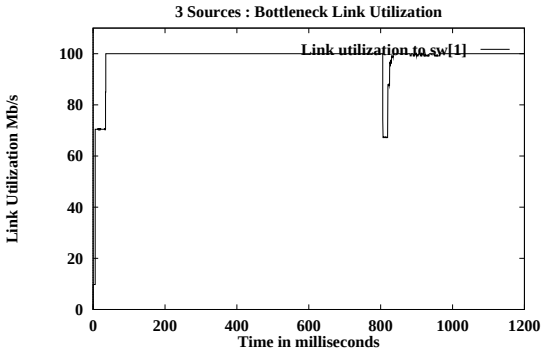


(c) Utilization

Figure 5.8: Three sources: Case 3 + DQF simulation results



(a) ACRs



(b) Utilization

Figure 5.9: Three Sources (Transient) : ACR and utilization graphs.

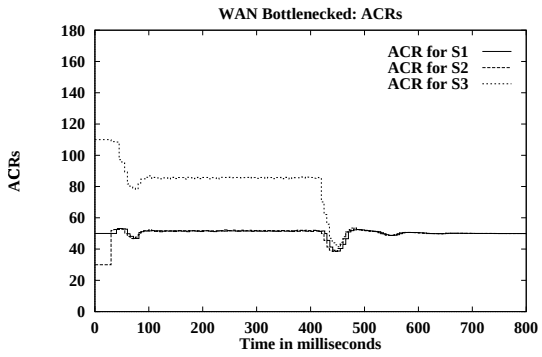
#	Src #	Weight func.	Expected fairshare (ntr)	Actual (ntr) share	Expected fairshare (tr)	Actual (tr) share
1	1	1	74.88	74.83	49.92	49.92
	2	1	NC	NC	49.92	49.92
	3	1	74.88	74.83	49.92	49.92
2	1	1	54.88	54.88	29.92	29.83
	2	1	NC	NC	49.92	49.92
	3	1	94.88	95.81	69.92	70.93
3	1	15	29.92	29.23	18.53	18.53
	2	35	NC	NC	49.92	49.92
	3	55	119.84	120.71	81.30	81.94

ntr - non-transient period, tr - transient, NC - not converged

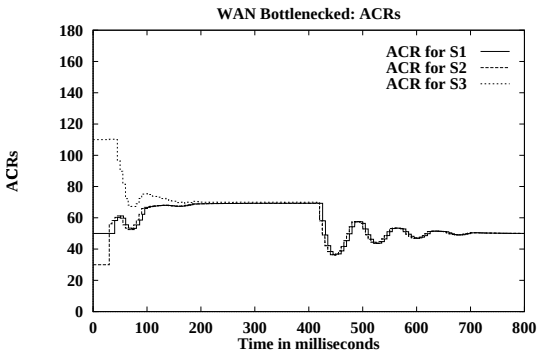
Table 5.2: Three sources transient configuration simulation results

during the presence of source bottleneck (i.e., before 400 ms). The correct fairness is achieved only when the measured source rates are used. When the source bottleneck disappears after 400 ms, the sources achieve GW fairness (fairshare value is same as in the simple configuration), both when CCR is used as source rate and when source rates are measured.

Figure 5.10 (a) shows the ACR graph for the simulation of case 1 using the source rate from the CCR field of RM cell. Figure 5.10 (b) shows the same case using measures source rates. When the CCR value from the RM cells is used as the source rate, the algorithm is not able to estimate the actual rate at which the source is sending data. As a result, it does not estimate the correct GW fairshare values in presence of source bottlenecks. When measured source rate is used it calculates the correct fairshare even in the presence of source bottlenecks.



(a) Case 3 + DQF + source rate from CCR field

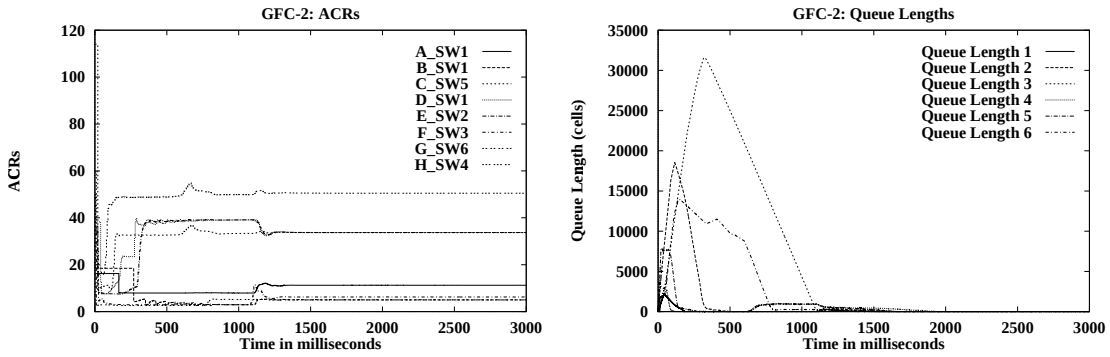


(b) Case 3 + DQF + measured source rate

Figure 5.10: Three Sources Bottleneck: ACR graphs

5.8.4 Link Bottleneck: GFC-2

In this configuration each link is a bottleneck link. An MCR value of 5 was used for all A type VCs. All other VCs have an MCR of 0. The "MCR plus equal share of excess bandwidth" was chosen as the fairness criteria. The dynamic queue control function was used in this simulation. The expected share for VCs of type A, B, C, D, E, F, G, H are 11.25, 5, 33.75, 33.75, 33.75, 6.25, 5, and 50.625 Mbps, respectively. The actual allocation for these VCs in the simulation was 11.25, 5, 35.67, 35.75, 35.75, 6.25, 5, and 50.5 Mbps respectively, which agree well with the expected allocations. Figure 5.11 (a) shows the ACR graphs for each type of VCs. Figure 5.11 (b) shows the queue length graph at various bottleneck links between the switches. From the figure and actual allocations it can be seen that the VCs converge to their expected fairshare. The queue length graphs show that the initial queue buildup occurs before convergence and its maximum queue length depends on the ICR (initial cell rate)



(a) ACR graph

(b) Queue graphs at different links

Figure 5.11: GFC-2 configuration: ACR and queue graphs

and round trip time. This simulation demonstrates that the algorithm works in the presence of multiple link bottlenecks and different round trip times.

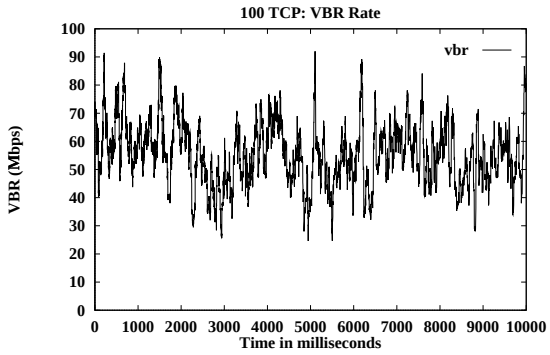
5.8.5 100 TCP Sources with VBR Background

The VBR VC carrying multiplexed MPEG source traffic has higher priority over TCP sources running over ABR. The VBR traffic generated is highly variable as shown in Figure 5.12 (a). The TCP sources are infinite TCP sources. Initially, the TCP traffic is bursty since its congestion window is limited by ACR and slow start protocol. Once the congestion window reaches the maximum value the TCP sources become equivalent to persistent sources. All TCP sources start sending data at same time, so the load phases (active and idle periods) of multiple sources coincide. Source-25 has MCR value of 1 Mbps, source-50 has MCR of 1.5 Mbps, and source-100 has MCR value of 2 Mbps. All other TCP sources have an MCR value of 0.5 Mbps. A value of 10 was used for parameter 'a' of the weight function (a+MCR). Hence the GW fairness criteria here is MCR plus proportional to MCR.

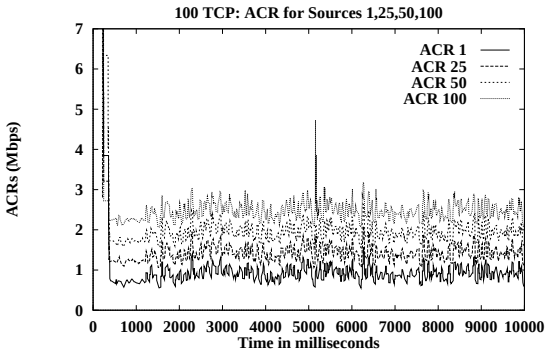
Case #	Src #	Weight	Expected fairshare function	Using CCR in RM cell	Using Measured CCR
1	1	1	69.92	51.50	69.29
	2	1	69.88	51.80	69.29
	3	1	69.88	85.94	69.29
2	1	1	39.88	43.98	39.58
	2	1	59.88	52.06	59.57
	3	1	79.88	85.85	79.76
3	1	15	19.96	42.72	19.19
	2	35	53.32	51.62	53.28
	3	35	86.64	86.16	86.37

Table 5.3: Three sources bottleneck configuration simulation results

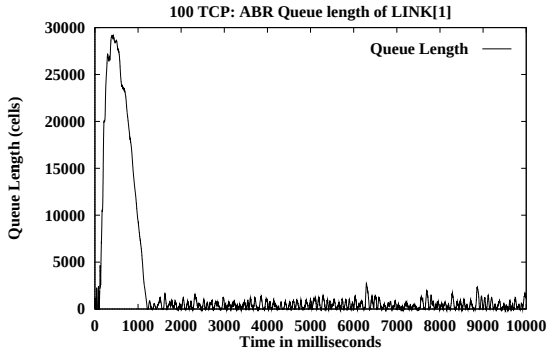
Figures 5.12 (b), (c) and (d) show the ACRs, the queue length and link utilization respectively, which are ATM-level metrics. Figures 5.12 (e) and (f) show the congestion window and average throughput respectively, which are TCP-level metrics. Though the system does not have a steady state, the queues are controlled and utilization is high. The expected throughput received by the TCP sources when the congestion window is maximized, is 1.02 Mbps for source 1, 1.54 Mbps for source 25, 2.07 Mbps for source 50 and 2.59 Mbps for source 100, according to the GW fairness criteria (MCR plus proportional MCR in this case). The average throughput values as shown in Figure 5.12 (f) is slightly different from the expected throughputs. This is due to the varying VBR capacity and since the average throughputs include measurement during initial burstiness of TCP sources, where the congestion windows have not yet reached the maximum value. This simulation demonstrates that the algorithm is robust and works well even in the presence of large number of sources.



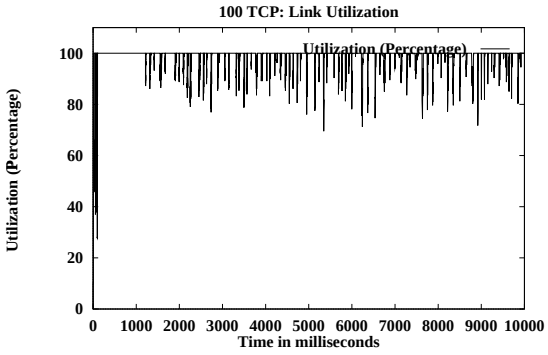
(a) VBR capacity



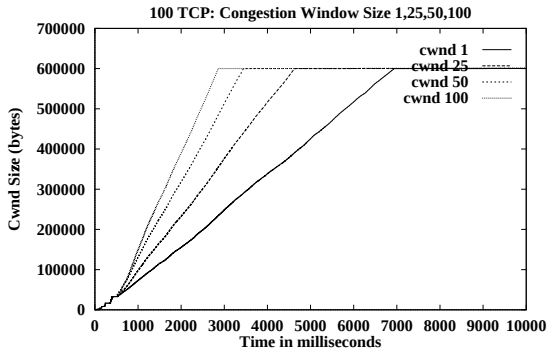
(b) ACRs



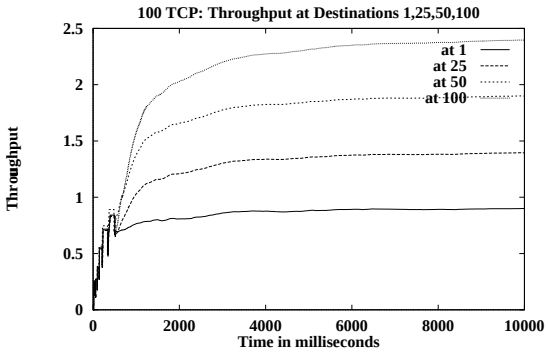
(c) Queue length



(d) Link utilization



(e) TCP congestion window



(f) Average TCP throughput

Figure 5.12: 100 TCP + VBR background simulation graphs

5.9 Proof of Convergence of Algorithm GWFairERICA+

We make the following assumptions:

- Synchronous update of source rates
- Queue control function is a constant function
- Infinite (greedy) sources, which always have data to send. Though there might be a source or link bottleneck present.
- If a source bottleneck is present, it does not change it bottleneck rate during convergence.
- $\sum_{s \in S_l} \mu_i \leq A_l$, i.e., Sum of MCRs is less than available ABR capacity (connection admission policy)
- Load factor $z > 0$ and $ER < A_l < LinkRate$

Lemma 1 *The Algorithm GWFairERICA+ converges to the GW fair allocation for a session bottlenecked by a link.*

Proof: The proof technique used here is similar to the one used in reference [68]. Let l_b be the link which is bottlenecked. Without loss of generality assume that the first k sessions through the link l_b are bottlenecked (either link bottlenecked or source bottlenecked) elsewhere. Let $n = \mid S_{l_b} \mid -k$. Let $r_{b1}, r_{b2}, \ldots, r_{bk}$ be the bottleneck rates and $r_1, r_2, \ldots, r_n$ be the rates of non-bottlenecked (under-loaded) sources. Let $A_b = \sum_{i=1}^{k} r_{bi}$ be total capacity of bottlenecked links. These non-bottlenecked sources are bottlenecked at the current link l_b. According to the GW fairness definition, fair

allocation rates g_i is given by:

$$g_i = \mu_i + \frac{w_i(A_l - A_b)}{\sum_{j=1}^{n} w_j}$$

Assume that the bottlenecks elsewhere have been achieved. Therefore, the rates $r_{b1}, r_{b2}, \ldots, r_{bk}$ are stable. For simplicity, assume that the MCRs of these sources are zero. Proof for the bottlenecks having non-zero MCRs is a simple extension.

We show that the rates allocated at this switch converge to $r_{b1}, r_{b2}, \ldots, r_{bk}$ and $g_1, g_2, \ldots, g_n$ and that the load factor converges to $z = 1$.

Case 1: Load factor $z < 1$. Here the link is under-loaded. Due to the $VCShare$ term $SourceRate(i) - \mu_i/z$, all the rates increase. If $n = 0$, i.e. all the sessions across this link are bottlenecked elsewhere, there are no non-bottlenecked sources, the GW fair allocation is trivially achieved. Assume that $n \geq 1$. Because of the $VCShare$ term (in the step for calculating ER in Algorithm GWFairERICA+), the rates of non-bottlenecked sources increase. This continues until the load factor reaches a value greater than or equal to one. Hence we have shown that if load factor is less than one, the rates increase till the load factor becomes greater than one.

Case 2: Load factor $z > 1$. In this case, if the link is not getting its $ExcessFairshare$ then its rate increases, which might further increase z. This continues until all the sessions achieve at least their $ExcessFairshare$. At this point, the allocation rates are decreased proportional to $1/z$ due to the first term. As in the previous case the z decreases, till it reaches a value of 1 or less.

From the above two cases, it can be seen that the load factor oscillates around one and converges to the value of one. Assume that the load factor is $z = 1 + \delta$, then the number of round trip times for it to converge to one is given by $log_{1+\delta} \mid S_l \mid$. Henceforth, we assume that the network is near the steady state and that the load

factor is near one. This implies that

$$\sum_{i=1}^{k} r_{bi} + \sum_{i=1}^{n} r_i = A_l \rightarrow \sum_{i=1}^{n} r_i = A_l - A_b$$

Let $A_m = \sum_{i=1}^{n} \mu_i$ be the total allocation for MCRs of the non-bottlenecked sources. Define $\alpha_i = r_i - \mu_i$, then we have:

$$\sum_{i=1}^{n} \alpha_i = A_l - A_b - A_m = A$$

We have to show that:

$$\alpha_i = \frac{w_i A}{\sum_{j=1}^{n} w_j}$$

Case A: $n = 0$, i.e., there are no bottleneck sources. From the step for calculating ER in algorithm GWFairERICA+, we have:

$$\alpha_i = \max(ExcessFairshare(i), \alpha_i / z)$$

We observe that this equation behaves like a differential equation in multiple variables [74]. The behavior is like that of successive values of root acquired in the Newton-Ralphson method for finding roots of a equation. Hence, the above equation converges and the stable values of α_i are given by:

$$\alpha_i = ExcessFairshare(i) = \frac{w_i A}{\sum_{j=1}^{n} w_j EAL(i)}$$

Since we have assumed greedy sources and no bottlenecks in this case, the excess activity level is one for all sessions. Hence,

$$\alpha_i = ExcessFairshare(i) = \frac{w_i A}{\sum_{j=1}^{n} w_j}$$

which is indeed the desired value for α_i.

Case B: $n \neq 0$, i.e., there are some bottleneck sources. Let β_i be the allocated rate corresponding to r_{bi}. Let w_{bi} be the weight for session s_{bi}. Let $W_b = \sum_{i=1}^{K} w_{bi} EAL(b_i)$ and $W = \sum_{i=1}^{n} w_i$. We know that the equation for the rate allocation behaves as a stabilizing differential equation. In the steady state, all the above terms such as W, W_b and rates stabilize. For sources bottlenecked elsewhere, the algorithm calculates a rate β_i which is greater than r_{bi}, otherwise, the bottlenecked session would be bottlenecked at the current link. For non-bottlenecked sources, the rate at steady state is given by:

$$\alpha_i = \frac{w_i(A_l - A_m)}{W_b + W}$$

Since the link has an overload of one at steady state, we have

$$\sum_{i=1}^{n} \alpha_i = A_l - A_m - A_b$$

which implies that

$$\frac{(A_l - A_m)\sum_{i=1}^{n} w_i}{W_b + W} = A_l - A_m - A_b$$

Substituting W for $\sum_{i=1}^{n} w_i$ we get:

$$W_b = \frac{W A_b}{A_l - A_m - A_b}$$

Using the above value for W_b we get:

$$\alpha_i = \frac{w_i(A_l - A_m)}{\frac{W A_b}{A_l - A_m - A_b} + W} = \frac{w_i(A_l - A_m - A_b)}{W}$$

which is the desired values for the α_i. Hence, the sessions bottlenecked at the link, l_b, do indeed achieve the GW fairness. □

Theorem 5.1 *Starting at any arbitrary state of the network, if only greedy sources and source bottlenecked or link bottlenecked sources are present the algorithm GW-FairERICA+ converges to GW fair allocation.*

Proof: The convergence of the distributed algorithm is similar to the centralized algorithm. Assume that the centralized algorithm converges in M iterations. At each iteration, there are set of links $\mathcal{L}_i$ which are bottlenecked at the current iteration. $\cup_{i=1}^{M} \mathcal{L}_i = \mathcal{L}$.

Using lemma 1, we know that each link $l \in \mathcal{L}_i$ does indeed converge to the general fair allocation $\mathcal{G}_l$. The distributed algorithm converges in the above order of links until the whole network is stable and allocation is $\mathcal{G}$. The number of round trips taken to converge is bounded by $M \times O(\log S)$, since each link takes $O(\log S_l)$ round trips for convergence. □

5.10 Chapter Summary

In this chapter, we have given a general definition of fairness, which inherently provides MCR guarantees and divides the excess bandwidth proportional to predetermined weights. Different fairness criterion such as max-min fairness, MCR plus equal share, and proportional MCR can be realized as special cases of this general fairness. We showed how to realize a typical pricing policy by using appropriate weight function. The GW fairness can be achieved by using the $ExcessFairshare$ term in the switch algorithms. The weights are multiplied by the activity level when calculating the $ExcessFairshare$ to reflect the actual usage of the source. The MCR guarantee and the generalized fairness can be used for achieving throughput continuity of multimedia ABR connections.

We have shown how the ERICA+ switch algorithm can be modified to achieve this general fairness. The proof of convergence of the algorithm GWFairERICA+ was given. The simulation results show that the modified algorithm achieves the

general fairness in all configurations. In addition, the results show that the algorithm converges even in the presence of both source and link bottlenecks and is quick to respond in the presence of transient sources. In the source bottlenecked configuration, the value of the CCR (source rate) from the RM cells maybe incorrect. Hence, it is necessary to use the measured source rate in the presence of source bottlenecks. The algorithm is robust and scalable as demonstrated by simulation results using the one hundred TCP sources plus VBR background configuration.

CHAPTER 6

OVERLOAD BASED SWITCH SCHEMES

In the previous chapter, we proposed a general definition of fairness, and gave an overload based ABR switch scheme that provides MCR guarantees by modifying ERICA+ algorithm. In this chapter, we propose three new algorithms that use the overload factor to calculate the explicit rate feedback. All the proposed algorithms provide MCR guarantees and achieve generalized fairness. These algorithms give an additional choice when choosing which rate control scheme should be used when transporting multimedia over ABR connections.

The load factor (also referred to as "overload factor" or "overload") is the ratio of the measured input rate to the available ABR (available bit rate) capacity. The ERICA+ switch scheme monitors the load on the link and calculates feedback based on the load. It tries to achieve unit load on links to efficiently use the link and also converge to max-min fairness [42]. Max-min fairness assumes zero MCR values. In this chapter, we have used the generalized fairness with non-zero MCRs as defined in section 5.2.

6.1 The Switch Schemes

The general structure of the algorithms proposed are similar to the ERICA+ which was discussed in chapter 3. The three switch algorithms proposed have the same structure and differ in the way in which the end of interval accounting is done, and the manner in which the feedback explicit rate is calculated.

6.1.1 Overload Based Algorithm: General Structure

The three different switch schemes have the following common algorithmic structure. They differ in the manner in which the feedback rate is calculated and in accounting. The algorithms perform the following steps for calculating the feedback rate:

1. The problem of non-zero MCRs is reduced to one with zero MCRs. The MCR is subtracted from each connection's current cell rate (CCR(i)) to obtain the excess rate of each source (SR(i)) over MCR. If the source is transmitting at a rate less than its MCR, an excess rate of zero is used.

2. The switch algorithm for zero MCRs is applied to these excess rates to obtain the feedback rate. The excess available capacity $(A - \mu)$ is divided in proportion to the pre-determined weight.

3. The MCR for each source is added to the explicit feedback rate calculated in the previous step. The resulting rate is indicated in the ER field of the BRM cells.

The sources transmit data initially at their initial cell rate (ICR) value. After one round trip time, the feedback from the switches arrives at the sources, and sources

Overload Based Algorithm Structure:

At the End of Each Averaging Interval:

$$\begin{aligned}
\text{ABR Capacity} &\leftarrow \text{Link Capacity} - \text{VBR Capacity} \\
&\quad - \sum_{i=0}^{n} \min(SR(i), \mu_i) \\
\text{TargetABRCap} &\leftarrow f(Q) \times \text{ABR Cap} \\
\text{Input Rate} &\leftarrow \text{ABR Input Rate} - \sum_{i=0}^{n} \min(SR(i), \mu_i) \\
z &\leftarrow \text{Input Rate/TargetABRCap}
\end{aligned}$$

End_of_Interval_Accounting()

When an FRM is received:

$$\text{CCR(i)} \leftarrow \text{CCR}_{RM\_Cell}$$

When a BRM is received:

$$\begin{aligned}
\text{Ex_ER} &\leftarrow \textbf{\textit{Calculate_Excess_ER()}} \\
\text{ER} &\leftarrow \mu_i + \text{Ex_ER} \\
\text{ER}_{RM_Cell} &\leftarrow \text{Min(ER}_{RM_Cell}\text{,ER,Target ABR Cap)}
\end{aligned}$$

Figure 6.1: Pseudo code for the overload based algorithm structure.

adjust their allowed rate accordingly. When a constant queue control function is used, (say $Fraction = 0.9$), rates converge to the GW fairness allocation. When a dynamic queue control function is used, the available capacity varies depending on the queue length. Therefore, the feedback rate also varies until the queues are drained. Once the queues are drained, the feedback rates converge to the GW fair allocation. We present simulation results using both the constant queue and dynamic queue control functions.

