ROUTING AND END-TO-END QUALITY OF SERVICE IN MULTIMEDIA NETWORKS

BY

Sanjeev Rampal

A THESIS SUBMITTED TO THE GRADUATE FACULTY OF

NORTH CAROLINA STATE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

RALEIGH

August, 1995

APPROVED BY:

CHAIR OF ADVISORY COMMITTEE	CO-CHAIR OF ADVISORY COMMITTEE

To my parents.

Acknowledgements

This thesis is dedicated to my parents Mr. Debdutta Rampal and Mrs. Rajesh Rampal. This thesis is a result of their efforts towards providing me with a good education and a loving home. I thank them from the bottom of my heart. My wife Neeti, has been a pillar of support for me over the last year. I am indebted to her for her love and encouragement. I am deeply grateful to all my family members and friends for their love and support.

I have been fortunate to have had a very helpful, knowledgeable and dedicated advisory committee. My advisors Dr Douglas Reeves and Dr Dharma Agrawal have been instrumental in shaping this thesis to its current form. Our invigorating discussions have always helped bring out important research topics and possible solutions. It has been a very fruitful learning experience for me. Dr Ioannis Viniotis has contributed significantly to my learning through our many discussions. Dr Mladen Vouk's involvement has helped me present my ideas in better ways. I thank all my committee members for their advice and guidance.

The different people who have contributed directly or indirectly towards the successful completion of this thesis are too many to thank individually. I thank all my friends, fellow researchers and colleagues for their help and support.

Author's Biography

Sanjeev Rampal was born in New Delhi, India. He obtained Bachelor's and Master's degrees in Electrical Engineering (in 1988 and 1991 respectively), both from the Indian Institute of Technology, Bombay. During this period, he was also with the Center for Microwave Research, Bombay, and with Messrs Godrej and Boyce, Bombay, involved in hardware and software design. Since October 1995, he has been with the Network Architecture group at IBM in Research Triangle Park, North Carolina.

Table of Contents

List of Tables viii

List of Figures x

List of Symbols xi

1 Introduction 1
- 1.1 Resource Management and Quality of Service in Multimedia Networks . . . 3
- 1.2 Overview of Thesis . . . 6

2 Routing Algorithms for Real-time Traffic 8
- 2.1 Introduction . . . 8
- 2.2 Admission Control Schemes for Real-time Traffic . . . 10
  - 2.2.1 Earliest Due Date . . . 11
  - 2.2.2 Stop& Go . . . 12
  - 2.2.3 Weighted Fair Queuing . . . 13
  - 2.2.4 Traffic Shaping and Effective Bandwidths . . . 14
- 2.3 Routing Methods . . . 15
  - 2.3.1 Conventional Algorithms . . . 16
  - 2.3.2 An Algorithm Based on Circuit-switched Network Routing . . . 16
  - 2.3.3 Real-time Algorithms . . . 17
  - 2.3.4 Optimal Routing . . . 19
  - 2.3.5 Summary . . . 20
- 2.4 Experimental Method . . . 21
- 2.5 Experimental Results and Analysis . . . 24
  - 2.5.1 Experiments Involving Homogeneous Traffic . . . 24
  - 2.5.2 Experiments Involving Heterogeneous Traffic . . . 27
  - 2.5.3 Other Experiments . . . 31
- 2.6 Summary . . . 36

3 Path Level Bandwidth Reservation for End-to-end QoS Guarantees **37**
3.1 Introduction . . . 37
3.2 The Multiple Hop QoS Problem : Some Insights . . . 40
3.2.1 Some Counter Intuitive Properties of Cell Loss over Multiple Nodes 40
3.2.2 Bandwidth Sharing versus Bandwidth Partitioning . . . 44
3.3 An Architecture Based on Virtual Path Level Bandwidth Reservation . . . 46
3.3.1 Multiplexing Potential of Real-life Sources . . . 48
3.4 The Multi-hop Bandwidth and Buffer Assignment Problem . . . 50
3.4.1 Queuing Model and Definitions . . . 51
3.4.2 Simultaneous Bandwidth and Buffer Assignment . . . 54
3.4.3 Bandwidth Allocation with Fixed Buffers . . . 56
3.4.4 Average Case Performance of the MGF policy in the Fixed Buffers Case . . . 59
3.5 An Implementation of the Proposed Architecture . . . 60
3.5.1 Operation of the WRR Server . . . 61
3.5.2 Call Admission for a Single VP . . . 64
3.5.3 Use of Buffer Sharing . . . 66
3.5.4 Implementation of the WRR Scheme . . . 69
3.6 Summary . . . 71

4 Dynamic Resource Allocation For Providing QoS Guarantees **72**
4.1 Introduction . . . 72
4.2 Description of the Algorithm . . . 76
4.2.1 Queuing Model and Definitions . . . 76
4.2.2 An Algorithm for Resource Allocation Based on QoS Measurements 77
4.3 Numerical Results . . . 80
4.3.1 Simulation Model and Procedure . . . 80
4.3.2 Variation of Convergence Time with Initial Update Interval . . . 82
4.3.3 Variation of Convergence Time with Buffer Size . . . 88
4.3.4 Variation of Convergence Time with Burst Length . . . 90
4.3.5 Variation of Convergence Time with Peak-to-Mean Ratio . . . 91
4.3.6 Variation of Convergence Time with CLP Requirement . . . 91
4.3.7 Convergence Times at Video Rates . . . 93
4.3.8 Resource Allocation Algorithms for Other/Multiple QoS Measures . 93
4.4 Comparison with Other Approaches . . . 95
4.4.1 Comparison of Steady State Performance with Equivalent Capacity Formulas . . . 95
4.4.2 Comparison of the Dynamic Behavior with Alternative Approaches . 97
4.5 Summary . . . 99

5 Conclusions **101**
5.1 Future Research . . . 103

List of References 105

A Monotonicity of Cell Loss with Service Rate 112

B Monotonicity of Cell Loss with Buffer Size 115

C Optimality of MGF policy in the Variable Buffer Allocation Case 117

List of Tables

1.1 Typical Data Rates and QoS Requirements for Multimedia Traffic 2

2.1 Average Path Lengths and Link Utilizations for EDD Server 24
2.2 Average Link Utilizations for Different Admission Control Policies 35

List of Figures

1.1 Relation between NP, NRM and CAC functions 4

2.1 Use of a Traffic Shaper with a Rate-based Server 14
2.2 Graph of Network used for Routing Experiments 22
2.3 Voice and Video Source Models for Routing Experiments 23
2.4 Call Acceptance Probabilities for Homogeneous (Video) Traffic, EDD Server 25
2.5 Illustration of Tradeoff of Fairness with Utilization in Routing 26
2.6 Call Acceptance Probabilities for Heterogeneous Traffic, EDD Server 28
2.7 Call Acceptance Probabilities for Heterogeneous Traffic, Stop& Go Server . 29
2.8 Call Acceptance Probabilities for Heterogeneous Traffic, WFQ Server 29
2.9 Relative Acceptance Probabilities of Voice and Video 30
2.10 Comparison of Admission Control Policies under Loose Delay Bounds . . . 31
2.11 Comparison of Admission Control Policies under Stringent Delay Bounds . 32
2.12 Call Acceptance Probabilities for Hypercube Network 33
2.13 Effective Bandwidth Versus Shaper Delay for Voice and Video 34
2.14 Call Acceptance Probabilities for Homogeneous Traffic with Traffic Shaping 35

3.1 The Two Basic Techniques for Obtaining High Bandwidth Utilization . . . 39
3.2 Problems in Determining End-to-end QoS 40
3.3 Non-monotonic Behavior of Losses with Service Rate in a Network 42
3.4 Non-monotonic Behavior of Losses with Buffer Size in a Network - I 43
3.5 Non-monotonic Behavior of Losses with Buffer size in a Network - II 43
3.6 Diminishing Returns from Statistical Multiplexing 45
3.7 Illustration of Path Level Bandwidth Reservations 46
3.8 Illustration of Statistical Multiplexing Potential of Voice Sources 49
3.9 Illustration of Statistical Multiplexing Potential of Video Sources 49
3.10 Queuing Model of a VP Used for Analysis 51
3.11 Worst Case Ratio of Bandwidth Requirement of MGF and Optimal Policies 58
3.12 Total VP Bandwidth for Two Hop Path for Different Allocations at Hop 1 . 60
3.13 Model of Output Buffered ATM Switch 62
3.14 Logical Model of WRR Scheduler 63
3.15 Emulation of Partitioned Buffers by a Shared Buffer 67
3.16 Variation of Cell Loss Probability with Unit Server Bandwidth 70

4.1 Queuing Model for Dynamic Bandwidth Allocation 77
4.2 Algorithm for varying K_n and U_n at decision instant t_n 79
4.3 Source Model used for Simulations of Dynamic Bandwidth Allocation . . . 81
4.4 Convergence Time Variation with Initial Update Interval (Absolute Convergence Time) 83
4.5 Convergence Time Versus Initial Update Interval (Number of Updates) . . 84
4.6 Sample Paths of Rate for Different Initial Update Intervals 85
4.7 Initial Portion of Sample Paths of Rate 85
4.8 Sample Paths of Cumulative CLP 86
4.9 Sample Paths of Current Loss Probability 87
4.10 Error in Transient CLP for Constant Rate Server 87
4.11 Convergence Time Variation with Buffer Size 89
4.12 Sample Paths of Rate for Buffer Variation 89
4.13 Convergence Time Variation with Source Burst Lengths 90
4.14 Convergence Time Variation with Source Peak-to-mean Ratio 92
4.15 Convergence Time Variation with Target CLP 92
4.16 Convergence Time Variation with Update Interval for Video Model 94
4.17 Use of REQS for Alternate QoS Definition 95
4.18 Bandwidth Savings Over Equivalent Capacity Formulas 96
4.19 Problem with Approaches Based on Error in Cumulative QoS 98

C.1 Networks Compared for Optimality Proof of MGF Policy 119

List of Symbols

List of symbols on routing algorithms

$d_{i,n}$	Local delay bound for channel i on link n.
D_i	End-to-end delay bound for channel i.
$T_{min,i}$	The minimum packet inter-arrival time for channel i.
$X_{i,n}$	The maximum transmission time of a packet of channel i on link n.
$X_{max,n}$	The maximum transmission time of a packet of any channel using link n.
H_i	The number of hops in the path of channel i.
T_g	The duration of frame size g.
C_n	The bandwidth of link n.
C_n^g	The total bandwidth assigned to frame g at link n.
r_i	The rate assigned to channel i in a Stop & Go scheme.
$l_{max,n}$	The maximum packet size over all channels using link n.
β_i	Token size of the leaky bucket for channel i.
ρ_i	Token generation rate for the leaky bucket for channel i.

List of symbols on end-to-end QoS

$\mathcal{V}_j$	Virtual path j.
B_j^h	The buffer size reserved for VP j at its h'th hop.
n_j	The number of slots per WRR cycle reserved for VP j.
T	The duration of a cycle of the WRR server.

d_s^h	The maximum delay in the switch fabric at the h'th hop of a VP.
H	The number of physical hops (in the path of a VP).
T_{prop}^h	The propagation delay at the h'th hop of a VP.
D	The specified end-to-end delay bound for a VP.
ϵ	The specified bound on end-to-end loss probability for a VP.
W	The end-to-end waiting buffer space assigned to a VP in the queuing model.
B^h	The buffer space at hop h in the queuing model of a VP.
C^h	The service rate allocated to node h in the queuing model of a VP.

List of symbols on dynamic resource allocation

$\mu(t)$	The instantaneous service rate of the queue.
$\mathcal{Q}$	The specified Quality-of-Service bound.
U_n	Update interval n.
μ^*	The steady state value of the rate.
$\mathcal{A}_n$	The number of arrivals during update interval n.
$\mathcal{L}_n$	The number of losses during update interval n.
$\mathcal{P}_n$	The loss probability during update interval n.
$\mathcal{P}_{0..n}$	The cumulative loss probability up to and including interval n.
$\mathcal{P}^*$	The steady state value of the cumulative loss probability.
$\mathcal{Q}_l$	The specified bound on average cell loss probability.
K_n	The error scalar for dynamic rate control in update interval n.
S_0, S_1	The 2 states of the input MMBP source.
λ_0, λ_1	The average arrival rates in states S_0 and S_1 respectively.
γ_0, γ_1	The average durations of states S_0 and S_1 respectively.

Chapter 1

Introduction

Packet/cell switched data networks are increasingly being utilized to carry multimedia traffic (e.g., video and voice). This trend is expected to continue with the deployment on a widespread scale of high-speed networking technology, such as ATM [46]. The new networking technology is intended to fully support both real-time (or time-sensitive) traffic such as interactive multimedia sessions and non-real-time traffic such as bulk file transfers. Transport of multimedia sessions requires the network to provide certain "Quality of Service" (QoS) guarantees to users. Typical QoS measures include bounds on latency of information transfer and information loss rate. Whenever a user requests service, the QoS guarantees calculated by the network can be compared with the QoS levels desired in order to decide whether the user can be admitted and receive the requested level of service. Once admitted, each user can expect to receive the level of service specified. For instance, such an approach ensures that a user participating in a video teleconference does not temporarily lose his picture whenever a large file is downloaded in another part of the network.

The two main types of networks existing at present are circuit-switched networks and packet-switched networks. The main advantage of circuit-switched networks such as telephone networks is the assured performance. As an example, when a user picks up the telephone, a certain level of voice quality can be expected. The main advantage of packet-switched data networks on the other hand is their efficiency. It costs very little to send e-mail for example. However, there is no guarantee as to when a file sent by e-mail will actually be delivered to its destination. In some sense, the next generation multimedia networks are expected to combine the best features of both these types of networks viz. assured

	Peak Rate	**Mean Rate**	**Allowable Loss Probability**
Voice	32 Kb/s	11.2 Kb/s	10**(-2)
Video	11.6 Mb/s	3.85 Mb/s	10**(-5)

(a) Source models for voice and video, and typical acceptable loss rates.

CCITT G.114 Delay Recommendations	
One-Way Delay	**Characterization of Quality**
0 to 150 ms	"acceptable for most user applications"
150 to 400 ms	"may impact some applications"
above 400 ms	"unacceptable for general network planning purposes"

(b) End-to-end delay requirements for voice and video.

Table 1.1: Typical data rates and QoS requirements for multimedia traffic.

performance levels and high efficiency. As is shown later, the difficulty in optimizing the performance of such networks arises mainly from these conflicting requirements.

The need is being felt to provide support for multimedia applications even in existing packet switched data networks which were not designed specifically to carry such traffic. For instance, the next generation version (IPv6) ([48]) of the Internet Protocol which is currently used over most of the Internet, will use resource reservation protocols designed specially for the support of real-time multimedia applications. The use of such applications will only increase with the proliferation of broad band networking technology. Further, even tariff policies in such networks are expected to depend on the level of performance desired by each user [13]. When each user pays according to the desired level of service, it is important to be able to guarantee the level of service delivered. Hence, the need for being able to provide QoS guarantees will be even greater than in current networks. Aras et al[3] and Kurose [36] provide good surveys of recent research towards support of real-time multimedia applications in packet-switched networks.

Table 1.1 shows the data rates and typical QoS requirements of voice and video traffic [37][46][55] (loss probabilities are for cells in ATM networks).

1.1 Resource Management and Quality of Service in Multimedia Networks

A key problem facing the network service provider is that of providing QoS guarantees to users while utilizing network resources as efficiently as possible. Typically, a tradeoff exists between the efficiency of use of network resources and the ability to accurately characterize each user's level of service. Kurose [36] outlines some of the key issues involved in providing QoS guarantees along with high efficiency.

Network resources typically include link bandwidth, buffer space at the switching nodes and processing capability at the switching nodes. The service provider has a number of controls using which network performance can be tuned. These include the routing, bandwidth allocation, call admission control (CAC) and packet/ cell scheduling functions. The routing function determines the physical path taken by packets/ cells on their way from source to destination. Clearly, by selecting appropriate paths, the routing function can efficiently utilize network resources. Bandwidth allocation refers to the service rate assigned to a source at a physical link. A higher service rate normally implies better service (lower delays) but lower efficiency. The call admission control function controls the acceptance of calls[1] into the network. A new call is admitted only if its QoS requirements can be met while meeting those of all existing calls in the network. A good CAC algorithm should result in the maximum possible utilization of resources. Finally, scheduling policies applied at the cell or packet level enable provision preferential treatment to certain traffic classes (e.g. those that have stringent QoS requirements) over others so that the QoS requirements of all calls can be met if possible.

Mitrou et al [38] provide a generalization of the different kinds of controls available to a service provider in order to optimize network performance. Figure 1.1 (from [38]) shows three different kinds of control functions which are typically involved in network resource optimization. Using terminology defined by the ITU for ATM networks [7], traffic and congestion control functions may be classified into Network Provisioning (NP), Network Resource Management (NRM) and Call Admission Control (CAC). The NP function controls long-term configuration of network resources mainly at the physical level, e.g. assigning

[1] In this thesis the terms 'call', 'channel' and 'virtual circuit' are used equivalently.

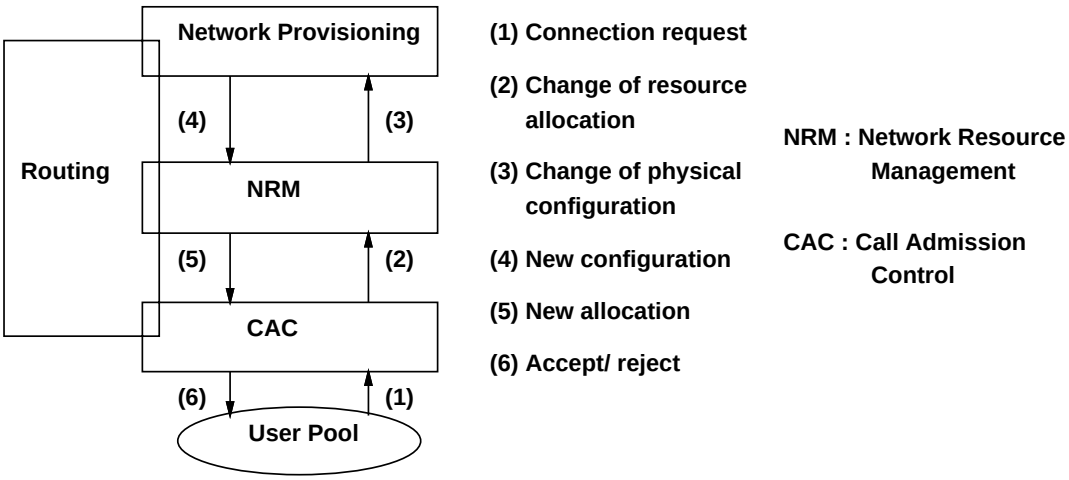

Figure 1.1: Relation between NP, NRM, CAC functions

trunk capacity between nodes or adding new nodes. The CAC function operates at the time scale of individual calls as described above. NRM functions operate over intermediate time scales by providing for resource allocation to classes of calls. An example of NRM controls is the control of the configuration and resource allocation for Virtual Paths (VPs) in an ATM network. Other controls such as packet/ cell scheduling operate over even smaller time scales to optimize performance experienced during the duration of a call. This thesis concentrates on controls which are classified in the NRM and CAC categories and on controls which operate at the cell level.

Typically, tradeoffs exist in using multiple controls at the same time. For instance packet scheduling and CAC algorithms exist which provide end-to-end [2] QoS guarantees but compromise on bandwidth allocation, resulting in low utilization [19], [23], [44]. On the other hand, packet scheduling schemes have been suggested which exploit statistical resource allocation but do not allow the to network provide end-to-end guarantees e.g. [12] or which provide approximate guarantees but only at a single node e.g. [25].

Some of the problems in providing QoS guarantees along with high resource utilization in a multiple node environment are as follows.

[2] In this thesis, the term end-to-end refers to a physical path comprising several switching nodes of a homogeneous network. Neither are heterogeneous "internetworks" considered nor the effects of the higher layer software at the originating and destination host. Analysis of the complete end-to-end path from the originating application software to the destination application software is outside the scope of this thesis.

- Due to the "bursty" and statistically variable nature of typical multimedia traffic, high efficiency can be obtained mainly by statistical sharing of network resources. Allowing several bursty connections to share a given resource improves the utilization of the resource, but makes it difficult to characterize the performance seen by the individual connections [36]. The ratio of peak to mean rates in Table 1.1 is an indication of the variability in traffic generation rate of typical multimedia sources.
- Statistical resource sharing techniques rely on accurate source traffic models. Multimedia sources are very difficult to model in the first place ([29]). Additionally, any model used at the entrance to a network may not be valid in general deep within a network after some buffering and multiplexing with other streams at different nodes. Hence resource allocation techniques within a network may result in performance levels different from those calculated using the source model at the edge of the network.
- The admission control function has to be executed on-line as and when each connection comes in. The use of exhaustive numerical analysis is thus not feasible.
- The problem of providing end-to-end (i.e. multiple node) QoS guarantees is inherently much more difficult than the single node case. This is because even basic relations at a single node between control actions and resultant performance are not valid in the multiple node case. For instance in a latter chapter, it is shown that increasing the service rate of a single node in isolation always decreases its losses, while if this node is part of a network of nodes, then increasing its service rate may actually increase overall losses. Hence simple gradient-type optimization procedures do not yield optimal solutions.
- The large delay-bandwidth product in typical broad band networks exacerbates the difficulty of end-to-end control since each node does not accurately know the state of downstream nodes in general.
- Typical performance measures for multimedia applications can be quite stringent. Trying to guarantee cell losses to occur with precisions as low as 1e-5 or lower can be expected to be a difficult task in any case.

Such properties make the problem of providing end-to-end QoS guarantees while utilizing resources efficiently, a very difficult one. The results from this thesis provide some contributions in this area.

1.2 Overview of Thesis

In this thesis, three related problems in the area of efficient resource management while providing QoS guarantees in multimedia networks, are analyzed. Most of the terminology used is specific to ATM networks; however most results conceptually apply to any packet switched network which requires provision of QoS guarantees to users.

In **chapter 2**, a joint performance evaluation of routing and admission control algorithms is performed. The performance of three standard admission control schemes which provide end-to-end QoS guarantees is analyzed. The CAC schemes studied provide deterministic end-to-end guarantees on cell delay (i.e. guarantees on the maximum possible queuing delay) and guarantee no cell loss. New routing algorithms which exploit knowledge of the constraints of the CAC function are developed. It is shown that these algorithms result in improved call acceptance as compared to conventional approaches in which the routing function and admission control function operate separately. Thus these techniques enable some improvement in network efficiency while retaining the ability to guarantee QoS. This is the first detailed study of the joint performance evaluation of routing and admission control functions. A new algorithm, the "Shortest Cost" algorithm which addresses the constraints of the CAC algorithm, yields the best performance in terms of the call acceptance probability. The routing algorithms are also evaluated for *fairness* properties when two different classes of calls are present. The algorithms are tested under different simulated conditions. An approach is suggested which improves the link utilizations in such schemes by employing traffic shapers at the network edge for loss-tolerant applications.

In **chapter 3** the problem of improving resource utilizations while providing end-to-end QoS guarantees is addressed. The technique introduced in chapter 2 uses traffic shaping to improve the achievable utilizations. In chapter 3, this is extended to additionally exploit statistical multiplexing between channels traversing the same set of links resulting in even better utilization. A new approach is introduced viz. virtual path (VP) level reservations.

In this approach, deterministic bandwidth reservations are made on groups of calls rather than individual calls as in the schemes studied in chapter 2. End-to-end QoS guarantees can still be provided along with improved utilization levels over the schemes discussed in chapter 2. Using a queuing model of a VP traversing a multi-hop path, the problem of distributing end-to-end resources into per-hop resource allocations in order to optimally satisfy end-to-end QoS specifications is addressed. The advantages of an approach (the 'Maximal Gain First' policy) are compared with other approaches in terms of efficiency in use of given resources for meeting an end-to-end specification on cell loss. The working of cell scheduling policy (Weighted Round Robin (WRR)) to implement the proposed scheme is outlined. Some implementation issues are discussed.

Chapter 4 addresses the problem of dynamic resource allocation for meeting a specified QoS constraint. An algorithm is developed to determine the minimal bandwidth needed to satisfy a specified cell loss probability. The main advantage of this algorithm is that it obtains the minimum bandwidth requirements for a source with an arbitrary (stationary) traffic model and loss specification. Both the dynamic and steady state behavior of the algorithm are studied in detail under different simulated conditions. The effectiveness of this approach in determining the optimal bandwidth requirements in order to satisfy other types of QoS definitions is also demonstrated. The performance of this algorithm is compared with other approaches including approaches based on the popular "equivalent capacity" formulas. This algorithm is applicable for bandwidth allocation in the architecture proposed in chapter 3. Additionally, it can be used to determine the optimal allocation level for other types of resources and QoS definitions.

Overall conclusions, a summary of the results and suggestions for future research are presented in **chapter 5**.

Chapter 2

Routing Algorithms for Real-time Traffic

2.1 Introduction

In this chapter, the use of the routing function is investigated in order to to improve efficiency of network usage. The ability of the network to provide end-to-end QoS guarantees is retained through the use of an appropriate packet scheduling and CAC mechanism. A performance evaluation of 3 standard CAC schemes which provide end-to-end QoS guarantees is also presented. Finally, an approach using traffic shapers at the edge of the network is introduced to improve the utilizations achievable with these schemes.

The CAC method used by a network directly determines which channels are accepted, and which are blocked. Thus, admission control is of vital importance to both the network provider and the users. The *routing* function finds a path in the network from a source to a destination and hence is an important control parameter for network performance. In a connection-oriented service, all packets from a call follow this same path from the source to the destination. Routing itself does not accept or reject calls. However, the choice of route directly influences the likelihood that a call will be accepted. As an example, if a routing method selects a route which has very few resources available, the request for a connection will probably be rejected. Once a route is selected, the availability of resources along other paths is of no use in call admission. As a result, routing is also important to the network provider and users.

Multimedia traffic has special QoS requirements which can only be met by appropriate admission control and packet scheduling methods. Thus, it is important to know how

existing routing algorithms will perform on this type of traffic and with such admission control methods, and if they don't perform well, to devise algorithms which will work well. Little work has been done on this important topic. This previous work includes the following.

- Parris and Ferrari [45] proposed a routing algorithm for real-time applications (described in Section 2.3 as the SP algorithm) but did not investigate its performance.
- Kompella et al [35] have investigated some routing problems for multicasting of multimedia data. They show that finding an optimal multicast tree which meets delay bound constraints is an NP-complete problem, and present a heuristic approximation.
- Ahmadi et al.[2] investigated dynamic routing algorithms for the Paris networking project. They proposed a method which computed the shortest path (where distance was a function of link utilization) satisfying the delay bound, and having the minimum possible hop count. This algorithm was shown to be much better than a minimum-hop algorithm, and somewhat better than a pure shortest-path algorithm. The routing algorithm was evaluated only in combination with the admission control method used in the Paris project. This CAC method is based on statistical multiplexing and the computation of equivalent capacity.

The goal of this study is to examine the impact of routing algorithms on the blocking probability of real-time channels. Several new routing algorithms have been formulated and implemented to specifically address the QoS requirements and admission control constraints of multimedia traffic. An experimental investigation (using simulation) of the relative performance of the algorithms is performed for a variety of admission control algorithms and traffic loads. This study is the first in-depth investigation of this important topic. As a secondary issue, the performance of several admission control policies for multimedia data is analyzed. The study is restricted to deterministic policies which can provide strict *a priori* guarantees of QoS; no new policies are proposed herein. This represents the first detailed, quantitative comparison of these policies for a variety of traffic conditions. The experiments yield a number of important insights into router and admission controller behavior.

The organization of the chapter is as follows. In the next section the three admission

control methods used in the experiments are reviewed, and the use of traffic shaping with those methods is also described. Section 2.3 describes the routing algorithms which are evaluated. Section 2.4 describes the simulation method and assumptions, and Section 2.5 presents the results of the experiments. The last section summarizes the findings.

2.2 Admission Control Schemes for Real-time Traffic

In this section, three well-known deterministic methods of admission control are described. These three methods were used to investigate the interaction between routing and admission control of multimedia traffic.

In this study, the use of the term "call admission" refers to the service policy at the switches, as well as the algorithm for controlling the admittance or rejection of calls; this is not standard usage, but is convenient for our purposes. The resources required for a call are calculated from some characterization of the properties of that call. To make the calculation simple, only a few traffic parameters are used, such as the packet peak and mean arrival rates, and the mean burst length.

This study exclusively used deterministic methods of admission control and multiplexing. No admission control schemes exist as yet, which employ complete statistical multiplexing at all nodes of the network and still provide end-to-end QoS guarantees [3]. The only effective way to calculate the QoS produced by statistical methods in multi-hop networks is to simulate the end-to-end transmission of every packet. This can require a huge amount of CPU time for high-speed networks. In addition, a fair comparison of policies requires all performance parameters except one to be fixed. For example, to meaningfully compare the delays resulting from two different admission control policies, it must be ascertained that the loss rates for the two policies are the same. However, there is no known way to accurately fix the end-to-end loss rate that a given statistical policy will produce.