The procedures *End_of_Interval_Accounting()* and *Calculate_Excess_ER()* are the key steps that differentiate the algorithms.

6.1.2 Algorithm A: ExcessFairShare/Overload

The *ExcessFairShare* term is defined as follows:

$$ExcessFairShare(i) = \frac{w_i(A-\mu)}{\sum_{j=1}^{n} w_j}$$

This divides the excess available bandwidth $(A-\mu)$ (i.e., ABR capacity) in proportion to the weights $w(i)$. An allocation of $\mu_i + ExcessFairShare(i)$ for each source i is the GW fair allocation. A source might be bottlenecked at some other link, hence using less than its fair share of link capacity. By using the notion of activity level, the effective weight of the bottlenecked source can be calculated.

The activity level for a given VC is defined as follows:

$$AL(i) = min\left(1, \frac{SourceRate(i) - \mu_i}{ExcessFairShare(i)}\right)$$

The activity level can be used to accurately estimate the effective number of VCs. The effective number of VCs is given by the following expression:

$$\text{Effective number of VCs} = \sum_{j=1}^{n} AL(j)$$

The VCs that are bottlenecked by this link will have an activity level of 1 and will be counted as one VC. The VCs bottlenecked elsewhere will are counted as fractional VCs depending on their activity level. The proof that the above expression accurately estimates the effective number of VCs is given in [32].

We extend the notion of activity level to the weighted case by multiplying the weight function with the activity level of the *ExcessFairShare* term. Therefore, the *ExcessFairShare* is:

$$ExcessFairShare(i) = \frac{w_i(A - \mu)}{\sum_{j=1}^{n} w_j AL(j)}$$

In this algorithm, the *Ex_ER* is calculated based on $ExcessFairShare$ and the overload factor z. For each source, the activity level and the $ExcessFairShare$ are updated at the end of each interval. When a BRM cell arrives, the feedback rate is calculated as the $ExcessFairShare$ term, divided by the overload. The source rate ($VCShare$ term) is not used in the feedback rate calculation.

If the network is overloaded ($z > 1$) the Ex_ER decreases since fairshare is divided by the overload factor z. In underload conditions ($z < 1$) the sources are asked to increase their rate. Due the recursive definition of $ExcessFairShare$ and activity levels these value converge.

As the network reaches steady state, the overload will become one and the *Ex_ER* will converge to the required $ExcessFairShare$, achieving GW fair allocation. The proof of convergence is similar to the proof given in [117]. Since the overload factor varies every averaging interval, the rate oscillations increase for this algorithm. If the "averaging interval" is greater than the feedback loop, then the switch adjusts to the feedback rate before the next averaging interval. If the averaging interval is smaller, then before the source can adjust to the feedback rate, multiple feedbacks are given. In this situation, $ExcessFairShare$ calculation is not accurate because the source rate does not reflect the feedback rate. Exponential averaging of the $ExcessFairShare$ term is used to overcome this problem. Exponential averaging is done as follows:

$$ExcessFairShare = \alpha\ ExcessFairShare + (1 - \alpha) PrevExcessFairShare$$

where the $PrevExcessFairShare$ is the value calculated for the previous averaging interval. The parameter α is called the decay factor. In the simulations a value of $\alpha = 0.9$ was used because this gives more weight to the current estimate.

End_of_Interval_Accounting():

foreach VC i do

$$AL(i) \leftarrow min\left(1, \frac{\text{SourceRate}(i) - \mu_i}{ExcessFairShare(i)}\right)$$

$$ExcessFairShare(i) \leftarrow \frac{(\text{TargetABRCap})w_i AL(i)}{\sum_{j=1}^{n} w_j AL(j)}$$

endfor

Calculate_Excess_ER():

$$\text{Ex\_ER} \quad \leftarrow \quad \frac{\text{ExcessFairShare(i)}}{z}$$

6.1.3 Algorithm B: MaxAllocation/Overload

Let the GW fair allocation be $(g_1, g_2, \ldots, g_n)$ for n bottle-necked sources. Consider two sources i and j, the excess bandwidth is divided proportional to weights $(g_i - \mu_i)/w(i) = (g_j - \mu_j)/w(j)$. Let m be the VC such that term $(g_m - \mu_m)/w(m)$ is the maximum of such terms of all VCs. An allocation which assigns rates as $\mu_i + w(i)(g_m - \mu_m)/w(m)$ will achieve GW fairness. The term $(g_m - \mu_m)/w(m)$ is defined as the weighted maximum allocation. In algorithm B, the feedback is calculated proportional to the weight of the VC and the weighted maximum allocation is divided by overload. The overload in the denominator increases or decreases the allocation depending on the load. The source rate is not used in the feedback calculation. This algorithm can give rise to large queues if the weighted maximum allocation is

measured incorrectly. This problem of large queues occurred during the simulation of this algorithm using GFC-2 configuration.

End_of_Interval_Accounting():

$$\text{WtMaxAllocPrev} \leftarrow \text{WtMaxAllocCur}$$

$$\text{WtMaxAllocCur} \leftarrow 0$$

Calculate_Excess_ER():

$$\text{Ex\_ER} \leftarrow \frac{\text{w(i)WtMaxAllocPrev}}{z}$$

$$\text{WtMaxAllocCur} \leftarrow \text{Max (WtMaxAllocCur,Ex\_ER/w(i))}$$

6.1.4 Algorithm C: VCShare and MaxAllocation

In this algorithm, the *End_of_Interval_Accounting()* is the same as in the previous algorithm (algorithm B). The *Ex_ER* is calculated based on the weighted maximum previous allocation ($WtMaxAllocPrev$) and $VCShare$ under lightly loaded conditions ($z < 1 + \delta$). For overloaded conditions, the $VCShare$ is given as the feedback rate. The problem of large queues explained in the previous algorithm is not expected to occur since the maximum previous allocation is given as feedback only if the link underloaded. As the overload converges to one, the $VCShare$ converges to $w(i)(g_m - \mu_m)/w(m)$, achieving the GW fair allocation.

Calculate_Excess_ER():

$$\text{VCShare} \leftarrow \frac{max(0, SourceRate(i) - \mu_i)}{z}$$

$$\text{IF } (z > 1 + \delta)$$

$$\text{THEN Ex\_ER} \leftarrow \text{VCShare}$$

$$\text{ELSE Ex\_ER} \leftarrow \text{max(w(i) WtMaxAllocPrev, VCShare)}$$

WtMaxAllocCur $\leftarrow$ max(WtMaxAllocCur,Ex_ER/w(i))

6.2 Simulation Configurations

We used simple link bottleneck and source bottleneck configurations to test the proposed algorithms. Infinite sources were used (which have an infinite amount of data to send, and always send data at ACR) in all the simulations. The rates are expected to converge to GW fair allocation values in the presence of infinite sources. The algorithms are expected to give minimum rate guarantees for poisson or self-similar sources. These types of sources were not used since the GW fair allocation for source with varying rates is also dynamically varying. The data traffic is only one way, from source to destination. Using two-way traffic would produce similar results, except that the convergence time would be larger since the RM cells in the backward direction would travel with traffic from destination to source. All the link bandwidths are 149.76 (155.52 less the SONET overhead), except in the GFC-2 configuration.

6.2.1 Three Sources

This is a simple configuration in which three sources send data to three destinations over two switches and a bottleneck link (see Figure 5.3). This configuration is used to demonstrate that the switch algorithms can achieve GW fairness.

6.2.2 Source Bottleneck

In this configuration, the source S1, is bottlenecked at a rate of 10 Mbps, which is below its fairshare (50 Mbps) for the first 400 ms of the simulation (see Figure 5.4). This configuration tests whether the fairness criterion can be achieved in the presence of source bottleneck.

Configuration Name	Link Dist	Averaging interval	Target Delay	Weight Function
Three Sources	1000 Km	5 ms	1.5 ms	1
Src Bottleneck	1000 Km	5 ms	1.5 ms	1
GFC-2	1000 Km	15 ms	1.5 ms	1

Table 6.1: Simulation Parameter Values

6.2.3 Generic Fairness Configuration - 2 (GFC-2)

This configuration is a combination of the upstream and parking lot configurations (see Figure 5.5). In this configuration, all the links are bottlenecked links. This configuration is explained in reference [108].

6.2.4 Simulation Parameters

The simulations were done using an extensively modified version of the NIST ATM simulator [54]. The parameter values for different configurations are given in Table 6.1. The algorithms were simulated using both a constant queue control function ($Fraction = 0.9$) and using a dynamic queue control function. For dynamic queue control, hyperbolic functions was used, with curve parameters $a = 1.15$ and $b = 1$. The $QDLF$ value was set to 0.5.

Exponential averaging was used to decrease the variation in measured quantities such as overload and the number of VCs. Exponential averaging of the overload factor and the number of VCs were done with a decay factor of 0.8 for algorithm C. The algorithms A and B are more sensitive to overload factor. So, the decay factor of 0.4 was used to average overload in algorithms A and B.

Config. and Queue Control	Src 1	Src 2	Src 3
Simple and CQF	24.93	44.93	64.93
Simple and DQF	29.92	49.92	69.92
Src Bottleneck + CQF (0-0.4s)	32.39	52.39	72.39
Src Bottleneck + DQF (0-0.4s)	39.86	59.86	79.86
Src Bottleneck + CQF (0.4-0.8s)	24.93	44.93	64.93
Src Bottleneck + DQF (0.4-0.8s)	29.92	49.92	69.92

Table 6.2: GW fair allocation for different configurations

A weight function of one was used in all configurations. This corresponds to MCR plus equal share of excess bandwidth. The value of $\delta = 0.1$ was used for algorithm C. In reference [117] it was shown that an overload-based explicit rate algorithm achieves GW fairness for various weight functions.

6.3 Simulation Results

In this section we present the simulation results of the algorithms using different configurations. Table 6.2 gives the expected GW fair allocation with a constant queue control function (shown as configuration name and CQF) and with a dynamic queue control function (shown as configuration and DQF) for each simulation using the three source configuration. The queues, when using a constant queue control function, took a longer time to drain. The bottleneck link utilization was, as expected, 90% when constant queue control was used. With the dynamic queue control, the algorithms achieved 100% link utilization for bottlenecked links.

6.3.1 Three Source: Results

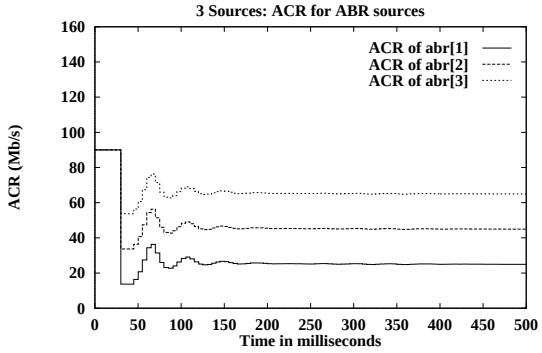
The MCR values for the three source configuration are 10,30,50 Mbps for the source 1, source 2 and source 3 respectively. For the simulation with queue control, the excess bandwidth, (149.76 - 90 = 59.76) is divided equally among the three sources. The expected allocation is (10+59.76/3, 30+59.76/3, 50+59.76/3) = (29.92, 49.92, 69.92). Figure 6.2(a)-(c) shows the ACRs for algorithms A,B, and C (with dynamic queue control) respectively. From the graphs, it can be seen that the expected allocation is, achieved by all the three algorithms.

6.3.2 Source Bottleneck: Results

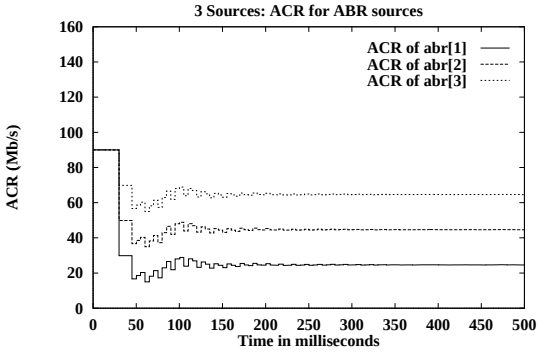
In this configuration, the MCR values of (10,30,50) Mbps were used. The total simulation time was 800 ms. The source, S1, is bottle-necked at 10 Mbps for the first 400 ms of the simulation. It always sends data up to 10 Mbps even when its ACR larger than 10 Mbps. Figures 6.3 (a)-(c) shows the ACRs for algorithms A, B, and C with constant queue control respectively. The expected allocation is (32.39,52.39,72.39) for the first 400 ms and it is (29.92,49.92,69.92) between 400 ms and 800 ms. The algorithms do converge to the expected allocation. The algorithm A has fewer rate oscillations and converges faster than algorithm B and C.

6.3.3 GFC-2: Results

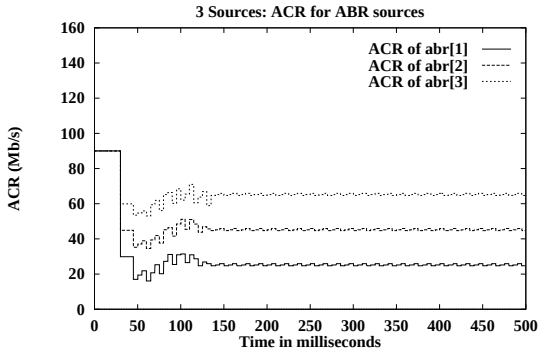
An MCR value of zero was used for all sources expect for type A sources which had a MCR value of 5 Mbps. The expected allocation for each type of VC using a constant queue control and a dynamic queue control is given in the Table 6.3. Figure 6.4 (a)-(c) show the ACRs of each type of VCs A through H for algorithms A, B, and C with constant queue control respectively. Algorithm A converges to the



(a) Algorithm A and DQF



(b) Algorithm B and DQF



(c) Algorithm C and DQF

Figure 6.2: Three Sources: ACR graphs for algorithms A, B, and C.

expected allocation. Algorithm B with constant queue control, does not converge to expected GW fair allocation within the simulation time as seen in Figure 6.4(f). This is due to the presence of sources with different round trip times sharing the same bottleneck link. In such a case, the sources with larger round trip times take a long time to adjust the rates compared to the ones that have smaller round trip times. For example, the maximum allocation for A type VC that has large round trip time was assigned for seven VCs of type G which have small round trip time. This also led to large switch queues and slow convergence of algorithm B. The input rate at the link between SW6 and SW7 is overloaded by a factor of seven which gives rise to the huge queues.

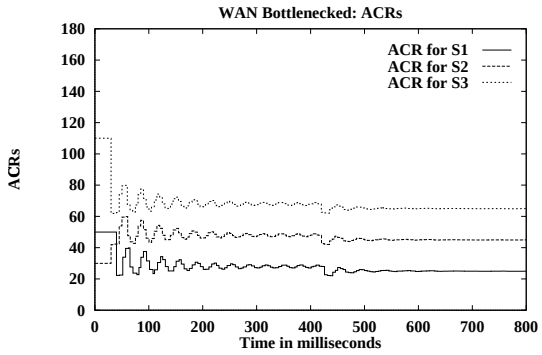
A	B	C	D	E	F	G	H
11.25	5	33.75	33.75	33.75	6.25	5	50.62
9	4.5	31.5	31.5	31.5	9	4.5	47.25

Table 6.3: GFC-2 configuration: Expected allocations when using DQF and CQF for each type of VC

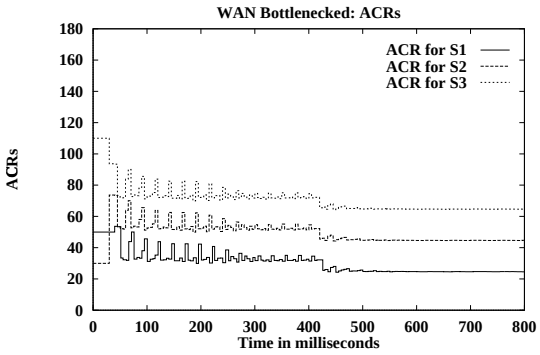
6.4 Comparison of Switch Schemes

Table 6.4 gives a comparison of the algorithms.

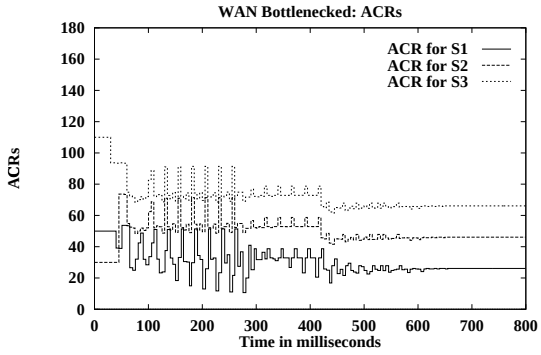
Algorithm A has higher complexity than the other two algorithms it does $O(N)$ computations at the end of interval. Algorithm B results in large switch queues in the presence of multiple round trip time. Algorithm C is the best of the proposed algorithm since it has O(1) complexity both at the end of each interval and when a BRM cell arrives. Also, algorithm C has smaller queues compared to algorithm B.



(a) Algorithm A with CQF

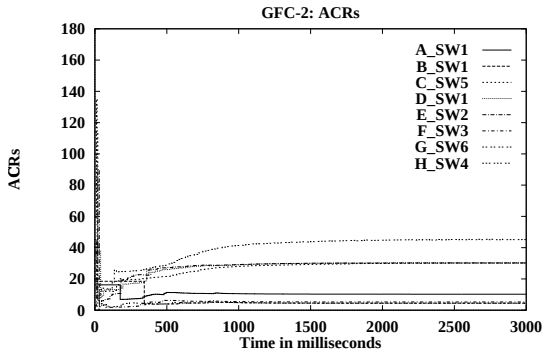


(b) Algorithm B with CQF

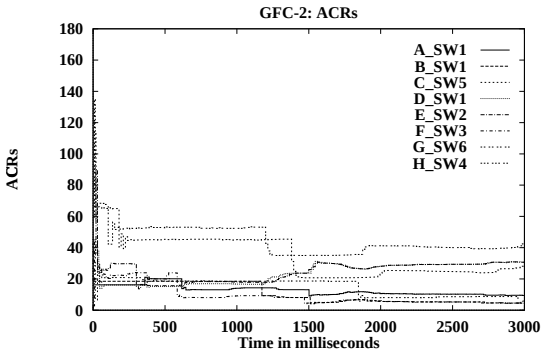


(c) Algorithm C with CQF

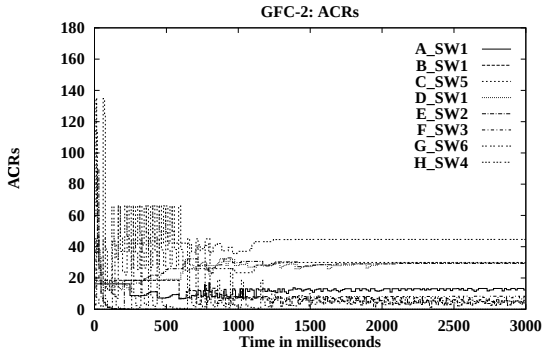
Figure 6.3: Source Bottleneck: ACR graphs for Algorithm A, B, and C.



(a) Algorithm A and CQF



(b) Algorithm B and CQF



(c) Algorithm C and CQF

Figure 6.4: GFC-2 config: ACR graphs for algorithms A, B, and C.

Algo-rithm	End of Interval Complexity	Feedback Complexity	Max Q Len	Sensitivity to Q cntrl
A	O(N)	O(1)	Med	High
B	O(1)	O(1)	Large	Low
C	O(1)	O(1)	Med	Low

Table 6.4: Comparison of the algorithms

6.5 Chapter Summary

In this chapter, we have presented three algorithms which achieve GW fairness and provide MCR guarantees. The algorithms monitor the load on the link and calculate the overload factor. The overload and other quantities ($ExcessFairShare$ or $WtMaxAllocPrev$) are used to calculate the feedback rates.

The algorithms proposed have similar structure in that they perform end of interval computations and give feedback when BRM cells arrive. The algorithms differ in the end of interval accounting and feedback calculation. Simulations show that the algorithms converge to GW fairness in most cases. Queue control can be done using a constant function and dynamic (hyperbolic) function. Algorithm A has O(N) complexity for the end of interval calculations. Algorithm B can give rise to large queues if the configuration has sources with different round trip times sharing the same link. Algorithm C, which uses the $VCShare$ and $WtMaxAllocPrev$, is the best because it has O(1) complexity and is less sensitive to queue control function.

CHAPTER 7

DESIGN AND ANALYSIS OF DYNAMIC QUEUE CONTROL FUNCTIONS

The goals of a rate allocation scheme are to maintain high utilization, small queuing delay, small cell loss, and fairness among competing sources. In order to support multimedia video sources over ABR service, it is also desirable in the steady state to have the rates and queuing delay constant. One way to achieve high utilization and low queuing delay is to vary the target rate as a function of queue length. The function should be a decreasing function of queue length. The function should also be *simple* so that it can be implemented in hardware.

In this chapter, we study several *queue control* functions which satisfy the above needs. We present an analytical explanation for the performance of these functions. Then, we present simulation results that are consistent with the analysis. The various trade-offs between the queue control functions are studied using appropriate metrics. The *ERICA+* switch scheme is used in the simulations [71].

7.1 Switch Scheme Model

This section gives an overview of the switching scheme model on which this study is based.

- An ABR switch scheme achieves the goals by giving explicit feedback to the sources to adjust their source rates. This type of switch is usually known as a *Explicit Rate Feedback* switch.

- One way to achieve high utilization (100%) and control queuing delay by quick draining of queues is to vary the target ABR rate dynamically. During steady state, the target ABR rate is 100% while it is lower during the transient state. Higher overloads result in even lower target rates (thereby draining the queues faster). In other words:

$$\text{Target rate} = f(\text{queue } lngth) \times \text{function}(\text{current rate}, \text{link rate}, \text{HPR})$$

  HPR is the total rate of higher priority classes like VBR (variable bit rate) and CBR (constant bit rate). The "f(queue length)" has to be a decreasing function of the queue length. The switch scheme uses the above queue control function to adjust the allocated rate depending on the current switch queue size.

- The switch measures the load and queue length and gives explicit feedback of target rate at fixed intervals. This interval is called the "averaging interval". The measurements are done using the *FRM* cells and the feedback is given using the *BRM* cells. We assume that only one feedback is given in each averaging interval to the sources. This avoids unnecessary conflicting feedback received at the sources.

The *ERICA+* algorithm used in this study fits the above model. There are many other ABR switch schemes which may be used instead of ERICA+ ([71, 99, 8, 4, 10]).

7.2 Queue control functions

In this section the relationship between the queue length and queue control function is presented for the above switch model. Then various queue control functions to achieve the desired goals are presented.

The terms are used in the discussion are given in Table 7.1.

Term	Description
N	number of sources.
t_s	"averaging interval", the period at which feedback to the sources is calculated at the switch.
$r_i(t)$	rate of source i.
$er_i(t)$	explicit rate of source i calculated at the switch.
t_p	propagation time from the source to switch.
t_f	feedback delay is twice t_p.
R_l	available ABR capacity. For simplicity we assume the higher priority traffic such as CBR and VBR are not present. Hence, R_l is same as the link rate.
$Q(t)$	switch queue length (in cells)
$R(t)$	aggregate *input rate* seen at the switch. $R(t) = \sum_{i=1}^{N} r_i(t)$
$C(t)$	(conversion function) number of cells transmitted in time t at the link rate. $C(t) = (R_l \times t)/424$ if R_l is given in Mbps.

Note : X(t) denotes that X is a function of time.

Table 7.1: Description of terms used in discussion of queue control functions.

7.2.1 Queue Length Function

To simplify the analysis, we assume that the propagation delay, t_p, from source to the switch is same for all sources. Due to propagation delay t_p, the rate seen at the switch at time t is the same as source rate at time $t - t_p$. The sources adjust their rates to the explicit rate indicated in the BRM cells. Due to propagation delay t_p,

the sources adjust their rate at time t to the explicit rate generated at from interval $t - t_p$.

In one averaging interval $Q(t)$ is drained by $R_l \times C(t_s)$ cells. The queue builds up at *input rate*. Then, $Q(t)$ can be expressed as follows :

$$Q(t) = Q(t - t_s) + (\sum_{i=1}^{N} r_i(t - t_p) - R_l)C(t_s)$$

$$Q(t) = Q(t - t_s) + (\sum_{i=1}^{N} er_i(t - t_f) - R_l)C(t_s)$$

$$Q(t) = Q(t - t_s) + (R(t) - R_l)C(t_s)$$

The switch scheme tries to adjust the input rate $R(t)$ to match the output rate depending on current queue size, i.e., $R(t) = f(Q(t)) \times$ Available ABR Capacity. We assume no higher priority traffic, so $R(t) = f(Q(t)) \times R_l$. Hence,

$$Q(t) = Q(t - t_s) + (f(Q(t - t_s)) - 1)R_l C(t_s)$$

and $(f(Q(t - t_s)) - 1)R_l$ is the rate at which the queue changes in one averaging interval. The function $f(Q)$ is the queue control function. The queue length increases if $f(Q) > 1$, remains constant if $f(Q) = 1$, and decreases if $f(Q) < 1$. Hence, by using an appropriate function $f(Q)$, the queue length can be controlled.

7.2.2 Explicit Rate Feedback

The explicit rate is calculated as follows

$$er_i(t) = f(Q(t - t_f)) \times F(r_i(t - t_p), \text{Link Rate}, \text{HPR})$$

The sources adjust their rates $r_i(t)$ based on the explicit rate feedback from the switch. The feedback reaches the switch after time t_p. Hence, $r_i(t)$ (source rate) can be expressed using the following equation :

$$r_i(t) = er_i(t - t_p) = f(Q(t - t_f)) \times F(r_i(t - 2t_p), \text{Link Rate}, \text{HPR})$$

Since $t_f = 2t_p$, we get

$$r_i(t) = f(Q(t - t_f)) \times F(r_i(t - t_f), \text{Link Rate}, \text{HPR})$$

For simplicity we have assumed there is no HPR traffic (Note in the presence of bursty VBR sources there might not be any steady state of the system). So the above function becomes

$$r_i(t) = f(Q(t - t_f) \times F(r_i(t - t_f), \text{Link Rate})$$

For the *ERICA+* scheme, the above function is as follows

$$r_i(t) = f(Q(t - t_f)) \times maximum(\frac{r_i(t - t_f) \times \text{Link Rate}}{\text{Input Rate}}, \frac{\text{Link Rate}}{n})$$

where *Input Rate* is the ABR input rate measured at the switch. The scheme tries to match the input rate to the link rate by over allocating the rates if the queue is small. If the queue is large, it is drained quickly by using the (1-f(Q)) part of the link capacity.

For other schemes the following modification can be done to incorporate queue control function with that scheme. Let $er_A(t)$ be the explicit rate calculated by an algorithm A. Then add the following as the last step in the algorithm A:

$$er_A(t) = maximum(f(Q(t - t_f)) \times er_A(t - t_f), PCR)$$

where PCR is the peak cell rate. It can be shown that if algorithm A converges to max-min rates, then the modified algorithm also converges to the max-min rate. Further, the queue control function $f(Q)$ can be chosen so that the queue length (and hence delay) is constant in the steady state.

7.2.3 Design of Queue Control Function

The design considerations for the queue control functions are as follows:

- If the queue length is very small, it should be increased so that the scheme can maintain a few cells in the queue which can used when the link is under utilized. This implies that $f(Q)$ should be greater than one for small queue lengths.

- In the steady state, we desire that the queue length be constant and the target rate to be the max-min fairness rate. The function $Q(t)$ satisfies this goal if $f(Q) = 1$ in the steady state.

- If the queue is large, then part of the link capacity is used to drain the queue. Hence $f(Q)$ should be less than one. It is desirable not to use all the capacity to drain the queue. Therefore, there is a minimum threshold, the queue drain limit factor (QDLF), for $f(Q)$.

- The $f(Q)$ function has to be continuous. Discontinuities imply sudden changes which give rise to oscillations.

The queue control function with above properties will be of the form

$$f(Q) = \begin{cases} > 1 & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ < 1 & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where $Q_0 < Q_1 < Q_2 < \infty$

The following functions are possible candidates.

Step function

The step function has multiple thresholds (See figure 7.1). This is the simplest one to implement in hardware (lookup table).

$$f(Q) = \begin{cases} = s_a & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ = s_b & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where $s_a > 1$ and QDLF $< s_b < 1$ are step parameters. In general, it can have n steps. In the above case $n = 4$.

Linear function

The function $f(Q)$ has linear relationship with queue length. (See figure 7.1)

$$f(Q) = \begin{cases} = 1 - m_b \frac{(Q - Q_0)}{Q_0} & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ = 1 - m_a \frac{(Q - Q_1)}{Q_1} & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where m_b and m_a are slope of the linear portions. This function can be implemented in an efficient manner using shift operations, if m_a and m_b are of the form $1/2^k$ and the queue length is counted in terms of Q_0.

Hyperbolic function

The function $f(Q)$ is a hyperbolic function of the queue length. (See figure 7.1)

$$f(Q) = \begin{cases} = \frac{h_b Q_0}{(h_b - 1)Q + Q_0} & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ = \frac{h_a Q_1}{(h_a - 1)Q + Q_1} & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where h_a and h_b are the parameters which control the degree of curvature of the hyperbolic function. This function takes more time to calculate because it has a division operation. For a high value of h_a, the hyperbolic function becomes similar to the step function. For an h_a value near 1, the hyperbolic function approaches the linear function.

Inverse Hyperbolic function

The fraction $f(Q)$ is an inverse hyperbolic function of the queue length for overload conditions (see Figure 7.1). In the underload region, a hyperbolic function is used.

$$f(Q) = \begin{cases} = \frac{\lambda_b Q_0}{(\lambda_b - 1)Q + Q_0} & 0 \leq Q \leq Q_0 \\ = 1 & Q_0 < Q \leq Q_1 \\ = \frac{\lambda_a Q_1}{(\lambda_a + 1)Q_1 - Q} & Q_1 < Q \leq Q_2 \\ = \text{QDLF} & Q_2 < Q < \infty \end{cases}$$

where λ_a and λ_b are the parameters which control the degree of curvature of the inverse hyperbolic and hyperbolic functions. This function is continuous and smooth at both Q_0 and Q_1.

The curve used in the control function in the underload region is called the "a-curve" and the one used in over loaded region is called the "b-curve". The parameters used in the a-curve s_a, m_a, h_a, λ_a are called *a-parameters*. s_b, m_b, h_b, λ_b are called the *b-parameters*. Note that, since all the functions are continuous, at Q_2 we have the equation $f(Q_2) = QDLF$. As a result, Q_2 can be expressed in terms of $QDLF$ and *a-parameter* for linear, hyperbolic and inverse hyperbolic functions.

7.3 Metrics

To compare the performance of the queue control function the following metrics are chosen.

Convergence Time: The time the scheme takes to converge to steady state. To find the convergence time, the variance and standard deviation of the desired variable are calculated between $(i \times t_k, (i+1) \times t_k)$ for $i = 0, 1, \ldots$, where t_k (= 100 ms) is a small time interval. Initially, the standard deviation is large due to oscillations. The convergence time is $i \times t_k$ after which the variance is small.

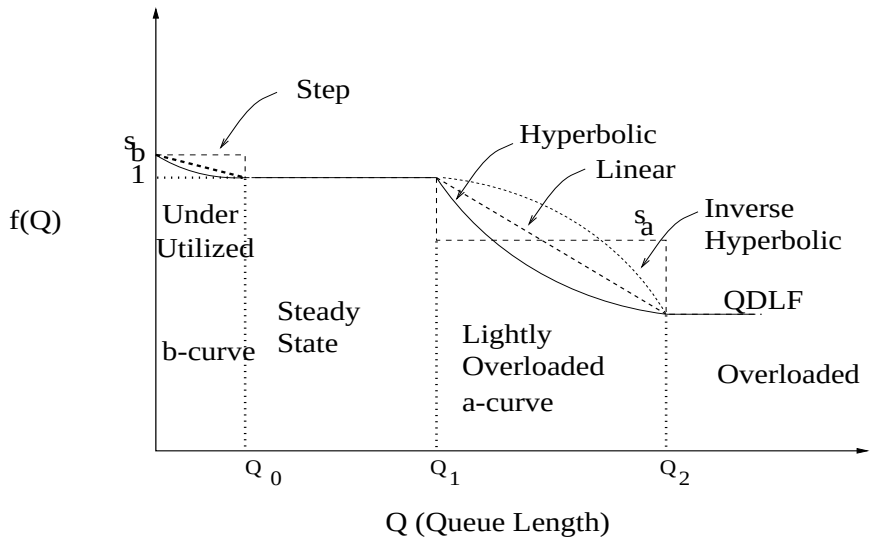


Figure 7.1: Queue control functions. 'b-curve' is used in under utilized region. In steady steady, $f(Q) = 1$ value is used. In lightly overloaded region 'a-curve', which is a decreasing function, is used. In overloaded region the $f(Q)$ value is limited to $QDLF$. Possible queue control functions are step, linear, hyperbolic and inverse hyperbolic.

Also, the graphs of (mean+standard deviation) value of the variable versus time are plotted. From the graph, the convergence time can be calculated.

Standard Deviation: Standard deviation of various quantities like ACRs, queue length and utilization are calculated. In order to separate the oscillations before steady state from affecting the measurement, the variance is measured both before and after steady state is achieved.

Visual inspection of the graphs also gives a good idea about the convergence time and the variations.

7.4 Analytical Explanation

In this section we analyze the behavior of the proposed queue control functions. We assume a simple configuration in our analysis. N infinite ABR sources (always having data to send) are sending data to N ABR destinations (See figure 5.3). The performance study under more stressful conditions is done by a simulation using the Generic Fairness configuration - 2 [108] in section 7.6.

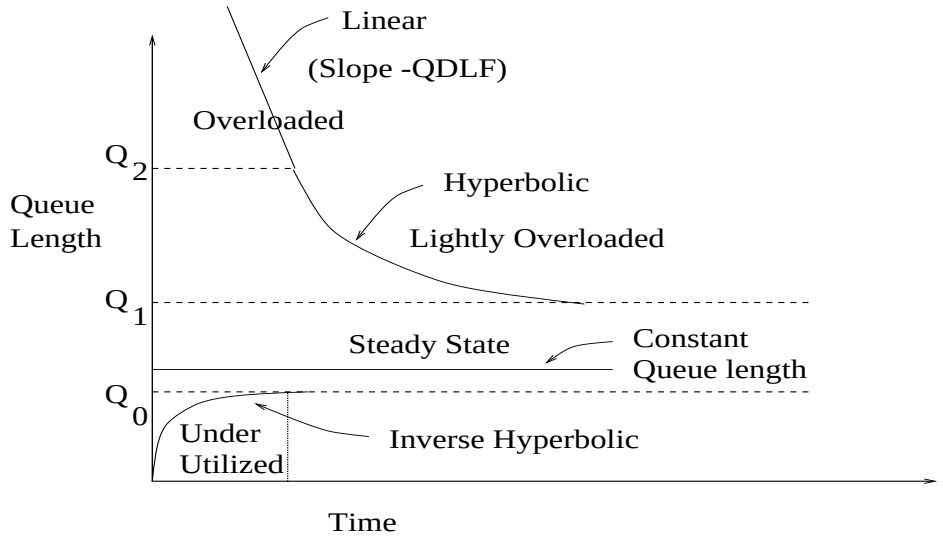


Figure 7.2: Queue Behavior for inverse hyperbolic function.

The change in queue length in a averaging interval t_s is given by:

$$\Delta Q = (f(Q(t - t_f)) - 1) \times R_l \times C(t_s)$$

- *Initial behavior:* In the beginning, the queue lengths grow depending on the initial ICR (initial cell rate). So the maximum queue depends on the ICR and round trip time and is independent of the queue control function used.

The feedback information reaches the sources who in turn adjust their rates accordingly.

- *Under utilized:* The switch initially estimates that the link is under utilized. The queue control function during under utilization is $f(Q) > 1$ for $Q < Q_0$. As a result, the switch gives feedback to the sources to increase their rates. Due to the increased rates, the queue length at the switch increase. The rate of increase of the queue length is dictated by the b-curve of the queue control function.

- *Over loaded:* Either due to the increase in the source rates or due to the high ICRs, an overload condition occurs and queues at the switch grow. Initially the feedback information is not accurate; hence, the queue might grow beyond the Q_2 threshold. In such a heavy overloaded condition, the queues are quickly drained by using the (1-QDLF) fraction of the link capacity. In the meantime the feedback control loop is established, and the switch gives reliable feedback to the sources. For the lightly over loaded region, i.e., queue length in range of $Q_1 \leq Q \leq Q_2$, the a-curve is the queue control function. The value of $f(Q)$ is less than one in this region, which effectively decreases the queue length.

- *Steady state:* The feedback information tries to match the input rate to output rate. As the input rate approaches the output rate, the oscillations die down and the network reaches steady state. In the steady state, the rates and the queue lengths remain constant, since $f(Q) = 1$

It is shown in reference [24] that additive increase and additive decrease lead to steady state, assuming non-dynamically changing environment. For all four queue

control functions $f(Q) > 1$ for $Q_0 \leq Q < Q_1$ and $f(Q) < 1$ for $Q_1 \leq Q < Q_2$. The change in queue length depends on the value of $f(Q) - 1$. Hence, the change in the queue length is an additive increase for the under utilized region and an additive decrease in the over loaded region. Therefore, the queue length converges to a value between Q_0 and Q_1. An interesting point to note is that the amount of increase and the amount of decrease itself is dictated by the current queue length and its distance from the steady state queue length range. The behavior of the queue length for the inverse hyperbolic function is shown in figure 7.2. The figure shows the queue length (y-axis) versus time (x-axis). The queue length decreases linearly in the range $Q_2 \leq Q < \infty$ (highly overloaded) with slope $QDLF$. For queue length in the range $Q_1 \leq Q < Q_2$ (lightly overloaded, near the steady state) the slope is given by $f(Q)-1$. Hence, the queue decreases hyperbolically (inverse of f(Q) function of a-curve) with respect to time. In the steady state, $Q_0 \leq Q < Q_1$, the queue length is constant. In the under utilized range $0 \leq Q < Q_0$, the queue increases inverse hyperbolically since the slope is $f(Q) - 1$ and $f(Q)$ is hyperbolic.

7.4.1 Step Function

If $Q(t - t_f) < Q_0$, then $f(Q(t - t_f) = s_b > 1$, so the queue grows until feedback information is passed to the sources asking them to decrease their rate. The queue grows for t_f time and it can be expressed as follows:

$$Q(t) = Q(t - t_f) + (s_b - 1) \times R_l C(t_f)$$

If the condition $Q_0 < Q(t) < Q_1$ is satisfied and the input rate matches the output rate, then the steady state is achieved, and the queue remains at this constant length.

If $Q_1 < Q(t) < Q_2$, then the $Q(t)$ starts decreasing with slope $-(1 - s_a)$. This decrease also takes place for t_f time, if the queue ends up between Q_0 and Q_1 and if the input rate is close to output rate then again the steady state is achieved. Therefore, for the system to achieve the steady state, the value of the queue length after one feedback interval should be within the range Q_0 and Q_1. This requirement is satisfied if the condition $Q_1 > Q_0 + (s_b - 1) \times R_l C(t_f)$ holds. Since the step function has discontinuities, it is very sensitive to queue length values near the thresholds and steady state might not be reached if the parameters are not set properly. If parameters are not set properly, then the queue grows from a value below Q_0 for t_f time, crosses Q_1, and decreases for t_f time to a value less than Q_0. Then, this pattern repeats.

7.4.2 Linear Function

If $Q(t - t_f) < Q_0$, then $f(Q(t)) > 1$. Similar to the step function, the queue keeps growing for t_f time with a slope of $(f(Q(t - t_f)) - 1) \times R_l$. But unlike the step function, the slope now depends on the value of queue length. After t_f seconds, if the queue $Q(t) > Q_1$, the queue length starts decreasing with a slope of $(f(Q(t) - 1) \times R_l$. The slope now depends on the value of the queue length so that there are no sudden changes in the slope. Therefore, the oscillations are fewer when compared to the step function. If the system is near the steady state, then the oscillations decrease, the queue length reaches a value between Q_0 and Q_1, and the system reaches steady state.

7.4.3 Hyperbolic Function

The analysis for this case is similar to above. If h_a and h_b parameter are close to one (typical values are $h_a = 1.15, h_b = 1.05$) the hyperbolic function has similar behavior as the linear function. If h_a is high, then the hyperbolic function is close

to the step function. Since the hyperbolic function has a larger curvature initially and then smoothes out, $f(Q_1 + \delta Q)$ value will be smaller than the corresponding $f(Q_1 + \delta Q)$ value obtained using the linear function. Hence, the fluctuations in the rates are greater, but the queue drains faster.

7.4.4 Inverse Hyperbolic Function

The behavior of the queue length versus time for the inverse hyperbolic function is shown in Figure 7.2. The behavior of this queue control function has been explained earlier in this section. Since the inverse hyperbolic function is continuous and smooth near Q_1, it gives rise to less oscillations compared to the other cases.

7.5 Simulation: Configuration and Parameters

In this section the two configurations used in the simulations are explained. In all, the simulation no higher priority (CBR and VBR) traffic is present.

7.5.1 Simple Configuration: N Source - N Destinations

In this configuration: (See figure 5.3)

- N infinite sources send data to N destinations
- The traffic is one way
- The initial values of ICR are chosen randomly in the range (0,link rate)
- All links are of length 1000 Km, which corresponds to a propagation delay of 5 ms.
- All links have a bandwidth of 149.76 Mbps (after accounting for SONET overhead).

- The sources start at a random time between (0, t_{RTT}), where t_{RTT} is the round trip time. t_{RTT} = 30 ms for the above configuration.

7.5.2 Generic Fairness Configuration - 2 (GFC-2)

This configuration (see Figure 5.5) was used to test the performance of the queue control functions and the switch scheme under more stressful conditions. The value of link distance, D, was chosen to be 1000 Km. In this configuration, the expected max-min fairness rate for the different VCs are: A VCs = 10 Mbps, B VCs = 5 Mbps, C VCs = 35 Mbps, D VC = 35 Mbps, E VCs = 35 Mbps, F VC = 10 Mbps, G VCs = 25 Mbps, H VCs = 52.5. This configuration is explained more in reference [108].

7.6 Simulation: Results

In this section the simulation results using the above two configurations are given. The graphs of ACR rates, queue length, and utilization are given. The tables and the graphs are used to study the performance of different queue control functions. In the simulations for both configurations, $QDLF$ was chosen to be 0.5.

7.6.1 Simple Configuration: Results

Table 7.2 shows the performance for different step values (parameters) of the step queue control functions as the queue threshold Q_1 is varied. The values of mean bottleneck link queue length, its standard deviation before one second and after one second (last two columns), are shown in the table. Note that Q_2 is fixed given the $QDLF$ and other parameters of the *linear* and *hyperbolic* functions. The number of sources, N, was 3.

The following observations can be made from the Table 7.2.

Queue Control	a param	b param	Q_1	Q_2	Convg time (secs)	Mean Q (cells)	Std Dev (before 1 sec)	Std Dev (after 1 sec)
Step	0.75	1.01	4 Q_0	26 Q_0	-	252.93	552.21	501.60
	0.90	1.01	4 Q_0	26 Q_0	-	98.04	651.82	241.43
	0.90	1.05	4 Q_0	26 Q_0	-	663.63	1226.70	840.36
	0.95	1.01	4 Q_0	26 Q_0	-	251.51	816.62	393.26
	0.95	1.05	4 Q_0	26 Q_0	-	124.11	805.32	240.04
	0.95	1.01	2 Q_0	26 Q_0	-	896.90	1386.87	1036.66
	0.95	1.01	8 Q_0	26 Q_0	-	483.20	1001.54	644.73
Linear	1/16	1/16	2 Q_0	26 Q_0	0.20	311.85	335.61	0.69
	1/16	1/16	4 Q_0	26 Q_0	0.32	403.52	457.90	0.69
	1/16	1/16	8 Q_0	26 Q_0	0.61	402.85	622.02	0.69
Hyperbolic	1.15	1.05	2 Q_0	26 Q_0	-	509.94	423.89	205.65
	1.15	1.05	4 Q_0	26 Q_0	0.32	214.19	500.14	0.86
	1.15	1.05	8 Q_0	26 Q_0	0.82	220.96	862.25	0.63
Inverse Hyperbolic	36	1.05	2 Q_0	26 Q_0	0.22	313.17	525.51	0.69
	16.5	1.05	4 Q_0	26 Q_0	0.25	209.50	580.15	0.50
	6.75	1.05	8 Q_0	26 Q_0	1.12	202.27	704.02	16.98

Table 7.2: Simple Configuration: Results

- The step function never converges entirely. The values are fluctuating near the target values so the standard deviation after one second is lower than in the first second.

- The linear hyperbolic and inverse hyperbolic functions reach steady state after one second since the standard deviation after one second is very small.

- As Q_1 increases, the convergence time increases for the linear, hyperbolic and inverse hyperbolic functions

- For $Q_1 = 2Q_0$, the linear function converges. The value of f(Q) for the hyperbolic function value is less compared to that of linear function so the queue is

drained faster and Q becomes less than Q_0. Therefore, for the hyperbolic function the queue length and rate values are oscillating near the target value. This is mainly because the hyperbolic function is not smooth (not differentiable) near Q_1. The inverse hyperbolic is smooth at Q_1 so the queue lengths do converge in this case.

- For $Q_1 = 8Q_0$, the convergence time for the hyperbolic function and the inverse hyperbolic is more than linear. This happens since the both hyperbolic and inverse hyperbolic functions vary slowly near the steady state. The queue length variance is more for the inverse hyperbolic function.

The graphs 7.3(a), 7.4(a), 7.5(a), 7.6(a) show the ACR rate of the three sources. The mean and standard deviation of the rates and the queue lengths are calculated for every 100 milliseconds. These are shown in figures 7.3(b), 7.4(b), 7.5(b) and 7.6(b) for ACR rates and in figures 7.3(d), 7.4(d), 7.6(d) for the queue lengths. From these graphs, the converging time can be estimated. In the steady state, the oscillations are small and the standard deviation is small compared to mean. So the quantity (mean + standard deviation) has a value close to the mean in the steady state.

For the step function, there are oscillations in all the graphs (rates, queue and utilization). For the linear and hyperbolic functions, the oscillations die down and the system reaches steady state. In the steady state, the rate and queue length are constant and utilization is 100%. Hence the *linear*, *hyperbolic* and *inverse hyperbolic* queue control functions fulfill the desired goal. This is consistent with the analytical explanation given in the previous section.

In all the cases when the queue length and the rates converge, the queue length is non-zero hence, the utilization at the steady state is 100%.

7.6.2 GFC-2 Configuration: Results

The following parameters were used in the simulations for this configuration.