By contrast, when a deterministic admission control method is used, the QoS can be calculated purely by simulating the admission control function. This requires several orders of magnitude less CPU time than packet-level simulation. One can also easily control the experiments to ensure that all admission control methods provide the same quality-of-service

(loss and delay)[1]. Thus, the results from multiple experiments can be directly and fairly compared.

The three admission control methods used in this study are Earliest-Due-Date (EDD)[19], Stop&Go[23], and Weighted-Fair-Queuing (WFQ)[44]. Each of these is described briefly below. The low utilization which is a drawback of deterministic methods can be improved through the use of traffic shaping, at the expense of some packet loss. A method of traffic shaping for multimedia traffic is also described.

2.2.1 Earliest Due Date

The Earliest Due Date (EDD) packet scheduling and admission control policy [19], [33], splits end-to-end delay bounds into local per-link delay bounds. The sum of the link delay bounds along the path from the origin to the destination must be no more than the end-to-end delay bound. Packets for an outgoing link are multiplexed according to the Earliest Due Date policy which is a *deadline*-based algorithm which uses the "logical" arrival time of a packet and its delay bound at this link.

Let the local delay bound of channel i on link n be denoted as $d_{i,n}$. Also denote the minimum packet interarrival time at the source for channel i as $T_{min,i}$, and the maximum transmission time of any packet of channel i on link n as $X_{i,n}$. If the K channels using a link n are ordered by increasing value of their local delay bounds for this link (so that $d_{1,n} \leq d_{2,n} \leq \cdots \leq d_{K,n}$), then the following necessary constraint must be satisfied to accept channel i on link n:

$$T_{min,i} \geq \sum_{j=1}^{K} X_{j,n}, \qquad (i = 1, \cdots, K) \tag{2.1}$$

Let $X_{max,n}$ be the maximum packet transmission time of any existing channel using link n. The requesting channel can be accepted without violating the guarantees for existing channels on the link if and only if

$$d_{i,n} \geq \sum_{j=1}^{i} X_{j,n} + X_{max,n} \qquad (i = 1, \cdots, K) \tag{2.2}$$

[1]Our experiments did not place any restrictions on jitter. There may be significant differences between the admission control methods and routers with respect to jitter; this is not studied here.

This equation also represents the maximum queuing delay which a packet for channel i will experience on link n. The end-to-end delay bound for this channel is then $D_i = \sum_{n=1}^{H_i} d_{i,n}$, where H_i is the number of hops[2] on the path from the origin to the destination of channel i.

The EDD policy is based on results from real-time scheduling theory. It is considered to be somewhat expensive to implement, but appears to be quite flexible. [3]

2.2.2 Stop& Go

The Stop&Go service and admission control policy [23] is designed to provide very tight bounds on end-to-end jitter in networks, as well as predictable end-to-end delays. Each requesting channel i is assigned to some *frame* of size (duration) T_g, and is allocated a transmission bandwidth r_i. During each interval of length T_g, i is allowed to transmit at most $r_i \cdot T_g$ bits; this is termed the (r_i, T_g)-smoothness property.

A link is assigned a set of G frame sizes, $T_1, \cdots, T_G$; assume the frame sizes are ordered in decreasing size. The total bandwidth of link n assigned to frame level g, denoted C_n^g, is the sum of the bandwidths (r_i's) for all channels assigned to frame level g. Let C_n be the total bandwidth of link n, and $l_{max,n}$ denote the maximum packet size over all channels using link n.

A requesting call can be accepted by this link, without violating the QoS guarantees given to existing calls using the link, only if the following necessary constraint is satisfied:

$$-C_n^{g0} + \sum_{g=g0}^{G} C_n^g (1 + \lceil T_{g0}/T_g \rceil) T_g / T_{g0} \leq \begin{cases} C_n - l_{max,n}/T_{g0}, & g0 = 2, \cdots, G \\ C_n, & g0 = 1 \end{cases} \tag{2.3}$$

When the ratios of the frame sizes are integers, instead of Equation 2.3 reduces to a simple capacity constraint [23]. The framing structure is essential to providing end-to-end QoS guarantees but also introduces quantization of the rates that can be assigned to a channel. This rate quantization can lead to significant wastage of bandwidth for low rate channels [23].

[2] In this thesis, the terms 'hop' and 'link' are used equivalently.

[3] The "simple" form of EDD has been described and experimented with here; the full version is even more flexible, and even more expensive to implement.

For a channel i routed over a path with hop-count H_i, and assigned to a frame of size T_g at every link along the path, the maximum end-to-end-delay D_i it will experience is guaranteed to be

$$H_i \cdot T_g \leq D_i \leq 2H_i \cdot T_g \tag{2.4}$$

Stop&Go provides very predictable service, and is not particularly difficult to implement. It does require sources to obey the smoothness property. Also note that the end-to-end delay is related to (i.e., dependent upon) the rate allocated to the channel.

2.2.3 Weighted Fair Queuing

The Weighted Fair Queuing (WFQ) method was proposed by Demers et al.[15] and analyzed by Parekh[44]. In WFQ, channels sharing an outgoing link are transmitted as if they were serviced in Round-Robin order on a bit-by-bit basis. Channels can receive differing amounts of bandwidth by using a Weighted Round-Robin service scheme ("weighted" so that some channels can transmit more than 1 bit on each round). Tight bounds on end-to-end delay can be achieved if the arrival process for the channel at the source is regulated by a token bucket with bucket capacity β_i and token generation rate ρ_i. Token buckets are not described here; a good introduction can be found in [46]. A channel i is assigned a transmission bandwidth or rate of $r_i \geq \rho_i$ on all links along its path.

A necessary constraint which must be satisfied in order to accept a new channel on a link n is that the sum of the rates of all the channels multiplexed onto this link be less than the capacity of the link. Thus, if K channels have been accepted on this link,

$$\sum_{i=1}^{K} r_i \leq C_n \tag{2.5}$$

The delay bound for a requesting channel is

$$D_i \leq \frac{\beta_i}{r_i} + (H_i - 1)\frac{l_i}{r_i} + \sum_{n=1}^{H_i} \frac{l_{max,n}}{C_n} \tag{2.6}$$

The rate assigned to a channel is a function both of the peak rate of the source (after regulation by the token bucket), and of the delay bound which is desired. WFQ has received a great deal of attention lately, and is considered to be both reasonable to implement, and to provide high quality of service (i.e. low delays).

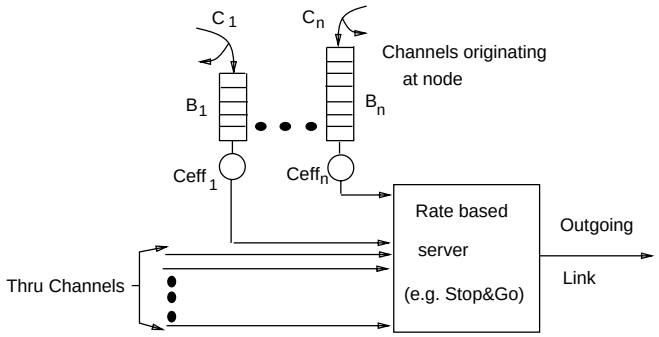

Figure 2.1: Use of traffic shaping with a rate-based server, for a single output link of a switching node.

2.2.4 Traffic Shaping and Effective Bandwidths

The *effective bandwidth* (also referred to as *equivalent capacity*) of a source represents a minimal bandwidth that must be provided to guarantee its required bound on average loss probability [17], [25]. (This bandwidth refers to the service rate of a queue at which the loss probability will be no more than a specified value when the given source is fed to it). Approximate formulas have been derived in [17] and [25] for the effective bandwidths for certain types of sources.

The enforcement of the effective bandwidth is achieved by *traffic shaping*. A traffic shaper delays the arrival of some packets in order to smooth the arrival process for the channel. This delay can be accomplished by mechanisms such as leaky and token buckets (again, see [46] for an introduction). If the buffer capacity is exceeded (input rate exceeds output rate for too long a period), some packets may be discarded. The token generation rate and bucket size are calculated to ensure that the loss probability does not exceed the user-specified QoS bound.

The benefit of traffic shaping is that it can improve network utilization while keeping delay and loss bounds within acceptable levels. Traffic shaping of a source can be combined with a deterministic admission control policy to yield strong end-to-end quality-of-service guarantees, with improved network utilization; see Figure 2.1. The traffic shaper introduces an additional delay which must be subtracted from the specified end-to-end delay bound to yield the allowable network delay. When the network delay bound is considerably less than the required end-to-end delay bound, traffic shaping can significantly reduce the effective

bandwidth. Since the deterministic admission control methods guarantee zero loss inside the network, all allowable loss occurs at the network interface and this loss can be engineered to be no more than the tolerable end-to-end loss. The effective bandwidth of a channel after shaping is regarded as the peak bandwidth of the channel, for purposes of admission control.

2.3 Routing Methods

The routing problem has been investigated extensively for both circuit-switched and packet-switched networks. Some recent summaries of routing techniques are by Bertsekas [5], Girard [22], and Perlman [47]. The goals for routing include maximizing the load accepted by the network while providing satisfactory QoS to channels and treating all call requests equally (i.e., being fair). These goals can be contradictory, so the search for optimal algorithms is in fact quite elusive.

Routing methods can be static or dynamic. Static methods are extremely simple to implement, but are unable to react to changing network traffic patterns. This can easily lead to unnecessary and very localized congestion. Dynamic methods attempt to balance the load across the network and thus accept more traffic with less delay variation across paths. Dynamic routers often optimize some metric (such as average packet delay) subject to certain constraints on the choice of links. Dynamic methods are more complex than static methods, and are also subject to problems of oscillation. For details, see any of the surveys cited above.

Most dynamic algorithms are based upon the concept of a shortest (or least-loaded) path. These are usually computed by Djikstra's algorithm or the Bellman-Ford algorithm. Considerably different routing goals can be accomplished by suitably defining the *length* of a link, while the algorithm executed remains exactly the same.

Little work has been done on routing of multimedia traffic with delay and loss requirements, as mentioned in Section 2.1. A wide variety of routing algorithms were implemented to determine what effect routing has on call acceptance. These algorithms can be roughly classified into three groups:

- "Conventional" algorithms which have been considered for packet-switched data

networks;

- Sequential algorithms used in circuit-switched networks and proposed for integrated networks; and,
- Real-time algorithms which promise to better fulfill the quality-of-service requirements of multimedia traffic.

Each of the routing algorithms is now described.

2.3.1 Conventional Algorithms

As a point of comparison, two simple generic routing algorithms used in packet-switched data networks were chosen.

The **C_STAT** algorithm is a static router. For each origin-destination pair, a minimum-hop path is found. If there are multiple minimum-hop paths between the origin and destination, the one which uses the lowest numbered link (according to a static link ordering) possible at each hop is chosen. The selected path for each destination is stored for each host. All calls originating at the same origin and having the same destination follow the same path, regardless of network loading conditions.

The **C_DYN** algorithm is a dynamic shortest-path router. The length of each link is set to be inversely proportional to the spare capacity fraction (i.e., 1-utilization) of that link. (This formulation corresponds to a link length equal to the average queuing delay in a network of $M/M/1$ queues [5]). A desirable by-product is that the load on the network tends to get evenly distributed.

2.3.2 An Algorithm Based on Circuit-switched Network Routing

Hwang et al [28] proposed the use of sequential routing algorithms, similar to those used in circuit-switched networks such as telephone networks, for routing in integrated services networks. Because multimedia sessions have many of the characteristics of circuits in circuit-switched networks (e.g. guaranteed service), it seems reasonable to expect that circuit-switched routing algorithms would perform well. These algorithms also have the advantage that they are completely distributed, and do not require updates of routing tables.

The best performing member of this group is known as the Crank back **(CB)** algorithm. In the Crank back algorithm, a path from the origin to the destination is found one hop at a

time. From the origin, the algorithm checks the first link on the static minimum hop path to the destination. If the call can be accepted on that link (according to the admission control constraint), that link is added to the path. Otherwise, each of the remaining links from the origin are tried, in (static) link order. The first one for which the call can be accepted is added to the path. Each time a link is added to the path, routing continues recursively from the node at the other end of the link. The algorithm terminates if the destination is reached. If a node is encountered for which no outgoing links can accept the requesting call, backtracking to the previous node on the partially-routed path is allowed. A different outgoing link is chosen from this node (again, in static order) to avoid the congested node. If this process exhausts all paths starting from the origin without successfully reaching the destination, failure to find a route is reported. Once a feasible path is found, the call is accepted if it will meet its end-to-end-delay bound along this path.

2.3.3 Real-time Algorithms

None of the above algorithms is formulated to specifically meet the needs of multimedia traffic. We use the term *real-time routing algorithm* for any method which considers the delay bounds and necessary constraints of the call admission function. Several such methods are presented below; except where noted, they are proposed here for the first time.

These algorithms are constrained to avoid links which will certainly lead to call rejection. Any link for which the necessary constraint of the admission control algorithm (Equations 2.1 for EDD, 2.3 for Stop&Go, and 2.5 for WFQ) is not fulfilled is eliminated from consideration by the routing algorithm. That is, the network configuration is pruned before the routing algorithm is run.

The Shortest Path **(SP)** algorithm is targeted to find a path which minimizes the delay bound which can be guaranteed to the requesting channel. Let $d_{i,n}^{min}$ denote the minimum delay bound which can be guaranteed for a requesting channel i on link n, without invalidating the guarantees for existing channels which use that link. $d_{i,n}^{min}$ is calculated as follows:

- EDD: The smallest value of $d_{i,n}$ is computed for which the admission control condition (Equation 2.2) is still true. Since deadlines of channels already accepted on the link are already ordered by value, finding this smallest value is

not difficult.

- Stop&Go: Let $T_{g_{min}}$ be the smallest frame for which the quantized bandwidth r_i can be accommodated, according to Equation 2.3. $T_{g_{min}}$ is not difficult to find, since in a real system there will likely only be a small number of frame sizes. $d_{i,n}^{min}$ is equal to $2T_{g_{min}}$.
- WFQ: The minimum value of $r_i \geq \rho_i$ which satisfies Equation 2.6 can be computed analytically. Given this value for r_i, the maximum delay for a packet of channel i on link n will be $\frac{l_i}{r_i} + \frac{l_{max,n}}{C_n}$.

The length of each link n is set to $d_{i,n}^{min}$ and then a shortest-path calculation is performed. The advantage of the SP algorithm is that worst-case delays are calculated directly from the admission control policy, rather than indirectly from the link utilization. The SP algorithm was first proposed in [45]. However, no quantitative evaluation was presented.

The Shortest Cost **(SC)** routing algorithm is targeted to meet server constraints, rather than minimizing end-to-end delay. This approach seems reasonable in the case where end-to-end delay bounds are not particularly difficult to achieve. In such a case, the constraint imposed by service policy is of more importance than the delay bound for a link. The length of link n is modeled in the following way for each admission control method:

- EDD: From Equation 2.1 it is clear that the probability of call i being blocked on link n increases as the value of the right hand side of the equation approaches that of the left hand side. Acceptance will be improved when this difference is as large as possible. Hence, the length of link n is set to $1/(T_{min,n} - \sum_{j=1}^{K} X_{j,n})$, where $T_{min,n}$ is the minimum of peak packet interarrival times over all channels using link n (including the requesting channel).
- Stop&Go: From Equation 2.3, it is seen that the blocking probability increases as the quantity on the left hand side approaches that on the right hand side i.e. the link capacity. The length of link n is set to $1/(C_n - (-C_n^{g0} + \sum_{g=g0}^{G} C_n^g(1 + \lceil T_{g0}/T_g \rceil)T_g/T_{g0}))$.
- WFQ: The necessary constraint for WFQ is just a bandwidth constraint. Blocking increases when utilization increases. Thus, the length of link n is set to $1/(C_n - \sum_{i=1}^{K} r_i)$.

The SC algorithm performs a shortest path calculation using one of the above definitions

of the link length. No consideration is given to meeting the end-to-end delay bound.

A minimum-hop version of algorithm SC, called the Modified Shortest Cost **(M-SC)** algorithm was also implemented. This algorithm finds the shortest path (with link lengths as described above for the SC algorithm) from among all feasible minimum-hop paths. The M-SC algorithm was implemented as an approximation of the algorithm of Ahmadi et.al. [2], in an attempt to compare their routing algorithm with others. This was done by weighting the cost of a link sufficiently smaller than unity (which represents the weight given to the length of the link (unity). The algorithm thus always selects a minimum hop path if possible under the given loading and then selects a shortest cost path from among all possible minimum hop paths.

Another promising idea for routing is to identify a path with minimum total length (where link lengths are as described for the SC algorithm), as well as delay less than the specified bound. Unfortunately this problem is NP-complete [20]. The Min Max Cost with Delay Bound **(MMCDB)** algorithm is a heuristic approximation to this ideal algorithm. A path is found for which the maximum length of any link in the path is minimized, while also satisfying the end-to-end delay bound. A sketch of the algorithm is as follows. The links are first ordered by increasing cost; the link with the ith smallest cost is renamed link i. Using a dynamic programming approach, on the i'th iteration the shortest path (using delays as link lengths) from the source to the destination is found which uses only links from those numbered 1 through i. This path is checked to see whether it will meet the end-to-end delay bound according to the admission control method (Equations 2.2, 2.4, and 2.6 for EDD, Stop&Go, and WFQ, respectively). The algorithm terminates the first time a path is found which satisfies the delay bound (success), or when no path is found after the Lth iteration (failure), where L denotes the number of links in the network.

2.3.4 Optimal Routing

One approach to routing which was *not* investigated was optimal routing using either the flow-deviation or gradient projection techniques. [5] While it is undoubtedly useful to compare heuristics to an optimal method, this method is not suited for high-speed networks for the following reasons:

- There are no constraints on packet delays or losses; only average delay is minimized.
- Packets in a single call can be routed along different paths, which is inconsistent with the CAC schemes studied here, all of which assume a connection-oriented model.
- This method requires all calls be routed at the same time, in a batch. Old calls may have to be rerouted when new calls arrive. This seems to be an expensive proposition.

To our knowledge, no optimal routing method which includes delay and loss constraints and routes all packets of a call along the same path has been proposed.

2.3.5 Summary

The eight routing algorithms described in this section represent a wide range of approaches that might be used to route multimedia data. The static algorithm is clearly the least expensive to implement, because it is run one time only. All other algorithms are executed once for each call that requests admission. Most of these algorithms involve a shortest path computation, which can be executed in $O((L+N)*\lg N)$ time, where L denotes the number of links in the network, and N denotes the number of nodes. The computation for algorithm MMCDB takes $O((L+N)*N)$ time in the worst case. Since CB has the potential to try all paths in the network that start from the source node, it could be very expensive (factorial in the number of links!) to execute in the case of a highly-connected network. However, it has been found practical for use on real telephone networks.

Any of the above algorithms should be practical to implement (in terms of running time) for networks with no more than a few thousand nodes. A multimedia call will frequently last for quite some time; for example, a video conference may last for an hour or more. The time to find a good route with any of these algorithms is insignificant relative to this call duration. For truly huge networks, or when the routing and call setup time must be minimized, a static or hierarchical routing approach will be preferable (even though the quality of the results will be lower). Another possible time saving technique is to perform call setup at the same time as the route is being selected; the CB algorithm is an example of such an approach.

Having described the algorithms, the evaluation procedure is now discussed.

2.4 Experimental Method

A series of experiments were run to determine the "goodness" of the above routing algorithms for multimedia traffic with stringent quality-of-service requirements. This allowed us to model the network much more accurately than analytical models would permit, at the expense of requiring much more CPU time to compute. The simulations were performed in a static environment with the assumption of infinite call holding times and negligible call processing times. An important point to make is that only the call admission process was simulated. Because the loss and delay characteristics of deterministic methods are completely predictable, it was not necessary to simulate the transmission of individual packets. In all cases confidence intervals were calculated for the 95% confidence level, after running the same experiment 20 times using different numbers output by the same random number generator. These confidence intervals are noted in each graph and table.

The figure of merit used to evaluate the routing algorithms was the call acceptance probability. As noted in the introduction, the service provider is interested in "profit", which increases (although perhaps not linearly) with the number of calls accepted. Maximizing the acceptance probability is also important to network users. In the experiments, the acceptance probability was calculated as the (number of calls accepted) / (number of calls requested) over the last R requests; this gives a probability at one discrete point, rather than a cumulative probability over all calls. In our experiments, R was set to 300. This acceptance probability is measured as a function of the load in the network, which is the accepted load rather than the offered (sum of accepted plus rejected) load. Load is measured as the sum of the peak bandwidths of accepted channels.

There are a number of reasons why profit may not be simply a function of the number of calls accepted. For instance, calls requiring only one or two links may generate a higher profit (and may be preferred) to calls requiring more links. Another plausible example is that calls requiring more bandwidth may generate more profit than calls requiring less. Several experiments (not shown) were run using a figure of merit which assigned a higher "value" to calls needing a longer path and a higher bandwidth. The relative performance of

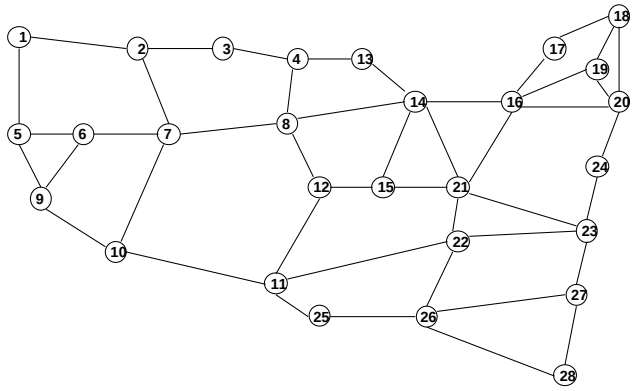

Figure 2.2: Graph of network used in the experiments.

the routers was unchanged for this alternative figure of merit, which gives us some confidence in the use of call acceptance probability for comparison purposes.

A good routing algorithm should also be fair, as well as produce high utilizations. Users will not be satisfied with a policy that consistently favors certain classes of traffic, unless they can be convinced that such a policy is reasonable. A further discussion of this subject may be found in [21]. Both fairness and utilization properties will be examined in the analysis below of the experimental results.

Figure 2.2 shows the topology of the network used in the experiments. This network was previously used in work by Murakami [40] and Grover [24], and is loosely based on the Internet backbone in the U.S. The diameter of this network is 7 and the average node degree is 3.2. Link bandwidths were set uniformly to 155 Mb/s, except for one experiment noted below. The sum of propagation and processing delay at each link was chosen to be uniformly 3ms. Buffer sizes were assumed to be infinite, so no packet loss occurred inside the network.

Packet sizes were uniformly set to the ATM cell size of 53 bytes. Two standard Markov-modulated Rate Process type models were used to generate voice and video traffic (from [55] and [37] respectively). Figure 2.3 shows the model parameters used in the experiments. These models translate to a peak rate of 32 Kb/s for voice, and 11.7 Mb/s for video. The peak packet interarrival times (assuming 48 payload bytes per packet) were 12 ms for voice, and 33 μs for video. Note that a small amount of variability (on the order of 5%) was introduced into the source peak rates to more accurately reflect real behavior. That is to

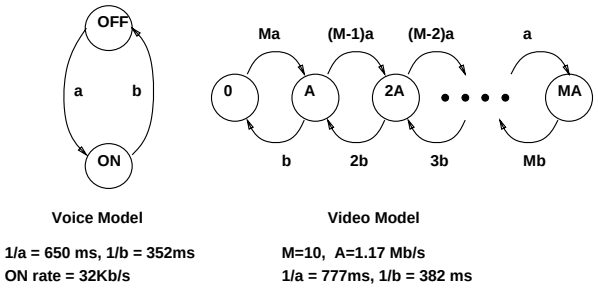

Figure 2.3: Voice and video models used in the experiments.

say, peak interarrival times for voice calls were evenly distributed between approximately 11.5 and 12.5 ms, while peak interarrival times for video were evenly distributed between approximately 32 and 34 μs. Allowable packet loss was set to 5% for voice and 1e-5 for video. Allowable end-to-end delay was assumed to be 350 ms, with the exception of one experiment described below. The higher delay figure is acceptable according to subjective studies, while a much shorter delay is required to eliminate the need for echo cancellation equipment, or where a more natural interaction is preferred.

Several details about our implementation of the admission control methods should be mentioned, as follows:

- For the EDD algorithm, the deadline on a link n was set to be a_n times the end-to-end deadline (i.e., 350ms except for one experiment). a_n was computed as the fraction of the minimum achievable end-to-end delay which was due to link n.
- No method of choosing frame sizes or allocating channels to frame sizes has been published for Stop&Go. The frame sizes were chosen to be 3ms, 6ms, 12ms, 24ms, 48ms, 96ms, and 192ms. This avoids bandwidth loss due to using frame sizes that don't have common divisors.
- For routing purposes, a channel was assigned to the maximum possible frame size under which its end-to-end delay requirement could be met, under the assumption that the channel would be routed on a minimum-hop path to its destination.
- For WFQ, the channel rate r_i was set equal to the peak arrival rate for the channel and the token bucket size was set to 1. There was no attempt to

	MMCDB	CB	SP	M_SC	SC	C_DYN	C_STAT
Path Lengths	3.34	3.06	2.83	2.72	2.76	2.47	2.22
Length/Min Hop	1.37	1.23	1.13	1.10	1.11	1.04	1.00
Link Peak Util.	.692	.668	.669	.656	.668	.593	.422

Table 2.1: Average path lengths of accepted channels, average ratio of path length to minimum-hop path length, and average link utilizations based on peak arrival rates. For homogeneous traffic (video only), EDD server, 155Mb/s link speeds. Measured at the 75% acceptance probability for each routing algorithm. Confidence intervals are no more than 1% for any number in this table.

discretize the channel rates into a set of allowable values for WFQ.

2.5 Experimental Results and Analysis

In this section the results of the simulation experiments are presented. The performance of the routing algorithms is investigated under different admission control policies and mixes of traffic. Note that a channel request can be blocked for any of the following reasons:

1. Bandwidth Rejection At least one link along the path does not have sufficient bandwidth available to accommodate the peak arrival rate of the channel.

2. Server Constraint Rejection For at least one link in the path, the necessary constraint imposed by the CAC function is not met.

3. Delay Bound Rejection The delay bound which can be achieved along this path violates the user-specified delay bounds for this channel.

2.5.1 Experiments Involving Homogeneous Traffic

As a base case for evaluating routers, the use of the EDD admission control policy was investigated for a network carrying only one kind of traffic (video). Figure 2.4 is a graph of the probability of call acceptance as a function of the accepted load in the network. Additional information about the performance of the routers is shown in Table 2.1.

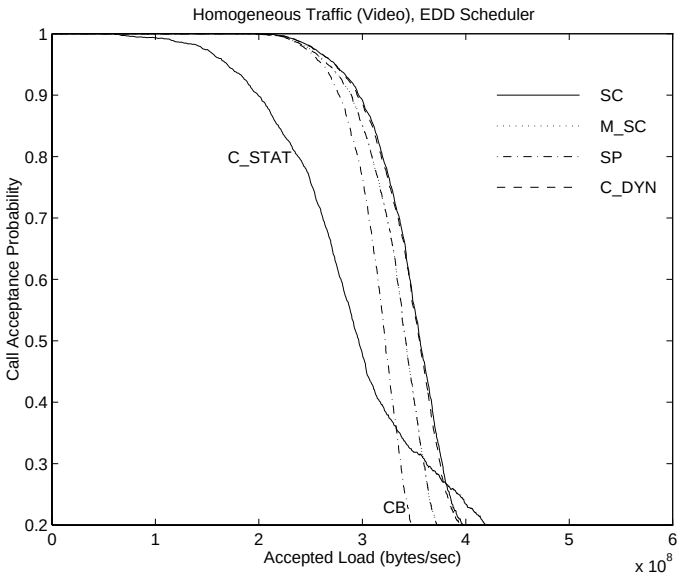

Figure 2.4: Call acceptance probabilities for homogeneous traffic (video only), EDD server, 155 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 2.7%.

For any acceptance probability above 60% (which is already lower than most users would consider tolerable), there is only a modest difference between the best and worst dynamic algorithms, while the static algorithm performs significantly worse, as expected. The best performing algorithm is the real-time algorithm SC, followed very closely by the C_DYN algorithm. The MMCDB algorithm is not plotted because its performance in this and other experiments was consistently the worst of the real-time algorithms. The reason can be seen in Table 2.1; the ratio of path length to minimum-hop path length is higher, and as a result link utilizations are higher. The CB algorithm, in this and other experiments, performs consistently worse than the real-time algorithms, outperforming only the C_STAT algorithm. The CB algorithm gains implementation robustness/simplicity by taking a much more local view of path optimization, and as a result does not perform particularly well. Note from the table that the C_DYN algorithm is almost a dynamic minimum-hop router, and as a result performs very well when delay bounds are easy to meet and the traffic is uniform.