- Thresholds: $Q_0 = 176$, $Q_1 = 4 \times Q_0$, $Q_2 = 26 \times Q_0$, $QDLF = 0.5$
- Step: $s_a = 0.95$, $s_b = 1.01$
- Linear: $m_a = 1/16$, $m_b = 1/16$
- Hyperbolic: $h_a = 1.15$, $m_b = 1.05$
- Inverse Hyperbolic: $\lambda_a = 16.5$, $\lambda_b = 1.05$

Table 7.3 shows the performance for the three queue control functions. The table shows the $H(1)$ VC's mean rate, switch queue length for SW5 and its convergence time, standard deviation before one second and after one second. The queue length variation is present in all three cases. The rate variation is much less in the *linear* and *hyperbolic* functions compared to the *step* function. This is also evident from the graphs that are explained in the next section.

7.6.3 GFC-2 Configuration: Graphs

The graphs in Figure 7.7 and 7.9 were obtained by simulating the GFC-2 configuration using the *step*, *linear*, *hyperbolic* and *inverse hyperbolic* queue control functions. Graphs 7.7(a), 7.8(a), 7.9(a) and 7.10(a) show the ACR rate for one VC of each of A through H type VCs versus time when different queue control functions are used. From these graphs it can be seen that the expected rates are obtained when linear, hyperbolic and inverse hyperbolic functions are used for queue control.

Queue Control Control	Quantity	Converg. Time (secs)	Mean	Std Dev (before 1 sec)	Std Dev (after 1 sec)
Step	H(1) ACR	-	72.81	18.4	4.46
	SW5 Queue	-	284.28	878.63	281.85
Linear	H(1) ACR	1.25	52.46	14.38	1.08
	SW5 Queue	1.30	455.46	1043.71	220.42
Hyperbolic	H(1) ACR	1.45	52.77	13.57	0.58
	SW5 Queue	1.30	361.32	968.27	201.86
Inverse Hyperbolic	H(1) ACR	1.51	52.38	14.04	0.92
	SW5 Queue	2.00	1443.72	3829.17	999.62

Table 7.3: GFC-2 Configuration: Results

The (c) graphs have the queue length for all the switches. The maximum queue is due to the initial overload before the first round trip time. Once the feedback control loop is established, the $f(Q)$ value is $QDLF$ and queues are drained quickly.

When the step function (Figure 7.7(b)) is used the oscillations are more compared to the oscillations when other functions are used. The graphs 7.7(b), 7.8(b), 7.9(b), 7.10(b) plot mean plus standard deviation for VC (ACR) rates. The figures 7.7(d), 7.8(d), 7.9(d), 7.10(d) plot corresponding (mean+standard deviation) graphs for the queue lengths.

Note that in the graphs when the step function is used, some of the VCs do not get their max-min fair share rates and the VCs near the fair share have considerable oscillations. The step function is very sensitive to the queue length variation near the thresholds. Since the configuration is complex, with large number of VCs passing through each of the switches, the queue length and hence the rates vary. For the graphs 7.8(a), 7.9(a) and 7.10(a) the oscillations are only present before steady state.

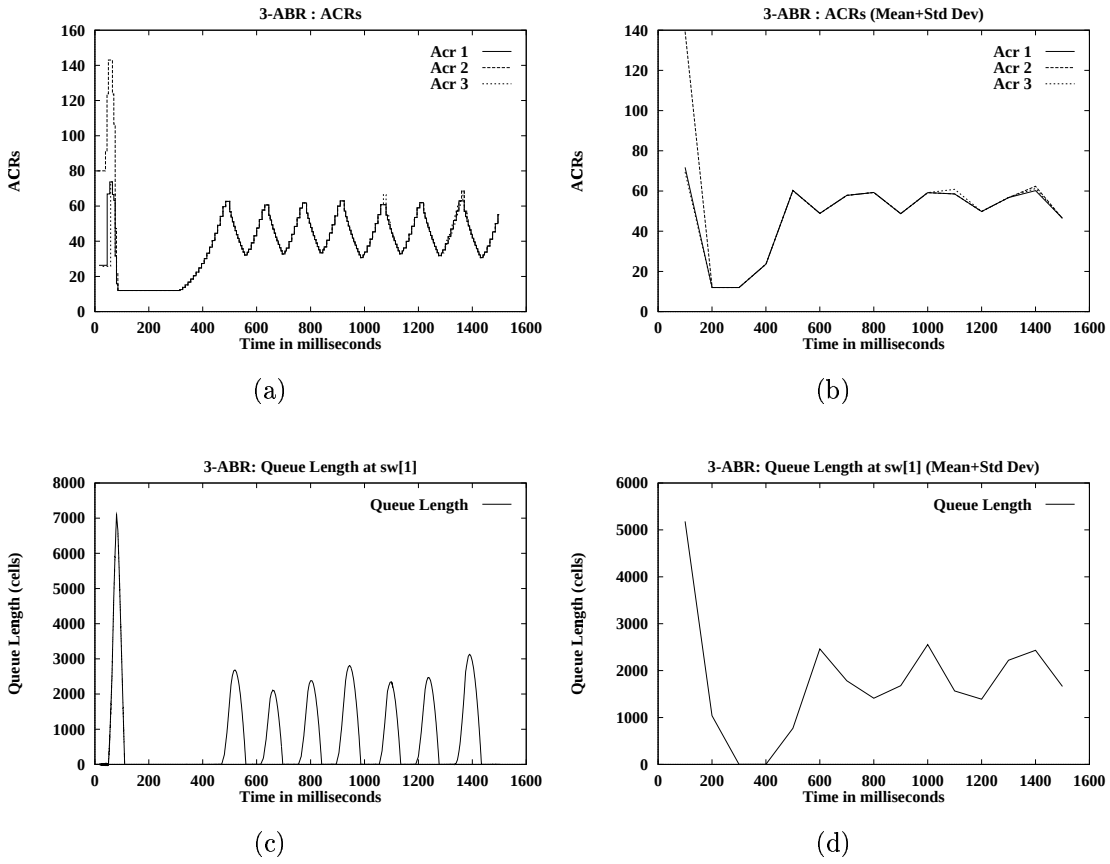


Figure 7.3: Simple Configuration: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using the step queue control function

The oscillations die down and the rates become steady since the function $f(Q)$ changes smoothly. The maximum queue length is same for all queue control functions because this depends only on the ICR. When the inverse hyperbolic function is used, the queues are larger since the steady state queue length is near Q_1.

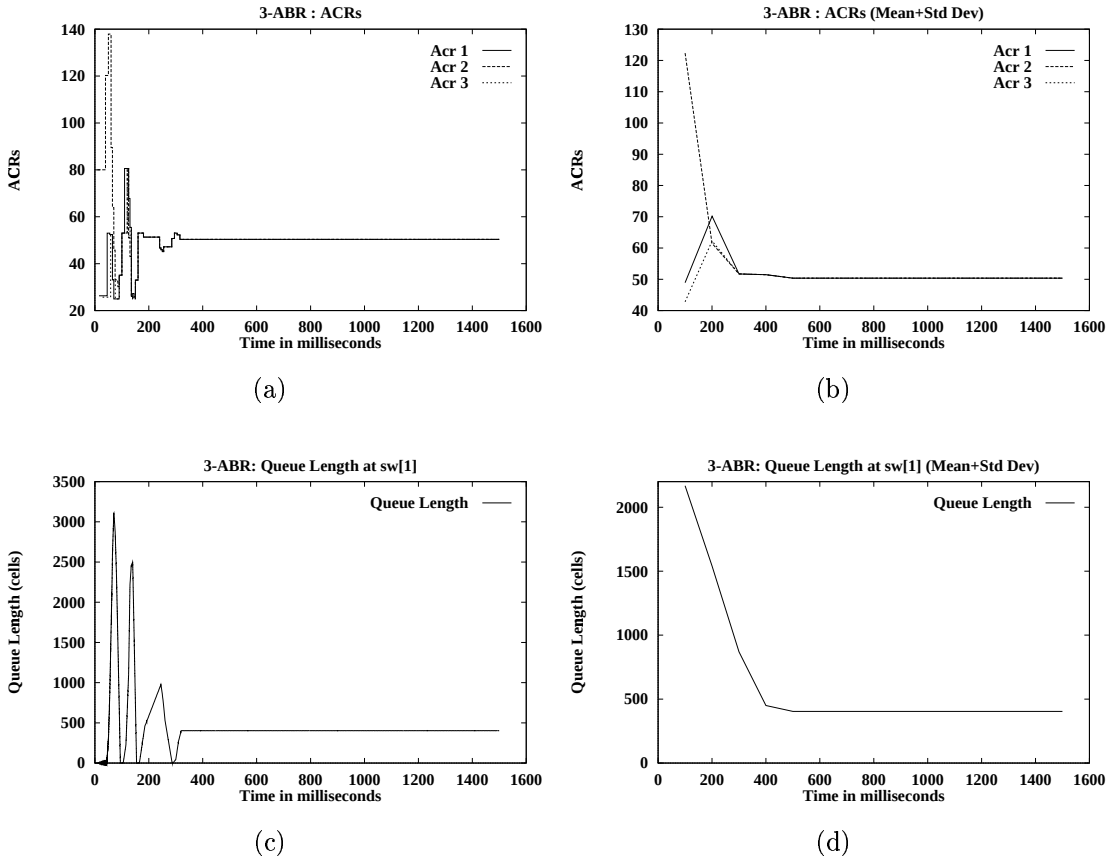


Figure 7.4: Simple Configuration: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using the linear queue control function

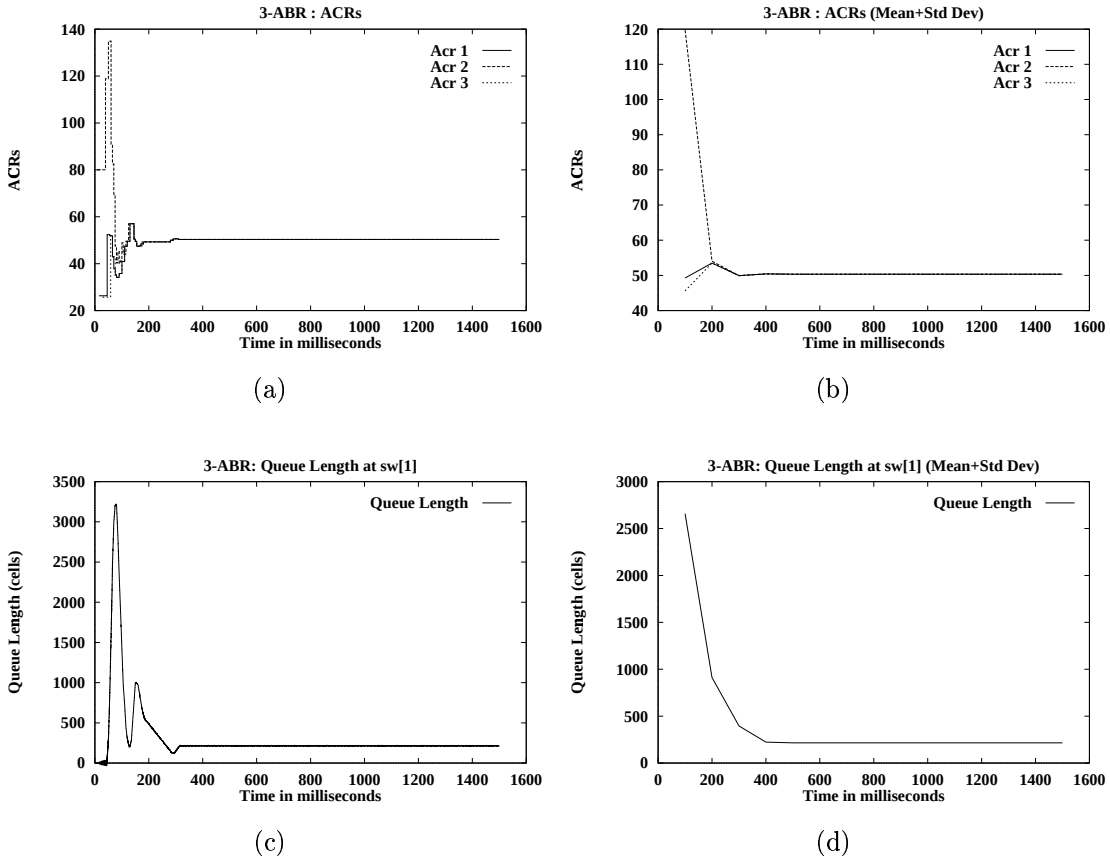


Figure 7.5: Simple Configuration: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using hyperbolic queue control function

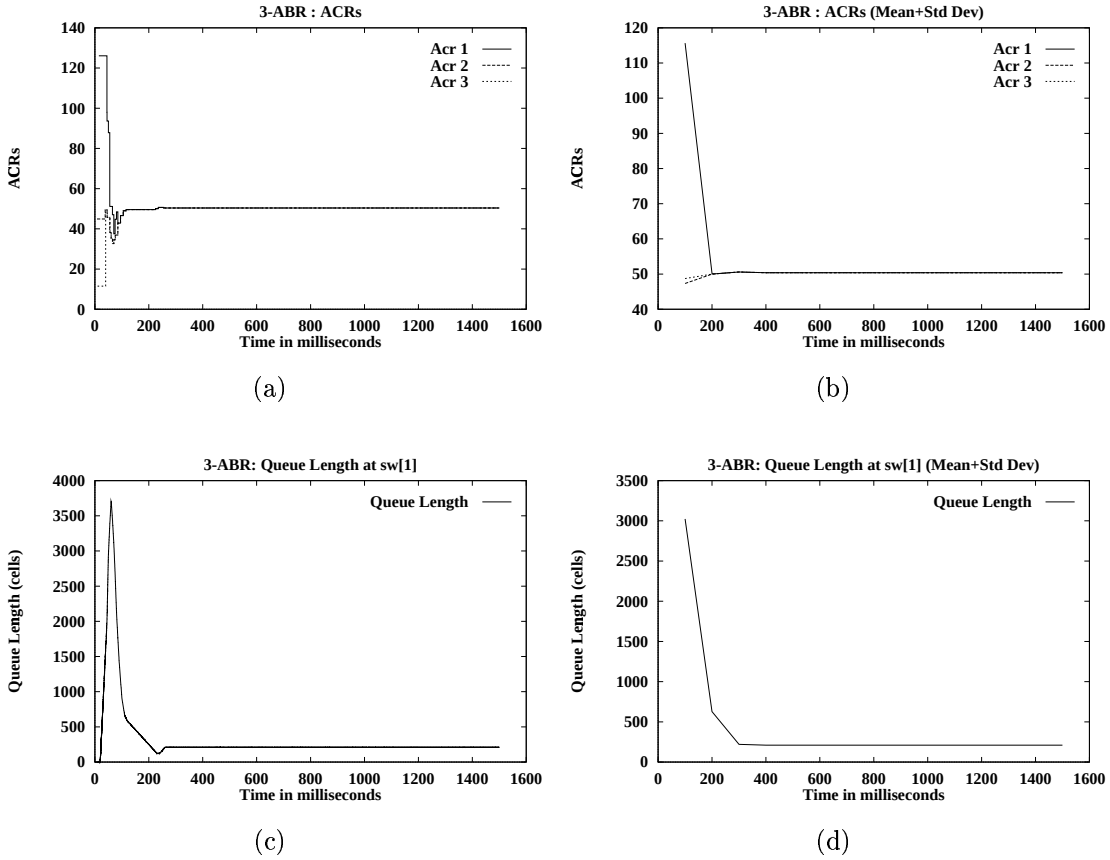


Figure 7.6: Simple Configuration: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using the inverse hyperbolic queue control function

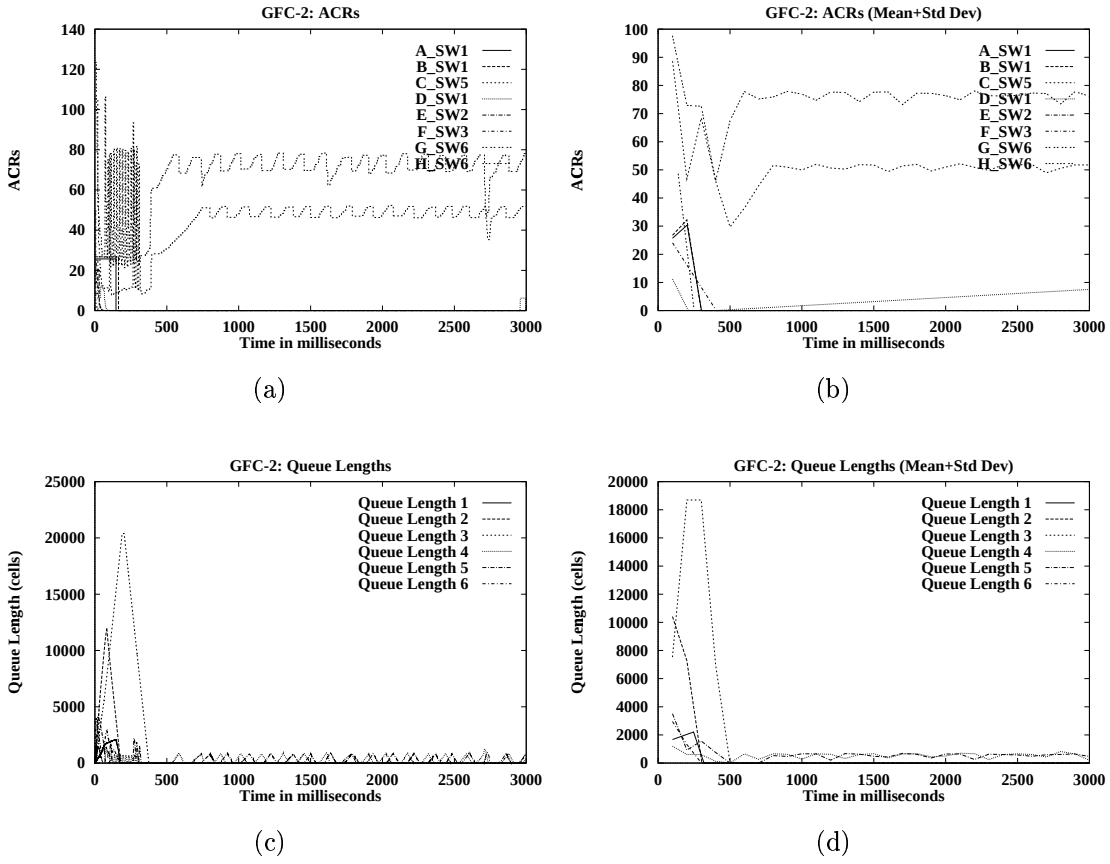


Figure 7.7: GFC-2 Configuration: Rate, Queue and corresponding mean plus standard deviation graphs when using the step queue control function

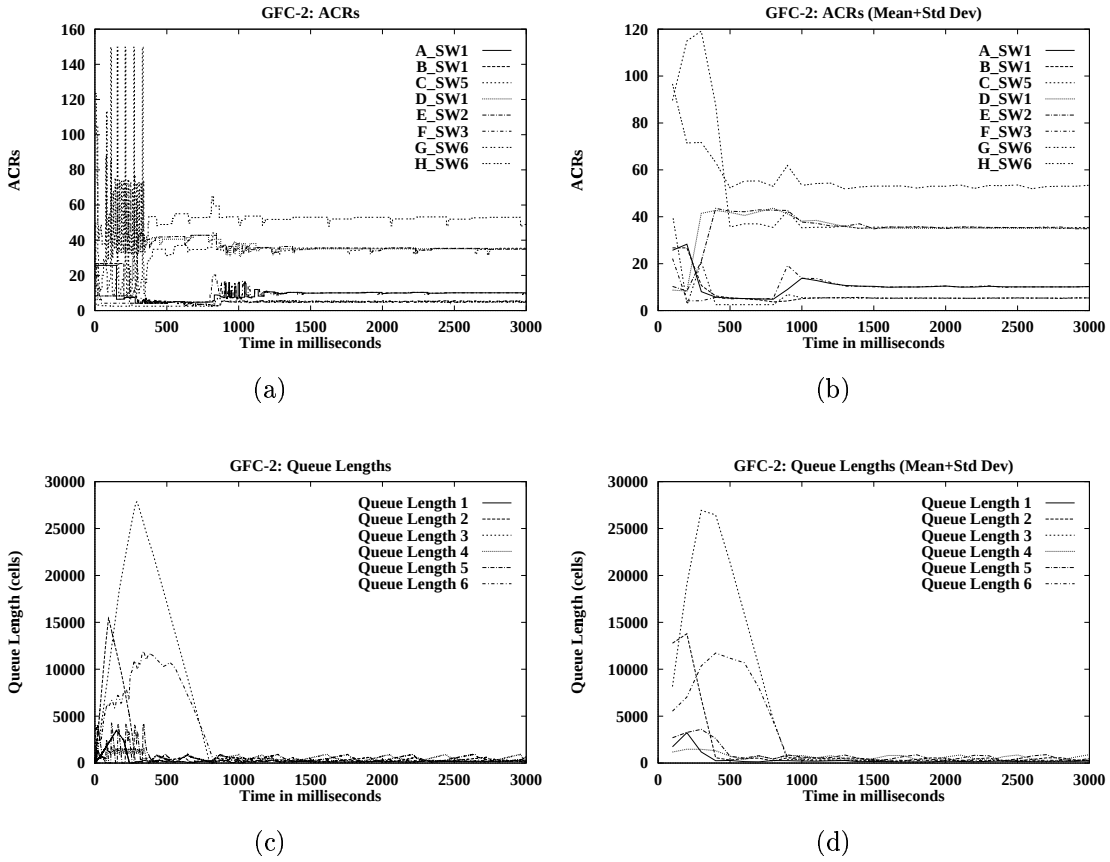


Figure 7.8: GFC-2 Configuration: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using the linear queue control function

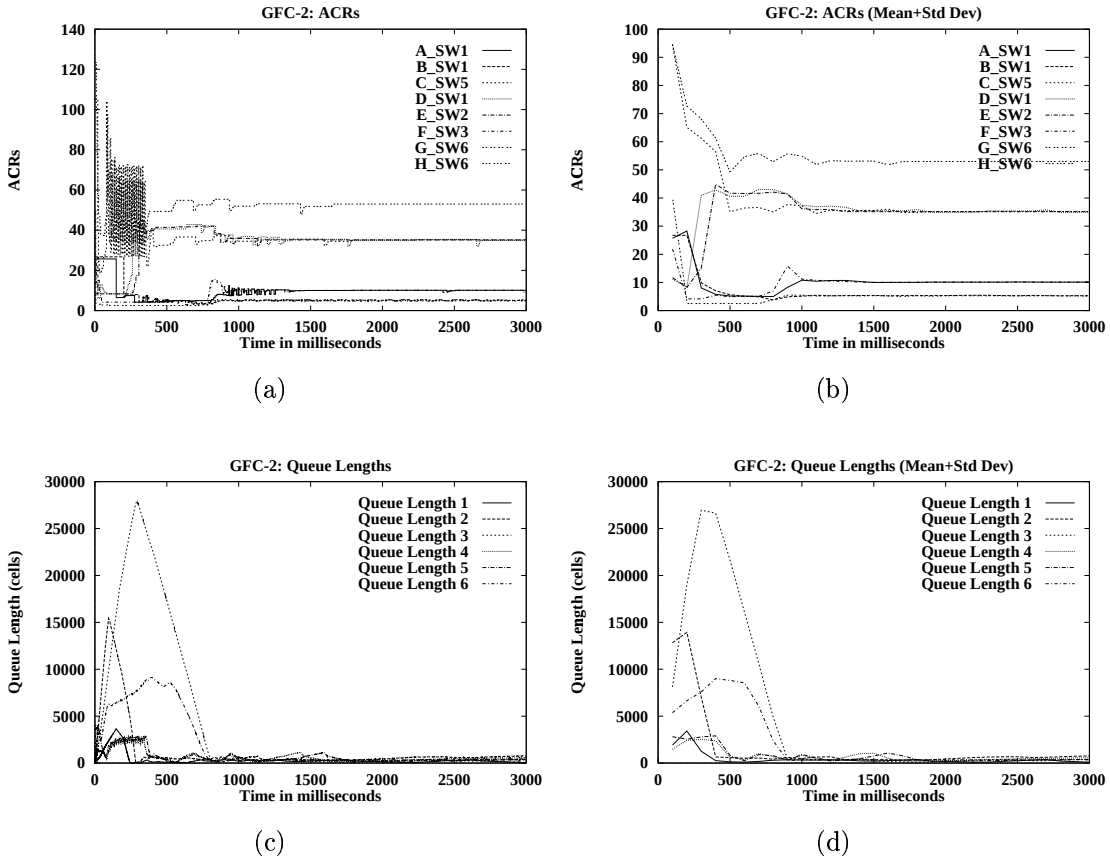


Figure 7.9: GFC-2 Configuration:: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using the hyperbolic queue control function

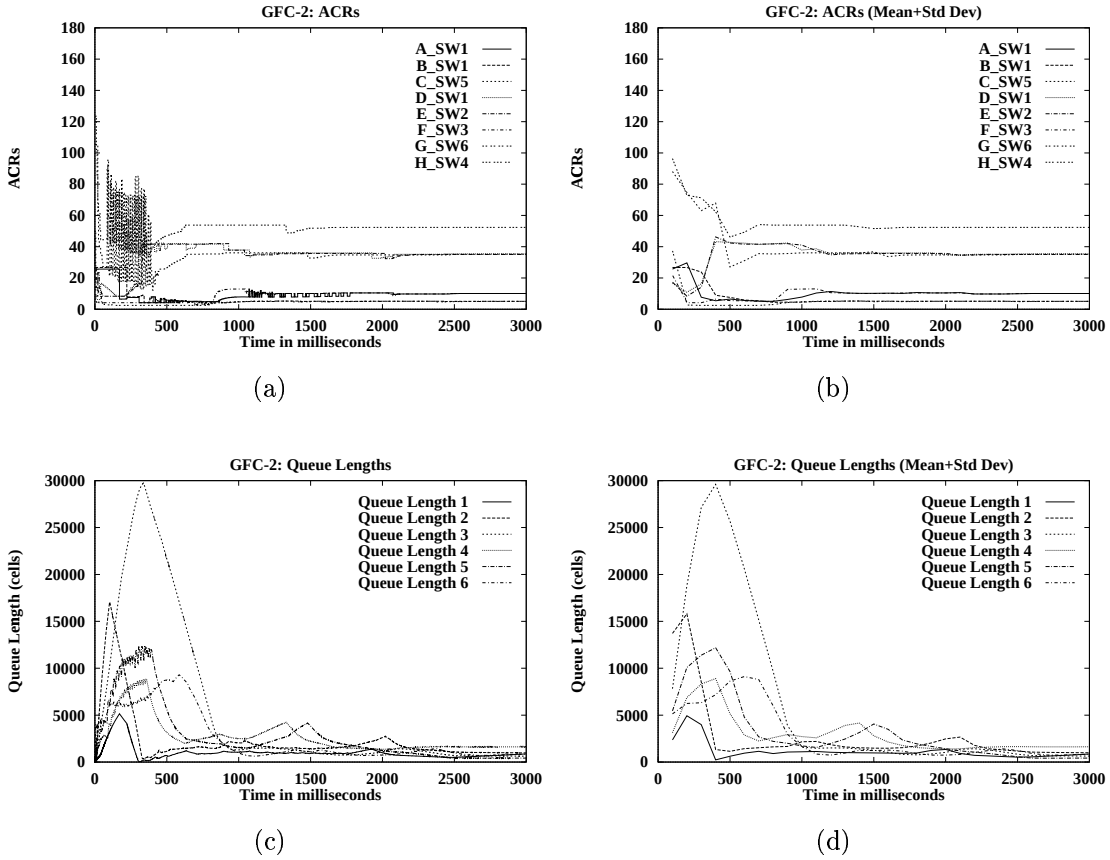


Figure 7.10: GFC-2 Configuration:: Rate, Queue and corresponding mean plus standard deviation graphs obtained when using the inverse hyperbolic queue control function

7.6.4 Summary of Results

The simulation results obtained by using different queue control functions in the simple and the GFC-2 configurations are consistent with the analytical explanation. The step function is sensitive to queue thresholds (Q_0, Q_1, Q_2) used. The other functions are not sensitive to these queue thresholds. A small steady state queuing delay can be achieved by choosing nearby values for Q_0 and Q_1.

7.7 Chapter Summary

In this chapter we have considered the problem of designing a simple and robust queue control function for switch schemes. A switch scheme tries to maximize utilization, minimize queuing delay and give max-min fair rates to the sources. It is also desirable to have fewer oscillations in rates and queue length to support multimedia over ABR service. We assume a switch scheme model that dynamically adjusts the rate of the sources to match the output rate and drain large queues. The design considerations were discussed with analytical explanations. Four different queue control functions were analyzed. The choice of parameters for the queue control functions were explored analytically and through simulation. The simulations showed that even in a complex configuration (like GFC-2), the system behavior was consistent with of the analytical explanation. When the step function is used, the system oscillates and does not converge in most cases. From both the analytical and the simulation results, it can be concluded that the inverse hyperbolic is the best queue control function, followed by the hyperbolic and the linear queue control functions. For simpler implementation complexity, the linear function is recommended.

CHAPTER 8

APPLICATION LAYER TECHNIQUES FOR ADAPTIVE STREAMING OF MULTIMEDIA

The Internet was designed for best-effort data traffic. With the development of high-speed technologies such as ATM, gigabit Ethernet, and frame relay, user expectations have increased. Real-time multimedia applications including *video-on-demand*, *video-broadcast*, and *distance education* are expected to be supported by these high-speed networks. Organizations such as the IETF, ATM Forum, ITU-T are developing new protocols (e.g., real-time transport protocol), service models (e.g., integrated services and differentiated services), and service classes (e.g., ATM Forum service categories and ITU-T transfer capabilities) to satisfy the requirements of multimedia applications.

The Internet is a heterogeneous environment connecting various networking technologies. Even with networking support through service classes, the available network resources to multimedia applications will change dynamically over time. For example, the network conditions may change due to difference in link speeds (ranging from 28.8 Kbps modem links to 622 Mbps OC-12 links) or the variability in a wireless environment caused by interference and mobility. One way of achieving the desired

quality of service in such situations is by massively over-provisioning resources for applications. This solution, however, leads to inefficiency. Without over-provisioning, network resources can be used efficiently if applications are capable of adapting to changing network conditions.

Adaptation of multimedia applications can be done at several layers of the network protocol stack. At the physical layer, adaptive power control techniques can be used to mitigate variations in a wireless environment. At the data link layer, error control and adaptive reservation techniques can be used to protect against variation in error and available rate. At the network layer, dynamic re-routing mechanisms can be used to avoid congestion and mitigate variations in a mobile environment. At the transport layer, dynamic re-negotiation of connection parameters can be used for adaptation. Applications can use protocols such as the real-time streaming protocol (RTSP) [104] and the real-time protocol (RTP) [103]. At the application layer, the application can adapt to changes in network conditions using several techniques including hierarchical encoding, efficient compression, bandwidth smoothing, rate shaping, error control, and adaptive synchronization. This chapter focuses mainly on the application layer techniques for adaptation.

8.1 Issues in Supporting Adaptive Multimedia

Before presenting the various application level adaptation techniques we first present the issues and problems related to supporting adaptive multimedia applications. Multimedia applications have different characteristics and requirements compared to traditional data applications. Multimedia applications are delay sensitive, to a certain extent loss tolerant, and may be interactive unlike the data applications.

8.1.1 Challenges of Supporting Adaptive Multimedia

The problem of supporting multimedia applications is challenging because of the following network and system related phenomena:

Bandwidth: Multimedia applications have varying bandwidth requirements (e.g., amount of data varies form frame to frame in a video application). The available network capacity is also a varying quantity. Hence, matching the varying available bandwidth with dynamically changing bandwidth requirement poses a challenging problem.

Error Rate: Multimedia applications are sensitive to loss. For good quality they need certain minimum guarantees on error rate. Typical error requirements for audio and video are in the range of $10^{-4}-10^{-5}$ and $10^{-3}-10^{-4}$ respectively [112].

Delay and Jitter: *Interactive* multimedia applications need minimum delay bounds to be guaranteed. Delay jitter[1] (variation in delay) degrades multimedia applications' performance since they need data in a periodic manner (e.g., MJPEG application with frame rate of 30 frames/second need complete frame data every 33 milliseconds.)

Synchronization: Synchronization of different media arriving at different times is necessary at the user end.

Heterogeneity: Multimedia servers have to simultaneously handle receivers which have different characteristics and requirements.

[1] For non-interactive multimedia applications, the jitter can be mitigated by buffering at the client side.

User Interface: The application programming interface (API) has to be developed to control the adaptation process. Ideally, the user interface should be something simple such as one knob for controlling each feature of the multimedia application.

8.1.2 Client/Server Issues

The client/server model depicted in Figure 8.1 shows the various components at the server and client side of the system. The figure also shows the flow of feedback, which can be used at various layers of the network protocol and at the application layer to adapt to changing conditions.

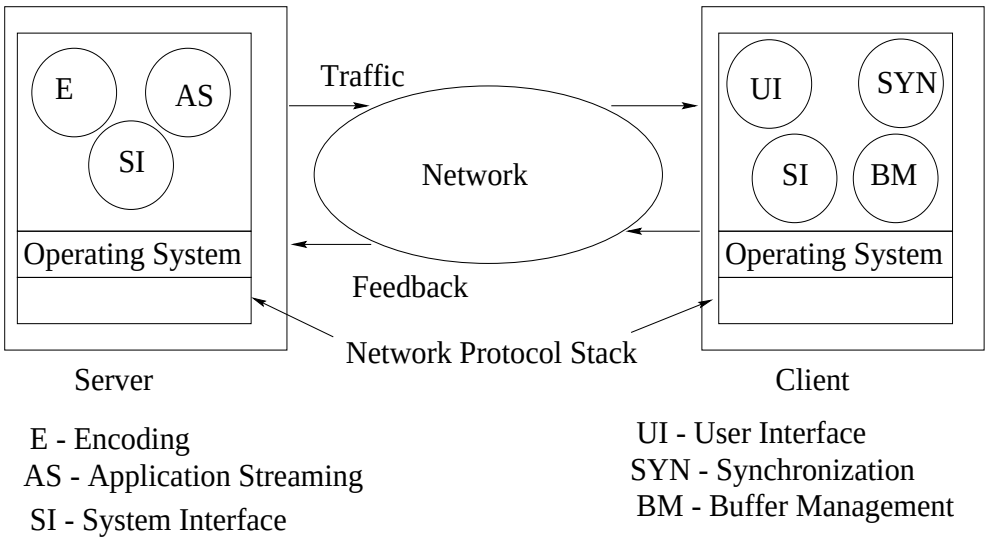


Figure 8.1: Client/Server model showing the various components at the server and client to support adaptive multimedia applications.

Table 8.1 gives the list of techniques that can used in the various components of client/server model for adaptation. These techniques are described in detail in the

subsequent sections. The *C/S* column indicates whether the technique is a client side or server side issue. The *dependency* column of the table explains how each technique relates to other components. The issue addressed and the advantage of the technique are also shown in the table.

Support for multimedia applications in the current Internet is at its initial stages. In corporate intranets where bandwidth is reasonably abundant, ranging from 380 Kbps to 100 Mbps, multimedia applications such as video conferencing can be easily supported. In public networks such as the Internet, however, the quality of service achieved by multimedia applications is much smaller than in intranets. With IETF working to solve the QoS problem, we expect the support for multimedia applications to improve considerably in the coming years. We believe (video based) interactive multimedia applications will be the next killer application after Voice over IP (Internet Protocol). With the advent of optical networks which increases bandwidth dramatically, increasing processing speeds, and advances in compression technologies networked multimedia applications of the future may be able to support HDTV (high definition television) quality of video.

There are also several issues, which are not tackled by the application level techniques discussed here and or by techniques used in other layers of protocol stack discussed elsewhere. A few of these are mentioned below:

Middleware: Designing and implementing middleware to support adaptive multimedia applications is still unaddressed. Middleware should provide simple interfaces between the application, operating system and the network.

CPU to NIC bottleneck: Network interface cards (NIC) are designed to keep up with the growing link capacity. CPU speeds are also doubling every eighteen

Comp-onent	C/S	Technique	Depend-ency	Issue tackled	Advantage
Encoding	S	Compr. algorithms	Streaming scheme	Match source to available rate	Independent of network Layer
Appln Streaming	S	Multiple Layer trans.	Encoding	Mitigate effect of congestion	Can support heterogeneous receivers
	S	Rate Shaping	Encoding, Network Layer	Match source to available rate	Use network feedback
	S&C	Error Control	Encoding, Network Layer	Mitigate effect of losses	Use network feedback
	C	Synchro-nization	Network Layer	Synchro-nizing different media	Solves synchro-nization problem
	S&C	Smoothing	Network Layer	Change trans. scheme based on a-priori knowledge	Better use of resources
OS System	S&C	Process Scheduling	Network Layer	Treat multimedia application preferentially	Can satisfy real-time requirements requirements

Table 8.1: Adaptation techniques that can be used in the different components of the client/server (C/S) model

months. But there is a bottleneck at the I/O bus, which connects the CPU to the NIC. Traditionally NICs are treated as peripheral devices, but if the system has to keep up with the increasing speeds then they should be considered as peer devices to the CPU.

Scalability: Wide scale deployment and usage of multimedia applications is still at its the early stages. The problem of scalability will have to be addressed as the demand and capability of the multimedia applications grows.

8.2 Compression Level Methods

While the rest of the chapter deals with multimedia in general, in this section we briefly examine video compression algorithms and their features which are useful for adaptation. The two reasons for focusing on video are: (1) video requires larger bandwidth (100 Kbps to 15 Mbps) than audio (8 Kbps - 128 Kbps), and (2) humans are more sensitive to loss of audio than video, because audio is important for comprehension. Therefore, audio is given higher priority, as a result only the video can be used for adaptation. Hence, we generally should bias towards adapting the video part of the multimedia application.

Transmitting raw video information is inefficient. Hence, video is invariably compressed before transmission. The three main compression techniques used for video are: (1) discrete cosine transformation (DCT) based, (2) wavelet transforms based, and (3) proprietary methods. Other methods of compressing video not discussed here include vector quantization [51, 50] and content-based compression. Adapting to changing network conditions can be achieved by a number of techniques at the compression level (video encoder) including layered encoding, changing parameters

of compression methods, and using efficient compression methods. In the event of bandwidth not being available to the video source, it can reduce its encoding rate by temporal scaling (reducing frame rate) or spatial scaling (reducing resolution).

8.2.1 MPEG Compression Standard

DCT is the compression method used in the popular MPEG (Moving Picture Experts Group) set of standards [1]. MPEG standards are used for both audio and video signals. MPEG-2, MPEG-1 and JPEG (an earlier standard for still images) all use discrete cosine transformations. The transformed coefficients are quantized using scalar quantization and run length encoded before transmission. The transformed higher frequency coefficients of video are truncated since the human eye is insensitive to these coefficients. The compression relies on two basic methods: intra-frame DCT coding for reduction of spatial redundancy, and inter-frame motion compensation for reduction of temporal redundancy. MPEG-2 video has three kinds of frames: I, P, and B. I frames are independent frames compressed using only intra-frame compression. P frames are predictive, which carry the signal difference between the previous frame and motion vectors. B frames are interpolated, i.e., encoded based on the previous and the next frame. B frames achieve the greatest compression at the the expense of decoding time. MPEG-2 video is transmitted in a group of pictures (GoP) format that specifies the I, P, and B frames sequence used in the video stream.

There are several aspects of the MPEG compression methods that can be used for adaptation. First, the rate of the source can be changed by using different quantization levels and encoding rate [28, 16]. Second, DCT coefficients can be partitioned and transmitted in two layers with different priorities. The base layer carries the

important video information and additional layer improves the quality. In the event of congestion, the lower priority layer can be dropped to reduce the rate [30, 89, 90].

8.2.2 Wavelet Encoding

In wavelet compression, the image is divided into various sub-bands with increasing resolutions. Image data in each sub-band is transformed using a wavelet function to obtain transformed coefficients. The transformed coefficients are then quantized and run length encoded before transmission. In a sense, wavelet compression results in progressively encoded video. Two common approaches for wavelet compression are to use a motion-compensated 2-dimensional (2-D) wavelet function [115] or a 3-D wavelet [93].

Wavelet compression overcomes the blocking effects of DCT based methods since the entire image is used in encoding instead of blocks. An important feature of wavelet transforms is the support of scalability for image and video compression. Wavelet transforms coupled with encoding techniques provide support for continuous rate scalability, where the video can be encoded at any desired rate within the scalable range [26]. A wavelet encoder can benefit from network feedback such as available capacity to achieve scalability [23].

8.2.3 Proprietary Methods

Commercial applications such as Real Networks Inc.'s RealVideo and Intel's Indeo use proprietary methods for compression and adaptation. These proprietary schemes use both DCT based and wavelet based techniques for compression. A distinguishing feature of these methods is that they are optimized to work for particular bandwidths

such as 28.8 Kbps and 56 Kbps. Some of the techniques used for adaptation by these applications are discussed later in the chapter.

8.3 Application Streaming

Most multimedia applications transmit data in streams, and at the user end the data is used as soon as it is received. Application streaming is the technique used by such multimedia applications for sending multimedia data streams. At the application streaming level, adaptation techniques include layered encoding, adaptive error control, adaptive synchronization and smoothing. These methods can be classified into *reactive* and *passive* according to their approach towards adaptation. In *reactive* methods, the application modifies its traffic to suit the changes in the network. In *passive* methods, the application aims to optimize the usage of network resources. Rate shaping and smoothing of stored video are example applications of reactive and passive methods respectively.

8.3.1 Layered Encoding

In layered encoding, a passive method, the video information is encoded into several layers. The base layer carries important video (lower order coefficients of DCT) and critical timing information such as timestamps. The higher layers improve the quality of video progressively. The receiver can get a reasonable quality with the base layer, and quality improves with the reception of higher layers. The encoder assigns priorities to the encoded layers with the base layer having the highest priority. When the network transmits layered video, it can drop lower priority (higher) layers in the event of congestion.

A discussion of adaptive transmission of multi-layered video is given in reference [119]. Here the layered encoding method is made reactive by adding or dropping layers based on network feedback. The authors discuss both credit-based and rate-based approaches for providing feedback.

An optimal data partitioning method for MPEG-2 encoded video is discussed in reference [30]. In this method, the data in the MPEG-2 stream is partitioned into two layers (which can be done even after encoding). The two streams are transmitted over an ATM-based network, with ATM cells of the lower priority stream having their CLP (cell loss priority) bit set. The paper discusses an optimal algorithm to partition the MPEG-2 video stream. The problem is posed as an optimization problem and the Lagrangian optimization technique is used for finding the optimal partitioning for I, P, and B frames. Data partitioning methods can benefit from network feedback. For example, if the network indicates that more bandwidth is available, more data can be sent in the base layer, and conversely data in the base layer can be reduced when bandwidth is scarce.

8.3.2 Receiver Driven Multicast

Receiver driven Layered Multicast (RLM), a reactive method, was the first published scheme which described how layered video can be transmitted and controlled [80]. In RLM, receivers dynamically subscribe to the different layers of the video streams. Receivers use "probing" experiments to decide when they can join a layer. Specifically, if a receiver detects congestion, the receiver quits the multicast group of the current highest layer (drops a layer), and when extra bandwidth is available, it joins the next layer (adds a layer). Network congestion is detected by packet losses. Extra

capacity is detected by join experiments. In a join experiment, the receiver measures the packet loss after joining. The join experiment fails if the packet loss is above a certain threshold. Receiver join experiments are randomized to avoid synchronization. The overhead of join experiments in the presence of a large number of receivers is controlled by the receivers learning from the join experiments of others, instead of initiating their own.

Currently, extra capacity is only estimated in RLM. The low-level network feedback can aid the receivers in measuring precisely the available capacity. Hence, this scheme will benefit if network layer feedback is used.

Layered video multicast with retransmission [125] is another method which uses layered video. The issue of inter-session fairness and scalable feedback control of layered video is discussed in reference [124].

8.3.3 Rate Shaping

Rate shaping techniques are reactive and attempt to adjust the rate of traffic generated by the video encoder according to the current network conditions. Feedback mechanisms are used to detect changes in the network and control the rate of the video encoder.

Video has been traditionally transported over connections with constant bit rate (e.g., telephone or cable TV networks). The bandwidth requirements of the video sequence changes rapidly due to scene content and motion. The variable rate video is sent to a buffer that is drained at a constant rate. In such a situation, the video encoder can achieve constant rate by controlling its compression parameters based on

feedback information such as the buffer occupancy level. This is commonly done in the H.263 and MPEG video standards.

A similar technique is used for adapting video and audio to network changes. In these cases, the feedback from the network is used instead of local buffer information. Control mechanisms for audio and video are presented in reference [16, 15].

coarserlyRate shaping of the IVS video coder (which uses the H.261 standard) is discussed in [16]. The rate shaping can be obtained by changing one or more of the following:

- **Refresh rate:** Refresh rate (frame rate) is the rate of frames which are encoded by the video encoder. Decreasing the refresh rate can reduce the output rate of the encoder.

- **Quantizer:** This specifies the number of DCT coefficients that are encoded. Increasing the quantizer decreases the number of encoded coefficients and the image is more encoded.

- **Movement detection threshold:** For inter-frame coding, the DCT is applied to signal differences. The movement detection threshold limits the number of blocks which are detected to be "sufficiently different" from the previous frames. Increasing this threshold decreases the output rate of the encoder. Again, this results in reduced video quality.

Two modes are used for controlling the rate of the encoder. In Privilege Quality mode (PQ mode), only the refresh rate is changed. In Privilege Rate mode (PR mode), only the quantizer and movement detection threshold are changed. PQ mode

control results in higher frame rates, but with lower SNR (signal-to-noise ratio) than PR mode.

The packet loss information is used as feedback. The receiver periodically sends its current loss rate. The following simple control algorithm is used to dynamically control the rate of the video encoder:

If *median loss* > *tolerable loss*

$$max\_rate = \max(max\_rate/GAIN,\ min\_rate)$$

else

$$max\_rate = \max(max\_rate + INC,\ min\_rate)$$

In the presence of congestion, indicated by loss, the rate is quickly decreased due to multiplicative factor *1/GAIN*. Otherwise, the rate is slowly increased by the additive factor *INC*. This multiplicative decrease, additive increase mechanism adapts well to network changes.

Two dimensional scaling changes both the frame rate and the bit rate based on the feedback [84]. Experimental results show that the system performs well in a rate constrained environment such as the Internet. A heuristic (success rate) is used to decide whether the rate can be increased. The low-level network feedback information, if available, can replace this heuristic.

Rate shaping mechanisms use similar methods but differ in how the rate shaping is achieved. Other rate change approaches include block dropping [130] and frame dropping [97].

8.3.4 Error Control

The error rate is variable in a wireless network due to interference, and the loss rate is variable in the Internet due to congestion. Multimedia applications need to adapt to changes in error and loss rates. Two approaches to mitigate errors and losses are Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). ARQ is a closed-loop and reactive mechanism in which the destination requests the source to retransmit the lost packets. FEC is an open-loop and passive method in which the source sends redundant information, which can partly recover the original information in the event of packet loss. ARQ increases the end-to-end delay dramatically in networks such as the Internet. Hence, ARQ is not suitable for error control of interactive multimedia applications in the Internet. It may be used in high-speed LANs where round trip latencies are small.

Other error control methods include block erasure codes, convolutional codes, interleaving and multiple description codes.