An interesting phenomenon in this and other experiments is that at some point the conventional static algorithm appears to outperform the best of the real-time algorithms(!).

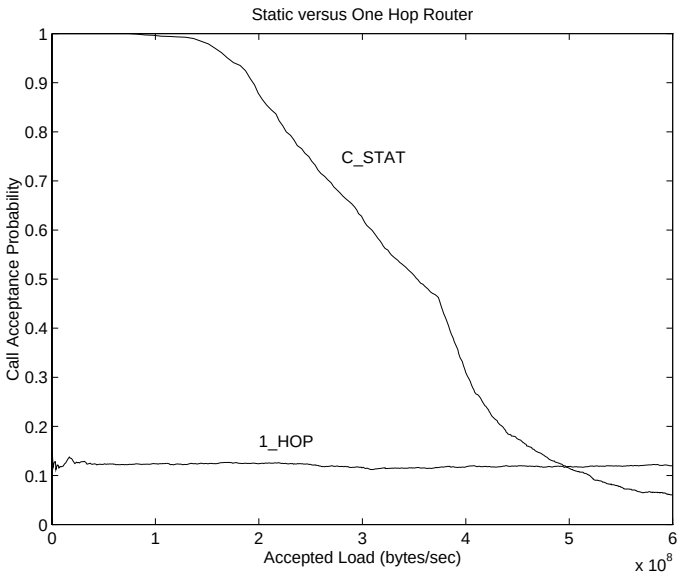

Figure 2.5: Call acceptance probabilities for homogeneous traffic (voice only), EDD server, 155 Mb/s links. 95% confidence intervals for any curve are no greater than 1.7%. (Illustration of tradeoff of fairness with utilization).

This conclusion is somewhat misguided for the following reason. Let $H_{i,j}$ be the number of hops on any minimum hop path between origin i and destination j. Table 2.1 shows that the conventional algorithms favor origin-destination pairs with significantly lower values of $H_{i,j}$. That is, channels which require a longer path (regardless of the routing algorithm) are being rejected at a significantly higher rate than channels which require shorter paths. When this behavior is continued for a long enough time, more channels will be accepted, since fewer resources are used for shorter paths (as shown clearly by the link utilization figures). Our conclusion is that the conventional algorithms (particularly C_STAT) are not as *fair* as the other algorithms.

Another illustration of this phenomenon is shown in Figure 2.5. A routing algorithm which only generates one-hop paths, and fails to find a path if the origin/destination nodes are not adjacent, is manifestly unfair to a large class of requesting calls. However, it clearly uses the minimum resources possible for each channel that is admitted, and at high network loads will surpass most other algorithms by our measure. This same observation has been made in [2]. To the categories of fairness identified in [21], one more that is appropriate

for broad band networks can be added: different types of calls should have roughly equal probability of acceptance.

The experiment shown in Figure 2.4 was repeated two times, once substituting the Stop&Go admission control policy for EDD, and once substituting the WFQ admission control policy for EDD. The graphs were almost identical to those for EDD, and are not shown. For this combination of traffic type + QoS requirements + network bandwidth, the choice of admission control policy makes little difference.

2.5.2 Experiments Involving Heterogeneous Traffic

The previous experiments indicate a modest benefit from the use of a real-time routing algorithm. This is consistent across the three admission control policies, as long as the traffic mix is very homogeneous. The power of real-time scheduling theory is the ability to discriminate between tasks or messages with differing timing requirements. To explore the effect of a traffic mix with widely varying requirements, experiments were run in which 80% of the traffic was voice, and 20% of the traffic was video. The peak rate of video is about 365$\times$ the peak rate of voice, which is an indication of the heterogeneity of the traffic mix.

Figure 2.6 shows the comparison of the routing algorithms when the admission control policy is EDD. Acceptance of a single video channel has a large impact on link utilization. Since packet sizes are assumed to be equal for all channels, the necessary constraint of EDD (Equation 2.1) dictates that no more than $(155 \times 10^6)/(11.7 \times 10^6) \approx 13$ channels can be accepted on any link for which at least one video channel is accepted. A link over which one video channel and only 12 voice channels are routed will obviously suffer a tremendous waste of bandwidth.[4] Under these more stringent traffic conditions the advantages of the real-time router SC is much more pronounced. A real-time routing algorithm such as SC can provide substantially better network utilization when the particular constraints of the admission control algorithm are so important (i.e., are difficult to satisfy). Note that SC is also much better than the other previously-proposed real-time routing algorithm, the M_SC algorithm.

[4] To be fair, we note again that the "simpler" version of EDD[19] had been implemented. The full version would not suffer such drastic waste of bandwidth for this condition. However, it imposes a constraint on the channels which is much more expensive to compute.

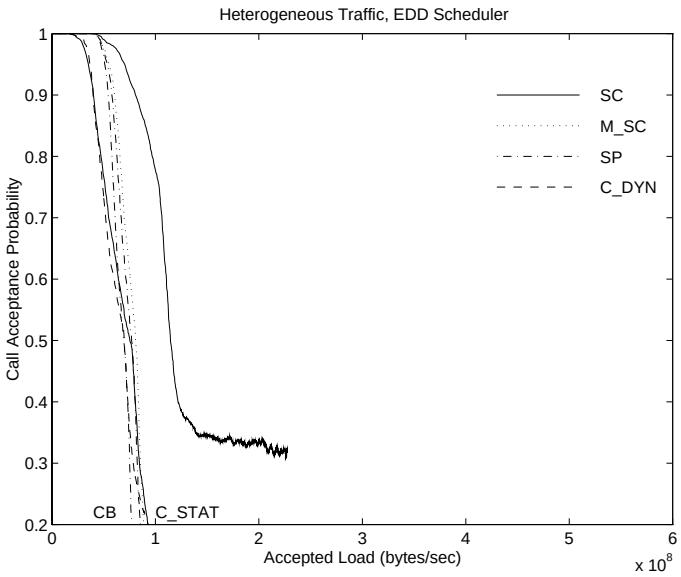

Figure 2.6: Call acceptance probabilities for heterogeneous traffic (20% video, 80% voice), EDD server, 155 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 5.3%.

Figure 2.7 shows the results of an experiment for the same network and traffic conditions, but using the Stop&Go admission control policy. The real-time routing algorithms perform in the same order as for the case of homogeneous traffic: SC best by a considerable margin, then M_SC and SP (indistinguishable from each other), and CB worst. Both of the conventional algorithms (C_DYN and C_STAT) cross over the real-time algorithms at higher loads, which can be attributed to the difference in fairness. The loads accepted by the Stop&Go policy are several times that accepted by the EDD policy. As mentioned, the necessary constraint of EDD (Equation 2.1) can lead to severe bandwidth waste for the heterogeneous traffic conditions that were simulated. There is no such waste for Stop&Go under these conditions. The Stop&Go necessary constraint (Equation 2.3) is almost a check that the sum of the peak bandwidths does not exceed the link bandwidth, under these conditions.

This experiment was run again using the WFQ admission control policy. The results are shown in Figure 2.8. While the relative rank of the routers for this experiment remains the same, the shape of the curves is somewhat peculiar. To understand this shape, refer to

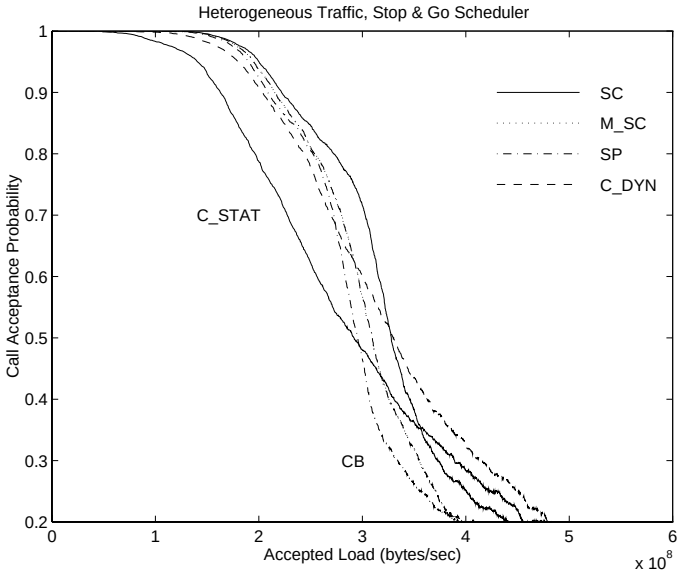

Figure 2.7: Call acceptance probabilities for heterogeneous traffic (20% video, 80% voice), Stop&Go server, 155 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 7.6%.

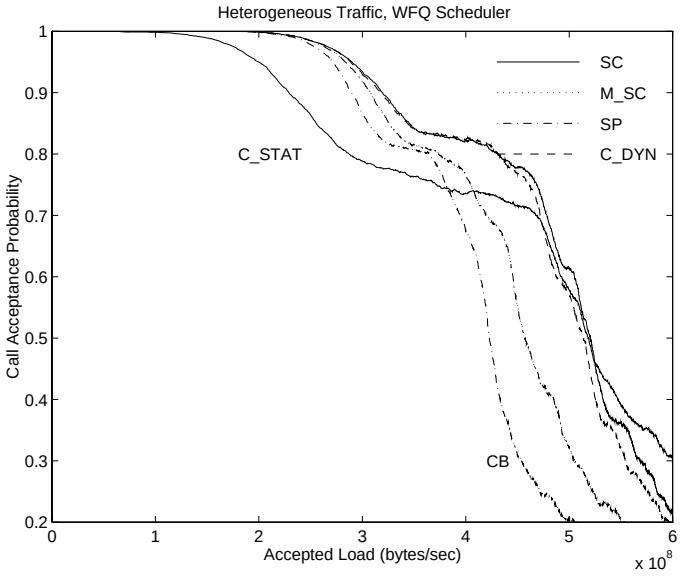

Figure 2.8: Call acceptance probabilities for heterogeneous traffic (20% video, 80% voice), WFQ server, 155 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 8.9%.

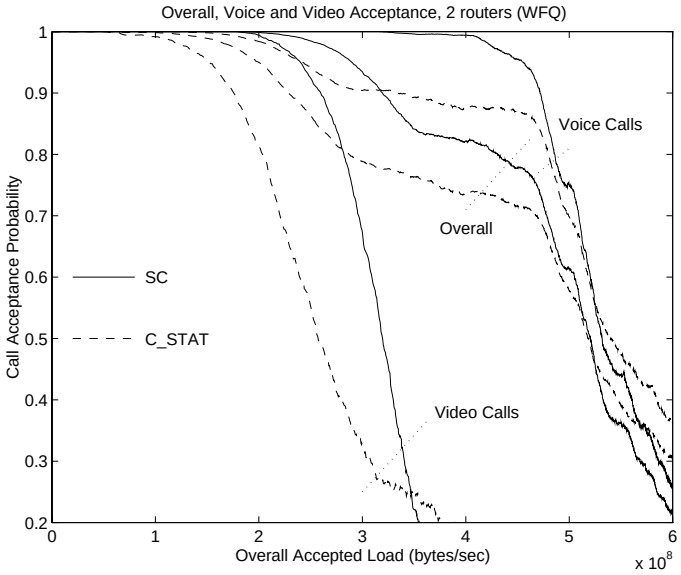

Figure 2.9: Call acceptance probabilities for heterogeneous traffic (20% video, 80% voice), WFQ server, 155 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 8.9%.

Figure 2.9. In this figure only two of the routing algorithms are plotted. The acceptance ratios of the two classes of traffic (voice and video) are graphed separately, as well as the combined total overall. Video calls are rejected while network utilization is still relatively low; significant rejections of voice calls occurs at a much higher load. When these two classes of traffic are combined, the resulting acceptance exhibits the expected first drop (video acceptance goes down to 0), followed by a leveling off (100% of voice is still being accepted), followed by the second drop (when voice acceptance starts falling).

The crossover of the C_STAT algorithm relative to the others can again be explained in terms of fairness (or lack of it). Routing a video channel over a long path consumes a lot of resources. An algorithm such as C_STAT (which favors channels requiring short paths) permits a much larger number of channels to be accepted at high network loads. However, rejections occur much earlier (particularly for video traffic) for the static algorithm, due to its inability to avoid local congestion.

The WFQ admission control policies accepts a significantly higher load than even the Stop&Go policy under these conditions. The primary reason is that a fixed set of packet

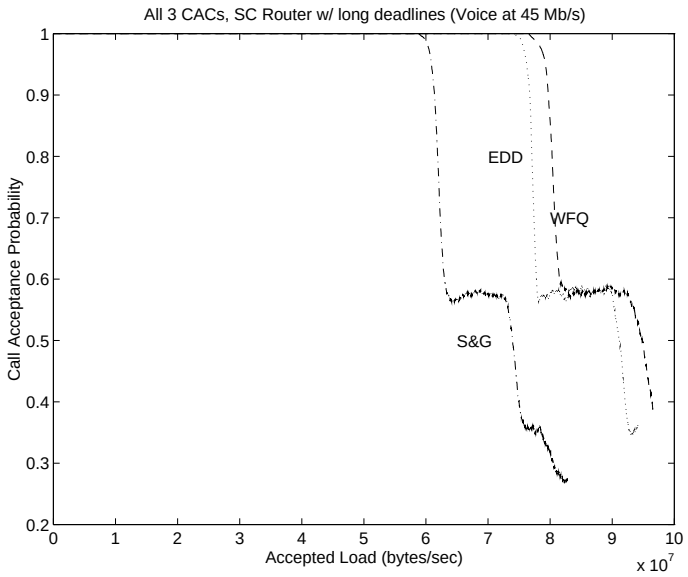

Figure 2.10: Comparison of admission control policies for loose end-to-end delay bounds, homogeneous traffic (voice only), SC routing algorithm, 45 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 6.7%.

transmission rates (frame sizes) was available with the Stop&Go policy. Calls that don't fit one of these rates exactly will cause rate overallocation, which is wasted bandwidth. Our implementation of WFQ placed no constraints on rates and allocated exactly the bandwidth needed by a call.

2.5.3 Other Experiments

In the experiments described above, the end-to-end delay bound for all calls was assumed to be 350 ms. These delay bounds are easy to meet for all traffic conditions/networks/admission control policy combinations found in those experiments. Another experiment was run for purposes of comparing the admission control algorithms under conditions where the delay bounds are much more stringent. This experiment simulated a network with 45 Mb/s links, voice traffic only, and one routing algorithm (SC), for each of the three admission control policies. The network was simulated once for calls with long delay bounds (350 ms), and once for calls with short delay bounds (varied uniformly between 20 and 50 ms). The results are shown in Figures 2.10 and 2.11.

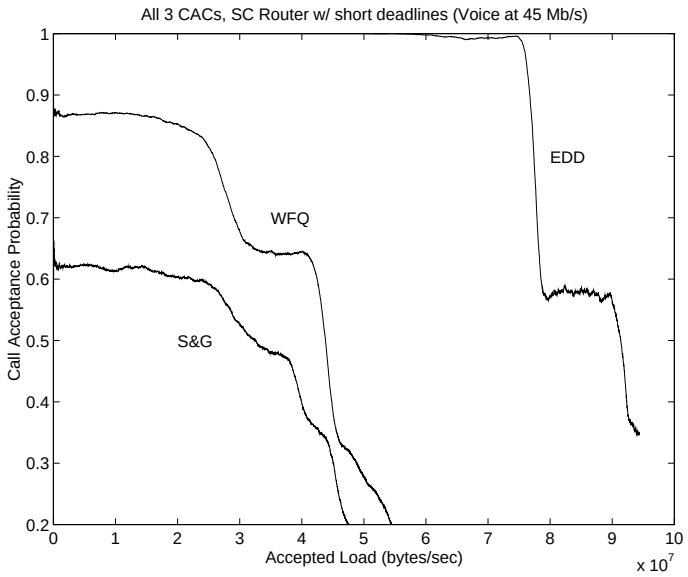

Figure 2.11: Comparison of admission control policies for stringent end-to-end delay bounds, homogeneous traffic (voice only), SC routing algorithm, 45 Mb/s link speeds. 95% confidence intervals for any curve are no greater than 8.8%.

Since this network has a diameter of 7 hops and each hop has a propagation delay of 3 ms, just the propagation time for some paths will be 20 ms or more. The delays due to framing (Stop&Go, Equation 2.4) and fair queuing (WFQ, Equation 2.6) for voice at 45 Mb/s are substantial, and must be added to this. Achieving the shorter delay bounds is therefore rather challenging.

For the given conditions, the behavior of the EDD policy is essentially unaffected by the shorter delay bounds; EDD is considered to be flexible in its ability to cope with stringent delay bounds. The Stop&Go policy, on the other hand, suffers a substantial drop in accepted load. An examination of the delay bound condition for Stop&Go (Equation 2.4) indicates smaller end-to-end delays can only be achieved by assigning channels to smaller frame sizes. Reserving one packet slot every T_g seconds for a channel in which peak packet interarrival time is greater than T_g naturally implies bandwidth is wasted, leading to the lower accepted loads. WFQ shares with Stop&Go a dependence between the allocated rate and achievable delay bound, as indicated by Equation 2.6. To achieve lower delays with WFQ also requires overallocating the bandwidth assigned to a channel. Our implementations of WFQ and

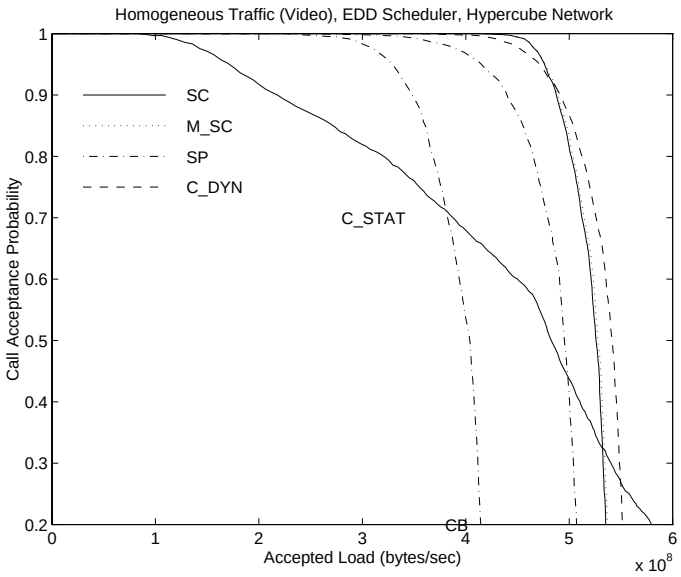

Figure 2.12: Call acceptance probabilities for a hypercube network topology, with homogeneous traffic (video only), EDD admission control method, and 155 Mb/s link speeds. 95% confidence intervals for any curve are less than 1.5%.

Stop&Go limited the rate assigned to a channel to be no more than a maximum amount. As a result, channels for which the required delay bound can only be met by allocating more than the maximum rate are simply rejected. This leads to immediate rejections even for an unloaded network. It also tends to reject channels requiring longer paths (since delay is proportional to hop-count, by Equation 2.6), which is a form of unfairness.

All of the experiments discussed above were conducted on the network shown in Figure 2.2. It is possible that network topology strongly influences the behavior of the routers. Accordingly, an experiment was conducted on a network whose topology was a four-dimensional hypercube. A hypercube is a completely symmetric network with a high degree of connectivity between the nodes. The network link speeds were 155 Mb/s and the EDD admission control policy was used. The results are shown in Figure 2.12. The results can be compared with Figure 2.4, for identical conditions except the difference in network topology.

The differences in router performance are more substantial for the hypercube network. This is because in a highly-connected network there are more opportunities for optimization, justifying the use of a more intelligent router. The relative ranking of the algorithms

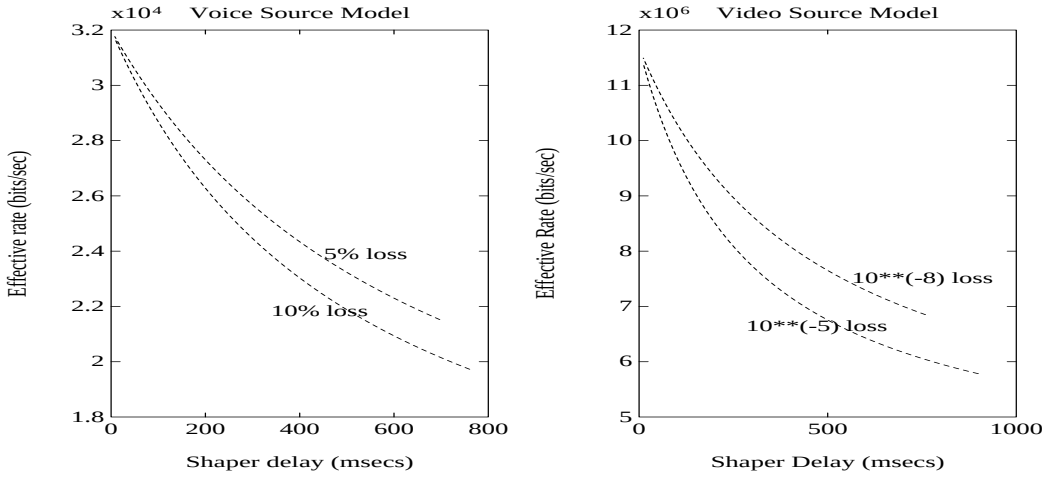

Figure 2.13: Effective bandwidth vs. shaper-induced delay for voice (peak = 32Kb/s, mean = 11.24Kb/s) and video (peak = 11.7Mb/s, mean = 3.85Mb/s) source models.

(except for C_DYN) remains the same; algorithm SC does particularly well in this network. The steep fall-off in acceptance probability is due to the symmetric nature of the hypercube. In such a network, a load-balancing routing algorithm results in all links filling up at essentially the same moment. C_DYN does particularly well for this network topology. Under these conditions (uniform network bandwidth and topology, loose delay bounds, homogeneous traffic characteristics), a dynamic algorithm based solely on utilization compares very favorably with real-time algorithms.

Section 2.2 mentioned that a common criticism of deterministic admission control/ multiplexing methods is the low average utilization they can achieve. This is directly due to the high peak/mean ratio of multimedia traffic sources, such as compressed video and voice. Traffic shaping reduces this peak/mean ratio, which should lead to improved utilization even while preserving strong guarantees on quality-of-service. For the voice and video models used in this study, the effective bandwidth computations yield the curves shown in Figure 2.13.

An experiment was conducted to investigate this hypothesis. This experiment was run for the original network of Figure 2.2, with 155 Mb/s link speeds, the Stop&Go admission control algorithm, and video traffic only, with traffic shaping of the video sources. A simple calculation indicates that no channel should incur an end-to-end delay in the network of

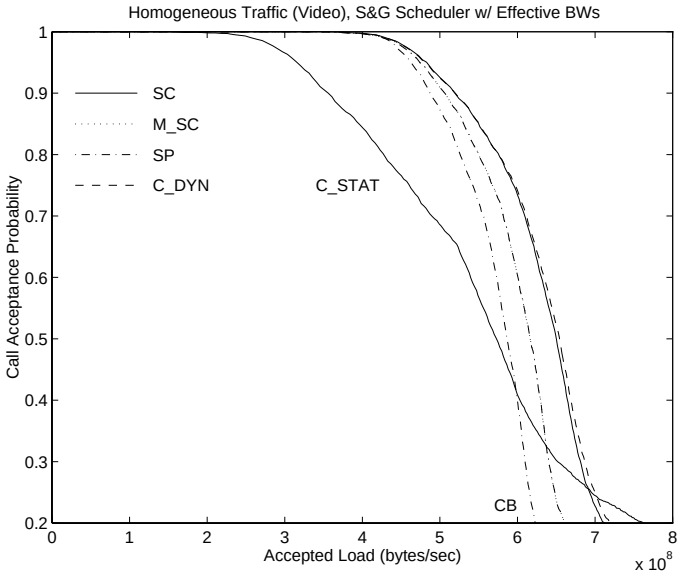

Figure 2.14: Call acceptance probabilities for homogeneous traffic (video) with traffic shaping, Stop&Go admission control, and 155 Mb/s link speeds. 95% confidence intervals for any curve are less than 2.4%.

	EDD	**Stop&Go**	**Stop&Go with Shaping**	**WFQ**
Peak Traffic Rates	.637	.676	.690	.687
Average Traffic Rates	.212	.226	.382	.230

Table 2.2: Average link utilizations for peak and average source traffic rates, for homogeneous traffic (video only), the SC routing algorithm, and 155 Mb/s link speeds. Measured at a call acceptance probability of 75%. 95% confidence intervals for all numbers are less than 1.5%.

greater than 50 ms for these conditions, leaving 300 ms (out of the specified delay bound of 350 ms) for traffic shaping. From Figure 2.13, the effective bandwidth for a video source can be decreased by more than 25% for a shaper delay of 300 ms and acceptable loss of 10^{-5}. Figure 2.14 shows the results of this experiment, and should be compared with Figure 2.4. The use of shaping substantially improves the absolute performance of all routers, while not affecting their relative performance. Table 2.2 shows the increase in average network utilization which is achieved for one specific acceptance probability. The use of traffic shaping improves average utilization by almost 70% for this specific combination of network conditions and admission control and routing policies.

2.6 Summary

The performance of several routing algorithms and three admission control policies has been evaluated for multimedia traffic in packet-switched networks. The study included routing algorithms which have been suggested or proposed as being well-suited for this purpose, and a number of new algorithms as well. The new routing algorithms specifically target the real-time requirements of multimedia. The algorithms have been compared on their ability to accept new calls at a given network load. This is the first comprehensive study of real-time routing and deterministic admission control algorithms for voice and video traffic. It is also the first study of the close relationship between routing and admission control.

It has been found that a dynamic real-time routing algorithm (SC) which addresses the constraints of the admission control policy performs best overall. The benefits of this router are greatest when quality-of-service requirements are difficult to meet, the network is highly connected, and admission control constraints are stringent. A conventional dynamic algorithm based solely on utilization also performs reasonably well, although it is not as fair to channels requiring long paths. Sequential routing algorithms (as used in circuit-switched networks such as telephone networks) perform noticeably worse than shortest-path algorithms.

The relative and absolute performance of the admission control policies depended very much on bandwidth quantization effects, end-to-end delay requirements, and the traffic mix. Careful attention should be given to these factors in selecting and implementing any one of these policies, or low utilizations will result. The use of traffic shaping, if allowable under the user-specified bounds on delay, can significantly improve the link utilizations achieved by deterministic admission control / multiplexing policies.

The achievable improvement in utilization with the use of a traffic shaper at the input of a network is limited by the tolerable end-to-end delay. In the next chapter, this technique is extended and an architecture developed in which link bandwidth utilization is improved even further. This is done by exploiting statistical multiplexing in addition to traffic shaping, resulting in higher efficiency.

Chapter 3

Path Level Bandwidth Reservation for End-to-end QoS Guarantees

3.1 Introduction

In this chapter, an architecture is developed for packet/cell switched networks such as ATM, in which strong guarantees on end-to-end QoS can be provided to users. In addition, link utilizations are improved over conventional techniques. In chapter 2, it has been shown that schemes such as EDD, Stop&Go and WFQ as proposed, result in link utilizations of no more than about 25% or so, with typical multimedia sources. This is because these rely on peak bandwidth allocation to ensure no cell loss and guarantee end-to-end delays. For typical sources, the peak rates can be several times the mean rates, leading to poor utilization.

In chapter 2, the use of traffic shapers at the edge of the network has been shown to improve link utilizations as compared to direct peak bandwidth allocation. The traffic shaper uses the "slack" in allowable end-to-end delay to reduce the peak rate of the source using buffering. Strong guarantees on end-to-end delay and loss probability can still be provided because the deterministic scheduling scheme provides delay guarantees and ensures no cell loss, while the shaper rate at the source can be designed to guarantee the end-to-end cell loss probability.

In this chapter, this notion is extended further and an architecture is developed in which strong end-to-end guarantees can still be provided, and utilization is improved even further. Essentially, deterministic bandwidth enforcement is performed on groups of channels rather

than individual channels. The two basic techniques to obtain high utilization in the presence of bursty traffic are Statistical Multiplexing and Traffic Shaping. The use of traffic shapers at the edge as in the previous chapter does not exploit statistical multiplexing between sources which limits the improvement in utilization. The architecture developed in this chapter on the other hand exploits both these techniques resulting in even better utilizations. The ability to provide strong end-to-end QoS guarantees is not compromised as shown later.

Figure 3.1 demonstrates these two basic techniques for improving utilizations. The waveforms shown represent the instantaneous arrival rates of bursty sources. The traffic shaping technique relies on the use of a buffer to smooth out bursts of arrivals. As shown, the buffer can be serviced at a rate lower than the peak rate of the arrival process so that the queue can operate at high utilization. Statistical multiplexing relies on the independence of different bursty streams. The probability of several sources transmitting at their peak rate simultaneously is small. Hence the bandwidth allocated to a group of sources can be smaller than the sum of the peak bandwidths of individual sources. The utilization achievable using techniques which rely only on traffic shaping are limited by the tolerable delay since the smoothing process introduces additional delay. This has been seen earlier in chapter 2 also. Techniques which rely on statistical multiplexing on the other hand encounter difficulties for predicting the end-to-end performance of individual streams since the multiplexing operation can alter the traffic characteristics of individual streams in a way which is difficult to quantify in general.

The architecture proposed in this chapter exploits both these techniques in order to obtain high bandwidth utilization.

There has not been much work towards improving the utilization of the schemes evaluated in chapter 2, while retaining their ability to provide end-to-end QoS guarantees. In [34], the authors propose the Deterministic Bounding Interval-Dependent traffic model. Instead of using the peak rate, the rate averaged over an interval of time is used for bandwidth allocation. Increased utilization is obtained because the averaged rate can be significantly lower than the peak rate. Conceptually, this technique is similar to the traffic shaping technique introduced at end of the previous chapter since peak rate reduction is obtained by averaging the arrivals over sufficiently long periods of time. However, this approach requires a complex source characterization which is non-standard to typical traffic descriptors (viz.

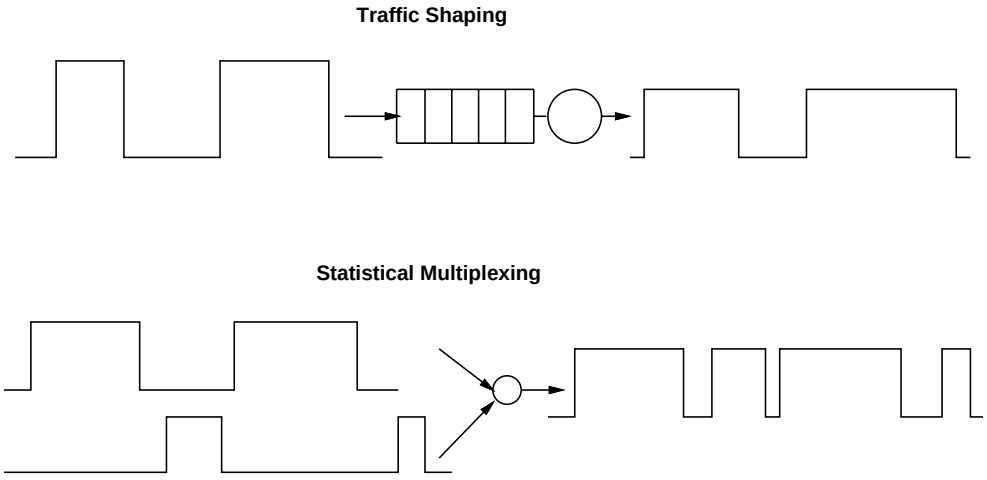

Figure 3.1: The two basic techniques for obtaining high utilization in the presence of bursty traffic

the rate-interval curve [34]). Further, the gains achievable by reducing the rate of individual sources are limited by allowable delay as in the shaper-based scheme of chapter 2. The approach outlined in this chapter exploits statistical multiplexing in addition to traffic shaping resulting in even better utilization. Additional comments on relevant literature are presented at the appropriate points in the chapter.

Section 3.2 develops the rationale behind the suggested approach by demonstrating some important properties of QoS measures such as cell loss in a multi-hop environment. The difference between providing QoS guarantees in a multi-node environment versus in a single node case is brought out to motivate the approach taken in this chapter for providing multi-node QoS guarantees. The Virtual Path (VP) level bandwidth reservations scheme is introduced in section 3.3. Section 3.4 then analyzes the problem of distributing the resources reserved for a VP over the different physical links traversed by it. Using a queuing model of a VP in the proposed architecture, some advantages of a certain technique (the 'MGF' policy) are brought out. This technique reserves resources such that cell loss occurs only at the first physical link traversed by a VP. In the next section, a suggested implementation of this scheme using a Weighted Round Robin server is outlined. Motivated by the results from the previous section, cell loss is allowed to occur only at the first physical link of a VP. Some implementation issues are discussed, including the use of cell spacers to implement this

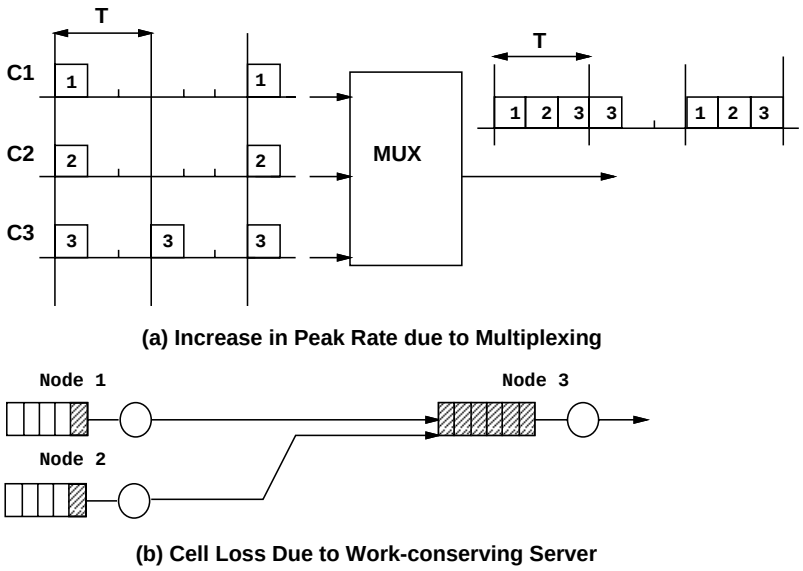

Figure 3.2: Problems in determining QoS in the Multi-hop case

architecture and the use of buffer sharing techniques for improved performance. Section3.6 summarizes the results of this chapter.

3.2 The Multiple Hop QoS Problem : Some Insights

In this section, some basic properties of QoS measures such as cell loss in the multiple hop environment are demonstrated. This helps motivate the development of the scheme based on Virtual Path level bandwidth reservations outlined in the following section.

3.2.1 Some Counter Intuitive Properties of Cell Loss over Multiple Nodes

Approximate schemes for providing QoS guarantees in the presence of statistical multiplexing exist for a single node (e.g. equivalent capacity formulas [25]). However, unlike some the schemes like WFQ which provide end-to-end guarantees, such results cannot be directly extended to the multi-node case.

Figure 3.2 illustrates the difficulty of extending admission control based on statistical multiplexing to the multi-hop case. In Figure 3.2a (from [36]), connection 3's minimum inter-cell arrival time changes from 3 slots to 1 slot (thereby changing its Peak Cell Rate

or PCR), because of the work-conserving multiplexor (*Note:* a server is work-conserving if it does not idle as long as there is a cell waiting for transmission). The PCR is one of the most important traffic descriptors of a source [1]. This example shows that the source model at the edge of the network is not valid in general in the interior of the network. Statistical resource allocation techniques are strongly dependent on statistical models of traffic sources, hence cannot be accurately applied on an end-to-end path without excessive numerical analysis.

For a single node, it is intuitive that when all cells require the same amount of service, a work conserving cell transmission policy minimizes the total losses at a node. However, Figure 3.2b shows that in the multi-hop case, work conserving servers can cause loss of cells where non-work-conserving servers would not have done so. Hence, unlike the single node case, use of work conserving service policies is not optimal for minimizing total losses in the multiple node case.

Cruz [14] has shown that with simple work-conserving FIFO, as loss tolerance becomes 0, the downstream buffer requirement grows exponentially with the number of hops (H) in the path. Equivalently, if smaller buffers are used, more than the peak bandwidth may need to be allocated to ensure zero loss. In comparison, the use of a non-work-conserving scheduler can always guarantee zero losses, while requiring only $O(H)$ buffer requirements and peak bandwidth allocation [23]. The difference between the single node and multiple node case is further illustrated by the following theorem.

Theorem 1 *For a single node with constant length cells, if the service rate is increased, total losses can only decrease for any arbitrarily fixed sample path of arrivals. However in case of a network of such nodes, total losses can* increase *if the service rate of a node is increased.*

Proof: A sample path proof of the first part is in appendix A. The second statement can be simply illustrated using an example. Figure 3.3 shows a two hop network with a two units of buffer space at queue 1 (including the space for the cell being served) and one unit at queue 2, in which overall losses increase when the service rate at queue 1 is increased. □

A similar result exists for the relation between losses and buffer size.

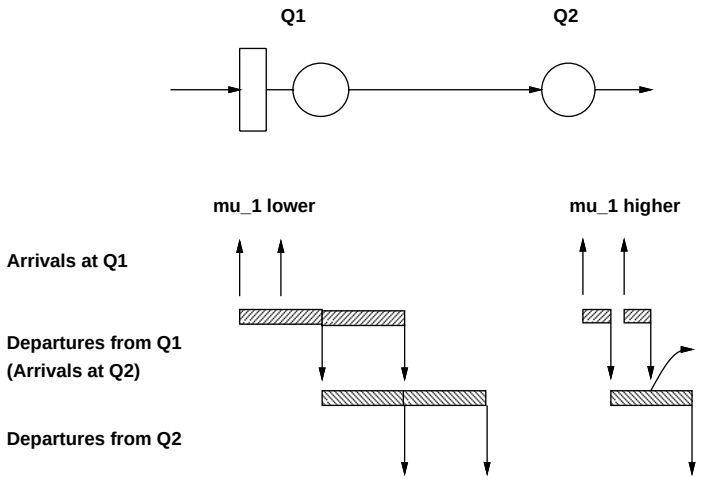

Figure 3.3: Network loss rate can increase if service rate of a queue is increased

Theorem 2 *For a single node with constant length cells, if the queuing buffer size is increased, total losses can only decrease for any arbitrarily fixed sample path of arrivals. However in case of a network of such nodes, total losses can* increase *if the buffer size of a node is increased.*

Proof: A sample path proof of the first part is in appendix B. The second part is again illustrated using an example. In Figures 3.4 and 3.5, the buffer size at hop 2 is increased from 2 units to 3 units (including the space for the cell in service), but still overall losses increase for the given sample path of arrivals! □

These results have shown that simple monotonic relations between the amount of resource required and the QoS seen by a source do not hold as one moves from the single node to the multiple node case. Without proper control of network resources (say bandwidths), one can end up increasing resource allocations to a source and still *degrading* its QoS. As a consequence, algorithms which obtain resource allocations levels on the basis of a gradient driven search, may not result in optimal solutions. This is because several locally optimal solutions can exist for obtaining the optimal resource requirements needed to meet specified QoS requirements.

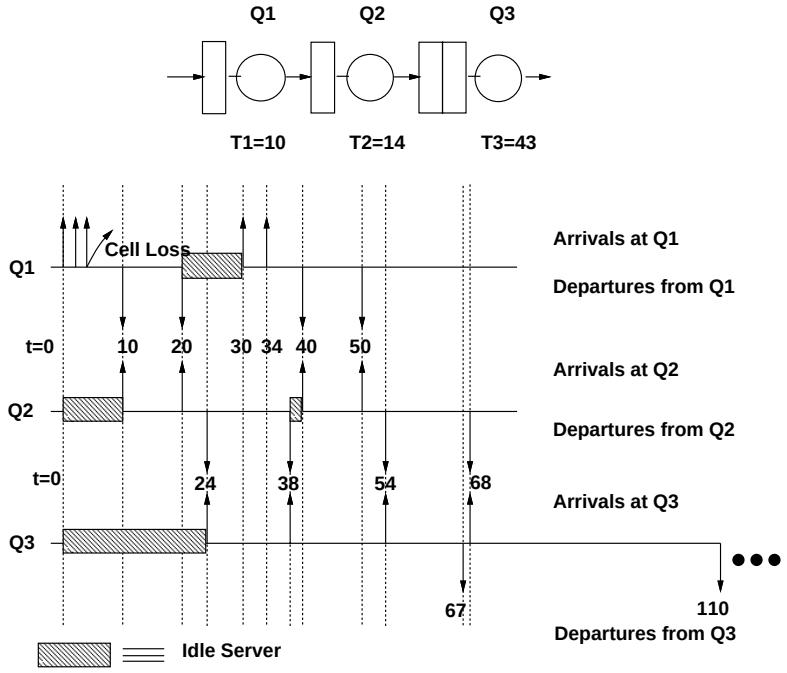

Figure 3.4: Overall Losses can increase with increase in Buffer size - Part I (B1=2)

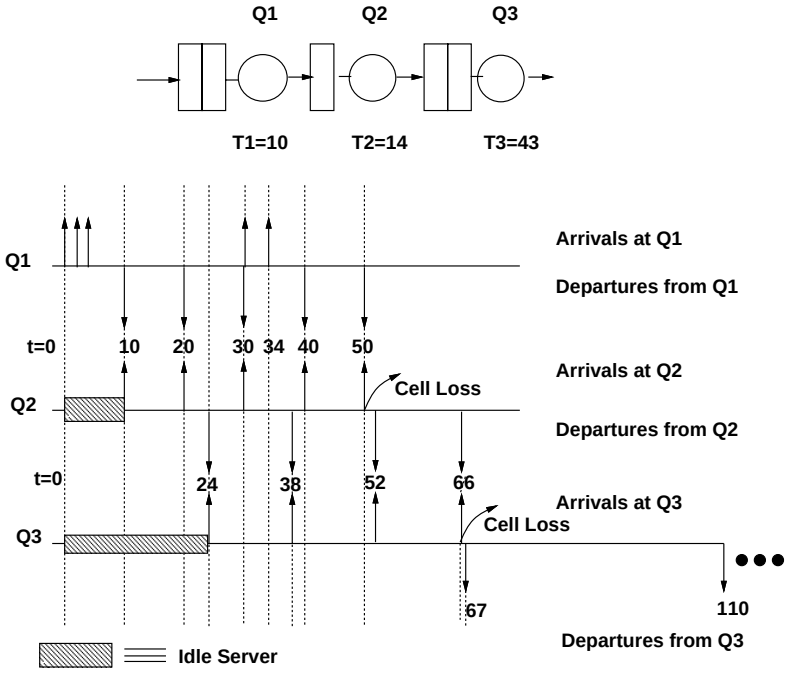

Figure 3.5: Overall Losses can increase with increase in Buffer Size - Part II (B1=3)

3.2.2 Bandwidth Sharing versus Bandwidth Partitioning

The relative advantages and disadvantages of statistical multiplexing vis-a-vis segregation of available bandwidth for different sources are discussed in this section. Several studies on statistical multiplexing in ATM networks have indicated that when using a simple FIFO service policy, either bandwidth sharing or bandwidth partitioning can lead to lower total bandwidth requirement for a set of sources depending on different scenarios of relative burstiness, QoS requirements, arrival rates of the sources being multiplexed and the cell scheduling policy used. See for example [10], [4], [6].

Additionally, the gains achieved by statistical bandwidth sharing typically diminish as the multiplexing level is increased. For instance Figure 3.6 shows the total bandwidth requirement for a set of 200 voice sources at different partitioning levels; an approximation based on Gaussian approximation of the distribution of the aggregate input rate (from [25]) is used for this plot. [1] For each partition size, this approximation is used to calculate the bandwidth requirement of each partition and the total requirement is obtained as the linear sum of individual partition bandwidths.

It can be seen that increasing the size of the partitions results in bandwidth savings initially, but once there are more than about 30 or 40 sources per partition, the total bandwidth requirement does not change very much. The bandwidth requirement for 4 partitions of 50 sources is only about 10% more than the full sharing case. Similar behavior can be expected with other types of sources. Hence if bandwidth partitioning is performed at a sufficiently "coarse" level, the loss of utilization as compared to a full sharing scheme can be quite small.

Deterministic bandwidth partitioning (i.e. a strict TDM-like enforcement of rate) also has other advantages as compared to full sharing. De Veciana [51] has shown that serving a source at a deterministic rate (as in a deterministic bandwidth partitioning scheme) results in minimal burstiness of the output traffic for a given average service rate, and consequently the effective bandwidth of the source at downstream nodes is minimized. Equivalently, in [49], it is shown that for a given average service rate, the smoothing of a source is maximal

[1] This approximation uses the probability of the instantaneous input rate exceeding the link rate as an measure of the loss probability. The central limit theorem of statistics [54] is invoked to characterize the aggregate bit rate as having a Gaussian distribution. The standard voice model, as in Figure 2.3 has been used here.

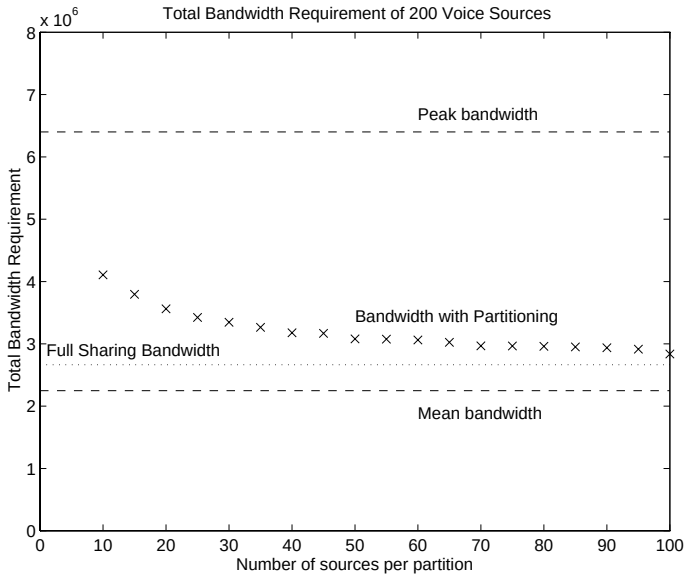

Figure 3.6: Illustration of diminishing returns from statistical multiplexing. Total bandwidth requirement for 200 voice sources at different partitioning levels.

when it is serviced in a deterministic manner as is done in a partitioned bandwidth scenario. The phenomenon of traffic smoothing can be quantitatively illustrated using the squared coefficient of variation of the inter-departure times of an $M/D/1$ queue, which equals $1-\rho^2$, where ρ is the utilization. At a utilization of 90%, this equals 0.19, so that the output process may be approximated as a CBR process (a squared coefficient of variation in inter-arrival times of less than 0.2 is normally approximated to be a deterministic process). Finally, several common bandwidth allocation algorithms (such as equivalent capacity [25]) anyway perform bandwidth allocation by linear addition of bandwidth requirements of individual sources (such as in a partitioned bandwidth model), even when bandwidth sharing is being employed. Thus the achievable utilization in a partitioned scheme may be comparable to that with a fully shared scheme employing linear bandwidth allocation.

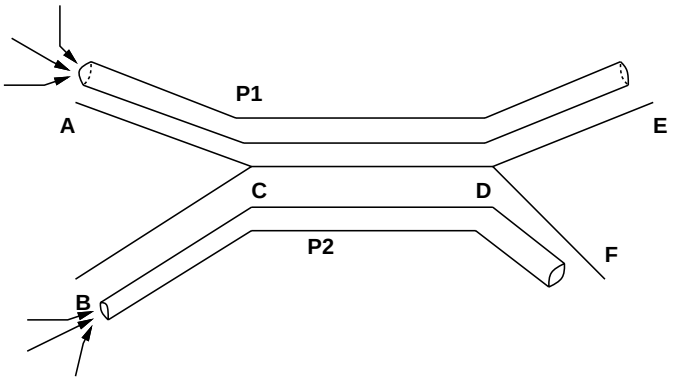

Figure 3.7: Illustration of virtual path level bandwidth reservations

3.3 An Architecture Based on Virtual Path Level Bandwidth Reservation

In the earlier section, the relative advantages and disadvantages of bandwidth partitioning versus sharing have been examined with respect to achievable utilization, and end-to-end QoS predictability. It has been shown that with work-conserving service disciplines, making optimal end-to-end QoS guarantees is much more difficult in the multiple node case than in the single node case.

In this section, an architecture for providing end-to-end QoS guarantees is introduced motivated by some of the properties of QoS over multiple hops discussed in the previous section. In the proposed architecture, strict bandwidth partitioning is performed at the Virtual Path (VP) level i.e. the bandwidth assigned to a VP is deterministically enforced. All channels (VCs) which use a VP however, can freely use this bandwidth by employing statistical sharing. This is illustrated in Figure 3.7. Here, two virtual paths (P1 and P2) are shown for which bandwidth is deterministically reserved and enforced in the network. Thus these VPs are isolated from each other even at link CD which is common to them.

Because, each VP is isolated from all other VPs, and since all VCs within a VP traverse the same path, end-to-end QoS guarantees can be provided without getting affected by cross-traffic. In general, a VC may traverse several VPs from its origin to destination nodes. However, this thesis concentrates on analyzing the case where a VC traverses a single VP.

Analysis of a queuing model of a single VP in such an architecture (reported in the following section) shows that all the cell loss and a large fraction of the end-to-end queuing delay occurs at the first hop of the VP. Hence to determine the end-to-end loss only the losses at the first link of a VP need be analyzed. For a single queue, problems such as those outlined in section 3.2 are not encountered, so conventional single node techniques can be used to provide end-to-end QoS guarantees. This is a major advantage of this approach in being able to provide end-to-end guarantees on cell loss while exploiting statistical multiplexing of sources.

It may be seen that this architecture employs a preventive form of congestion control since the bandwidth assignments to VPs are strictly enforced. Other approaches for end-to-end control include feedback-based approaches (e.g.[18]). Such schemes employ reactive congestion control. The performance achieved by such schemes typically depends on the delay-bandwidth characteristics of a network and the end-to-end performance is not guaranteed but rather, best-effort. In contrast, the proposed architecture provides a guaranteed end-to-end transport mechanism irrespective of physical network characteristics such as propagation delays and link transmission speeds.

As outlined in section 3.1, the two basic techniques for improving utilization in the presence of bursty traffic are traffic shaping and statistical multiplexing. The proposed approach exploits both these strategies in order to obtain high utilization. Statistical multiplexing is exploited by multiplexing several VCs onto the same VP. Each VC is allowed to fully use the bandwidth allocated to a VP. The actual bandwidth enforcement is performed only at the VP level. The burstiness of the aggregated traffic is expected to be no more than that of the individual streams. Further, traffic shaping is exploited on the aggregated traffic by using an appropriately sized buffer at the first node of a VP. This further reduces the burstiness of the aggregated traffic.

Since the proposed architecture exploits both the techniques mentioned, the bandwidth utilization achieved can be significantly better than schemes for end-to-end QoS which require peak bandwidth allocation e.g. WFQ [44] or which exloit only the traffic shaping technique e.g. D-BIND [34] and the technique presented earlier in chapter 2. This scheme is not claimed to be optimal in terms of providing end-to-end guarantees with the best possible resource usage. However, the utilizations achieved are expected to be higher than

existing schemes as mentioned above.

As outlined earlier, one disadvantage of employing statistical multiplexing is that the ability to characterize the performance seen by individual streams is lost. This is true even in the proposed architecture. Consequently, the performance levels (i.e. QoS) attained by a VP must conform to the worst case QoS specified from among any of the VCs using this VP. This is outlined in a latter section where a call admission procedure for a VC in the proposed architecture is outlined. An approach (not analyzed the thesis) which can improve performance over this worst case approach, is to use a priority scheme between the VCs in a single VP. Since, end-to-end analysis is reduced to analysis of a single node in this case, and several results on optimal priority queuing for a single node exist in literature (see for example [31]), these schemes can be incorporated into the proposed architecture and their results extended to give end-to-end guarantees.

3.3.1 Multiplexing Potential of Real-life Sources

In the proposed architecture, statistical multiplexing is exploited only at the first hop since the peak arrival rate at downstream hops is never more than the reserved bandwidth. These plots indicate that with typical sources, a single multiplexing operation can result in a process with a sufficiently low ratio of peak to mean rates that it can be approximated as a CBR process requiring a deterministic bandwidth reservation.

Figures 3.8 and 3.9 show the per-source effective bandwidth requirement for a multiplexed set of voice sources and a multiplexed set of video sources, respectively. These curves are obtained by using the equivalent capacity approximation [25], [17] for a loss probability of 1%. The delay shown in the curves is the maximum queuing delay that would be encountered in the first hop of the VP onto which the sources are multiplexed. The voice and video models used are two standard markov modulated fluid type models ([50] [37]).

From these curves the multiplexing potential of both video and voice sources can be clearly seen. For example, for 20 voice sources and an allowable queuing delay of 100 msecs the per-source effective rate is about 16 Kb/s; this represents an average utilization of 70%. For 20 video sources and an allowable delay of 100 msecs, the utilization is even better (almost 90%). For 20 voice sources and 200 msecs of delay, 80% utilization can be achieved. Typical end-to-end acceptable delay limits are up to 250-300 msecs (see Table 1.1). The

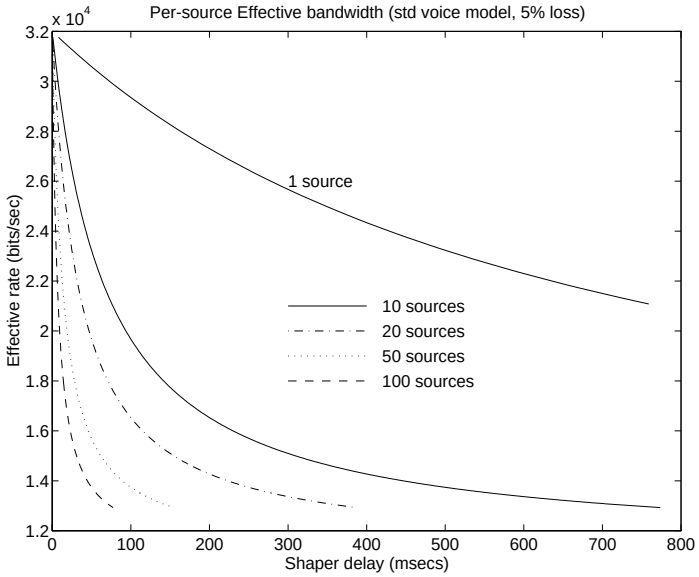

Figure 3.8: Per-source effective bandwidth requirement for multiplexed voice sources (Fluid-flow model, peak 32Kb/s, mean 11.24 Kb/s)

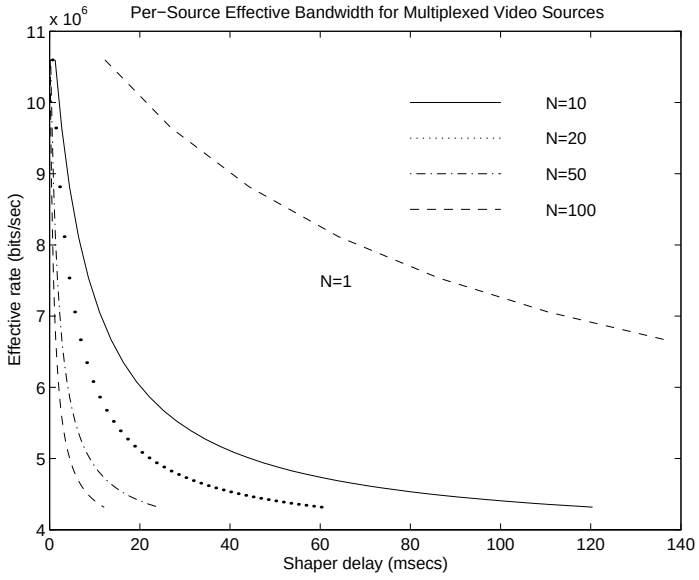

Figure 3.9: Per-source effective bandwidth requirement for multiplexed video sources (Fluid-flow model, peak 11.7 Mb/s, mean 3.85 Mb/s)

equivalent capacity formulas are in fact conservative over-estimates and thus in practice even higher utilizations will be achieved [17]. Clearly, for both video and voice traffic a single level of multiplexing is able to achieve high utilization. Since the peak rate of the aggregation of sources is very close to the mean, the output of such an aggregation of sources fed to a deterministic server is very close to being a CBR process and hence can be treated deterministically without significant loss of utilization. In these curves, the effective bandwidth requirement of a source decreases as the number of sources being multiplexed increases, even though the effective bandwidth formulas as proposed by Guerin et al [25] and Elwalid [17] do not depend on the number of sources being multiplexed. This is because a larger aggregation of sources can also be assigned a larger buffer for shaping for given bounds on maximum queuing delay. Consequently, the effective bandwidth of each source decreases as a function of this larger buffer [25].

3.4 The Multi-hop Bandwidth and Buffer Assignment Problem

In the previous section, an approach based on deterministic reservation of link bandwidths and buffer space for each VP has been proposed. In this section we analyze a queuing model of a VP in such an architecture. The proposed architecture requires deterministically reserving resources for each VP. However, an unresolved issue is that given the resources reserved for a VP, how should one optimally assign these over the different physical hops of a VP. For instance, how should a given amount of buffer space reserved for a VP in the proposed scheme, be distributed over the different physical hops that the VP traverses ?

In this section, the performance of different "Multiple Hop Resource allocation Policies" (referred to as MHRPs) will be compared. An MHRP is an algorithm for deciding the per-hop resource assignments (such as bandwidths and buffer space) given a constraint on end-to-end QoS measures (such as loss probability) and end-to-end resources such as buffers and bandwidths. The objective is to assign the per-hop resources while optimally satisfying some end-to-end QoS constraints. For instance, one problem addressed in a following section is that of minimizing the total bandwidth (summed over all hops) of a VP given the end-to-end buffer space and an end-to-end loss specification.