Adaptive FEC for Internet Audio

An adaptive FEC-based error control scheme (a reactive method) for interactive audio in the Internet is proposed in reference [65]. The FEC scheme used is the "signal processing" FEC mechanism [94]. Assume that the packets are numbered, $1, 2, \ldots, n$. In this scheme, the n^{th} packet includes, in addition to its encoded signal samples, information about previous packet $(n-1)$ which can be used to approximately reconstruct that packet if it is lost. The IETF recently standardized this scheme to be used in Internet telephony. The scheme works only for isolated packet losses, but can be generalized to tolerate consecutive packet losses by adding redundant versions of

additional (previous) packets ($n-2$ and $n-3$). The FEC-based scheme needs more bandwidth, so it should be coupled with a rate control scheme. The joint rate/FEC scheme can be used to adaptively control the rate and the amount of redundant information to be sent by the FEC method. The inventors of the scheme formulate the problem of choosing the FEC-method to use under the constraints of the rate control scheme as an optimization problem. A simple algorithm is used to find the optimal scheme. Actual measurements of the scheme for audio applications between France and London have shown that the scheme performs well and the perceptual quality of the audio is good.

Adaptive FEC for Internet Video

An adaptive FEC-based scheme for Internet video is discussed in reference [14]. The packet can carry redundant FEC information for up to four packets, i.e., packet n carries redundant information about packets $n-1$, $n-2$, $n-3$. Let $n-i$ indicate that packet n includes information about $n-i$. The different possible combinations of these methods are: (n), (n, n-1), (n, n-2), (n, n-1, n-2) and (n, n-1, n-2, n-3). These are numbered as combination-1 through combination-5, respectively. Different combinations can be used to adapt to network changes. The network changes are detected through packet loss, and a loss threshold (high loss) is used in the algorithm for adaptation. The following simple adaptation algorithm was used:

If $loss \geq high\ loss$

$Combinaton = \min(Combination+1,4)$

else

$Combinaton = \max(Combination\text{-}1,0)$

This algorithm adds more error protection when there is more loss and less protection when the losses are low.

One way to use network feedback in this method is to couple the rate available and the FEC *Combination* used. For example, information about available rate and loss rate received as feedback from network can be used to choose the FEC combination for error protection.

8.3.5 Adaptive Synchronization

Synchronization is an important problem for multimedia applications. Synchronization problems arise due to clock frequency drift, network delay, and jitter. Adaptive synchronization can be used for multipoint multimedia teleconferencing systems [79, 81]. The adaptive synchronization technique proposed in reference [79] is immune to clock offset and/or clock frequency drift, does not need a global clock, and provides the optimal delay and buffering for the given QoS requirement. The adaptive synchronization technique can be used to solve both intramedia (in a single stream) synchronization and intermedia (among multiple streams) synchronization.

The main idea used in the synchronization algorithm is to divide the packets into *wait*, *no wait* and *discard* categories. Packets in the *wait* bucket are displayed after some time, *no wait* packets are displayed immediately, and *discard* category packets are discarded.

The basic adaptive synchronization algorithm requires the user to specify the acceptable synchronization error, maximum jitter, and maximum loss ratio. The sender

is assumed to put a timestamp in the packets. At the receiver, the playback clock (PBC) and three counters for *no wait*, *wait* and *discard* packets are maintained. The algorithm specifies that when packets arrive early and enough wait packets have been received, the PBC is incremented. Similarly, when a threshold of *no wait* or *discard* packets are received, the PBC is decremented. This adaptive algorithm is shown to be immune to clock drift. Achieving intramedia synchronization is a straight-forward application of the basic algorithm. For intermedia synchronization, a group PBC is used, which is incremented and decremented based on the slowest of the streams to be synchronized.

The network delay is only estimated in this adaptive algorithm. The adaptation can benefit if the low-level feedback provides accurate information on the delay experienced in the network.

8.3.6 Smoothing

One way to mitigate the rate variations of the multimedia application is to perform shaping or smoothing of the video information transmitted. Recent studies show that smoothing allows for greater statistical multiplexing gain.

For live (non-interactive) video, a sliding-window of buffers can be used, and the buffer can be drained at the desired rate. This method is used in SAVE (smoothed adaptive video over explicit rate networks) [28], where a small number of frames (30) is buffered in a window. The video is transmitted over the ATM ABR (available bit rate) service, where the feedback from the network is indicated explicitly. The SAVE algorithm (a reactive method) uses this feedback information to dynamically change

the quantizer value of the MPEG-2 encoder. Note that this method already uses the low-level network feedback. Similar approaches have been proposed in [77, 78].

For pre-recorded (stored) video, the a-priori video (frame) information can be utilized to smooth the video traffic at the source. Bandwidth smoothing, a passive method, can reduce the burstiness of compressed video traffic in video-on-demand applications.

The main idea behind smoothing techniques is to send ahead of time large frames so that the bandwidth out of the buffer can augment the network bandwidth. There has been considerable research in this area resulting in several smoothing algorithms [35, 40, 33, 34, 101]. These differ in the optimality conditions achieved, and whether they assume that the rate is constrained or the client buffer size is limited. A good comparison of bandwidth smoothing algorithm is given in reference [39]. In the next subsection, we discuss the idea of bandwidth smoothing in more detail.

Smoothing Algorithms

A compressed video stream consists of n frames, where frame i requires f_i bytes of storage. To permit continuous playback, the server must always transmit video frames ahead to avoid buffer underflow at the client. This requirement can be expressed as:

$$F_{under}(k) = \sum_{i=0}^{k} f_i$$

here $F_{under}(k)$ indicates the amount of data consumed at the client when it is displaying frame k $(k = 0, 1, \ldots, n-1)$. Similarly, the client should not receive more data than its buffer capacity. This requirement is represented as:

$$F_{over}(k) = b + \sum_{i=0}^{k} f_i$$

where b is client buffer size. Consequently, any valid transmission plan should stay within the river outlined by these vertically equidistant functions. That is,

$$F_{under}(k) \leq \sum_{i=0}^{k} c_i \leq F_{over}(k)$$

where c_i is the transmission rate during frame slot i of the smoothed video stream.

Generating a bandwidth plan is done by finding m consecutive runs which use constant bandwidth $r_j, (j : 1, \ldots, m)$. Within the j^{th} run, the frames are transmitted at the constant rate r_j. The rate change occurs to avoid buffer overflow or buffer underflow. Mathematically, the runs of bandwidth plan must be such that the amount of frame data transfered forms a monotonically increasing, piecewise linear function.

Different bandwidth smoothing algorithms result from choosing the rate changes among the bandwidth runs. Several optimal bandwidth allocation plan-generating algorithms are discussed in reference [35]. These algorithms achieve optimal criteria such as minimum number of bandwidth changes, minimum peak rate requirement, and largest minimum bandwidth requirement. We discuss below an online smoothing algorithm, a proactive buffering mechanism and two algorithms which combine smoothing and rate shaping techniques.

Online Smoothing

Live video applications such as broadcasting of a lecture and news are delay tolerant, in the sense that the user does not mind if the video is delayed in the order of few seconds (or even a minute). For these live video applications, smoothing techniques (passive methods) can significantly reduce the resource variability.

Several window-based online smoothing algorithms (passive methods) are discussed in reference [98]. In the first approach, a hop-by-hop window smoothing

algorithm is used. Here, the server stores up to a window of W frames. The smoothing algorithm is performed over this window of frames taking into consideration the server and client buffer constraints. After the transmission of W frames, the smoothing algorithm is performed for the next set of W frames. This algorithm does not handle an inter-mixture of large I frames among P and B frames, since only in the first window the transmission of I frame is amortized. The consecutive windows can be aligned with an I frame at the end of each window.

While in the hop-by-hop algorithm, the server cannot prefetch data across window boundaries, the sliding-window method SLWIN(α) uses a sliding window of size W for smoothing. The smoothing algorithm is repetitively performed for every α frames time units over the next W frames. The sliding-window performs better but is more complex, since the smoothing algorithm is executed more times than in the hop-by-hop method.

Proactive Buffering

Another passive method, rate constrained bandwidth smoothing for stored video, is given in reference [37]. Here, the rate is assumed to be constrained to a given value (for example, the minimum cell rate (MCR) in the ABR service). The algorithm proactively manages buffers and bandwidth. This method uses the rate constrained bandwidth smoothing algorithm (RCBS) [33], which minimizes the buffer utilization for a given rate constraint. In RCBS, the movie frames are examined in reverse order from the end of the movie. The large frames that require more than the constrained rate are prefetched. These prefetches fill the gaps of earlier smaller frames.

The proactive method identifies feasible regions of the movie. A feasible range is where the average rate requirement of the range is less than the constrained rate. The

movie frames are examined in the reverse order and feasible regions are identified. The algorithm keeps track of the expected buffer occupancy at the client side and finds the maximal feasible regions. When the rate constraint is violated, frames are dropped. The dropped frames are placed apart to avoid consecutive frame drops. The proactive method results in maximizing the minimum frame rate for a given rate constraint.

The low-level network feedback can be used in this method as follows: assume that the network can guarantee the constrained rate and inform the source through feedback if any extra bandwidth is available. This extra bandwidth and current buffer occupancy level can be used to decide if additional frames can be sent.

Bridging Bandwidth Smoothing and Adaptation Techniques

Passive adaptation techniques like bandwidth smoothing algorithms take advantage of a-priori information to reduce the burden on the network, however, they do not actively alter the video stream to make them network sensitive. The reactive techniques usually do not take advantage of the a-priori information and hence, may not provide the best possible quality video over best effort networks. Some recent work has focussed on augmenting reactive techniques to take advantage of this a-priori knowledge.

A priority-based technique (reactive method) is used to deliver prerecorded compressed video over best-effort networks in reference [38]. Multiple level priority queues are used, in addition to a window at each level to help smooth the video frame rate while allowing it to change according to changing network conditions. The scheme uses frame dropping (adaptation technique) and a-priori knowledge of frame sizes.

The scheme tries to deliver the frame of highest priority level (base layer) before delivering the frames of enhancement layers. Adaptation is accomplished by dropping frame at the head of the queue if enough resources are not available.

Another algorithm which combines the smoothing and rate changing technique (frame dropping) is discussed in reference [131]. An efficient algorithm to find the optimal frame discards for transporting stored video over best-effort networks is given. The algorithm uses the selective frame discarding technique. The problem of finding the minimum number of frame discards for a sequence of frames is posed as an optimization problem. A dynamic programming based algorithm and several simpler heuristic algorithms are given to solve this problem.

8.4 Example Adaptive Applications

In this section, we present two commercial adaptive applications: (1) the Real Networks suite, and (2) Vosaic: video mosaic. These commercial applications are currently available and incorporate some of the adaptation techniques discussed in the previous section. They also incorporate additional optimization techniques. There are several other adaptive multimedia applications and architectures developed by academia such as Berkeley's vic [80], video conferencing system for the Internet (IVS) [16] developed by INRIA in France, Berkeley's continuous media tool kit (CMT) [92], OGI's adaptive MPEG streaming player [121], and MIT's View Station [114]. Most of these applications have contributed to the research results of adaptation methods discussed in earlier sections.

8.4.1 Real Network Solutions

The Real Networks company provides commercial player (free) and server software for streaming applications. Their products include a number of features such as scalability (can support 500 to 1000 simultaneous streams using IP multicast), bandwidth negotiation, dynamic connection management, sophisticated error control, and buffered play. In this section, we review some of the streaming techniques used in Real Networks products for adaptively transmitting multimedia streams over the Internet.

RealVideo: Adaptive Techniques

RealVideo [85] uses the RTSP streaming protocol and can run over both UDP and TCP protocols. RealVideo uses a robust version of UDP to reduce the impact of packet losses. It uses damage-resistant coding to minimize effects of packet loss in video. It also uses FEC-based methods when frame rates are low. The RealVideo supports two encoders: realvideo standard and realvideo fractal. The realvideo standard encoding can support a range of encoding from 10 Kbps to 500 Kbps. This encoder is specifically optimized to work over 28.8 Kbps and 56 Kbps modem lines.

SureStream: Multiple Stream Encoding

One approach to counter the changing network conditions at the server is to reduce the amount of data by dropping frames (stream-thinning). A limitation of this approach is that the resulting video is not of the same quality as the one optimized to the lower rate. The SureStream mechanism [86] overcomes this limitation by using two methods. First, it supports multiple streams encoded at different rates to be stored in a single file. Second, it provides a mechanism for servers and clients to

detect changes in bandwidth and choose an appropriate stream. Changes in the bandwidth are detected by measurements of received frame rate.

SureStream uses the Adaptive Stream Management (ASM) functionalities available in the RealSystem API. ASM provides rules to describe the data streams. These rules provide facilities such as marking priorities and indicating average bandwidth for a group of frames. This information is used by the server for achieving adaptability. For example, the server might drop lower priority frames when the available rate decreases. A condition in the rule can specify different client capabilities. For example, it can indicate that the client will be able to receive at 5 to 15 Kbps and can tolerate a packet loss of 2.5 percent. If the network conditions change, the clients can subscribe to another appropriate rule.

The techniques used in RealVideo and SureStream can benefit from low-level network feedback. For example, instead of detecting bandwidth changes through measurements, the server can use the available bandwidth information from the lower network layer to choose the appropriate stream to transmit.

8.4.2 Vosaic: Video Mosaic

The design and implementation of *video mosaic* (*Vosaic*) is given in reference [22]. The HTTP (hypertext transfer protocol) supports only reliable transmission and does not support streaming media. Vosaic uses the VDP real-time protocol that can be used to support streaming video in a WWW (world wide web) browser. Vosaic is built upon the NCSA Mosaic WWW browser. The URL (uniform resource locator) links of Vosaic allows specification of media types such as MPEG audio and MPEG video.

VDP uses two connections: an unreliable one for streaming data and a reliable one for control. In the control channel, the client application can issue VCR-like instructions such as play, stop, fast forward, and rewind. An adaptation algorithm is used to adjust the rate of the stream according to network conditions. The clients indicate two metrics: frame drop rate and packet drop rate as measured at the client, to the server as feedback using the control channel. The server initially transmits frames at the recorded rate and adjusts the frame rate based upon the feedback received from the client side. Experimental results show that the frame rate improves considerably when the VDP protocol and the adaptation algorithm are used (e.g., frame rate improved to 9 frames/sec from 0.2 frames/sec).

Vosaic can definitely benefit from low-level network feedback. Currently, the network condition is detected by measurements of the received frame rate at the client and sent to the server. Instead, the server can use network feedback such as available rate to dynamically adjust its frame rate.

8.5 Operating System Support for Adaptive Multimedia

Conventional operating systems are not designed for multimedia applications. For example, playback applications need to access CPU resources periodically during playout. This entails that the operating system provide ways for multimedia applications to access resources. To develop adaptive multimedia applications, there is need for an operating system capability that can provide information about available resources. In this section, we discuss some techniques that are used in operating systems to support adaptive multimedia streaming.

8.5.1 Integrated CPU and Network-I/O QoS Management

Multimedia applications use multiple resources and these resources such as CPU availability and bandwidth change dynamically. An integrated QoS management system to manage CPU, network and I/O resources is proposed in reference [75]. This cooperative model enables multimedia end-systems and OS to cooperate dynamically for adaptively sharing end-system resources. The thesis of this work is that end-system resources should be allocated and managed adaptively. The proposed OS architecture called AQUA (Adaptive Quality of service Architecture) aims to achieve this objective.

In AQUA, when an application starts, it specifies a partial QoS (for example, a video application can specify frame rate and may not specify bandwidth requirement). The OS allocates initial resources such as CPU time based on this QoS specification. As the application executes, the OS and the application cooperate to estimate the resource requirements and QoS received. Resource changes are detected by the measuring QoS. Then, the OS and the application renegotiate and adapt to provide predictable QoS with current resource constraints. To enable these functionalities, the AQUA framework includes a QoS manager, QoS negotiation library, and usage-estimation library.

The application specifies an adaptation function when the connection is set up. The QoS manager calls this function when it detects changes in QoS. Using this methodology a CPU-resource manager and network-I/O manager have been implemented in AQUA. A composite QoS manager uses the services of both CPU and network-I/O managers. This facilitates an integrated way to manage resources in AQUA.

The AQUA framework can use low-level network feedback to detect current availability of network resources. The QoS measuring function can be enhanced by using the network layer feedback.

8.5.2 Adaptive Rate-Controlled Scheduling

Multimedia applications need to periodically access resources such as the CPU. The operating system needs to schedule multimedia applications appropriately to support such needs. The CPU requirement of a multimedia application might dynamically change due to the frame rate change caused by scene changes or network conditions. A framework called Adaptive Rate-controlled (ARC) scheduling is proposed to solve this problem in reference [127]. It consists of a rate-controlled online CPU scheduler, an admission control interface, a monitor, and a rate adaptation interface.

ARC operates in a operating system which supports threads (Solaris 2.3). The threads can be of three types: RT (real-time), SYS (system) or TS (timesharing). RT threads have the highest priority in accessing the CPU. The online CPU scheduler schedules the threads belonging to these classes based on their priorities.

Adaptive rate-controlled scheduling is achieved as follows: multimedia threads register with the monitor thread during connection setup. The monitor thread is executed periodically (every 2 seconds). It estimates for each registered thread whether the CPU usage *lags* (how fast the thread is running ahead of its rate) or *lax* (measures how much of the CPU is unused). This estimation is given as feedback to the multimedia application that increases or decreases its CPU access rate accordingly.

This method can benefit by low-level network feedback. For example, the multimedia application can use the available bandwidth indicated in network layer feedback and change its encoding rate and also change its CPU access rate.

8.6 Related Work

In this section, we present a summary of some related work. Most of these works are related to supporting multimedia, though not directly dealing with the problem of adapting multimedia streaming to changing network conditions. When appropriate, we identify if the work could be used for achieving adaptation of multimedia streaming application.

- **MPEG-2 Error Resilience Experiments:** A study of the error resilience of MPEG-2 encoded video streams is given in [83]. Here, the error resilience MPEG-2 system layer is studied whereas the previous studies only concentrate on video layer. In the MPEG-2 system layer, several program streams might be combined to form a transport stream. The transport stream is multiplexed over the network, in this case an ATM network. The study shows that packet loss rates almost doubles when the impact of the system layer is included in some error rates. This study will be useful in designing and studying adaptive error control scheme for streaming multimedia transmitting MPEG-2 encoded video.

- **Fast Buffer Fill Up Scheme:** A fast buffer fill up scheme for video-on-demand applications running over ATM ABR service is presented in reference [133]. The MCR to be chosen for the application is based on a estimated value that depends on the GoP (group of pictures) used in the MPEG-2 encoding. To obtain a fast

buffer fill-up during mode changes, such has fast forward and rewind, a high PCR (peak cell rate) is used. The PCR/MCR value is found from the expected value of the ACR (allowed cell rate).

- **Indeo Video product, Intel:** The Intel's Indeo video product [64] provides support for progressively downloading video clips. The video file is stored in a hierarchical manner. The video is retrieved at lower frame rate using lower resolution quickly. If the user decides to continue, the frame rate and quality is increased as the download progresses. This supports users who have lower bandwidth to retrieve the video and download it based on what they see in the initial part of the video clip. The file encoding format provides the flexibility of specifying which frames of the video are key frames. The video producers can specify the scene-change frames as key frames.

- **Adaptive Transport Protocol for Multimedia:** A new transport protocol HPF, that supports multiple interleaved reliable and unreliable streams is presented in reference [29]. Currently, the Internet does not support heterogeneous flows. Multimedia applications have to get the different media as different streams and then synchronize them. The HPF protocol supports interleaved flows which is required by multimedia applications. The congestion control and reliability mechanism (which are integrated in TCP) of the transport are decoupled. This allows support for interleaving of reliable and unreliable streams. Priorities are used within sub-streams as indicated by the application-defined hints. This enables the scheduler to drop low priority packets in the event of congestion. An adaptive multimedia application can use the HPF protocol.

- **Resilient Multicast:** IP multicast is a popular delivery mechanism of streaming media of conference applications. Contrary to the belief that continuous media cannot do recovery of lost packets, one can show that there is a tradeoff between desired playback quality and degree of interactivity [126]. A new model of multicast called *resilient multicast* is proposed. In this scheme each client can decide its own tradeoff between reliability and real-time requirements. A resilient multicast protocol STORM (Structure-Oriented Resilient Multicast) is designed, in which groups self-organize themselves into distribution structures and use the structure information to recover lost packets from adjacent nodes. The distribution structure is dynamic and a lightweight adaptation algorithm is used to adapt to changing networking conditions. This scheme can be used to design and implement adaptive multicast streaming applications.

- **Scalable and Adaptable Video-on-Demand:** The design and implementation of a scalable and adaptable video-on-demand server is discussed in [12]. In this system, the movies are striped across multiple disk drives to boost I/O, reduce startup latency and prevent hot spots. Variable sized striping is used since MPEG encoding has variable-sized frames. The video stream is preprocessed and a RTP-like header which contains necessary information for the client is added. RTSP is used in the connection setup phase and RSVP is planned to be used for reserving resources along the path. An initial testbed to evaluate the system has been built and experiments demonstrate that the system reduces startup latency, buffering requirements and maximizes the number of concurrent users.

- **Efficient User-Space Protocols with QoS Guarantees:** Multimedia applications running on traditional OSs have limitations due to overhead of data movement and inflexible CPU scheduling. A discussion of protocol implementation in user-space using real-time up-calls (RTU) which can provide QoS guarantees is given in reference [55] . The RTU mechanism has minimal overhead for concurrency control and context switching compared to thread-based methods. Efficient data movement techniques such as batching of I/O operations to reduce context switches, direct movement of data from applications to network adapter, and header-splitting at the receiver to maintain page alignment are augmented with the RTU mechanism. Experiments on the RTU implementation show that QoS can be guaranteed even in the presence of other competing best-effort applications. This framework is amenable to support adaptive multimedia applications since RTU protocols are implemented in the user-space.

- **Rate-Adaptive Packet Video:** A framework for sharing available bandwidth to transport rate-adaptive video is presented in reference [59]. The framework uses an ABR-like flow control and switch algorithm to give feedback to the rate-adaptive sources about the currently available bandwidth. The switch scheme used decouples the rate used in the flow control and the actual rate of the source. The source has the flexibility to adjust its rate only at certain time intervals which is controlled by a parameter. The work also discusses how to renegotiate the MCR parameter and use weight functions to reflect the changing requirements of video applications.

- **Video Staging using Proxies:** In LAN environments large bandwidth is available due to high speed technologies (Gigabit Ethernet, ATM), but bandwidth is a scarce resource in a WAN environment. A proxy server can be used in a LAN to overcome the resource restrictions when retrieving multimedia streams over a WAN [122]. Using a proxy server reduces the backbone bandwidth requirement. The paper also discusses how smoothing techniques can be used at the proxy server when clients have playout buffers.

- **Prefix Proxy Caching:** In this work, the authors propose a prefix caching technique to mitigate effects of resource variability and startup latency [111]. In this technique a proxy server stores the initial frames of a long movie clip. When the client requests for the stream in future, the proxy server starts sending the initial frames. In addition, the proxy server also contacts the video server and requests the transmission of later frames. The proxy also performs *workahead smoothing* into the client's playout buffer.

8.7 Chapter Summary

Only best-effort service is widely supported in the current Internet. The Internet is also a heterogeneous network and is expected to remain heterogeneous. Lots of efforts are being made by standardization bodies (IETF, ATM Forum, ITU-T) to support QoS in the Internet. Even with QoS support from the network the multimedia applications need to be adaptive.

Adaptation to changing network conditions can be achieved at several layers of the network protocol stack. In this chapter, we surveyed several techniques for achieving adaptation at the application layer. A summary is as follows:

- Compression methods: An overview of DCT and wavelet compression methods was given. Aspects of these methods that can be used for adaptation were discussed.

- Application Streaming: The application methods are broadly classified into *reactive* and *passive* methods. Reactive methods are those where the application changes its behavior based on the environment. Passive methods aim to reduce the network burden by optimization techniques.

- Rate Shaping: Video encoder parameters (e.g., frame rate, quantization level) are changed to meet the changing network conditions.

- Error Control: These techniques use FEC-based methods to provide protection against changing error conditions.

- Adaptive Synchronization: An adaptive synchronization method for solving intermedia and intramedia synchronization problem was discussed.

- Smoothing: Smoothing techniques attempt to reduce the variability in the resource requirements of multimedia applications.

- Example Adaptive Applications: Techniques used for adaptation in real-world multimedia applications such as Real Networks' products (RealPlayer and RealServer) and Vosaic were presented.

- Operating System Support: Multimedia applications need periodic access of CPU and other system resources. Operating systems need to support such needs. Techniques to achieve this include real-time upcall, adaptive scheduling, and CPU management.

For each of these techniques, we discussed whether it could benefit from low-level network feedbacks. When appropriate, we discussed how the low-level feedback can be used for enhancing the adaptation technique.

CHAPTER 9

SOFTWARE SWITCH

In this chapter we discuss the software based testbed that we developed for doing experimental analysis of the schemes developed in this thesis.

9.1 Motivation

Various metrics such as packet loss, delay and throughput achieved can be used to measure the QoS, but improvements of QoS delivered to multimedia (especially video) can be best evaluated only through perceptual quality, i.e., by actually seeing the video frames. Though one could transport actual video frames through simulated components, it is a computationally intensive task. For best performance ideally one should implement the methods in a hardware switch. But it is not cost effective and it is difficult to acquire and modify code of a hardware switch since they are proprietary. Therefore, we decided to build a software based switch in which we can implement our methods and perform experiments to show the improvements in QoS. We chose to build the switch using PCs running Linux operating system since there was already considerable work done in this environment. Linux/ATM project and Kansas State University's ATM switch project were used as a starting point of our software switch project.

9.2 ATM on Linux Project

Linus Torvalds, a Finnish student in 1991 wrote the Linux operating system [63], modeled after the UNIX operating system and gave it away for free under the GNU general public license. Since then, due to the contribution of programmers from all over the world, it has gathered momentum and now is even being considered as rival of Microsoft's Windows operating system.

The ATM on Linux project was started by Werner Almesberger in 1995 [5]. This project currently supports the following features:

- Raw ATM connections, both permanent virtual connections (PVC) and switched virtual connections (SVC)
- IP over ATM
- LAN emulation (LANE)
- Multiprotocol over ATM (MPOA)
- CBR and UBR service categories
- AAL5 (ATM Adaptation Layer 5) and AAL0 (raw ATM cell) connections.
- ATM API (application program interface) using standard UNIX socket interface.
- Utilities for performing diagnostics and simple network management tasks.

9.3 Virtual ATM Switch

Ricardo Sanchez developed a virtual ATM switch using the ATM on Linux distribution as a part of the Rapidly Deployable Radio Network project at the Kansas State university [102]. The first version was released in March 1998 and the latest version 6.0 has been released in December 1999. In our project we have used the version 0.4 of the virtual switch. This version uses version 2.1.90 of Linux kernel and version 0.36 of the ATM on Linux project.

The virtual switch software project has the following features:

- Virtual switch (implemented in `switch.c`) which performs the switching operation in software.
- Virtual ATM network interface which enables a standard Ethernet card to act as a virtual ATM network interface card.
- A binary distribution of Bellcore's Q.Port software supports signaling for the virtual switch.
- Utilities to maintain the virtual ATM device and virtual switch.

9.3.1 Running the Virtual Switch

Now we briefly discuss how to actually run the virtual switch. We assume a simple configuration (Figure 9.1) in which three PCs are used and each pair is connected to each other using back-to-back cable[2]. Host A, acts as the source and host B which has two ATM NICs, runs the software switch and host C acts as the destination.

[2]The cable for this has to be specially made by connecting wires 1-2, 2-1, 7-8, 8-7 of the standard category 5 cable

Assume that the ATM NIC is numbered '0' in hosts A and C. Switch machine (host B) has two ATM NICs numbered '0' and '1'. The steps to run the virtual switch are the following:

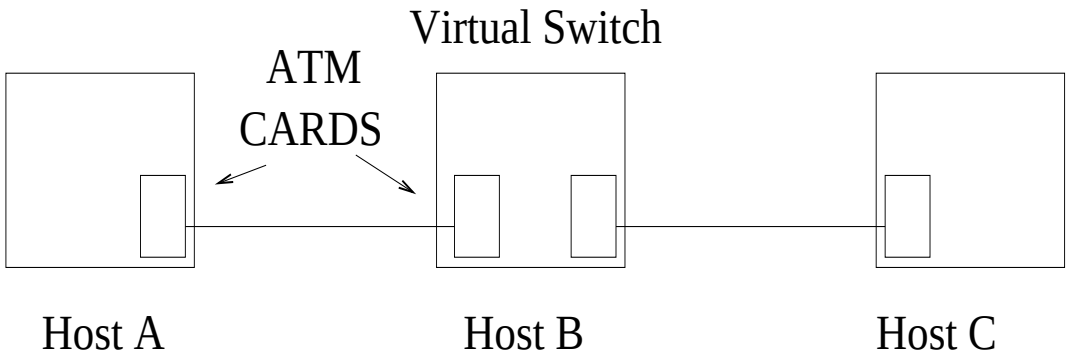


Figure 9.1: Back-to-Back switch configuration.

1. In host B create two ports corresponding to the two NICs by running the 'vsw_ctl -c 0' and 'vsw_ctl -c 1' commands. 'vsw_ctl' is the command to manage the virtual switch.

2. A VC is created by running the 'vsw_ctl -s 0 40 1 50' command. Now 0/40 (VPI/VCI) (Virtual path identifier/Virtual circuit identifier/) in host A is connected to 0/40 at host B. Host B will perform the switching of cells received in 0/40 to 1/50. This in turn is connected to 0/50 at host C.

3. Run 'aread 0.0.40' in host A. 'aread' commands reads continuously the cells received at the specified VPI/VCI of the given card.

4. Run 'awrite 0.0.50 <chars to be sent in a cell>' in host C. This command transmits an ATM cell consisting of the words give as argument to 'awrite' via 0/50 VPI/VCI pair.

5. Switch machine B, continuously switches cells received in 1/50 to 0/40 and vice versa. Host A, should receive the packet sent and display the characters (or number corresponding to the ASCII values of the characters) on its monitor.

We developed utilities that send cells continuously and sends a specified file to be used in our experimental analysis.

9.3.2 Virtual Switch Modules

The virtual switch is implemented in the `linux/net/atm/switch.c` file of the Linux kernel distribution. We give a brief overview of the key modules of this file. This is necessary to understand our extensions that implement the rate allocation scheme and queue control functions.

atm_push_raw: This function is called whenever the switch machine receives a cell in one of its NICs. The following are done in this module: 1) It finds the outgoing port entry which has the outgoing VCI information, 2) It changes the VCI field in the cell header, 3) It increments counters, such as number of cells counter, and 4) It calls the `atmsw_rx` function which queues the received cell.

atmsw_rx: Checks if the backlog (queue) size is less the maximum allowable queue size. It drops the cell if the queue is full. Otherwise, it inserts the received cell at the end of the queue and marks the bottom half[3] interrupt handler.

[3]In Linux, some slower interrupt events are handled using a two part handler. The top half quickly performs essential tasks and marks a bit to indicate that bottom half has to be called. The kernel periodically checks the interrupt registers and calls bottom half handler whose bit is set

atmsw_bh: The bottom half handler retrieves the cells from the backlog queue and calls the send routine of the appropriate ATM device driver to transmit the cell. This step is done repeatedly till the backlog queue becomes empty.

9.4 Implementation of ABR Service

The ABR service was implemented by developing modules that perform the operation of the source and destination end systems. Two files (`abrsource.c` and `abrdest.c`) implement the source and destination rules according to the ATM Forum traffic management specification version 4.0. Ideally, the ABR service's feedback control loop should be implemented in the NIC card. But currently there is no ABR card that is supported by the ATM on Linux project. We did obtain Olicom's RapidFire ABR card, but it works only in the Windows NT environment. Due to the difference in the signaling mechanisms we were not able to interface between a Windows NT as the end system and PC running Linux as the switch. Therefore we chose to implement ABR service over the AAL0, which sends raw ATM cells.

abrsource.c: Implements the source end-system rules. It sends the data cells at the allowed cell rate (ACR). It keeps track of number of data cells sent and inserts a 'forward RM' cell for every $Nrm - 1$ (default value for Nrm is 32) number of data cells. It asynchronously reads the socket interface in the reverse direction for receiving 'backward RM' cells. When a BRM is received it adjusts its ACR value based on the source rules and the explicit rate field value of the RM cell.

abrdest.c: The destination receives the data cells and passes it up to the application layer. When it receives an RM cell it turns it around based on the destination end-system rules.

9.5 Implementation of Switch Schemes

The modules of the `switch.c` were modified extensively to implement the ABR rate allocation schemes and queue control functions. We now discuss the main modifications to existing routines and new modules written.

atmsw_push_raw: Modified this to call the 'atmsw_rx' function with additional parameters.

atmsw_rx: It first updates counters such as the total number of cells received in the current averaging interval. If the cell is an RM cell, it calculates the explicit rate by calling 'calculate_ER' function. The 'calculate_ER' evaluates the ER using the monitored values of load, $VCShare$, and $ExcessFairShare$ and returns it. If cell is a data cell, this routine checks if the ABR queue is not full and inserts the cell at the end of the ABR queue.

atmsw_bh: Checks if VBR, ABR, or UBR queue is empty in that order. It retrieves the cell from the first non-empty queue and schedules it to be sent via the NIC corresponding to the output port.

eoi_calc: This function implements the 'End of Interval' part of the rate allocation schemes. It uses the measured value of input rate and estimated value of output rate to calculate the overload. Then the appropriate fairshare and activity level values are calculated depending on which version of the algorithm is being tested.

9.6 Experimental Analysis

In the first experiment we have measured the effective bandwidth available when two machines are connected back to back with a single link. This value will be used later in the testing of the switch schemes implemented. We found that the receiver end system works slower than the sending end system. Therefore, we introduced a delay (as empty for loop) at the receiver end. Table 9.1 shows the results obtained in this experiment. The 'starwars.mpg' of size 1.75 MBytes (corresponds to 36480 cells), which runs for about 2 minutes at 24 frames/sec, was used in this experiment. All cells were received without loss when the receiver did not write the received data into a file. If the receiver tried to store the cells in a file then there were considerable losses even for small value of delay. This is due to the slow disk I/O. This table will be used to estimate the delay value that should be used to achieve the desired bandwidth indicated by the ACR at the source. We have verified separately that each of the source, destination and switch components work independently. Next, we integrated components and tested them to make sure that they work together.

The following experiments were performed:

Video over UBR: In this simple experiment the 'starwars.mpg' file was transmitted over the UBR service. Since there was no feedback control and the receiver was operating at a slower speed, there were packet losses which manifested as glitches and frame freezes of the video clip.

Video over ABR: Here, the same video clip was transmitted over ABR service. Due to feedback control the source was throttled which avoided packet losses.

Delay	SentTime	Cells/s	Mbps	Rcv Time	Cells/s	Mbps
100	2.24	16261.14	6.24	27.83	1310.60	0.50
1000	3.36	10852.16	4.16	4.30	8477.81	3.25
10000	8.45	4315.41	1.65	9.95	3665.56	1.40
20000	15.51	2351.35	0.90	24.16	1509.43	0.57
30000	22.10	1650.20	0.63	30.83	1183.18	0.45
40000	28.47	1281.29	0.49	37.14	982.02	0.37
50000	35.32	1032.79	0.39	47.30	771.12	0.29
60000	41.85	871.54	0.33	52.21	698.70	0.26
70000	49.06	743.54	0.28	57.47	634.74	0.24
80000	55.40	658.48	0.25	65.17	559.75	0.21
100000	63.37	575.61	0.22	65.21	559.38	0.21

Table 9.1: Measurement of sender and receiver speeds with varying delay at sender.

Initially, the frame rate was slightly less and reached the required value as the feedback control loop was established.

Convergence of ERICA+: The bottleneck link bandwidth was set to 1.5 Mbps. A constant queue control function with target utilization of 90% was used. Two infinite ABR sources share the bottleneck link. First, only the first source was activated and it soon gets the entire available bandwidth of 1.35 Mbps (0.9×1.5). When the second source was activated, the algorithm quickly converges and both receive fairshare allocation of $1.35/2 = 0.675$ Mbps. This experiment demonstrates the convergence of the ERICA+ algorithm.

9.7 Chapter Summary

In this chapter we discussed the implementation of the software based ATM switch. The testbed consists of source, destination and switch modules. Preliminary experiments have been conducted to measure the throughput of an ATM link and test the convergence of the switch algorithms. These experiments show that the implemented algorithms converge and that video over ABR can get acceptable quality. Further experiments will be conducted to test the performance and comparison of different rate allocation schemes.

CHAPTER 10

SUMMARY AND OPEN ISSUES

The Internet is currently growing at an exponential rate. The next generation Internet will possess an order of magnitude more bandwidth and computing power will also increase tremendously. These will make the Internet and computing ubiquitous. Towards this goal, advanced research is being carried out in varying fields such as wearable computers, visualization, medical imaging, multicasting, scalable networking, active networks, wireless networks, security, and connecting appliances to the network. The impact of such advanced research will lead to a scalable, adaptable, and efficient Internet. One of the main components of such an network is good traffic management. We identified that multimedia is going to become a significant portion of the Internet traffic and for efficiency it has to be adaptable. In this dissertation, we have explored the traffic management methods to enhance the quality of service of adaptable multimedia applications.

We chose to explore the problem of enhancing QoS in ABR service since we believe that future multimedia applications will be adaptive. We proposed a new generalized fairness definition that can be used to bias the excess available bandwidth towards the multimedia connections. The ERICA+ switch scheme was modified to provide MCR guarantees and provide generalized fairness. Then we developed three new

switch schemes based on the overload factor. All the new algorithms provide MCR guarantees and converge to generalized fair allocations. These algorithms were compared based on their convergence time and algorithmic complexity. We argued that the varying portion of end-to-end delay is due to the queuing delay, and by controlling the queuing delay we can reduce the end-to-end delay. For doing this, we developed several new queue control functions and compared their behaviors using simulation and analysis. We also studied the state of the art techniques available at the application layer to adapt to network changes. This demonstrates that multimedia applications can be adaptive and can hence be transported over the ABR service. Finally, to perform experimental study of the proposed schemes we developed a software switch. This comprised of a software-based switch and end-system components that act as multimedia ABR source and destinations.

Next in this chapter, we give a summary of the results obtained and then discuss open issues and proposed approaches to tackle them, which will provide direction to our future work.

10.1 Summary of Key Results

The dissertation has tackled the problem of developing methods to enhance QoS of multimedia in ABR service. Several groups of problems such as those dealing with providing throughput continuity, fair allocation, controlling delay were explored. To summarize, in this dissertation we have:

- Given a new generalized fairness definition that inherently guarantees MCR allocation. This new definition has been included in the current version of ATM Forum traffic management specifications.

- Showed how to modify an existing switch algorithm to provide MCR guarantee and achieve generalized fair allocation.
- Designed new rate allocation schemes based on an overload factor.
- Developed guidelines for designing queue control functions that can be used by a rate allocation scheme to control queues.
- Following these guidelines, designed and analyzed four different queue control functions: step, linear, hyperbolic and inverse hyperbolic, and found that inverse hyperbolic function performs the best in all scenarios.
- Performed a survey on the state of the art application layer techniques for adaptively streaming multimedia traffic. We also identified how low-level network feedback information can be used in these methods when appropriate.
- Developed a flexible software-based testbed to perform experimental analysis of developed methods. This testbed used PCs running the Linux operating system and used off-the-shelf ATM network interface cards. Preliminary results in the testbed show that the methods proposed in this dissertation are indeed enhancing the QoS available to multimedia applications.

10.2 Open Issues and Future Work

In this dissertation, we have explored methods that can be used for enhancing QoS of multimedia applications. We studied these methods in the context of ATM ABR service. This research process has led to several open issues that need to be resolved both in the ATM networks and the Internet. In the subsequent subsections we outline these issues and discuss some approaches for tackling each of them.

10.2.1 Enhancing Throughput Continuity

How to enhance throughput continuity? In this dissertation we argued that throughput continuity is be provided by using the MCR guarantees of ABR service. There is need for exploring methods to further enhance throughput guarantees to support interactive multimedia applications.

One method to ensure throughput continuity is to use a separate queue for those ABR connections which transport multimedia, and to service this queue at a priority higher than the other ABR traffic. A part of the bandwidth may also be reserved exclusively for multimedia ABR connections. Using a separate queue also isolates the multimedia application traffic from other of ABR traffic, so the delay experienced is reduced.

10.2.2 Minimize Impact of Loss

How to minimize the impact of loss? End systems can code video and switches can adjust drop policies such that the impact of loss on the user's perceived quality is minimized.

The loss rate in ABR service is small, since it uses feedback to throttle sources in the event of network congestion. Nonetheless, losses can arise due to buffer overflow at the switches, or due to packets that do not arrive before the fixed delay bound. Packet loss affects the quality of service perceived by the user. So, it is necessary to reduce the impact of loss and variation in quality of service.

Two approaches to reduce the impact of losses are end system policies (at the source) and network level policies.

- *End-system Policies:* One end-system policy is to use hierarchical encoding (multi-layered encoding) of multimedia traffic [120]. The MPEG standard supports this encoding. The traffic can have two priorities, high and low. The high priority traffic can be sent without loss by reserving a higher MCR value, and the low priority traffic can be sent over a different VC with a lower MCR value.

- *Network level Policies:* Drop policies like simple drop tail (drop the packet when queue is full) or other intelligent ones like fair buffer allocation (allocating buffers proportionally) can be used to drop entire frames. For drop policies to work, the switch has to identify packet (frame) boundaries. If AAL5 is used, switches can easily recognize the frame boundaries by monitoring the EOM (end of message) bit in the cell headers. For other AALs, other methods may need to be used, for example, by sending additional RM cell at end of frame to indicate frame boundary.

10.2.3 Minimize Rate Variations

How to minimize the rate variations? Minimizing rate variations leads to reduction in variations in the quality of service. Users want a consistent quality of service in a multimedia application such as video. Hence, it is necessary to reduce the rate variations to provide low variation in quality of service.

As previously mentioned, ABR sources send a RM cell after every $Nrm - 1$ (usually $Nrm = 32$) data cells. Switch algorithms calculate a feedback value for every averaging interval (typically 5 ms). The sources adjust their rate when they receive these backward RM (BRM) cells.

One way of reducing the rate changes is to send RM cells less frequently, i.e., Nrm should be large (say 1024) instead of 32. We can conduct experiments to find an appropriate Nrm value. Sending RM cells at end of each frame is one possible option. Another method to reduce variation is to increase the length of the averaging interval.

10.2.4 Adapting Video Traffic

How to design adaptive multimedia video applications for transmission over ABR service? MPEG-2 is a currently a popular standard for compressing video traffic. Work is in progress to standardize MPEG-4 and MPEG-7. MPEG standards provide different parameters, such as the quantization value, which can be dynamically modified based on feedback to vary video source rate. Hence, techniques that specify how to modify these parameters (based on the feedback given by ABR) have to be studied.

One important set of multimedia applications is comprised of compressed video, video-on-demand, and movies-on-demand. Compressed video is typically bursty in nature. A challenging problem when ABR is used for transporting multimedia video traffic is how to control the video source rate when feedback is received from the network. The source rate can be adjusted if the video source is adaptable.

We should concentrate on using MPEG video sources, since it is the popular standard for transporting compressed video. The MPEG standard supports different kinds of scalability:

1. **Spatial Scalability:** Support of different picture resolutions (on the X and Y axes) in a single video stream.

2. **Temporal Scalability:** Ability to handle different picture rates in a single video stream.

3. **SNR Scalability:** Signal to Noise Ratio (SNR) scalability allows at least two different video qualities. The base layer carries vital information and the enhancement layer carries additional information.

We plan to use the above scalability features of MPEG to control the rate of MPEG sources when feedback is received. MPEG encoders also have a rate controller module which can adjust the quantization to achieve the desired rate. Various actions can be taken to tackle different degrees of congestion. To handle a slight reduction in rate (light congestion), the quantization level can be changed. Under severe congestion, the rate can be further reduced by dropping frames at the video source.

10.2.5 Adaptive Smoothing Techniques

How to use bandwidth smoothing to support video? It has been shown that bandwidth smoothing techniques can be used to transport prerecorded compressed video [36, 39]. Bandwidth smoothing results in much smoother traffic where rates vary slowly. The main issue is to study how to use these smoothing techniques adaptively to transport compressed video traffic.

Several bandwidth smoothing algorithms have been recently developed for prerecorded compressed video [36, 39]. These algorithms prefetch data and send it at a steady rate so that buffer underflow or buffer overflow does not occur at the destination play-out buffer. Reference [98] gives an online smoothing algorithm for delay tolerant applications such as videocast and news broadcast.

After smoothing the video source its network characteristics change, specifically the minimum rate and peak requirements decrease. We can study the problem of mapping these parameters to the ABR service parameters. Specifically, the problem of using appropriate MCR for transporting the smoothed video sources should be addressed.

In under-loaded conditions, the rate required by smoothed traffic can be satisfied. When congestion occurs, however, the source may not get enough bandwidth. An approach to solve this problem is rate re-negotiation of traffic parameters, such as MCR. If re-negotiation is not possible other rate reductions methods such as dropping frames can be used. We can use the following approach to adaptively drop frames. The sources have a look ahead buffer which holds a small number frames to be sent in the next Δt time interval. We use the following notations in our discussion.

$acr =$ Current allowed cell rate.

$r =$ Required rate to transmit the frames stored in the look ahead buffer.

$n =$ Number of frames in the look ahead buffer. For example, if $\Delta t = 2$ second and the video is being transmitted at 30 frames/sec, then $n = 60$. ahead buffer.

A frame drop rate (d) assuming that all the frames are of average size is calculated.

$$d = n\left(1 - \frac{acr}{r}\right)$$

The source reduces its rate to match the required rate by dropping d frames from the n frames to be sent. The frames to be dropped are evenly spaced among n frames to reduce variation in quality.

10.2.6 Adaptive Multimedia Applications

How to develop adaptive multimedia applications for the Internet? Given that Internet is heterogeneous, and will remain heterogeneous it is necessary that future multimedia be adaptive. But currently multimedia applications do not closely interact with the lower layer that is necessary for any kind of adaptation.

First, we believe that adaptive multimedia applications will need to interact closely with other layers such as the operating system and network protocol stack. For doing this effectively, one has to develop middleware that will act as the interface between these components.

Second, it is necessary to bring some form of feedback mechanism to the Internet. IETF has standardized the explicit congestion notification method to indicate congestion status using binary feedback mechanism. This needs to be further enhanced by adding more information such as either available rate or queue length status in the control mechanism. Recently, the end point congestion management (ECM), working group in IETF has been formed to develop protocol independent congestion control methods [95]. The congestion control will be done for a group of connections using a "congestion manager" module. We can introduce more elaborate feedback mechanism at the congestion manager level rather than at the end-systems which would obviate the necessity of changing the current end-systems.

BIBLIOGRAPHY

[1] ISO/IEC 13818-2. Generic coding of moving pictures and associated audio information. Technical report, MPEG (Moving Pictures Expert Group), International Organization for Standardization, 1994.

[2] B. Aboba. Introduction to Accounting Management. Work in progress, draft-aboba-acct-00.txt, August 1998.

[3] S. P. Abraham and A. Kumar. A Stochastic Approximation approach for a Max-Min Fair Adaptive Rate Control of ABR Sessions with MCRs. In *Proc. of INFOCOM*, April 1998.

[4] Y. Afek, Y. Mansour, and Z. Ostfeld. Phantom: A simple and effective flow control scheme. In *Proceedings of the ACM SIGCOMM*, August 1996.

[5] W. Almesberger. ATM on Linux project. http://lrcwww.epfl.ch/linux-atm/.

[6] J. Arkko. Requirements for Internet-Scale Accounting Management. Work in progress, draft-arkko-acctreq-00.txt, August 1998.