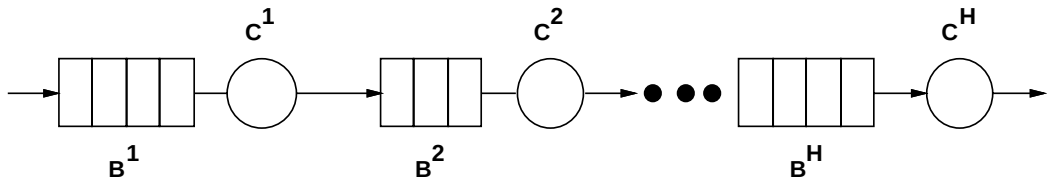

Figure 3.10: Queuing model of a VP in the proposed architecture used for analysis.

In this section, only one end-to-end QoS metric viz. the cell loss probability (CLP) is analyzed. Analysis of optimal resource allocation schemes for other QoS measures is an issue for further work. A similar problem was also addressed by Onvural and Liu [43]. However, in their model, the link bandwidth was fully shared by all VPs unlike our reservation-based approach.

3.4.1 Queuing Model and Definitions

Figure 3.10 shows a queuing model of a VP which is used for analyzing the properties of different MHRPs.

A VP in the proposed architecture is modeled as a tandem queuing network with H nodes corresponding to the physical hops traversed by this VP. There is no cross traffic and external cells arrive only at the first queue and depart from the last. The buffer space reserved at hop h is $B^h (\geq 1)$ cells and the service rate at hop h is C^h cells per unit time. Of the B^h slots at hop h, one slot is assumed required for the cell in service while the rest are required for waiting or queued cells. Each server is the standard work-conserving fixed rate server. This server models a deterministic bandwidth reservation scheme as in the proposed architecture.

This queuing model is consistent with the proposed approach of complete resource reservation for each VP. The rate of the server at hop h represents the bandwidth assigned to a VP at hop h and the size of the buffer at hop h represents the queuing buffer reserved for the VP at hop h. Changing these parameters is hence equivalent to changing the per-hop resource allocations of a VP. We will assume that only the size of the buffer for waiting cells can be changed and one unit of buffer space is always required at each hop for the cell in service. A model of a single VP can be analyzed in isolation because of the complete

separation of resources for each VP in the proposed architecture. All subsequent results are based on this queuing model.

Whenever a cell arrives at a full buffer at any hop in the path, it is lost due to overflow. Only the total *number* of lost cells determines the average loss ratio for a given sequence of arrivals. Hence, the total number of losses will be unaffected if a constant propagation delay is added between any two nodes. (If some fixed delay is added to link h say, then node $h+1$ will see the same arrival sequence as before, shifted in time by this amount. Since the inter-arrival times do not change, the total number of losses also remains unchanged. This follows from Lindley's recursion [54] according to which the cell losses at a queue depend only on the inter-arrival times and not on the actual arrival instants.). Propagation delays are hence assumed to be zero without any loss of generality.

Assigning different resources at different physical hops of a VP results in differing levels of QoS at each hop. For instance, from Theorem 1, increasing the bandwidth at some hop h can only decrease the total number of losses at this hop. But this may or may not decrease overall losses. The idea is to vary the per-hop assignments in order to optimally satisfy the end-to-end QoS requirement. Hence, the problem is equivalent to splitting the end-to-end QoS specifications into per-hop QoS specifications [41]. A policy which obtains the per-hop QoS specifications given the end-to-end QoS specification will be referred to as a QoS allocation policy. The MGF (Maximal Gain First) QoS allocation policy is defined as follows.

Definition 1 *Maximal Gain First Policy: The Maximal Gain First policy allots the entire end-to-end loss to the first physical hop of a VP. Transmission at all further downstream hops is required to be loss-less.*

The term "Maximal Gain" is motivated by the fact that by letting all the cell loss occur at hop 1, maximal reduction in bandwidth requirements is sought to be obtained at the first physical hop itself. This is a consequence of Theorem 1 which asserts the monotonicity of losses with service rate.

Theorem 3 (stated without proof here) asserts that setting service rates which are non decreasing from hop 1 towards hop H is sufficient to implement the MGF policy. This is because, when the service rate at a queue is at least as much as that at an upstream

queue, the time between two successive arrivals at a node is never more than the (fixed) transmission time at that upstream queue. Since each cell has the same transmission time for a given service rate, each cell must necessarily depart from the downstream queue before another cell arrives at this queue. Hence no queuing or cell loss occurs in such a case.

Theorem 3 *To ensure no cell loss at all nodes after the first for any arbitrary arrival process, it is sufficient to set the service rate of all servers to be at least as much as that at hop* 1.

For subsequent sections, for the case where per-hop buffer assignments are given and fixed, the MGF policy will be assumed to be implemented by setting the service rate at each hop to be equal. (From the above theorem, this is sufficient to restrict any cell loss to the first hop only). For the case where the per-hop buffer assignments are also controllable given a constraint on the total end-to-end buffer space, the MGF policy will be implemented by setting all service rates equal and assigning the entire available end-to-end waiting buffer space at hop 1 (so that at all downstream hops, only the unit buffer needed for the cell in service is present). For convenience, we will refer to both the QoS allocation policy of having all the loss occur at the first hop and the MHRP which implements it as the MGF policy.

A direct consequence of Theorem 3 is the following.

Corollary 1 *The total number of losses for a given (arbitrary) arrival sequence in a given tandem network configuration is unchanged, if the service rate of each queue i is set to the minimum of the rates of servers* 1 *through* i.

Hence, whenever a queue i has service rate no less than that of all queues 1 through $i-1$, it can in fact be eliminated from the network and its upstream node connected directly to its downstream node, without affecting the total number of losses, since such a queue only adds a constant delay (equal to one transmission time) to each cell. The inter-arrival times at downstream nodes and hence the losses are not affected. This implies that without any loss of generality, only networks in which the service rate is non-decreasing from source to destination need to be analyzed if only the total number of losses is of concern. The performance of the MGF policy is now analyzed with respect to other MHRP policies.

3.4.2 Simultaneous Bandwidth and Buffer Assignment

Consider the problem where the total amount of end-to-end queuing buffer space W and the service rate at the last hop (C^H) are given. It is required to find the per-hop buffer and service rate assignments in order to meet a specified bound on the end-to-end loss probability. W refers to the available end-to-end buffer space for waiting or queued cells (and not the cell in service) to be split up over different hops. If W_i represents the waiting buffer space allotted to hop i, then the total buffer space at each hop j is $B^j = W^j + 1$ (adding the unit buffer which is always present at each hop for the cell in service) and we must have $\sum_{i=0}^{H} W^i = W$.

In this case, as mentioned earlier, the MGF policy is implemented by setting all service rates equal (equal to C^H) and all W units of queuing buffer space to hop 1. Theorem 4 states that in this case, the MGF policy results in the minimal number of cells lost for any sample path of arrivals as compared to any other resource assignment under the same constraints.

Theorem 4 *Let the total amount of end-to-end queuing buffer space W and the service rate at the last hop H (C^H) be given. The MGF type configuration with $B^1 = W + 1$, $B^2 = B^3 = \cdots = B^H = 1$ and $C^1 = C^2 = \cdots = C^H$ results in the minimal number of losses for any arbitrarily given sample paths of arrivals as compared to any other assignment of rates and buffers under the same constraints.*

Proof: The proof is listed in appendix C. □

Theorem 4 is important since it indicates that end-to-end losses are minimized for any possible sequence of arrivals by an MGF type configuration given the total buffer space and service rate at the last hop. Translating this result to a VP in the proposed architecture, it means that given the bandwidth at the last hop, and full freedom to assign the per-hop buffers, by setting the bandwidth at all hops equal to that at the last hop and letting all the queuing (and cell loss) occur at hop 1, we obtain the minimum number of end-to-end losses for any possible sequence of arrivals.

Theorem 5 asserts that the MGF policy results in the minimal total bandwidth (sum of the service rates over the path) required to achieve a given end-to-end loss specification given only the end-to-end queuing buffer space. In fact, not just the total bandwidth but even

the bandwidth at each hop is minimized i.e. it is not possible to have another assignment of bandwidths and buffers which meets the specified loss bound and has a lower bandwidth at any single hop in the path.

Theorem 5 *Given the end-to-end queuing buffer space B, an arbitrarily fixed sample path of arrivals and a bound on total end-to-end losses, if some configuration of rate and buffer assignments can achieve the bound on losses then an MGF type configuration (with the same amount of total buffer space) will also achieve the same bound on losses while requiring no greater total bandwidth. Additionally, the bandwidth requirement at* each hop *will be no higher in the MGF configuration.*

Here total bandwidth is defined as the sum of the service rates of the H queues in the path.

Proof: Let if possible some configuration 'O' of service rates and buffer allocations achieve a given end-to-end bound on total losses. By reducing the rate of appropriate servers as in corollary 1, this can be reduced to a network with service rates decreasing monotonically from node 1 towards node H, without affecting the total number of losses. Hence we can assume that the rate at hop H (C^H) is the minimum rate of all servers in the system. Now, keep this rate fixed. From Theorem 4, if all other the service rates are made equal to C^H and all waiting buffer space moved to hop 1, the total number of losses can only decrease. Again, since C^H was the minimum rate in the system, this procedure involves only rate reductions. Hence, through a series of transformations involving only reductions in service rates, the configuration 'O' has been transformed into an MGF type configuration for which the total number of losses is no more than that with configuration 'O' for any arrival sequence.

Hence, the rate (equivalently the bandwidth) at each hop with the MGF configuration is smaller than the corresponding rate in configuration 'O' without violating the bound on end-to-end losses. Consequently, the sum of the rates over all hops (i.e. the total bandwidth) is also smaller in case of the MGF configuration. □

Theorem 5 essentially states that the MGF policy is the optimal policy when the total number of losses is of interest and the per-hop buffer assignments are variable given a constraint on the total end-to-end buffer space. In such a case hence, we only need restrict

attention to MGF type configurations.

Finally, the optimality of the MGF policy can be extended to the network level. Consider a network with several VPs in which there is no statistical multiplexing across VPs (as in our scheme). The term residual bandwidth at a link refers to the difference between the link bandwidth and the sum of bandwidths reserved for each VP using this link.

Corollary 2 *Given a network using the path level reservations architecture, given VP routing, let the total buffer allocation for each VP be given and let each VC allowed to traverse only one VP. Use of the MGF policy for each VP results in the maximum residual bandwidth at each link of the network.*

An assumption here is that service rates can be independently assigned to each VP at each link as long as the sum of the rates is no more than the link rate.

The above corollary is a direct consequence of the fact that employing the MGF policy within each VP results in the minimal bandwidth requirement for each VP at each hop in its path. At each link, the rate requirement for each VP is minimized by the MGF policy. Hence the sum of the rates is also minimized, which maximizes the residual bandwidth at each link in the network. The constraint on VCs traversing only one VP is necessary since the bandwidth requirements of VCs which traverse more than one VP cannot be predicted in general as outlined earlier, due to dependencies between the bandwidth allocations in the different VPs traversed.

Thus utilizing the MGF policy within each VP frees up the maximum possible bandwidth at each link of the network.

3.4.3 Bandwidth Allocation with Fixed Buffers

In general, buffer space assigned to a VP may not be moved around between different hops as required by the above results. Hence, the case where the reserved buffer space is fixed at each hop for the tandem network representing a VP is considered. Only the service rates (i.e. the VP bandwidths at each link) can be adjusted.

As mentioned earlier (in the discussion following Theorem 3), in this context, the MGF policy will be implemented with equal service rates at each hop. Here, the MGF policy is not optimal in terms of requiring minimal total bandwidth as it was for the variable

buffer assignments case. However, it is shown below that the total bandwidth requirement using the MGF policy is still within an easily computable constant factor of the optimal assignment. Further, the result of a simulation is presented which indicates that on the average the performance of the MGF policy can still be close to optimal.

For a given (arbitrary) arrival sequence, let R be the minimum service rate required at the first queue for the losses at the first hop to be no more than the specified upper bound on end-to-end losses. R could be computed off-line, from bandwidth-loss tables, queuing analysis or any other means. Let the mean rate of the arrival process be unity (R is hence the normalized equivalent rate). With the MGF policy, the rate of each server is set to R. This set of rates hence guarantees the specified end-to-end loss constraint.

The following theorem bounds the total bandwidth requirement of the MGF policy with respect to the optimal under the fixed buffers constraint. The end-to-end loss probability requirement is assumed to be sufficiently small (e.g. 1e-2 or less) so that the mean rate of the source is unchanged from input to output.

Theorem 6 *With a fixed per-hop buffer assignments, the total bandwidth requirement to meet a bound on end-to-end losses with the MGF policy is no more than* $\frac{RH}{R+H-1}$ *times the requirement with an optimal policy which results in the minimal total bandwidth.*

Proof: Since service rate and losses are monotonically related at a single node (Theorem 1), and the MGF policy lets all the loss occur at hop 1, the service rate at hop 1 with the MGF policy cannot be more than that with the optimal configuration. Hence the rate required at hop 1 even for the optimal policy must be at least R. To ensure no losses, the MGF policy requires a rate of R at all successive hops. At these hops, even the rate required by the optimal policy must be at least 1 (the mean rate). This is because the end-to-end loss probability is sufficiently small (typically 1e-2 or smaller) that the steady state mean rate of cells departing the network is essentially the same as the input mean rate. Hence the ratio of the total bandwidths in the two cases can be at most $\frac{RH}{R+H-1}$. □.

As an example with $R = 2$ over a 4 hop path (i.e. an equivalent capacity of twice the mean rate), the total VP bandwidth with the MGF policy is no more than 1.6 times the optimal. The average case performance of the MGF policy with respect to the optimal will in general be significantly better than above worst case bound, however, this bound enables

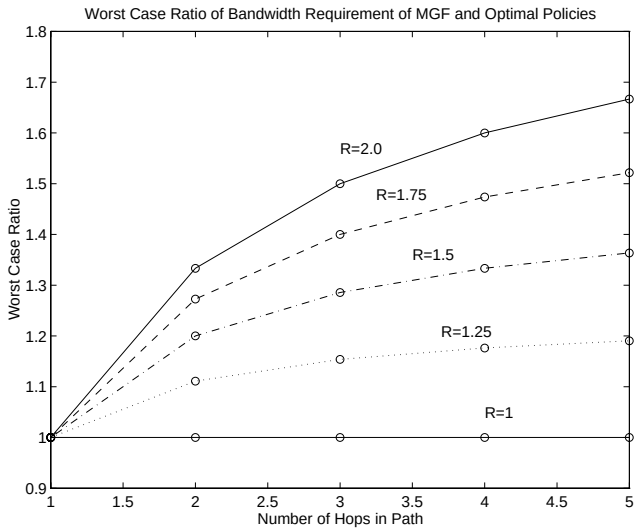

Figure 3.11: Worst case ratio of total VP bandwidth requirement with MGF to that with optimal policy for fixed buffers case.

a quick worst case estimate. Figure 3.11 plots the above ratio for typical values of R and H. For small values of R or H, the bandwidth requirement with the MGF policy can be seen to be no more than 50% more than the optimal typically.

In practice, finding the optimal configuration will be quite difficult because it requires characterization of traffic departing each node for performing bandwidth allocation. Further, as a consequence of the non-monotonic relations between service rate and loss rate in this case, several combinations of rates and buffers will result in local minima for the end-to-end losses. Finding the configuration of rates which results in the absolute minimal number of end-to-end losses will involve a complex global optimization procedure with repeated characterization of departure traffic (itself a complex procedure). Bandwidth allocation with the MGF policy on the other hand is very simple and requires no departure traffic characterization since no cell loss occurs at downstream nodes. At the first node, a source model is typically available. Hence, an approximate solution based on the MGF policy offers a simple approach for assigning resources over multiple hops while being able to guarantee end-to-end cell loss.

The next subsection demonstrates a simple simulation experiment which indicates that even in this case, typically the MGF policy may be quite close to the optimal in terms

of requiring the minimal total bandwidth. i.e. its average case performance may be much better than the above worst case performance ratio.

3.4.4 Average Case Performance of the MGF policy in the Fixed Buffers Case

Above, it has been shown that in the queuing model used for a VP, when the per-hop buffer allocations are fixed, the MGF policy (equal service rates at all hops) is not necessarily optimal for minimizing the total bandwidth requirement over all hops. However, an upper bound on the worst case ratio of the total bandwidth requirement with the MGF policy as compared to an optimal policy has presented.

One simulation result is reported here to examine the total bandwidth requirement with the MGF policy with realistic sources. It is found that even when per-hop buffers are fixed, the MGF policy performs very well in terms of minimizing the total bandwidth. A VP traversing two physical hops and carrying 15 voice sources has been simulated. The total bandwidth requirement for a single VP over two hops is plotted as a function of the end-to-end cell loss probability. This is shown in Figures 3.12. In these plots, each curve corresponds to a fixed value of the bandwidth assigned at hop 1. The value of the bandwidth at hop 2 is then varied to yield the curve.

The key observation is that in all figures, the different curves obey the property that a curve corresponding to a higher value of hop 1 bandwidth always lies above one corresponding to a lower hop 1 bandwidth. Consider a fixed value of the x-axis variable. The above observation implies that for a given value of actually observed end-to-end loss, the overall VP bandwidth is always smaller in an allocation in which the hop 1 bandwidth is smaller. This implies that the MGF policy can in fact be very close to optimal even in the fixed buffers case as is seen here. The bandwidth at hop 1 is incremented in steps of about 25 Kb/s. The plots suggest that the difference between the bandwidth requirements of the optimal and the MGF policy is of the order of 5 % at most in this case. The good performance of the MGF policy can be attributed to the fact that traffic multiplexed at hop 1 gets smoothed. As a result, very little rate reduction is possible at hop 2 without exceeding the end-to-end loss requirement. Hence, it is best to let all the loss occur at hop 1 and set the rate at both hops to be equal.

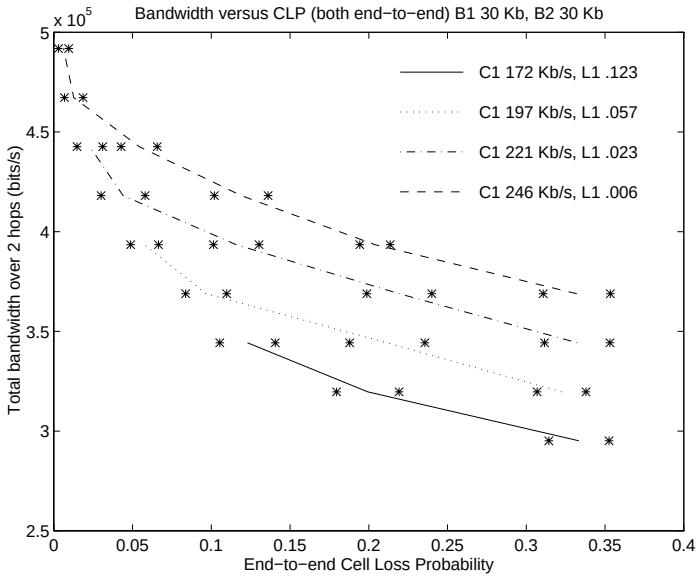

Figure 3.12: Total VP Bandwidth versus CLP (both end-to-end) for different values of hop 1 loss. Two hop path, 15 voice sources, Buffer1 = Buffer2 = 30 Kb

Several additional simulations are needed to confirm the performance of the MGF policy in the fixed buffers case. Based on this experiment, however, it is seen that even in the fixed buffers case, the MGF policy while not optimal, results in good performance.

The results in this section indicate that the MGF policy of configuring the end-to-end resources of a VP is optimal (in terms of resource usage) in some cases (outlined above) and performs well in other cases. A key advantage of this policy is that queuing analysis of the end-to-end path is greatly simplified since it is reduced to analysis of a single queuing node. An implementation of the VP level reservations architecture is now outlined in the following section.

3.5 An Implementation of the Proposed Architecture

Section 3.3 introduced an architecture in which deterministic bandwidth reservations are performed at the level of Virtual Paths in an ATM network. In general, this suggests that deterministic reservation be performed on groups of channels which traverse the same physical set of links. Next, section 3.4 analyzed some sample path properties of cell loss

over multiple hops in such a scheme and the advantages of a policy which lets all the end-to-end cell loss occur at the first physical link of a VP were demonstrated. A possible implementation mechanism for such an architecture is now outlined.

VP bandwidth guarantees are enforced using a non work conserving scheduler which enforces the assigned rates. In this section the use of a Weighted Round Robin (WRR) type scheduler (equivalent to a multi-rate time-division multiplexor) is investigated. The WRR scheduler is very simple to implement and analyze. Many other schedulers, such as Earliest Due Date [19] Stop&Go [23] or Weighted Fair Queuing [44] could also be used, since these too are based on the concept of rate control. However, these are significantly more complicated than WRR. The WRR based method is good enough to achieve high utilizations (as is shown in the following sections). The idea of round-robin type service of different traffic classes for ATM has been suggested by others also (see Sriram's Dynamic Time Slice Scheme in particular [50]). However, to our knowledge the performance achievable by this approach has not been quantified so far, particularly for the multi-hop case. Additionally, all the proposals for round robin service have employed a work conserving server in contrast to our non work conserving approach.

In this section, it is assumed that using a WRR server, an equal amount of bandwidth is provided to a VP at each physical hop. This is a consequence of the results from the previous section, where some of the advantages of the MGF policy were outlined.

3.5.1 Operation of the WRR Server

Each ATM switch in a path of a VP is modeled as an output-buffered multiple input multiple output switch as shown in Figure 3.13. In such a switch, queuing buffers are provided only on output ports of a switch and these directly feed the output links [46].

We assume that cell loss only occurs due to overflow of the output buffers and no cells are lost due to contention within the switch fabric. Let there be $K+1$ VPs being served by this server, denoted $\mathcal{V}_0, \mathcal{V}_1, \mathcal{V}_2, \ldots \mathcal{V}_K$. $\mathcal{V}_0$ denotes a VP carrying best-effort type traffic e.g. data files, network management traffic etc. Such traffic is not normally delay sensitive and the provision of an appropriate long-term average bandwidth for such traffic is assumed to be sufficient for such traffic. Let the number of slots reserved for $\mathcal{V}_j$ in each server cycle be denoted n_j.

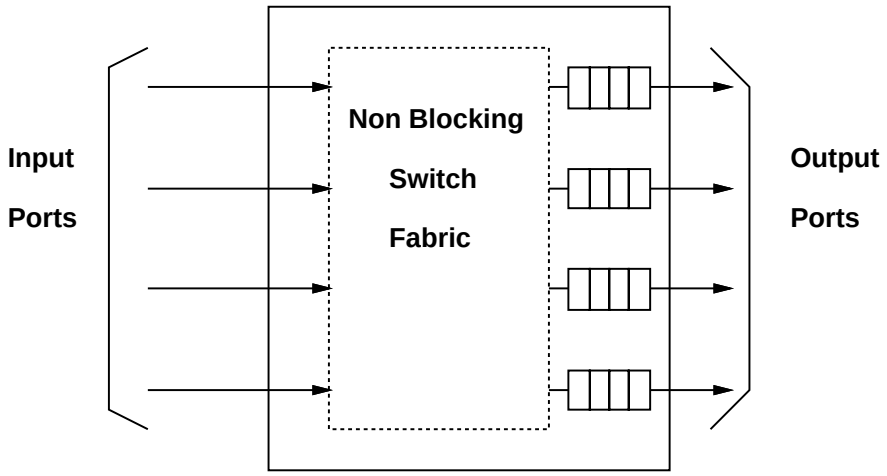

Figure 3.13: Model of output buffered ATM switch under consideration.

The output buffer at each port is logically partitioned such that a buffer of size B_j^h is *reserved* for $\mathcal{V}_j$ at the appropriate output buffer at the h'th switch in the path of $\mathcal{V}_j$ [2]. The server cycles through all VPs carrying guaranteed traffic (viz. $\mathcal{V}_1$ through $\mathcal{V}_K$) according to a preset deterministic schedule in a strict TDM-like manner. Let the length of a server cycle be T time units (the time unit is assumed to be the transmission time of a single cell). In each cycle, $\mathcal{V}_j$ is served for exactly n_j slots. If $\mathcal{V}_j$ does not have a cell to transmit, a cell from B_0 (the best-effort queue) is transmitted instead. If B_0 is also empty, no cell is transmitted and the server is idle.

These definitions are illustrated for 4 VPs in Figure 3.14, where $\mathcal{V}_1$ is assigned 2 slots, and $\mathcal{V}_2$ and $\mathcal{V}_3$ are assigned one slot each ($n_1 = 2$, $n_2 = 1$,$n_3 = 1$, $T = 4$). $\mathcal{V}_0$ is not assigned any slots in the cycle and only gets to transmit when a VP does not have a cell to transmit during its slot. $\mathcal{V}_1$ originates at the first node and itself consists of several bursty VCs. In general each VP sees a service "window" followed by a server "vacation" while other VPs are served.

Let d_s^h denote the maximum delay that can be encountered by a cell in the switch fabric of the $h'th$ switch in the path. It is assumed that the server cycle time T is larger than this delay.

[2] With an appropriate buffer management algorithm the same scheme can be implemented with a single buffer shared by all connections. This is discussed in a following section. The main requirement for providing the end-to-end guarantees is the deterministic bandwidth partitioning and not the buffer partitioning.

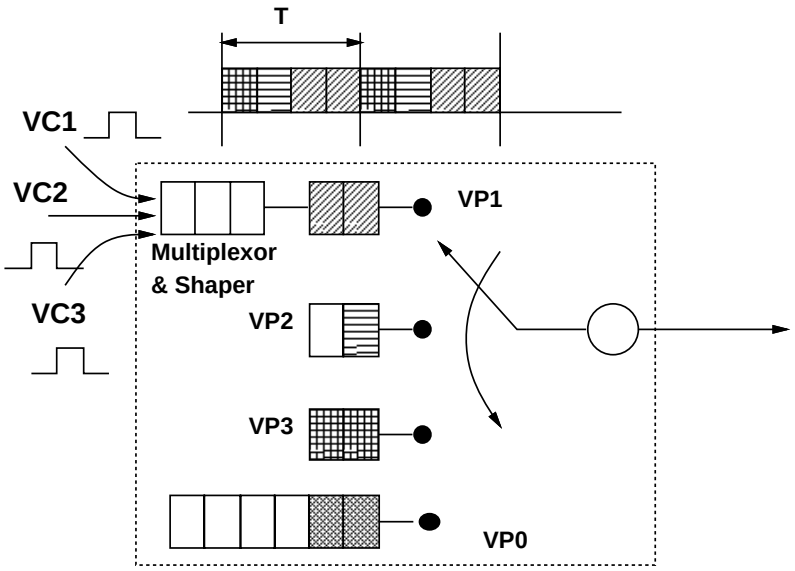

Figure 3.14: Logical model of single output buffer in WRR scheduling model.

Using the above notation and assuming that $d_s^h < T$, the following theorem indicates the necessary amount of buffer space to ensure no cell loss at each node after the first.

Theorem 7 *In a network employing the WRR scheduler described above at every node, no cell of a VP is lost due to buffer overflow at any hop after the first if at at each hop h, the buffer reserved for $\mathcal{V}_j$ (B_j^h) is at least $2 * n_j$ cells.*

Proof : The proof follows from the non work-conserving nature of the WRR server. Cell loss will occur at the output buffer at some hop h along the path of a VP only if the total number of arrivals over some time interval, exceeds the number of departures over the same interval by an amount greater than the buffer size. In any time interval $[t_1, t_2)$, the server guarantees $(\lfloor (t_2 - t_1)/T \rfloor) n_j$ departure opportunities to this VP. Further the maximum number of arrivals over the same interval is at most $(\lceil (t_2 - t_1)/T \rceil + 1) n_j$. (This is because cells transmitted during different upstream cycles may arrive closer together than T because of experiencing different delays in the switch fabric. Since $d_s^h < T$, at most one extra cycle worth of data can be input to B_j^h before a transmission opportunity becomes available). The difference between the total number of arrivals and total number of departures over any interval cannot exceed $2n_j$ (using the identity $\lceil x \rceil + 1 - \lfloor x \rfloor \leq 2, \forall x$). Hence a buffer of

$2n_j$ cells ensures no cell loss. □.

For simplicity, we have assumed that the server cycle is of the same length at each hop. In general, this need not be so. Due to the non-work-conserving service, zero cell loss can be similarly ensured with appropriate buffer sizing. Also, in the queuing model of a VP presented in the previous section, only a unit buffer was required at all hops after the first whereas in an actual implementation of the scheme using a WRR server, $2n_j$ cells of buffer space is required at downstream hops. The additional buffer space is needed to compensate for the worst case phase mismatch between the cycles at two successive hops and the delay variation in the switch fabric, neither of which are present in the queuing model. In general, the queuing buffer at the first hop will be significantly greater than $2n_j$. The entire end-to-end cell loss as well as a majority of the end-to-end queuing delay occurs at the first hop. Hence a VP with equal bandwidth at each hop can reasonably be modeled by the MGF configuration in the queuing model in section 3.4.

3.5.2 Call Admission for a Single VP

Using the above scheme, a simple call admission procedure can provide an accurate end-to-end QoS guarantee for any VC traversing a given VP. As mentioned earlier, in the proposed architecture, VCs may in general traverse multiple VPs. Development of admission control algorithms for such VCs is not addressed here. As discussed earlier, approaches which exploit statistical multiplexing compromise on the ability to accurately predict the performance of each individual stream being m ultiplexed. The approach taken in this section is to bound the QoS seen by the aggregated VP traffic and not the individual VCs within the VP. It is assumed that the QoS requirements for a VP are set to the most stringent requirements from among all the classes of calls that use the VP.

Consider QoS guarantees of the type $Prob(d_i > D) \leq \epsilon$, where d_i is the end-to-end cell delay of cells of a connection i using the given VP (say $\mathcal{V}_j$). This is converted into two guarantees viz. a deterministic guarantee on the maximum cell delay and an average cell loss probability guarantee. Hence the cells which get lost are the only ones which fail to meet the specified delay bound D. It can be seen that if the maximum cell delay can be bounded by D and the average cell loss probability by ϵ, it is *sufficient* to guarantee that the original QoS guarantee will also be met. Hence, the specified statistical delay guarantee is

converted into a deterministic guarantee on maximum delay and a guarantee on the average cell loss. This is a conservative approach to providing statistical delay guarantees.

An upper bound on the maximum end-to-end delay experienced by any cell of $\mathcal{V}_j$ which traverses H physical hops, can be easily calculated as

$$D_{max} = \lfloor B_j^1/n_j \rfloor * T + (B_j^1 - \lfloor B_j^1/n_j \rfloor * n_j)/C + 2*(H-1)*T + \sum_{h=1}^{H} T_{prop}^h \qquad (3.1)$$

where C is the physical link capacity (assumed same for all hops), and T_{prop}^h is the propagation delay for the h'th hop.

The first two terms represent the maximum queuing delay at the first hop of the VP. When n_j is small compared to the buffer size B_j^1, this can be approximated as,

$$\begin{aligned} D_{max} &\approx (B_j^1/n_j)*T + 2*(H-1)*T + \sum_{h=1}^{H} T_{prop}^h \qquad (3.2) \\ &\quad (n_j \ll B_j^1) \end{aligned}$$

Now the first term can be seen as simply the buffer size at hop 1 divided by the bandwidth reserved for this VP (n_j/T), while the second term implies a constant delay of up to T at each hop after the first.

The specific call admission procedure used to guarantee the cell loss is not defined here. The main advantage of this approach is that a cell loss calculation needs to be performed at only one node to determine the end-to-end loss. Further, since this calculation is performed at the first hop, a model of the source traffic is typically available. Any other approach which allows cell loss to occur at multiple nodes in the path would need to characterize the traffic departing the first node, (typically a complex queuing problem). At the first hop, any of several techniques can be used including equivalent capacity ([17], [25]) or simply off line computed bandwidth versus loss type tables [50]. Hence when a call i requests admission into VP $\mathcal{V}_j$, using the appropriate CAC algorithm determine whether cell loss at hop 1 will be less than specified. Determine maximum end-to-end delay from Eqn 3.1. Accept the call only if both the delay and loss bounds are satisfied, else reject the call.

The approach taken in this section has been that all VCs within a VP are guaranteed a worst case QoS according to which resources have been allocated to the VP. Further improvement in performance can be expected by using priority scheduling techniques between the different VCs of a VP which can guarantee differing QoS levels to different VCs within a VP. Optimal priority scheduling algorithms exist for satisfying certain QoS specifications at a single node (e.g. [31]). Since in the proposed architecture, all the cell loss and a large fraction of the queuing delay occurs at the first hop of a VP, these techniques can be used at the first hop of a VP to give end-to-end guarantees on delay and loss (the queuing delay at hops after the first can be considered to be constant equal to $2T$ for making end-to-end delay guarantees). Performance will improve at the expense of increased complexity of implementation.

Another possible approach for call admission is developed in the following chapter. In this, a call is admitted as long as the residual bandwidth on each link traversed by this call is at least the peak bandwidth requirement of this call. A dynamic resource re-allocation algorithm is then used to slowly reduce the VP bandwidth to the minimal level which still satisfies the specified QoS of all calls using this VP.

3.5.3 Use of Buffer Sharing

While outlining the working of the WRR server, it was assumed that both the link bandwidth and the output buffer before each link are partitioned in a static manner, with one partition for each VP. In fact, only the bandwidth partitioning is necessary for providing end-to-end QoS guarantees. The available buffer space at a switching node can be utilized more efficiently by sharing of the buffer space between cells from different VPs.

The key point here is that a shared buffer system can always do better than a partitioned buffer system with the same total buffer space for certain QoS measures, if the shared buffer space is managed appropriately. This is because with an intelligent buffer management algorithm which incorporates 'push out', a shared buffer system can always emulate a partitioned buffer system and can hence always perform at least as well. The term 'push out' refers to the technique of an incoming cell replacing an existing cell waiting in the buffer, under certain conditions. The advantages of buffer sharing are illustrated through a simple example below.

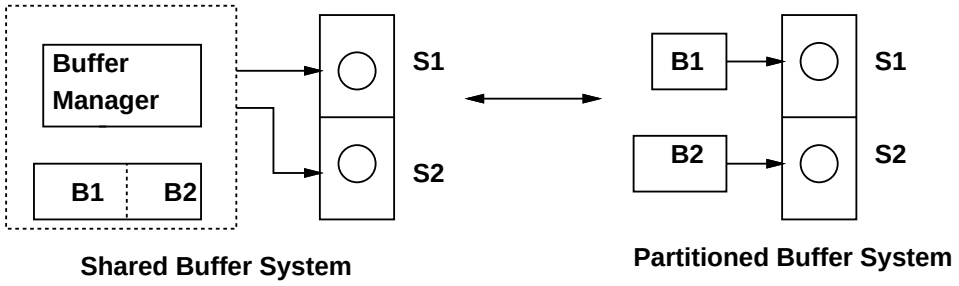

Figure 3.15: Comparison of partitioned and shared buffer schemes.

Figure 3.15 shows a queuing system with two fixed rate servers S_1 and S_2, once with a common buffer of size $B_1 + B_2$ cells and once with separate buffers of B_1 and B_2 cells respectively. In the partitioned case, the system essentially works like two separate queues, while in the shared buffer case, a buffer manager algorithm manages the storage space and sends cells to the servers when departures occur. Each cell can be sent only to a specific server. This models a WRR server according to the above definition, serving two VPs.

Consider the following strategy for the buffer manager. Perform a logical partitioning of the available buffer space into sections of B_1 and B_2 cells respectively. Now, operate this queuing system exactly as if the buffer was partitioned i.e. store cells to be served by S_1 in the partition assigned to S_1's cells and those to be served by S_2 in the other partition, as long as no arriving cell finds its partition full. When a cell departs from a server, any cell waiting in the corresponding partition is scheduled for service, else the server idles (even if there are cells waiting for the other server). If an arriving cell finds its partition full, and a slot is available in the other partition, the buffer manager temporarily stores it in that partition. Whenever a slot opens up in its correct partition, the cell is moved back to its partition. If an arriving cell finds cells from the other partition in its partition, those cells are pushed out and the arriving cell is stored in its proper partition. It can be seen that this strategy emulates the separate buffers system as far as possible and tries to do better whenever possible.

Consequently, using this strategy neither of two VPs experience higher losses in the shared buffer case than in the partitioned case, and in fact the losses may be lower.

Theorem 8 *With the buffer management strategy outlined above, the losses for each queue*

in the shared buffer system are no more than in the partitioned system.

This may be seen to be true from the following arguments. An arriving cell for each queue with the above strategy, is lost only when at least its partition is full, in which case it would be lost even in the partitioned case.

In the partitioned case, a cell will be lost if its partition is full even if another queue's partition is not. In the shared scheme however, a cell is lost only when the entire shared space is full, leading to lower losses.

The only way that this buffer management strategy can increase losses is if a cell which is lost in the partitioned buffer case is saved in the shared buffer case, but consequently causes higher losses due to the increased load taken on in the shared case. However, since all cells, have the same transmission time, an extra saved cell can cause the loss of at most one other cell. Hence the total number of losses for each queue can never be more than in the partitioned buffer case. These arguments are similar to those in the proof of Theorem 2, hence a formal proof is not presented for this case.

Hence, if the increased complexity of buffer management is acceptable, a buffer sharing scheme similar to that just outlined will always perform better than a partitioned buffer scheme. (Note however, that the delay bounds of Equations 3.1 and 3.3 are valid only for the partitioned case). Thus, at each output buffer of an ATM switch, all VPs originating at that node and using the same output port can be multiplexed into a single buffer for high buffer efficiency.

Further, these arguments apply even for shared buffer type ATM switches, where a common buffer is used for all output ports of the switch [46]. Theorem 8 continues to hold even if the two queues mentioned actually correspond to VPs leaving from different output ports of an ATM switch each of which employs the WRR server.

Hence, in a shared buffer type ATM switch employing the WRR server, a single multiplexing buffer can be used for all VPs originating at that node, irrespective of the output port they use. This will lead to very efficient use of switch buffer space and lower cell loss.

Theorem 8 was for the average loss rate QoS measure. Similar results may be shown for other QoS metrics such as queuing delay percentiles also, but the buffer management algorithm becomes accordingly more complex.

3.5.4 Implementation of the WRR Scheme

Cell spacers have been suggested for use at each switching node of an ATM network and have been shown to result in improved utilization by preventing loss by preventing buildup of delay jitter [26]. A cell spacer is nothing but a peak rate enforcer. The main requirement of our scheme is that a mechanism is needed for non-work-conserving enforcement of rate. This can easily be implemented by making the spacer-controller function in a non-work-conserving manner. Hence the complexity of implementation of our scheme is equivalent to that of schemes already proposed for use in ATM networks and the hardware requirements appear to be feasible. Details on the working of the spacer-controller algorithm may be found in [26].

An additional point of importance is that the manner in which the WRR server is operated affects the QoS observed even for fixed bandwidth and buffer allocations. This must be taken into account either by an appropriate modification to the admission control algorithm or with additional hardware as shown below.

The example simulation below shows that cell loss experienced by a VP can be very different even when a fixed amount of bandwidth and buffer (B_j^1) is reserved for it depending on the choice of the WRR cycle length. This is because, as defined above, the WRR server has some "vacation effects" since all n_j slots reserved for VP $\mathcal{V}_j$ appear in a burst during each cycle T. In some sense thus, the server is itself an "On-Off" type server with constant On and Off times and not a constant rate server.

Figure 3.16 shows the variation of loss probability of a VP with the unit bandwidth of the WRR server (which equals $1/T$ cells per second and is equivalent to varying T). Because of the deterministic isolation of VPs from one another, the loss probability of a single VP can be examined in isolation. The source is an aggregation of 15 voice sources conforming to the standard On-Off model [50]. The VP bandwidth is fixed at 200 Kb/s. The size of the multiplexing buffer is 10 Kb. For unit bandwidth of 8Kb/s or more, the CLP varies very slowly and is essentially unchanged. However, for small unit bandwidths (i.e. large values of T), cell loss increases sharply as T is increased. This is because with a large T, when n_j becomes comparable to the buffer size, many of the n_j departure opportunities that a VP has in 1 cycle will be wasted unless the queue is full.

This simulation shows that the effect of the server cycle length on cell loss can be taken

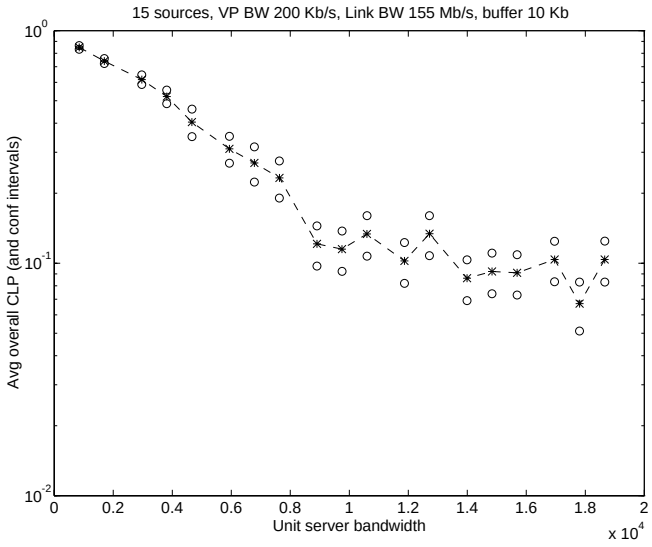

Figure 3.16: Variation of CLP with Unit Bandwidth (B_j^1 = 10 Kb).

care of by choosing a small enough cycle length according to the sizes of the shaping buffers of all VPs being served by this server. However, a small cycle length can lead to large bandwidth quantization granularities since each VP must be assigned a multiple of the unit bandwidth, which increases as the cycle length decreases. Another solution is to use a multi-level round robin mechanism such as Stop and Go [23] so that trade offs between these conflicting requirements can be resolved.

The effect of server vacations can be eliminated by adding an additional buffer of $2n_j$ cells between the queuing buffer B_j^1 and the WRR server at hop 1. Cells can now be moved from B_j^1 to this additional buffer in a synchronous manner, from where they are served by the WRR server. The queuing buffer B_j^1 now sees a perfectly uniform server and the cell loss observed is decoupled from the length of the server cycle (T) and depends only on the bandwidth allocated to the VP, thereby simplifying the call admission. An additional delay of T units is also added due to this rate matching buffer.

3.6 Summary

In this chapter, the traffic shaper based scheme introduced in chapter 2 has been extended to improve the link utilization achievable. The use of deterministic bandwidth reservation at the virtual path level has been proposed as an approach for achieving this. It is shown that such an architecture enables provision of end-to-end QoS guarantees in ATM networks, while maintaining high utilization in the presence of bursty traffic such as multimedia sources. Utilization is improved by exploiting statistical multiplexing between calls within the same Virtual Path in addition to traffic shaping. This approach is motivated by the fact that utilization does not suffer significantly between a full bandwidth sharing scheme and a partitioned scheme if the partitioning is sufficiently coarse. However, partitioning link bandwidth at the VP level enables provision of QoS guarantees over multiple hops and hence may well be worth the slight loss of utilization.

Several important properties of QoS measures such as cell loss over multiple hops have been analyzed. Using a queuing model of a VP under the proposed architecture, the problem of distributing end-to-end resources reserved for a VP over the different physical links traversed by the VP has been addressed. Some important properties (related to optimal resource usage) of a certain policy (the MGF policy) have been proved. Finally, the working of a simple round robin scheduler which implements the proposed approach has been outlined. The complexity of implementation of this approach has been shown to be similar to employing cell spacers which have already been proposed for use in ATM networks.

The proposed architecture requires efficient allocation of resources such as bandwidth to Virtual Paths. In the following chapter, a dynamic resource allocation algorithm is developed. This can be used to determine on-line, the optimal bandwidth allocation for a VP in the proposed architecture. In fact the technique is shown to be more general and can be applied to optimally control any resource whose allocation level controls the desired performance objectives in a monotonic manner.

Chapter 4

Dynamic Resource Allocation For Providing QoS Guarantees

4.1 Introduction

In chapter 3 a VP level resource reservation scheme has been proposed for providing end-to-end QoS guarantees along with high resource utilization. The problem of distributing given resources over different physical links of a VP has been addressed. However, it is not clear how to optimally allocate resources to a VP. Approximate techniques such as the equivalent capacity formulas [25] can be used for bandwidth allocation, but these are known not to be always accurate. Also, these are not applicable to resources other than bandwidth and performance measures other than cell loss.

In this chapter, a method is developed for determining the minimal resource requirements needed to guarantee a specified QoS measure. This technique is based on on-line measurements of the QoS actually received and feedback of this information for controlling the resource allocation. This technique is referred to as **REQS** for **R**esource-**E**fficient **Q**uality of **S**ervice. Thus this technique can be used for determining optimal allocation levels of resources other than just bandwidth and QoS definitions other than average cell loss probability. In this chapter, control of the resources of a single queue is addressed. This technique can directly be used to control end-to-end resources in case of a path level reservations based architecture such as the one proposed in chapter 3. In the architecture proposed in chapter 3, all the queuing essentially occurs at a single node of a path so that control of this node can be used to control end-to-end performance. For instance, the **REQS**

algorithm is used for determining the minimum service rate needed to satisfy a specified CLP constraint. Since, with the MGF policy outlined in chapter 3, cell loss occurs only at the first hop of a VP and the bandwidth reserved at all hops is the same, this algorithm can be directly applied for determining the minimum bandwidth assignment to a VP in order to guarantee an end-to-end CLP constraint.

The problem of providing QoS guarantees without being excessively conservative in utilizing resources is difficult for variable bit rate (VBR) sources. For such sources, the traffic offered to the network is "bursty" and somewhat unpredictable (as has been illustrated in Table 1.1). As an example, a video codec may generate data at a time-varying rate depending on the (unpredictable) video content. This does not, however, eliminate the need for guarantees on the end-to-end delay and losses when the video is transmitted across the network.

The conventional approach for dealing with this problem is to have each user declare in advance the characteristics of their traffic. As an example, the ATM Forum has adopted a standard set of 3 traffic descriptors: peak cell rate, sustainable cell rate and maximum burst size [1]. A policing mechanism such as the leaky bucket [1] ensures that the user's traffic conforms to the declared values for these descriptors. Using this information, an appropriate Call Admission Control (CAC) algorithm determines whether the QoS requirements of a new call can be met, without compromising the QoS levels of calls that have already been admitted.

There are problems with using this approach, including the following:

- Users have to be able to characterize their traffic generation process reasonably accurately; this is not always feasible. While a video sequence generated off-line can be analyzed very thoroughly, a user initiating a video conference may have very little idea of the expected traffic statistics.
- Even if users have the ability to model the traffic generation process, a small number of traffic descriptors may not be sufficient to characterize the traffic generation process well enough to predict the expected QoS accurately. For instance, El-Sayed and Perros [16] showed that two sources with the same peak rate, mean rate and mean burst length could experience very different cell losses because of differences in higher order statistics of the sources.

- Deriving an accurate CAC algorithm which can be executed on-line without excessive numerical computation is difficult even for simple traffic model approximations [25].
- Even if the characteristics of individual traffic sources are known, characterizing aggregations of traffic streams (such as VPs in the architecture proposed in chapter 3) is non-trivial and approximate. This approximation is likely to result in sub-optimal allocation.

A consequence of these problems is that the service provider can either end up over-allocating resources in order to guarantee QoS, thereby sacrificing efficiency, or under-allocating resources and providing poor service quality, thus losing customers.

In this chapter, we explore an alternate approach to efficient resource allocation and providing strong QoS guarantees. In this approach, resource allocations are adjusted dynamically on the basis of the QoS actually received. The main benefit of this approach is that very little information related to traffic characterization is needed from users. The method uses the minimum network resources required to guarantee the specified QoS requirements for any source with a stationary traffic generation model. The main drawback of this approach is that for a short initial period the requested QoS may not be provided. Also, some re-allocation of resources will be required during this initial period. However, the effects of these problems can be reduced through engineering solutions. This technique also involves a slight additional cost in implementation since provision must be made for measuring queuing performance on-line.[1]

A number of researchers have explored the use of dynamic and adaptive techniques for network resource allocation. Jeon and Viniotis [31] developed algorithms which dynamically vary the scheduling priority of different traffic classes in a multi-class queuing system, in order to meet QoS specifications such as loss probability and average queuing delay. They showed that these algorithms were able to achieve any desired performance vector which was feasible under the class of work-conserving service policies. Clark et al [12] developed the FIFO+ queuing discipline, in which the scheduling priority of a packet at a node is dynamically adjusted. This adjustment is based on the queuing delay seen at the previous node. They showed that this resulted in improved delay performance as compared to static

[1] However, performance monitoring modules are already expected to be deployed in ATM switches [25].

allocation of priorities. However, the improved resource utilization was obtained at the cost of not being able to *guarantee* the QoS seen by each user. They claimed that strong guarantees on QoS were not necessary for a type of service referred to as predictive service. Jamin et al [30] extended these notions and tested multimedia applications under predictive service. In [11], Chong et al examined dynamic bandwidth allocation algorithms for MPEG video. They examined two different approaches for predicting the rate required to transmit an MPEG frame based on data from the previous frames (one based on a Recursive Least Squares type prediction and the other using Hopfield Neural networks). They demonstrated the performance improvements over static bandwidth allocation techniques. However, in their study, they did not seek to determine a steady state "effective" bandwidth. Hence bandwidth re-allocation is continually required during the duration of a session. The algorithm developed in this chapter determines a steady state effective bandwidth so that once the algorithm has converged, no bandwidth re-allocation is necessary. In [27], Hsu and Walrand develop an algorithm for determining the effective bandwidth dynamically. They prove theoretically the convergence of the dynamic rate to the true steady state effective rate. However, the proof of convergence is limited to markov-modulated fluid type sources. Also, they do not present any performance analysis of the algorithm with real sources and do not address the issue of optimally selecting the algorithm parameters for practical implementation. In contrast, this study concentrates on performance analysis, tests the robustness of the algorithm developed under different conditions, and demonstrates the effect of algorithm parameters on the dynamic behavior. In [8], an algorithm for rate allocation for the ABR (Available Bit Rate [1]) service has been proposed. However, the emphasis there is not to guarantee any QoS measures, since applications which require ABR service are typically non-real-time (like file transfers) and only require a best-effort service. The algorithm developed in this chapter is applicable to real-time variable bit rate (VBR) type traffic which require QoS guarantees.

The performance analysis of this technique is limited mostly to the case where bandwidth is the resource of interest and average cell loss probability is the QoS measure of interest. Dynamic rate control of a single queue is studied, where the service rate of this queue is synonymous with the term "bandwidth". In case of flow control policies such as Weighted

Round Robin (outlined in the previous chapter) and Weighted Fair Queuing [44], the end-to-end bandwidth is simply this rate multiplied by the number of links traversed. An algorithm is developed which dynamically obtains the minimum bandwidth necessary to satisfy a specified QoS for a given (arbitrary) source. The effectiveness and robustness of the algorithm is demonstrated by varying the measurement frequency, source burstiness, buffer size and CLP specifications. It is shown that this algorithm can yield significant bandwidth savings over popular off-line approaches, such as the "equivalent capacity" method [17]. This approach can also be used to derive the minimum requirements for other resources (such as buffer space), and for other QoS measures (such as queuing delay percentiles). This approach dynamically determines the "effective resource" requirement to satisfy a given QoS measure for an arbitrary (stationary) source. Thus it may be seen as a generalization of the "effective bandwidth" approach. This method has implications for the design of traffic shapers, and for call admission in cell scheduling policies such as the WRR scheme introduced in chapter 3, Weighted Fair Queuing [44], Stop& Go [23], RCSP [56], HRR [32] etc, all of which are based on some form of rate allocation.

Section 4.2 explains the dynamic algorithm for bandwidth allocation. Section 4.3 presents simulation experiments, along with an analysis of the results. Section 4.4 compares the steady state and the dynamic behavior of this algorithm with some alternate approaches for bandwidth allocation. Section 4.5 summarizes the results of this study.

4.2 Description of the Algorithm

4.2.1 Queuing Model and Definitions

This chapter introduces an algorithm which dynamically adjusts the instantaneous service rate $\mu(t)$ of a finite buffer queue in order to meet a specified QoS performance measure $\mathcal{Q}$. In conformance with the ATM standard, all arriving cells are assumed to require the same amount of service. A cell which arrives at a full queue is lost due to buffer overflow. Figure 4.1 shows a model of the queuing system under consideration.

As shown, observations of the queuing behavior are used by a control algorithm to adjust the service rate of the queue. Let B be the (fixed) buffer size of the system. Decisions about adjustments to the resource allocation are made at discrete instants t_i; the interval between

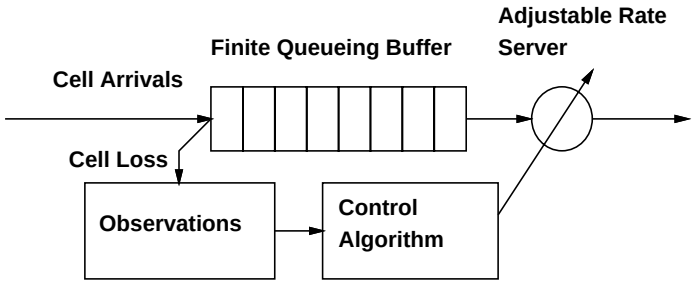

Figure 4.1: Model of the queuing system under consideration.

decision points t_n and t_{n+1} is referred to as the nth update interval U_n. The service rate over the nth update interval, denoted μ_n, is assumed to be constant.

For a given sample path of cell arrivals, let $\mathcal{A}_n$ denote the number of arrivals and $\mathcal{L}_n$ denote the number of cells lost due to buffer overflow in the nth update interval. The average loss probability in the nth interval is denoted $\mathcal{P}_n = \mathcal{L}_n/\mathcal{A}_n$. The cumulative loss probability for all intervals up to and including the nth interval is denoted $\mathcal{P}_{0..n} = (\sum_{i=0}^{n} \mathcal{L}_i)/(\sum_{i=0}^{n} \mathcal{A}_i)$. The cell loss probability during the nth interval, will be referred to by the term "current loss probability". The term "cumulative loss probability" refers to $\mathcal{P}_{0..n}$.

The goal for this method is the following. The instantaneous rate $\mu(t)$ should converge to a steady state rate μ^*, and the cumulative loss rate $\mathcal{P}_{0..n}$ should converge to a steady state value $\mathcal{P}^*$. $\mathcal{P}^*$ should be no more than the specified average cell loss probability $\mathcal{Q}_l$. Further, it is desired that the steady state rate μ^* be the minimum rate for which this statement is true. In addition to this goal for the steady-state behavior, the method should converge quickly, and be practical to implement.

4.2.2 An Algorithm for Resource Allocation Based on QoS Measurements

The algorithm is motivated by the basic result from Theorem 1 according to which the total number of losses at a queue increases monotonically as the service rate is decreased (if all cells have the length). The algorithm iterates over successive update intervals. The basic step at each iteration is given in Equation 4.1.

$$\mu_{n+1} = \mu_n + K_n E(\mathcal{P}_n, \mathcal{Q}_l) \tag{4.1}$$

In this equation, $E()$ is an error function which is a measure of how far the current cell loss probability is from the targeted QoS. We chose the error function to be the natural logarithm of the ratio of the current and desired loss probabilities i.e. $E(\mathcal{P}_n, \mathcal{Q}_l) = \log(\mathcal{P}_n/\mathcal{Q}_l)$. This means the rate is increased when the current loss probability is higher than the target, and decreased when it is smaller than the target. By Theorem 1, this results in moving the cumulative loss probability in the right direction. The logarithm function is appropriate because of the wide dynamic range (several orders of magnitude) of losses expected to be measured. In addition, an approximately logarithmic relation between service rate and loss probability has been suggested in the literature. Using this iteration, the instantaneous rate keeps changing as long as the current loss probability ($\mathcal{P}_n$) is not equal to $\mathcal{Q}_l$. Hence convergence of the instantaneous rate implies convergence of the current loss probability.

Figure 4.2 describes the entire algorithm using pseudocode. The algorithm has two distinct modes of operation. In the first mode, the algorithm tries to quickly converge to the neighborhood of the desired loss probability. In this mode, the lengths of the update intervals are kept fixed, and the scalar K_n is varied. In the second mode, the algorithm tries to gradually converge exactly to the desired loss probability. In mode 2, the scalar K_n is kept fixed, but the size of the update intervals is varied.

We call this the **REQS** (pronounced "rex") algorithm, which stands for **R**esource-**E**fficient **Q**uality of **S**ervice. The algorithm starts in mode 1 with an initial value K_0 for the scalar K_n and an initial update interval U_0. K_n is increased linearly until the sign of the error in the current loss probability changes in successive update intervals. K_n then decreases geometrically at each decision instant until it drops below some value K_∞, at which point the algorithm switches to mode 2 operation. In this mode, K_n stays fixed at K_∞. Each time the error changes sign in successive update intervals, the length of the interval is doubled. The algorithm operates in mode 2 for the remaining duration of the call.