[7] A. Arulambalam, X. Chen, and N. Ansari. Allocating fair rates for available bit rate service in ATM networks. *IEEE Communications Magazine*, 34(11), November 1996.

[8] A. Arulambalam, X. Chen, and N. Ansari. Allocating fair rates for available bit rate service in ATM networks. *IEEE Communications Magazine*, 34(11):92–100, November 1996.

[9] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for Traffic Engineering Over MPLS. Work in progress, draft-ietf-mpls-traffic-eng-00.txt, October 1998.

[10] A. W. Barnhart. *Example Switch Algorithm for TM Spec.* ATM Forum 95-0195, Feburary 1995.

[11] Y. Bernet, J. Binder, S. Blake, M. Carlson, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss. A Framework for Differentiated Services. Work in progress, draft-ietf-diffserv-framework-01.txt, October 1998.

[12] M. Berzsenyi, I. Vajk, and H. Zhang. Design and implementation of a video on-demand system. *Computer Networks and ISDN Systems*, 30:1467–1473, 1998.

[13] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. RFC 2475, December 1998.

[14] J-C. Bolot and T. Turletti. Experience with rate control mechanisms for packet video in the Internet. *Computer Communications Review*, 28(1), 1998.

[15] J.C Bolot and A.V. Garcia. Control mechanisms for packet audio in the internet. In *Proc. of IEEE INFOCOM*, November 1996.

[16] J.C Bolot and T. Turletti. A rate control mechanism for packet video in the internet. In *Proc. of IEEE INFOCOM*, November 1994.

[17] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol. Work in progress, draft-ietf-rap-cops-06.txt, February 1999.

[18] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP). RFC 2205, September 1997. http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2205.txt.

[19] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan. A Framework for Multiprotocol Label Switching. Work in progress, draft-ietf-mpls-framework-02.txt, November 1997.

[20] A. Charny. An algorithm for rate allocation in a cell-switching network with feedback. Master's thesis, Massachusetts Institute of Technology, May 1994.

[21] A. Charny, D. Clark, and R. Jain. Congestion Control with Explicit Rate Indication. In *Proceedings of ICC'95*, page 10 pp, June 1995.

[22] Z. Chen, S. Tan, R. Campbell, and Y. Li. Real Time Video and Audio in the World Wide Web. In *WWW4 (Available at: http://www.vosaic.com/)*, 1995.

[23] P. Cheng, J. Li, and C.-C.J. Kuo. Rate control for an embedded wavelet video coder. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(4), 1997.

[24] D. Chiu and R. Jain. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN*, 17(1), June 1989.

[25] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick. A Framework for QoS-based Routing in the Internet. RFC 2386, August 1998.

[26] David Taubman and Avideh Zakhor. A Common framework for rate and distortion based scaling of highly scalable compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(4), August 1996.

[27] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. In *Proc. of ACM SIGCOMM*, pages 1–12, Sept 1989.

[28] N.G. Duffield, K. K. Ramakrishnan, and A. R. Reibman. SAVE: An algorithm for smoothed adaptive video over explicit rate networks. In *Proc. of IEEE INFOCOM*, April 1998.

[29] D. Dwyer, S. Ha, J. Li, and V. Bharghavan. An adaptive transport protocol for multimedia communication. In *IEEE Conference on Multimedia Computing Systems*, 1998.

[30] A. Eleftheriadis and D. Anastassiou. Optimal Data Portioning of MPEG-2 Coded Video. In *First IEEE Int'l Conf. on Image Processing*, November 1994.

[31] S. Fahmy. *Traffic Management for Point-to-Point and Multipoint Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks.* PhD thesis, The Ohio State University, August 1999.

[32] S. Fahmy, R. Jain, S. Kalyanaraman, R. Goyal, and B. Vandalore. On determining the fair bandwidth share for ABR connections in ATM networks. In *Proceedings of the IEEE International Conference on Communications (ICC) '98*, volume 3, pages 1485–1491, June 1998. http://www.cis.ohio-state.edu/~jain/papers/neff.htm.

[33] W. Feng. Rate-constrained bandwidth smoothing for the delivery of stored video. In *Proc. of SPIE Multimedia Networking and Computing*, pages 316–327, November 1997.

[34] W. Feng. Time constrained bandwidth smoothing for interactive video-on-demand. In *Proc. of ICCC*, pages 291–302, November 1997.

[35] W. Feng, F. Jahanian, and S. Sechrest. An optimal bandwidth allocation strategy for the delivery of compressed prerecorded video. *ACM/Springer-Verlag Multimedia Systems Journal*, 1997.

[36] W. Feng, F. Jahanian, and S. Sechrest. An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video. *ACM/Springer-Verlag Multimedia Systems Journal*, 5(5):297–309, Sept 1997.

[37] W. Feng, B. Krishnaswami, and A. Prabhudev. Proactive buffer management for the delivery of stored video across best-effort networks. In *ACM Multimedia Conference*, September 1998.

[38] W. Feng, M. Liu, B. Krishnaswami, and A. Prabhudev. A priority-based technique for the delivery of stored video across best-effort networks. In *Proc. IS&T/SPIE Multimedia Computing and Networking*, January 1999.

[39] W. Feng and J. Rexford. A comparison of bandwidth smoothing techniques for the transmission of prerecorded compressed video. In *Proc. of IEEE INFOCOM*, pages 58–66, April 1997.

[40] W. Feng and S. Sechrest. Critical bandwidth allocation for delivery of compressed video . *Computer Communications*, 18(10), October 1995.

[41] ATM Forum. ATM Security Framework Version 1.0. Approved specifications, ftp://ftp.atmforum.com/pub/approved-specs/af-tm-sec-0096.000.pdf, Feburary 1998.

[42] The ATM Forum. The ATM forum traffic management specification version 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps, April 1996.

[43] The ATM Forum. ATM User-Network Interface (UNI) Signaling Specification 4.0. ftp://ftp.atmforum.com/pub/approved-specs/af-sig-0061.000.pdf, July 1996.

[44] The ATM Forum. Private network-network interface specification version 1.0. ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000.pdf, March 1996.

[45] The ATM Forum. ATM Trunking Using AAL1 for Narrow Band Services v1.0. Approved specifications, ftp://www.atmforum.com/approved-specs/af-vtoa-0089.000.pdf, July 1997.

[46] The ATM Forum. Circuit Emulation Service Interoperability Specification - 2.0. Approved specifications, ftp://www.atmforum.com/approved-specs/af-vtoa-0078.000.pdf, January 1997.

[47] The ATM Forum. (DBCES) Dynamic Bandwith Utilization in 64 KBPS Time Slot Trunking Over ATM - Using CES. Approved specifications, ftp://www.atmforum.com/approved-specs/af-vtoa-0085.000.pdf, July 1997.

[48] The ATM Forum. Voice and Telephony Over ATM to the Desktop. Approved specifications, ftp://www.atmforum.com/approved-specs/af-vtoa-0083.000.pdf, May 1997.

[49] The ATM Forum. Audio/Visual Multimedia Services: Video on Demand v1.1. Approved specifications, ftp://www.atmforum.com/approved-specs/af-saa-0049.001.pdf, December 1998.

[50] J. E. Fowler and S. C. Ahalt. Adaptive vector quantization of image sequences using generalized threshold replenishment. In *Proc. of 1997 IEEE ICASSP*, pages 3085–3088, April 1997.

[51] J.E. Fowler, K.C. Adkins, S.B. Bibyk, and S.C. Ahalt. Real-Time Video Compression Using Differential Vector Quantization. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(1):14–24, 1995.

[52] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One-Part five. RFC 2045-2049, Nov 1996.

[53] C. Fulton, San-Qi Li, and C. S. Lim. *UT: ABR feedback control with tracking.* Preprint, 1997.

[54] N. Golmie. *Netsim: network simulator.* http://www.hsnt.nist.gov/misc/hsnt/prd_atm-sim.html, 1998.

[55] R. Gopalakrishnan and G. Parulkar. Efficient user-space protocol implementations with qos guarantees using real-time upcalls. *IEEE/ACM Trans. no Networking*, 6(4):374–388, August 1998.

[56] R. Guerin, S. Kamat, A. Orda, T. Przygienda, and D. Williams. QoS Routing Mechanisms and OSPF extensions. Work in progress, March 1998.

[57] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327, April 1998.

[58] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. Work in progress, draft-ietf-diffserv-af-04.txt, January 1999.

[59] Y. Hou, S. Panwar, Z. Zhang, H. Tzneg, and Y. Zhang. On network bandwidth sharing for transporting rate-adaptive packet video using feedback. In *Proceedings of Globecom*, November 1998.

[60] Y. T. Hou, H. Tzeng, and S. S. Panwar. A Simple ABR Switch Algorithm for the Weighted Max-Min Fairness Policy. In *Proc. IEEE ATM'97 Workshop*, pages 329–338, May 1997.

[61] Y. T. Hou, Henry H.-Y. Tzeng, and S. S. Panwar. A Generalized Max-Min Rate Allocation Policy and its Distributed Implementation using the ABR Flow Control Mechanism. In *Proc. of INFOCOM*, April 1998.

[62] D. Hughes. *Fair share in the context of MCR.* ATM Forum/AF-TM 94-0977, 1994.

[63] Linux Online Inc. Linux online. http://www.linux.org/.

[64] Intel. Indeo video product. http://developer.intel.com/ial/indeo/.

[65] J. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC-Based Error Control for Interactive Audio in the Internet. In *Proc. of IEEE INFOCOM*, March 1999.

[66] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. Work in progress, draft-ietf-diffserv-phb-ef-01.txt. Also available at ftp://ftp.ee.lbl.gov/papers/ef_phb.pdf, November 1998.

[67] R. Jain. Quality of Service in IP Networks. Presentation, http://www.cis.ohio-state.edu/ jain/talks/ipqos.htm, May 1998.

[68] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and Ram Viswanathan. ERICA switch algorithm: A complete description. ATM Forum/96-1172, August 1996. http://www.cis.ohio-state.edu/~jain/amtf/a96-1172.htm.

[69] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. An efficient rate allocation algorithm for ATM networks providing max-min fairness. In *Proceedings of the 6th IFIP International Conference on High Performance Networking*, September 1995.

[70] S. Kalyanaraman. *Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks.* PhD thesis, The Ohio State University, August 1997.

[71] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore. The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks. To appear IEEE/ACM Transactions on Networking, Feb 2000. http://www.cis.ohio-state.edu/~jain/papers/erica.htm.

[72] S. Kalyanaraman, B. Vandalore, R. Jain, R. Goyal, S. Fahmy, and S. Kota. Performance of TCP over ABR with Long-Range Dependent VBR Background Traffic over Terrestrial and Satellite ATM networks. In *Proceedings of LCN*, October 1998.

[73] H. Kanakia, P. Mishra, and A. Reibman. Packet Video transport in ATM networks using single-bit binary feedback. In *Proceedings of Sixth International Workshop on Packet Video*, September 1994.

[74] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, 1978.

[75] K. Lakshman, R. Yavatkar, and R. Finkel. Integrated CPU and network-I/O QoS management in an end system. *Computer Communications*, 21:325–333, 1998.

[76] T. V. Lakshman, P. P. Mishra, and K. K. Ramakrishnan. Transporting Compressed Video over ATM Networks Control. ATM Forum AF-TM/97-0652, June 1997.

[77] T.V Lakshman, A. Ortega, and A.R. Reibman. Variable bit rate (VBR) video: Tradeoffs and potentials. *Proceedings of the IEEE*, 36, May 1998.

[78] S.S Lam, S. Chow, and D.K Yau. An alogrithm for lossless smoothing of MPEG video. In *Proc. ACM SIGCOMM*, pages 281–293, September 1994.

[79] C. Liu, Y. Xie, M.J. Lee, and T.N. Saadawi. Multipoint multimedia teleconference system with adaptive synchronization. *IEEE JSAC*, 14(7):1422–1435, 1998.

[80] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *ACM SIGCOMM*, Stanford, CA, August 1996.

[81] J. M. McManus and K. W. Ross. Video-on-Demand over ATM: constand-rate transmission and transport. *IEEE JSAC*, 1(6):1087–98, 1996.

[82] J. Mosley. *Asynchronous Distributed Flow Control Algorithms. Ph.D Thesis*, Dept. Electrical Engg., MIT, Cambridge, 1984.

[83] M.R. Frater, J.F. Arnold and J. Zhang. MPEG-2 video error resilience experiments: The importance of considering the impact of the systems layer. *Signal Processing: Image Communication*, 14:269–275, 1999.

[84] P. Nee, K. Jeffay, and G. Danneels. The performance of two-dimensional media scaling for Internet videoconferencing. In *Proc. of NOSSDAV*, May 1997.

[85] Real Networks. Realvideo technical white paper. http://www.real.com/devzone/library/whitepapers/overview.html/.

[86] Real Networks. Surestream technical white paper. http://www.real.com/devzone/library/whitepapers/surestrm.html/.

[87] K. Nichols, S. Blacke, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS field) in IPv4 and IPv6 Headers. RFC 2474, December 1998.

[88] K. Nichols, V. Jacobson, , and L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. Work in progress, December 1997.

[89] P. Pancha and M. Zarki. Prioritized Transmission of Variable Bite Rate MPEG Video. In *IEEE GLOBECOM*, pages 1135–38, December 1992.

[90] P. Pancha and M. Zarki. Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks. *IEEE Trans. on Circuits and Systems for Video Technology*, 3(3):190–198, June 1993.

[91] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control - the single node case. *IEEE Transactions on Networking*, 1(3):344–357, June 1993.

[92] K. Patel and L. Rowe. Design and performance of the berkeley continuous media toolkit. In *Multimedia Computing and Networking 1997, Proc. SPIE 3020*, pages 194–2–6, 1997.

[93] C. I. Podilchuk, N. S. Jayant, and N. Farvardin. Three-dimensional subband coding of video. *IEEE Transactions on Image Processing*, 4(2), February 1995.

[94] M. Podolsky, C. Romer, and S. McCanne. Simulation of FEC-based error control for packet audio on the Internet. In *Proc. of IEEE INFOCOM*, April 1998.

[95] End point congestion management. IETF working group. http://www.ietf.org/html.charters/ecm-charter.html, February 2000.

[96] K. Ramakrishnan and S. Floyd. A proposal to add explicit congestion notification (ECN) to IP. RFC 2481, January 1999.

[97] S. Ramanathan, P.V. Rangan, H.M. Vin, and S.S Kumar. Enforcing application-level QoS by frame-induced packet discarding in video communications. *J. of Computer Communications*, 18(10):742–54, Oct 1995.

[98] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, and D. Towsley J. Stankovic. Online smoothing of live, variable-bit-rate video. In *NOSSDAV*, pages 249–258, May 1997.

[99] L. Roberts. Enhanced PRCA (proportional rate-control algorithm). ATM Forum/94-0735R1, August 1994.

[100] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Work in progress, draft-ietf-mpls-arch-04.txt, February 1999.

[101] J.D. Salehi, Z.-L. Zhang, J.F. Kurose, and D. Towsley. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. In *ACM SIGMETRICS*, pages 221–231, May 1996.

[102] R. Sanchez. Virtual ATM switch driver for ATM on Linux. http://www.ittc.ukans.edu/ rsanchez/software/vswitch.html.

[103] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. audio-video transport working group. RFC 1889, Sept 1987.

[104] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326, April 1998.

[105] M. Seaman, A. Smith, E. Crawley, and J. Wroclawski. Integrated Service Mappings on IEEE 802 Networks. Work in progress, draft-ietf-issll-is802-svc-mapping-03.txt, November 1998.

[106] S. Shenker and J. Wroclawski. General Characterization Parameters for Integrated Service Network Elements. RFC 2215, September 1997.

[107] S. Shenker and J. Wroclawski. Network Element Service Specification Template. RFC 2216, September 1997.

[108] R.J. Simcoe. *Test Configurations for Fairness and other Tests.* ATM Forum/AF-TM 94-0557, 1994.

[109] K. Siu and T. Tzeng. Intelligent congestion control for ABR service in ATM networks. *Computer Communication Review*, 24(5):81–106, October 1995.

[110] J. E Smith and F. W. Weingarten. Research Challenges for the Next Generation Internet. Report resulting from Workshop on Research Directions for the Next Generation Internet (http://www.cra.org/Policy/NGI/research_chall.pdf), May 1997.

[111] S.Sen, J. Rexford, and D. Towsley. Proxy prefix caching for multimedia streams. In *Proc. of IEEE INFOCOM*, March 1999.

[112] M. Tatipamula and B. Khasnabish. *Multimedia communications networks : technologies and services.* Artech House, 1998.

[113] IP Telephony. IETF working group. http://www.ietf.org/html.charters/iptel-charter.html, February 1999.

[114] D. Tennenhouse and et al. The ViewStation: a software-intensive approach to media processing and distribution. *Multimedia Systems*, 3:104–15, 1995.

[115] J. Tham, S. Ranganath, and A. Kassim. Highly scalable wavelet-based video codec for very low bit-rate environment. *Journal of Selected Areas of Communications-Very Low Bit Rate Coding*, 1996.

[116] D. Tsang and W. Wong. A new rate-based switch algorithm for ABR traffic to achieve max-min fairness with analytical approximation and delay adjustment. In *Proceedings of the IEEE INFOCOM '96*, pages 1174–1181, March 1996.

[117] B. Vandalore, S. Fahmy, R. Jain, R. Goyal, and M. Goyal. A definition of general weighted fairness and its support in explicit rate switch algorithms. In *Proceedings of the Sixth International Conference on Network Protocols 1998 (ICNP '98)*, pages 22–30, October 1998. http://www.cis.ohio-state.edu/~jain/papers/icnp98_bv.htm.

[118] B. Vandalore, R. Jain, R. Goyal, and S. Fahmy. Dynamic queue control functions for ATM ABR switch schemes: Design and analysis. *Journal of Computer Networks*, 1999. http://www.cis.ohio-state.edu/~jain/papers/cnis_qctrl.htm.

[119] B. Vickers, C. Albuquerque, and T. Suda. Adaptive Multicast of Multi-Layered Video: Rate-Based and Credit-Based Approaches. In *Proc. of IEEE INFOCOM*, San Francisco, 1998.

[120] B. J. Vickers, M. Lee, and T. Suda. Feedback control mechanisms for real-time multipoint video services. *IEEE Journal Selected Areas Communications*, 15(2), April 1997.

[121] J. Walpole, R. Koster, S. Cen, C. Cowan, D. Maier, D. McNamee, C. Pu, D. Steere, and L. Yu. A Player for Adaptive MPEG Video Streaming Over The Internet. In *Proc. 26th Applied Imagery Pattern Recognition Workshop AIPR-97, SPIE*, pages 249–258, October 1997.

[122] Y. Wang, Z. Zhang, D. Du, and D. Su. A network conscious approach to end-to-end video delivery over wide area networks using proxy servers. *(Submitted to) IEEE/ACM Trans. no Networking*, 1998.

[123] J. Wroclawski. Specification of the Controlled-Load Network Element Service. RFC 2211, September 1997.

[124] X.Li, S. Paul, and M.H. Ammar. Multi-Session Rate Control for Layered Video Multicast. In *Proc. IS&T/SPIE Multimedia Computing and Networking*, January 1999.

[125] X.Li, S. Paul, P. Pancha, and M.H. Ammar. Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery. In *Proc. NOSSDAV'97*, May 1997.

[126] X.R. Xu, A.C. Myers, H. Zhang, and R. Yavatkar. Resilient multicast support for continuous-media applications. In *Proc. of NOSSDAV'97*, 1997.

[127] D.K.Y. Yau and S.S Lam. Adaptive rate-controlled scheduling for multimedia applications. *IEEE/ACM Transactions on Networking*, 5(4):475–487, 1997.

[128] N. Yin. Max-min fairness vs. MCR guarantee on bandwidth allocation for ABR. In *Proc. of IEEE ATM'96 Workshop*, August 1996.

[129] N. Yin and M. G. Hluchyj. On Closed-loop Rate Control For ATM Cell Relay Networks. In *Proc. of IEEE INFOCOM*, pages 99–108, 1994.

[130] W. Zeng and B. Liu. Rate shaping by block dropping for transmission of mpeg-precoded video over channels of dynamic bandwidth. In *ACM Multimedia*, 1996.

[131] Z. Zhang, S. Nelakuditi, R. Aggarwa, and R. P. Tsang. Efficient server selective frame discard algorithms for stored video delivery over resource constrained networks. In *Proc. of IEEE INFOCOM*, March 1999.

[132] Z. Zhang, C. Sanchez, B. Salkewicz, and E. Crawley. QoS Extensions to OSPF. Work in progress, March 1998.

[133] B. Zheng and M. Atiquzzaman. Multimedia over ATM: progress, status and future. In *Proc. of ICC*, 1998.

ACRONYMS

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACK	Acknowledgment
ACR	Allowed Cell Rate
ADTF	ACR Decrease Time Factor
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BECN	Backward Explicit Congestion Notification
BRM	Backward Resource Management cell
CAC	Connection Admission Control
CBR	Constant Bit Rate
CBT	Core-Based Tree
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
CRM	Missing RM-cell Count
CWND	TCP Congestion Window
DES	Destination End System
DVMRP	Distance Vector Multicast Routing Protocol
ECN	Explicit Congestion Notification
EFCI	Explicit Forward Congestion Indication
EOM	End Of Message
EPD	Early Packet Discard
ER	Explicit Rate
ERICA+	Explicit Rate Indication for Congestion Avoidance+
FIFO	First In First Out
FRM	Forward Resource Management cell
FRTT	Fixed Round-Trip Time
FTP	File Transfer Protocol
GCRA	Generic Cell Rate Algorithm

GEO	Geosynchronous Earth Orbit
GFR	Guaranteed Frame Rate
ICMP	Internet Control Message Protocol
ICR	Initial Cell Rate
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
ION	Internetworking Over Non-broadcast multiple access
IP	Internet Protocol
IRTF	Internet Research Task Force
ISP	Internet Service Provider
ISSLL	Integrated Services over Specific Link Layers
ITU	International Telecommunications Union
LAN	Local Area Network
LANE	Local Area Network Emulation
LEO	Low Earth Orbit
LIJ	Leaf-Initiated join
MBS	Maximum Burst Size
MARS	Multicast Address Resolution Server
maxCTD	Maximum Cell Transfer Delay
MCR	Minimum Cell Rate
MCS	Multicast Server
MFS	Maximum Frame Size
MOSPF	Multicast extensions to Open Shortest Path First
MPLS	Multiprotocol Label Switching
MPOA	Multiprotocol Over ATM
Mrm	Controls RM bandwidth allocation
NAK	Negative acknowledgment
NI	No Increase
Nrm	Number of cells between FRM cells
PCR	Peak Cell Rate
PIM	Protocol-Independent Multicast
PIM-DM	Protocol-Independent Multicast-Dense Mode
PIM-SM	Protocol-Independent Multicast-Sparse Mode
PNNI	Private Network to Network Interface
PPD	Partial Packet Discard
QoS	Quality of Service
RDF	Rate Decrease Factor
RED	Random Early Detection
RFC	Request For Comments
RIF	Rate Increase Factor
RIJ	Root-Initiated join
RM	Resource Management
RSVP	Reservation Protocol

RTT	Round-Trip Time
SCR	Sustained Cell Rate
SES	Source End System
SSTHRESH	Slow Start Threshold
TBE	Transient Buffer Exposure
TCP	Transmission Control Protocol
TCR	Tagged Cell Rate
Trm	Upper Bound on Inter-FRM Time
UBR	Unspecified Bit Rate
UDP	User Datagram Protocol
UNI	User to Network Interface
UPC	Usage Parameter Control
VBR-nrt	Non Real-Time Variable Bit Rate
VBR-rt	Real-Time Variable Bit Rate
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VS/VD	Virtual Source/Virtual Destination
WAN	Wide Area Network
WWW	World Wide Web