The motivation for this formulation of REQS follows. The error term which drives the algorithm is based on the loss probability only over the last update interval; that is, the current loss probability is used, not the cumulative loss probability. In mode 1, the algorithm tries to quickly get "reasonably" close to the target. The variation described for K_n first increases the sensitivity of the rate control and then decreases it. The linear increase

```
/* K_0, K_∞ and U_0 assumed to be given */
/* initial values: inc_flag = TRUE, mode = 1 */

curr_error ← log(P_n/Q_l);
prev_error ← log(P_{n-1}/Q_l);

if (mode = 1) {
        U_n ← U_{n-1};
        if ((curr_error × prev_error > 0) && (inc_flag = TRUE))
                K_n ← K_{n-1} + K_0;
        else {
                inc_flag ← FALSE;
                K_n ← K_{n-1}/2;
                if (K_n ≤ K_∞) {
                        K_n ← K_∞;
                        mode ← 2;
                }
        }
}
else {                  /* mode = 2 */
        K_n ← K_{n-1};
        if (curr_error × prev_error < 0)
                U_n ← 2 × U_{n-1};
        else
                U_n ← U_{n-1};
}
```

Figure 4.2: Algorithm for varying K_n and U_n at decision instant t_n

and geometric decrease in the scalar value has been found experimentally to result in the best dynamic behavior of several variations tried. However, the choice of these rules is not critical to the convergence of REQS. The geometric decrease in the scalar value corresponds to a binary search, since the service rate and loss rate are monotonically related.

An assumption in mode 1 operation is that loss probability measurements over each update interval are statistically significant. If the update interval is too small, however, the loss probability measured over this interval can be considerably far from the average value since losses typically occur in bursts and not uniformly. On the other hand, a very large update interval may increase convergence time. For this reason, the algorithm starts with a small update interval, and then increases it over time in mode 2. By limiting these increases to instants when the error changes sign, the update interval is prevented from becoming excessively large. During mode 2 the algorithm operates at minimum sensitivity (since the scalar is at its minimum value). A change of sign in the error is hence likely to be due to noisy measurements; so that increasing the update intervals yields better confidence in the loss measurements. Further motivation for the formulation of REQS is presented below in the discussion of experimental results.

4.3 Numerical Results

Analytical treatment of transient behavior of queues yields closed form solutions in very few cases. For instance, closed form solutions for time-dependent queue performance measures such as buffer occupancy are available only for simple queues such as the $M/M/1$ queue [39]. Transient analysis of a queue with variable service rate and bursty input traffic is consequently not expected to yield closed form results. Hence simulation has been used to analyze the performance of REQS.

4.3.1 Simulation Model and Procedure

The simulation model was shown earlier in Figure 4.1. A two-state Markov Modulated Bernoulli Process (MMBP) is used as the traffic source in most experiments. This model is used extensively in simulation and analytical studies of bursty multimedia traffic [16]. The 2 state MMBP is in either of two states $S0$ or $S1$. In $S0$ cells are generated according

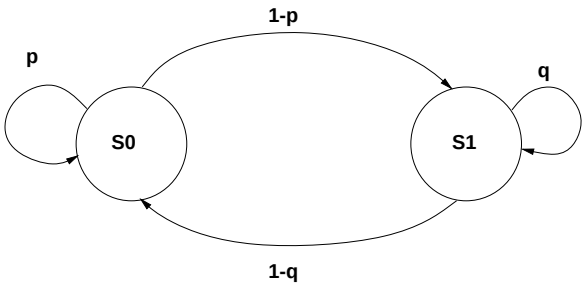

Figure 4.3: Model of MMBP Source used for simulations of dynamic bandwidth allocation algorithm.

to a Bernoulli Process with a mean rate of λ_0 cells/sec, and in $S1$ with a mean rate λ_1 cells/sec. The durations of the two states are geometrically distributed with means γ_0 and γ_1 seconds. A measure of the burstiness of the source is the squared coefficient of variation of the inter-arrival times, which can be determined from the above parameters [16]. For most experiments, values for these parameters are chosen as $\lambda_0 = 200$ cells/sec, $\lambda_1 = 1000$ cells/sec, $\gamma_0 = .02$ seconds, $\gamma_1 = .01$ seconds. These values reflect typical behavior for voice and video, but scaled down to reduce the simulation time required. In some of the experiments these values are varied to investigate the sensitivity of the algorithm to different traffic conditions. One experiment also uses parameter values based directly on MPEG-compressed videos, to test more realistic conditions.

To determine whether the dynamic algorithm converged, sample paths of the instantaneous rate and the loss probability have been examined. If the algorithm converges to the right steady-state cell loss probability, Theorem 1 guarantees that the instantaneous rate is the minimum that would achieve that QoS. Experimental evidence is needed, however, to confirm that convergence will occur. The *convergence time* is defined to be the time required to arrive at and stay within 5% of the steady state rate. The steady state rate is determined by running the simulation for long enough time (typically for five hours or more of simulated time), to ensure that no further rate changes would occur. This definition of convergence time corresponds to the term "relaxation time" used in transient analysis of queuing and control systems[53].

The method of batch means has been used to determine convergence time. In each batch, a number (typically 3-5) runs have been simulated of the queuing system with the dynamic

rate control for sufficiently long times to obtain convergence. The error with respect to the steady state rate for each run is averaged over the batch to obtain an averaged error curve. The convergence time obtained from this curve is taken as the convergence time of this batch of runs. A number of batches (typically 15-20) are executed for each setting in order to obtain the average convergence time along with the confidence intervals. 90% confidence interval levels are calculated in all cases, and are shown on the graphs. In this work it is assumed that the service rate can be set to any real value; there is no quantization effect due to being limited to a discrete set of rates. In all experiments, the initial service rate μ_0 is set to the mean rate λ_1 of the MMBP in its high state. λ_1 roughly represents the peak rate of this source, so this corresponds to a conservative initial allocation. The values of K_0 and K_∞ have been both set equal to about 1% of λ_1 in all cases. Note that K_∞ determines the final sensitivity of the rate control and hence must be small. Larger values of K_∞ have been found to result in improved convergence times, at the expense of accuracy in the steady state rate. The value of K_0 has not been found to be critical. Since the scalar is multiplied by the error term $\log(\mathcal{P}_n/\mathcal{Q}_l)$, adjustments in the rate are typically much greater than 1% of the peak rate. During intervals in which there is no cell loss, the value of $\mathcal{P}_n$ is set to an appropriately small non-zero value rather than zero. This value has been set equal to 2 orders of magnitude smaller than the target loss probability, in the results reported.

Simulation results on the performance of the algorithm under different conditions are now presented.

4.3.2 Variation of Convergence Time with Initial Update Interval

The first experiment measured the sensitivity of REQS to the choice of the initial update interval, U_0. The initial update interval was varied over the values .5, 5, 50, 200, and 500 seconds. The resulting convergence time has been plotted in Figure 4.4 for buffer sizes of 20 and 80 cells, and a target CLP of 1e-3. These graphs indicate that convergence time is relatively unchanged as long as the initial update interval is small enough.

The length of the update interval has a strong effect on the convergence time. A short update interval results in more opportunities for the rate to correct itself in a given length of time. However, since the algorithm is driven by the current loss probability, a smaller

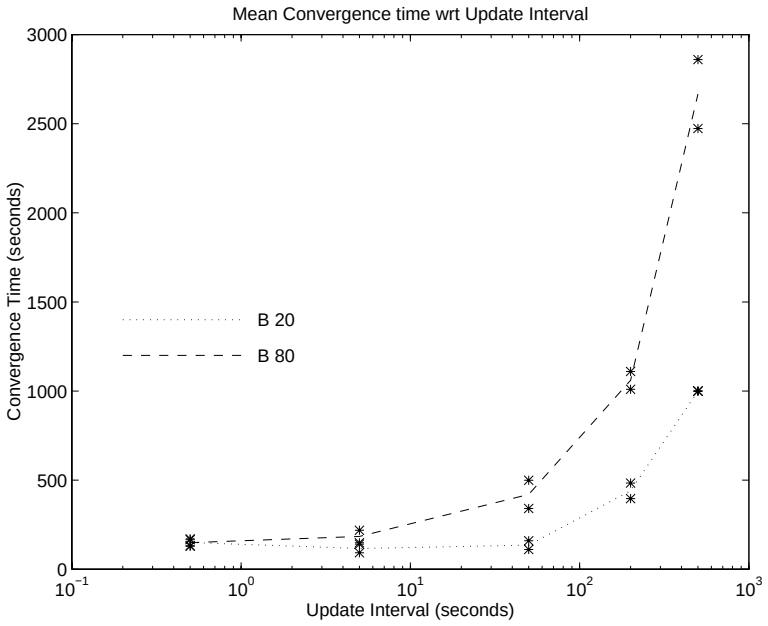

Figure 4.4: Convergence time with varying initial update interval. Standard source, target CLP 1e-3.

interval also implies that the measurements of current loss probability may not be statistically significant. A large update interval on the other hand, results in better estimates of the current loss probability but convergence time can increase due to the relative infrequency of making corrections. Clearly, determining a good value for the update interval is not straight forward and would likely depend on the source characteristics.

Since REQS is driven by the error in the loss probability, the convergence time is closely related to the inherent relaxation time of the time averaged loss probability of the given queuing system. The relaxation time of the loss process is quite difficult to predict, and can be significantly greater than the time for the arrival statistics to converge. Nevertheless, Figure 4.4 demonstrates the robustness of REQS to the choice of the initial update interval.

Figure 4.5 presents the convergence times for the same experiments ($B = 80$ only), but measured in terms of the required number of updates. This metric is important because reallocation of resources may be an expensive operation. A small initial update interval requires a larger number of rate changes, while a large initial update interval only requires about 5 updates. Note that the number of updates does not directly scale from the previous

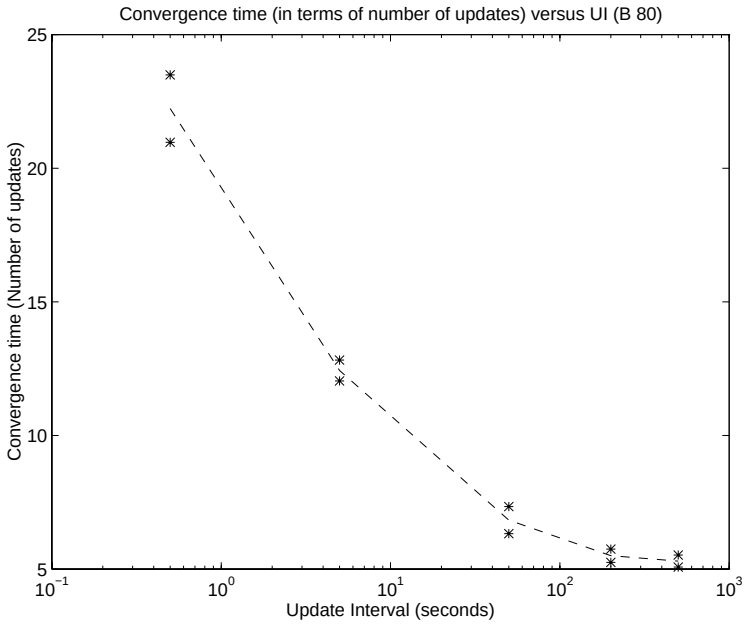

Figure 4.5: Convergence time (in terms of number of updates) with varying initial update interval. Standard source, target CLP 1e-3, $B = 80$.

graph, since the doubling of the update interval starts at different times for different values of U_0.

There is thus a tradeoff between absolute convergence time and required number of updates. Given an arbitrary source, a smaller U_0 should result in less convergence time, while a larger U_0 should result in fewer number of updates. In these experiments, an update interval of about 50 seconds represents a good compromise of these two factors. The best "balance point", however, is likely to depend on the exact source characteristics, which are assumed to be unknown.

Figure 4.6 shows sample paths of the instantaneous rate for four values of U_0 ($B = 80$ only). Although a formal proof of convergence has not been provided, data like this strongly supports our assertion that the algorithm converges. In fact, REQS has been observed to converge in every case tested, for all experiments listed here. The total simulated time is more than ten times the calculated convergence time, which is a strong indication that the rate has indeed converged. Figure 4.7 expands the initial portion of these graphs to illustrate the behavior of the method during convergence.

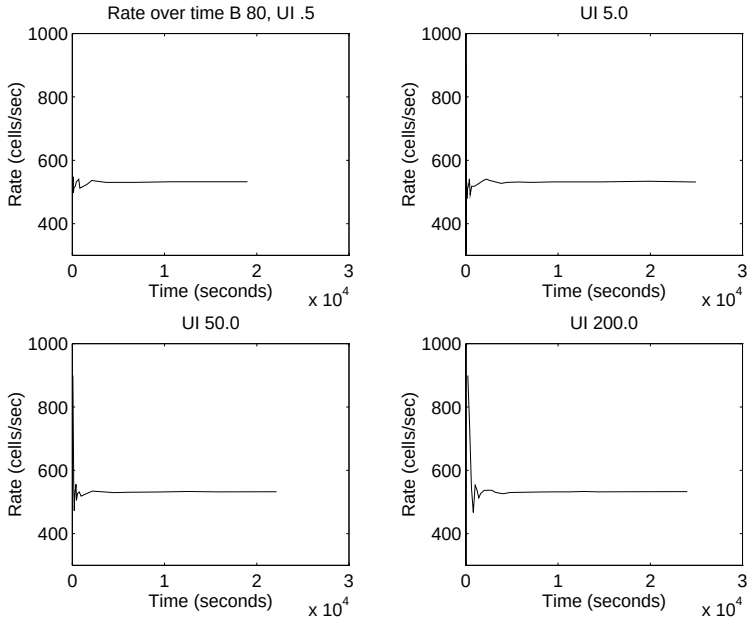

Figure 4.6: Sample paths of rate for different initial update intervals. Standard source, entire simulated time, target CLP 1e-3, $B = 80$)

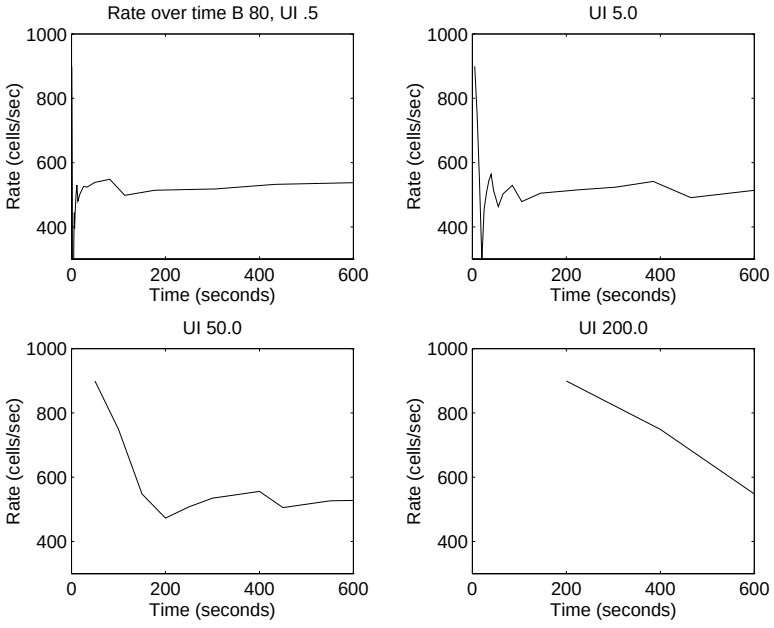

Figure 4.7: Initial portion of sample paths of rates for different initial update intervals. Standard source, target CLP 1e-3, $B = 80$

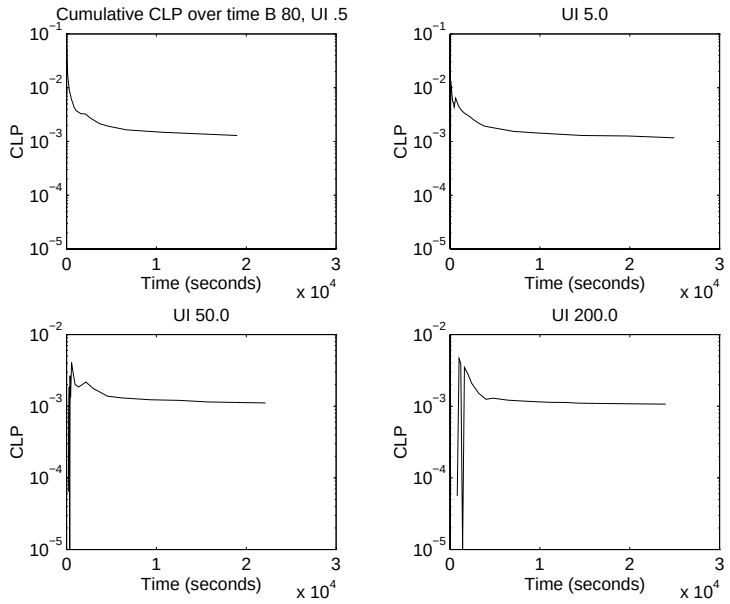

Figure 4.8: Sample paths of cumulative CLPs for different initial update intervals. Standard source, target CLP 1e-3, $B = 80$.

Data on the resulting losses incurred during these same experiments was also collected. Figure 4.8 shows the cumulative loss probabilities (i.e. $\mathcal{P}_{0..n}$) for the above sample paths,[2] while Figure 4.9 shows the loss probability in the current update interval (i.e. $\mathcal{P}_n$). These figures demonstrate that the cumulative loss probability is accurately controlled, even though REQS considers only the current loss probability. The accuracy with which the loss probability is controlled supports our claim of providing QoS *guarantees*.

Another experiment was conducted, with the same traffic source and buffer size, to determine how close to optimal the proposed algorithm is. In this experiment, the service rate was fixed from the beginning at the rate converged to in the previous experiment. The purpose was to monitor loss behavior of a queue which was initially allocated exactly the right service rate to achieve the desired CLP. The cumulative cell loss probability was then measured for each of 10 different runs, and the error in this CLP (percent difference from the desired CLP of 1e-3) was calculated. Figure 4.10 plots the error in the CLP for this "optimal" method. It can be seen that the time taken for the error to drop to less than 10%

[2]In the plots shown, the cumulative CLP is reset when the algorithm switched from mode 1 to mode 2

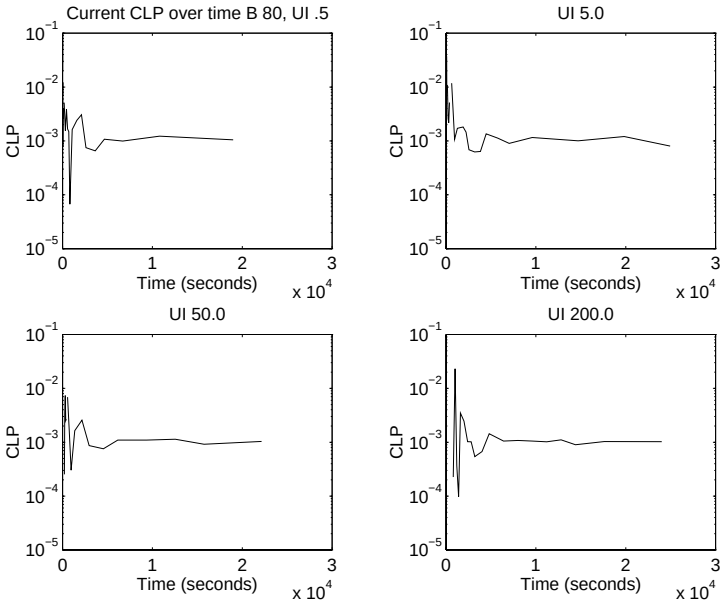

Figure 4.9: Sample paths of loss probability over current update interval for different initial update intervals. Standard source, target CLP 1e-3, $B = 80$

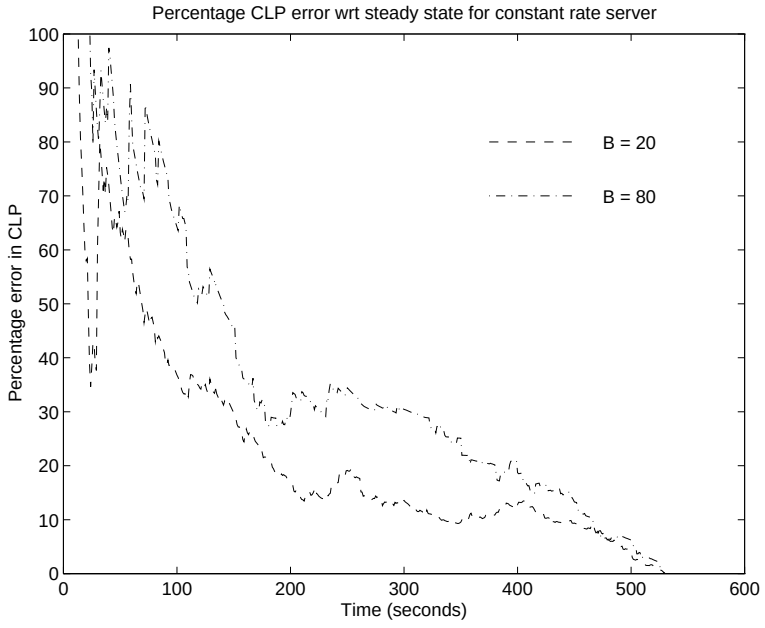

Figure 4.10: Fractional error of transient CLP with constant rate server. Standard source. Steady state CLP 1e-3.

is on the order of 200 seconds. Note that the convergence time of REQS for the same source is about 150 seconds with a small enough initial update interval. This is strong evidence that the convergence time is reasonable, and depends on the inherent relaxation time of the loss probability of the queue.

No closed form solutions are known for describing the transient measures of a queue with bursty arrivals such as MMBP sources. In [39], closed form solutions for the distribution of number of customers in an $M/M/1/K$ system conditioned on the initial number are presented. As a rule of thumb the author suggests a relaxation time of $1/(\sqrt{\mu} - \sqrt{\lambda})^2$ (where λ and μ are the average arrival and service rates respectively) for the distribution of the buffer occupancy of such a system. This formula indicates that relaxation time increases with utilization and decreases with the absolute value of the service rate. In this experiment, this is confirmed by the increased convergence time with a buffer of 80 cells versus 20 cells, as shown in Figures 4.4 and 4.10. As the buffer size is increased, a lower service rate can achieve the same average loss specification, leading to higher utilization.

Figure 4.10 also indicates that a few dozen seconds should be sufficient to approximate the long-term loss behavior of the queuing system. Thus, there is not much to gain by choosing U_0s much larger than 50-100 seconds, as has been seen in Figure 4.4.

4.3.3 Variation of Convergence Time with Buffer Size

The variation of convergence time with buffer size was studied next to determine sensitivity of REQS to this parameter. The size of the queuing buffer was varied over 5, 10, 20, 40 and 80 cells. The same source model and CLP specification (1e-3) were used as before. Figure 4.11 shows the results for two different values of U_0.

For U_0 = 5s, convergence time increases very slowly with increasing buffer size. It also indicates that convergence times should be reasonable even for very large buffer sizes. For U_0 = 200s, convergence time is longer, and exhibits a surprising behavior for small buffer sizes. The high convergence time for small buffer size is because the initial rate μ_o is actually below the final (converged) rate μ^*, as seen from the sample paths of the rates in Figure 4.12. For buffer sizes much smaller than the mean burst data length, it is known (e.g. [25]) that cell scale congestion during a burst is responsible for losses. The average

to negate the effect of the initial losses. However, this resetting is performed only once.

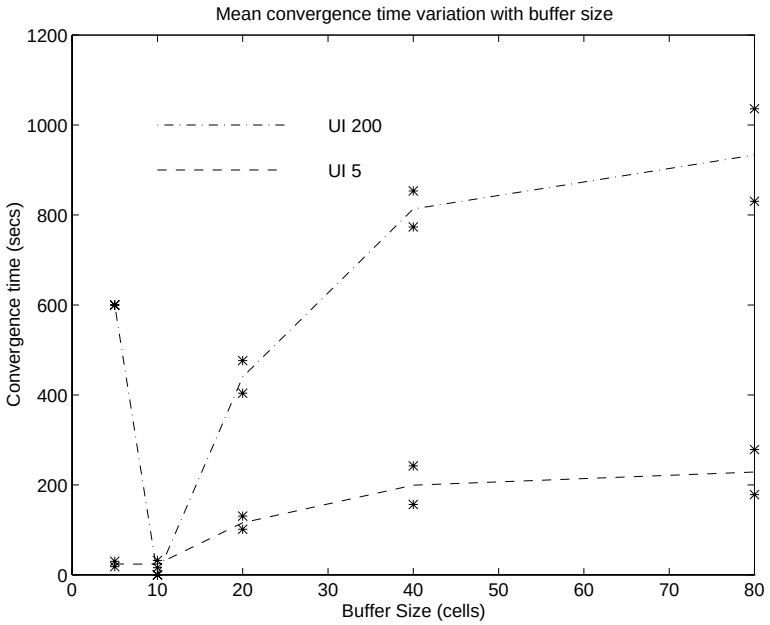

Figure 4.11: Convergence time versus buffer size. Standard source, target CLP 1e-3.

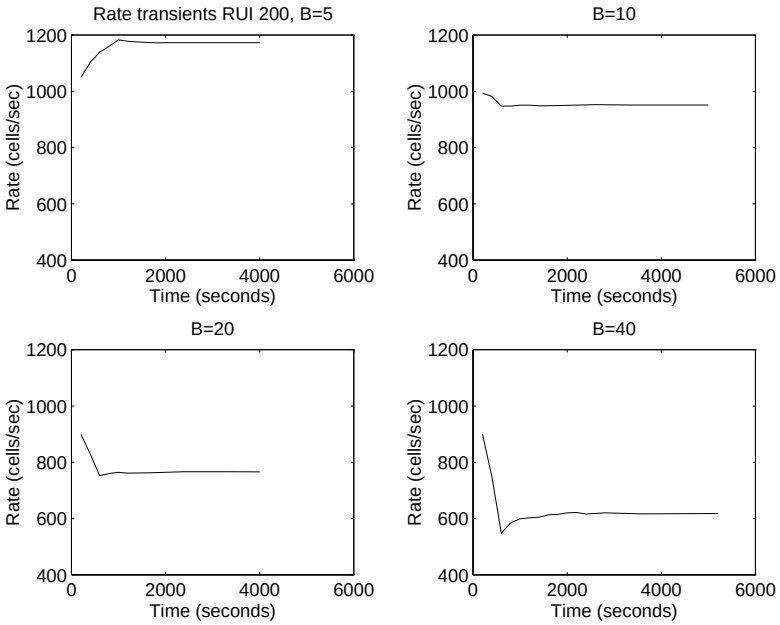

Figure 4.12: Sample Paths of instantaneous rates for buffer variation. Standard source, target CLP 1e-3, U_0 200s.

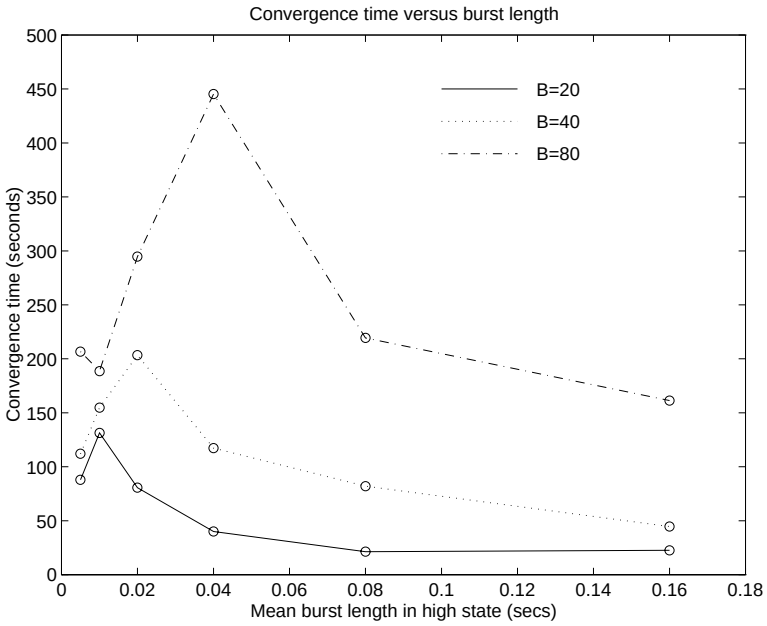

Figure 4.13: Convergence times for varying burst lengths. U_0 10s, Target CLP 1e-3. Confidence intervals not shown for clarity.

amount of data in a burst for this source was 10 cells which was twice the buffer size in this case. Hence, the effective rate can be greater than 1000 cells/secs. This unfortunate parameter choice demonstrates, however, that REQS converges to the correct value even when the initial rate is less than the final rate. Better convergence times for the case of small buffers can be obtained by setting the initial rate to a higher value.

4.3.4 Variation of Convergence Time with Burst Length

Another experiment investigated the sensitivity of the algorithm to changes in the characteristics of the traffic source. The burstiness of the source was varied, while keeping the same peak and mean rates. This was accomplished by varying γ_0 and γ_1 (the mean lengths of the two states), but keeping the ratio of the two constant.

The variation in convergence time with the burst length (duration of $S1$) is shown in Figure 4.13. Again, non-monotonic behavior is seen. Two opposing trends result in this non-monotonic behavior. As burst lengths increases, source burstiness increases (the squared coefficient of variation of the inter-arrival times increases for instance [16]). The relaxation

time of the loss probability can be expected to increase with source burstiness since the loss process will also become more bursty. Such an observation was also made by Wang et al [53] in their study of transient analysis of queues with bursty traffic. As source burstiness increases, queue service rates must increase (meaning lower utilization) to achieve a given loss rate. Lower utilization levels tend to reduce the inherent relaxation times, as described in section 4.3.2, and hence the convergence time of REQS. For very small burst lengths, the effect of increase in burstiness is dominant, while for larger bursts, the effect of reduced utilization is dominant. This results in the non-monotonic behavior seen. Analogous non-monotonic behavior was also seen in [53], where the maximum transient overshoot first increased, then decreased as the burstiness of the arrival process was increased.

4.3.5 Variation of Convergence Time with Peak-to-Mean Ratio

In this experiment, the peak-to-mean ratio of the source was varied to see how this affects convergence. This variation was accomplished by varying γ_0 (the mean duration of the low state), while keeping the rates in the two states as well as the mean duration of the high state fixed. The mean duration of the low state was varied over .01, .02, .04, .08 and .16 seconds. Here, U_0 was set to 10 seconds and the buffer size was 20 cells.

The variation of the convergence time is shown in Figure 4.14. The burstiness of the source increases with the peak-to-mean ratio. The final (converged) rates were found to decrease monotonically as peak to mean ratio was increased, so that utilizations increased with the peak-to-mean ratio. Hence both trends are now in the same direction, and convergence times increase monotonically with the peak-to-mean ratio.

4.3.6 Variation of Convergence Time with CLP Requirement

The sensitivity of REQS to the QoS specification, (in this case the target CLP) was also investigated. The target CLP was varied over 5 orders of magnitude (1e-1 through 1e-5), with a U_0 of 10 seconds and $B = 20$. The results are shown in Figure 4.15. Convergence time increases with lower loss requirement, since the algorithm must estimate the CLP with rarer events.

The sharp rise in convergence time for lower loss probabilities indicates convergence may be slow for very low loss requirements, such as 1e-8 or 1e-9. As discussed elsewhere

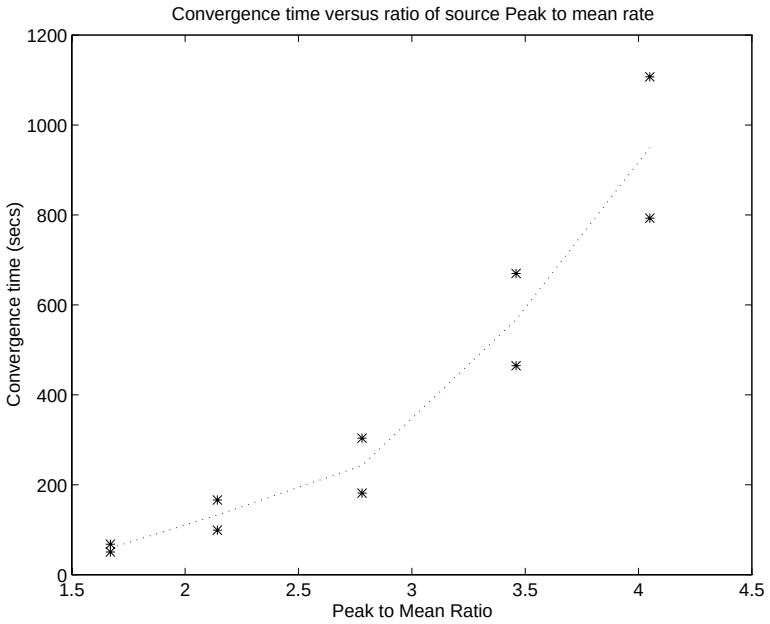

Figure 4.14: Convergence times for varying peak-to-mean ratio, (mean low state duration varied). Target CLP 1e-3, U_0 10s, B 20 cells

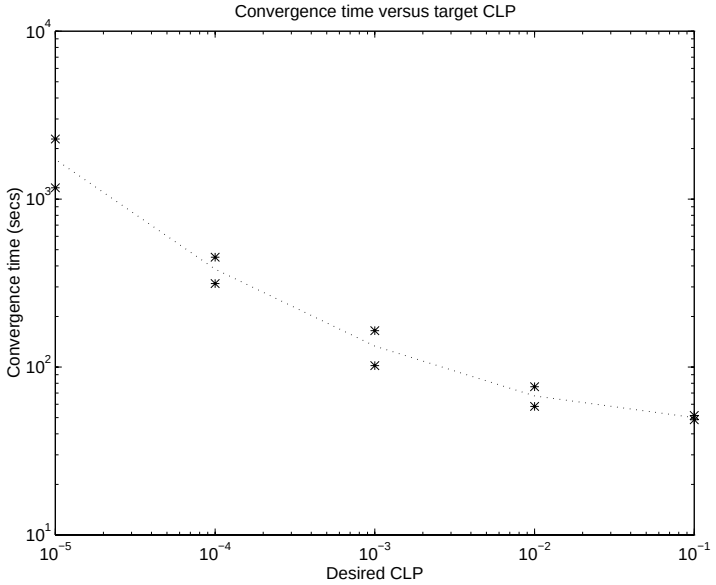

Figure 4.15: Convergence times for varying target CLP. Standard source, U_0 10s, $B = 20$.

(e.g. [42]), a steady state loss probability for very low probability events makes sense only over very long time scales. A user who specifies a very low CLP must understand this limitation. It is possible the convergence time can be reduced using techniques developed for fast simulation of rare events[52].

4.3.7 Convergence Times at Video Rates

An approximate 2 state MMBP model of a compressed video source was constructed and used as a traffic source for this experiment. This was done to get some idea of the performance and robustness of REQS at typical video rates.

The source was modeled as a 2 state MMBP with high rate of 28000 cells/sec and low rate 5000 cells/sec. Average On and Off times were set to 30 msec and 60 msec, to approximately model an IBBPBBI-type MPEG-compressed video. The rates in the two states are based on figures for the average amount of data in I and P frames and in B frames, respectively, as presented in [29]. The buffer was set to 1000 cells in order to be larger than the average burst length of 840 cells. The target CLP was set to 1e-4.

Figure 4.16 shows the behavior for this source as a function of varying initial update interval. Qualitatively the behavior is similar to that in Figure 4.4. Conclusions about the algorithm also seem to hold true for this more realistic traffic source. The convergence times (about 300 seconds) of REQS appear to be reasonable, since a video session is expected to last up to several hours.

4.3.8 Resource Allocation Algorithms for Other/Multiple QoS Measures

As mentioned in the beginning, the basic technique developed here can be used to determine optimal allocations for other kinds of resources and other forms of QoS specifications. In this experiment, REQS is used to determine the minimum resources to achieve a different type of quality of service. In this experiment, a bound of 25 msecs was set on the queuing delay, and it was required that no more than 1% of cells experience a queuing delay in excess of this bound. Such a QoS requirement may be specified for an interactive multimedia session in which bounding the fraction of excessively delayed cells is important. Instead of measuring the loss probability, the algorithm now measured the fraction of cells which

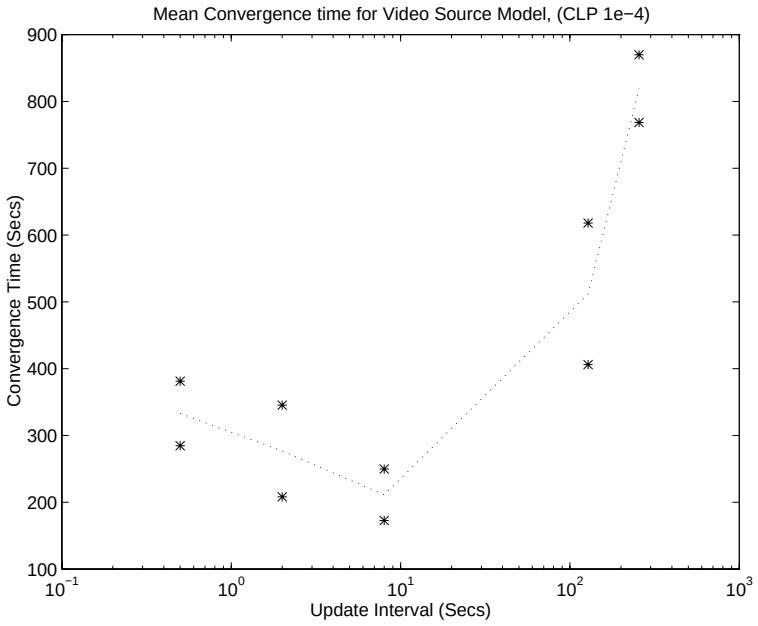

Figure 4.16: Convergence times for varying U_0. MMBP model at video rates, target CLP 1e-4, $B = 1000$.

experienced a queuing delay in excess of the given bound. This measurement was used to modify the allocated service rate.

The standard source for the earlier experiments was used, with a buffer size of 100 and an initial update interval of 10 seconds. One sample path of the instantaneous rate and current percentage of cells which experience queuing delay greater than 25 msecs is shown in Figure 4.17. Clearly, REQS can also be used to control a QoS specification defined in this manner.

Theorem 2 proved that the buffer size also monotonically controls cell loss. Clearly dynamic control of buffer size using the same technique can be used to obtain the "effective buffer" size needed to support a specified cell loss constraint. This illustrates that REQS can be used for determining "effective resource" requirements for different resource types and QoS definitions.

For the case where multiple QoS metrics are specified, REQS can be modified to obtain the minimum bandwidth at which all specified QoS measures can be met. This can be done by simultaneously tracking all the QoS measures of interest and setting the rate to be the

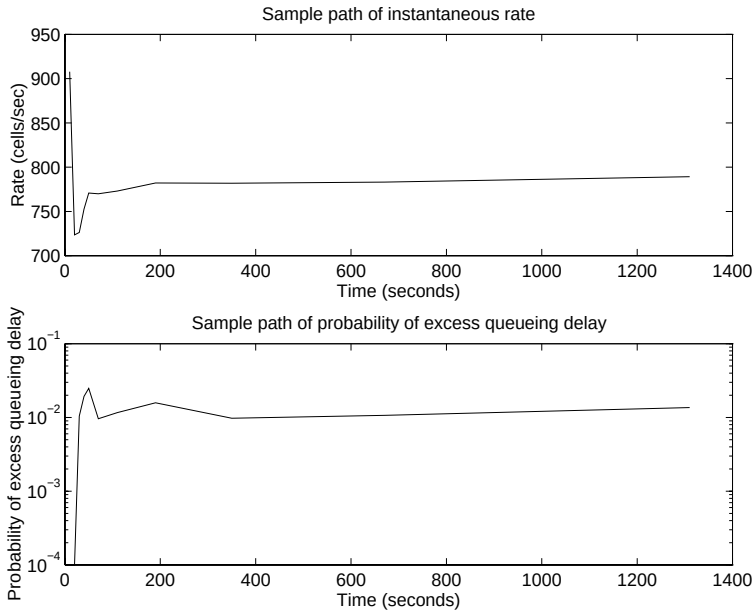

Figure 4.17: Example of use of REQS for a different QoS measure (percentile of queuing delay). Need fraction of cells with queuing delay greater than 25msecs no more than 1%. Standard source, $B = 100$, U_0 20 seconds.

maximum of that required to meet each of the measures independently. If the service rate monotonically controls all the measures, the rate will converge to the value corresponding to the most severe QoS requirement.

This concludes the experimental validation of the proposed algorithm. The next section compares REQS with other methods of bandwidth allocation.

4.4 Comparison with Other Approaches

4.4.1 Comparison of Steady State Performance with Equivalent Capacity Formulas

The equivalent capacity formula [17] is a widely-cited method for estimating the bandwidth needed to achieve specified loss, for certain types of traffic sources (markov-modulated fluids). The steady state service rate converged to by REQS is now compared with the rate calculated from the "equivalent capacity" formulas.

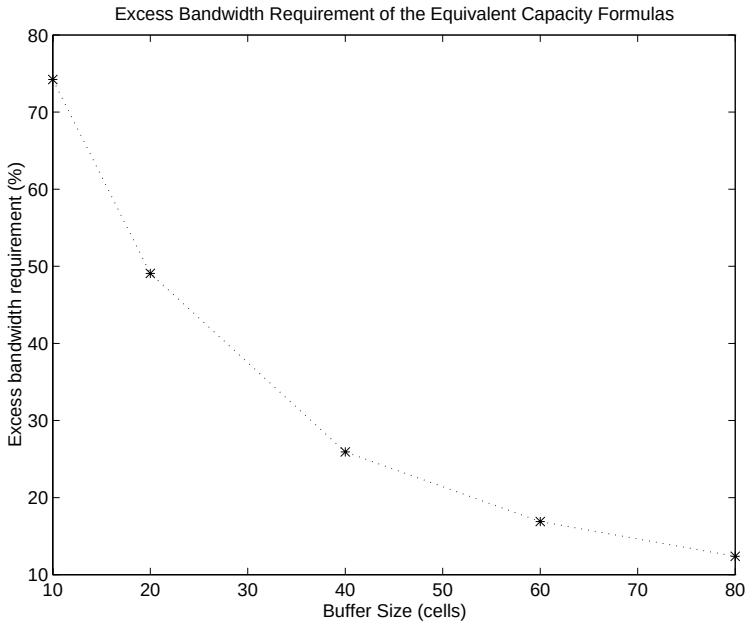

Figure 4.18: Percentage excess bandwidth predicted by Equivalent capacity formula. Standard source, deterministic distribution of state durations, target CLP 1e-3, $U_0 = 20s$.

The same standard source as in the earlier experiments is used, but with deterministically distributed durations of S_0 and S_1 (the two states of the bursty source). This experiment uses deterministic distributions for the state durations to illustrate one problem for the equivalent capacity formulations (which were derived for exponentially distributed state durations) which does not affect REQS.

The queuing system of Figure 4.1, with REQS as the method of resource allocation, was simulated to find the steady state rate. The equivalent capacity for this same traffic source was calculated using the method of [17], for a target loss rate of 1e-3. These two service rates were then compared. Figure 4.18 shows how much more bandwidth was required by the equivalent bandwidth formula, than predicted by REQS (for the same loss rate). This comparison has been performed for several buffer sizes.

The bandwidth savings is always non-negligible. For small buffer sizes, the equivalent capacity formulation overallocates the bandwidth by as much as 75%. Guerin et al. [25] have described a number of situations in which the equivalent capacity formula is either overly optimistic or overly pessimistic. REQS, in contrast, always converges to the minimal

rate to achieve a specified loss probability, for a wide range of traffic characteristics.

The equivalent capacity formulas ignore the potential for bandwidth reduction due to statistical multiplexing [25]. As a result, they result in overallocation of resources when applied to an aggregation of sources. REQS, on the other hand, can be used to allocate resources for a set of multiplexed sources, to yield an aggregate QoS for the multiplexed sources. In this case, the minimum total rate will be found which satisfies the specified aggregate QoS for the multiplexed sources[3]

Besides the bandwidth savings, an important advantage of REQS is that it can be applied for bandwidth allocation with different forms of QoS definitions as has been demonstrated earlier, while equivalent capacity formulas are known only for an average loss probability type of specification.

4.4.2 Comparison of the Dynamic Behavior with Alternative Approaches

Some alternate approaches have been tested for their dynamic performance, i.e., how fast do they converge relative to REQS. One alternative is to use a fixed value for the scalar K_n, rather than adapting it over time as described above. This alternative has been found to perform reasonably well when the initial update interval was small, but requires more time to converge when the initial update interval is large. With a large U_0, the algorithm must achieve convergence during mode 1 of operation; however, the constant scalar version of the algorithm performs poorly in mode 1.

A second alternative substitutes measurements of the cumulative loss probability, rather than the current loss probability, in the basic iteration of Equation 4.1. Other dynamic algorithms based on feedback (for example, [31]) have also used the error in the cumulative performance measure to adjust the resource allocation. This alternative was investigated experimentally. It was found that while the CLP converged very well to the desired value, the control parameter (in our case, the service rate) did not always converge to a steady state value. The reason is that over time the cumulative loss probability becomes less and less sensitive to changes in the instantaneous service rate. As a result, large changes in the service rate are needed to adjust the cumulative CLP. Figure 4.19 shows one pair of

[3]In this case, the QoS of individual sources may be more or less than the specified aggregate QoS. The user gives up some precision in specifying the desired QoS, in return for lower network overhead, and a reduction in the required resources.

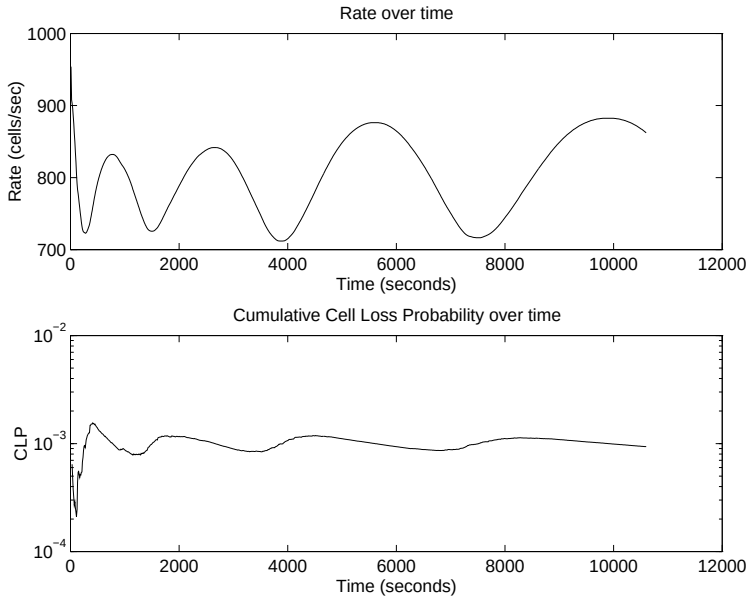

Figure 4.19: Problems with approachs based directly on error in cumulative QoS measures. Performance measure (CLP) converges but control (rate) exhibits strong oscillations. Standard source, target CLP 1e-3, U_0 10s, $B = 20$.

sample paths of the rate and cumulative loss probability, using this alternative. While the CLP appears to converge, the service rate oscillates, and in fact the magnitude of the oscillations increases. In [31] the control parameter (the scheduling priority) was not required to converge to a steady state value, so this was not a problem. Our experimental results indicate this approach will have difficulty converging both the QoS measure and the controlled resource. In contrast, REQS is able to control the cumulative loss very well, by measuring and using only the current loss probability. At the same time, the resource allocation converges to a steady state value in every case that was simulated.

A third alternative to REQS is to keep the update interval a constant length, rather than lengthening it as described. In experiments, this alternative also exhibited a tendency to oscillate. The reason is that the loss probability over any finite interval will never converge. An algorithm (such as REQS) which tracks the loss probability over the current update interval must necessarily lengthen the update intervals over time to ensure convergence.

Hsu and Walrand [27] have proposed an iterative algorithm which has many similarities

to our own. Some differences are that Hsu's algorithm uses a fixed update interval, and multiplies the error in the measured CLP by a factor of $1/n$, where n represents the index of the update. Hsu was able to prove this algorithm would converge to the desired CLP, under certain assumptions. The speed of convergence was not a primary issue in this work.

For an arbitrary traffic source, the multiplier $1/n$ may decrease too quickly. Before the actual QoS converges to the target QoS, the rate adjustment can become too small to make sufficient further progress. This happens if the update interval is chosen too small, resulting in an excessive number of updates. Through simulation, it has been found that convergence times vary significantly with the choice of update interval. No technique for selecting a good update interval was suggested in [27]. In contrast, the method of adjusting the update interval (during mode 2) proposed in this chapter ensures convergence, independent of the source characteristics.

4.5 Summary

This chapter has presented an algorithm for determining the minimum resource requirements needed to guarantee a specified Quality of Service measure. This algorithm is called REQS, which stands for **R**esource-**E**fficient **Q**uality of **S**ervice. The performance analysis focused on bandwidth as the resource of interest, and average cell loss probability as the QoS measure of interest. It has been also shown that other combinations of QoS measure and adjustable resource can be similarly controlled and optimized.

REQS determines the minimum steady state service rate which will satisfy the specified cell loss probability, for a given source. The main advantage of this approach is that essentially no traffic characterization of the source is required. This is important, considering the complexity of modeling and predicting source behavior. In addition, it eliminates the need to approximate (often very crudely) the resources needed to satisfy a QoS requirement. The proposed algorithm determined the minimum bandwidth necessary to satisfy a specified average cell loss probability, under a variety of different conditions. REQS converged quickly to the true "effective" rate in every case. REQS is simple to implement: only cell arrivals and losses need to be counted, and the rate adjustment calculation is very simple. In addition, the number of resource allocations required for convergence appeared to be very

reasonable (less than 10 for most cases tested).

Simulation showed that the convergence time of this algorithm increases as the burstiness of the source is increased, as the utilization level is increased, and as the loss constraint is made more stringent. Convergence time also increases with the size of the queuing buffer, though the rate of increase is fairly slow. The key advantage of REQS is that it is robust; it converged to the correct rate for every one of a wide range of conditions, and with almost no knowledge of the characteristics of the source.

We note that some engineering modifications may be needed in order to implement such a technique in practice. As an example, if the allocation level to a user or group of users is decreased during the transient phase of the algorithm, it may not consequently be possible to increase it if desired.

An engineering solution which takes care of this problem is to retain the initial conservative allocation during the transient and instead, perform the dynamic rate calculation on a "virtual server" with variable rate. Using a second buffer of equal size as the original and a "copy" of the actual arrival process, the dynamic rate calculation is essentially made in the background. None of the cells actually generated by the source are lost during the initial transient since these receive peak bandwidth.

When the rate of the virtual buffer converges to a steady state value, the rate of the actual server can be reduced to the converged value of the virtual server in one re-allocation. In this way, the rate is set to the optimal value in only one re-allocation step.

The work reported in this chapter has shown that a simple dynamic technique based on measurements of actual quality of service can result in accurate control of the QoS, while using network resources efficiently. This technique can be used for a wide variety of traffic sources and QoS specifications. The algorithm can be directly applied to the control of VP bandwidths in the architecture proposed in chapter 3. As discussed earlier however, it is applicable for determining the optimal resource allocation level for resources other than bandwidth and QoS definitions other than average cell loss probability.

Chapter 5

Conclusions

This thesis has presented methods for improving network resource efficiency while providing guarantees on Quality-of-Service to users. This is important for the efficient use of network resources in packet/cell switched networks such as ATM, designed to support multimedia traffic. Three related sub-problems have been addressed. The emphasis in all three problems has been to increase the utilization of network resources as much as possible, while continuing to provide strong guarantees on end-to-end QoS parameters. A number of important conclusions have resulted from these investigations.

The first problem is whether existing routing algorithms will work well for real-time traffic. It has been found that call acceptance rates into a network are improved if the routing function is designed to incorporate the constraints of the admission control function. An integrated routing and admission control function results in higher call acceptance than an approach in which routing and admission control operate independently of each other. It has been demonstrated that a new algorithm (the SC algorithm), which addresses the constraints of the admission control function, achieved the best call acceptance rates as compared to other algorithms studied. The benefits of the SC algorithm are greatest under conditions in which the QoS requirements are difficult to meet, the network is highly connected, and when admission control constraints are stringent (as in the presence of heterogeneous classes of traffic).

This has been the first joint study of routing and admission control algorithms for real-time multimedia traffic. The routing algorithms have been tested with three different call admission control schemes which have been proposed in literature for providing end-to-end

QoS guarantees. From comparisons of the admission control algorithms, it has been found that the choice of an appropriate CAC algorithm can have a greater impact on overall call acceptance than the choice of the routing algorithm. A deadline based flow control and admission policy (EDD) has been found to result in the best performance when QoS requirements on delay are tight, but is found to suffer in the presence of heterogeneous traffic classes. Finally, the use of a simple technique for improving the link utilizations achieved by these admission control policies has been introduced, which uses traffic shaping. By employing traffic shapers at the input of the network, utilizations were shown to improve while still providing strong end-to-end guarantees on maximum end-to-end delay and loss probability.

The second problem is at what level to reserve resources for guaranteed QoS. The main result of the second set of investigations is a design for a network architecture. This architecture deterministically reserves bandwidth for groups of channels traversing the same set of physical links (i.e. at the Virtual Path level in ATM networks), rather than for individual channels. This scheme exploits both traffic shaping and statistical multiplexing techniques to increase the utilization. Experimentally, utilizations of 70 to 80% are achievable using this technique. Using a queuing model of a VP, the advantages of letting all the end-to-end cell loss occur at the first physical link traversed by a VP have been formally proved. This approach has been shown to result in optimal bandwidth requirements at each link for a VP when the per-link buffer assignments are also controllable. A simple scheduling mechanism (the Weighted Round Robin or WRR server) which implements the proposed scheme has been suggested and examined. The complexity of implementation of the proposed scheme is equivalent to using cell spacers which are already in use in ATM networks. Buffer sharing techniques for improving performance without compromising on the ability to provide end-to-end QoS guarantees have been outlined.

The third problem is how to allocate network resources optimally with minimal user input. An algorithm has been developed for dynamically determining the minimum resources required to guarantee a specified quality of service for any given source. The main advantage of this method is that very little input traffic characterization is needed. This algorithm can be used whenever the resource being controlled monotonically controls the

QoS measures of interest. The algorithm has been tested to determine the minimum bandwidth needed to meet a specified bound on average cell loss probability. The algorithm utilizes measurements of current QoS levels to update the resource allocation. Extensive simulations have shown that the algorithm is extremely robust and results in the optimal bandwidth allocation in every case. Both the dynamic and steady state behavior of the algorithm have been tested. The performance of the algorithm has been analyzed under variations in the measurement frequency, source burstiness and desired QoS requirement. The superiority of this approach over approaches based on popular "equivalent capacity" formulas has been demonstrated. The use of this technique for determining the optimal resource requirement to satisfy QoS measures other than cell loss probability has also been demonstrated. This technique can be applied to control the bandwidth allocation to VPs in the architecture proposed in chapter 3, or for the control of any rate-allocating server.

5.1 Future Research

In the course of the investigations reported in this thesis, a number of interesting avenues have been uncovered which merit further research.

- Routing algorithms have been studied in the context of admission control schemes which are based on peak bandwidth allocation. The joint study of routing and CAC functions which allow for statistical multiplexing must consequently also be studied. The current study has concentrated on the interaction between the routing and admission control functions. Distributed implementation techniques for the real-time routing algorithms studied need to be investigated.
- In the architecture developed in this thesis, each VC using a VP is guaranteed the worst case QoS of all VCs using the VP. Further improvement in performance can be expected if QoS guarantees can be provided on a per-VC basis. The use of priority scheduling techniques between different VCs within a VP must be investigated to achieve such per-VC guarantees. Algorithms for optimal design of the configuration of VPs within a network and their resource allocation levels must also be developed.

- The performance of the dynamic resource allocation algorithm developed in this thesis must be studied in the presence of non-stationary source models and with traces from actual sources. Optimizing the performance of the algorithm, particularly for very stringent QoS constraints (such as loss probabilities of 1e-5 or lower) is an important topic for further work. The use of similar techniques for optimizing the allocation of other types of resources such as buffer space and the simultaneous control of multiple resources in a network in order to optimally satisfy end-to-end QoS requirements are also important issues for further work. A formal proof must be developed to establish the convergence of the algorithm in all cases.

List of References

[1] ATM Forum, "ATM User-Network Interface (UNI) Specification, Version 3.0, " 1993.

[2] H. Ahmadi, J. Chen and R. Guerin, "Dynamic Routing and Call Control in High-Speed Integrated Networks," *Proc. of the 13th International Teletraffic Congress*, Copenhagen, June 1991, pp. 397-403.

[3] C.M. Aras, J. Kurose, D. S. Reeves and H. Schulzrinne, "Real-time Communication in Packet-switched Networks," *Proceedings of the IEEE*, Vol.82, No.1, January 1994, pp. 122-139.

[4] J.J. Bae, T. Suda and R. Simha, "Analysis of a Finite Buffer Queue with Heterogeneous Markov Modulated Arrival Processes: A Study of the Effects of Traffic Burstiness on Individual Packet Loss," in *Proc. IEEE INFOCOM '92*, 1992, pp. 219-230.

[5] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, New Jersey: Prentice-Hall, 2nd Ed 1992.

[6] F. Bonomi, S. Montagna and R. Paglino, "A Further Look at Statistical Multiplexing in ATM Networks," *Computer Networks and ISDN Systems*, 26(1993), pp. 19-138.

[7] CCITT/ ITU Study Group XVIII, "Traffic Control and Resource Management in B-ISDN," *CCITT Recommendation I.371*, Geneva, 1992.

[8] A. Charny, "An Algorithm for Rate Allocation in a Packet-switching Network with Feedback," *Tech. Rep. MIT/TR-601*, MIT, Cambridge, May 1994.

[9] I. Chlamtac, A. Farago, and T. Zhang, "How to Establish and Utilize Virtual Paths in ATM Networks," *Proc. IEEE ICC'93*, Geneva, 1993, pp. 1368-1372.

[10] J.H.S. Chan and D.H.K. Tsang, "Bandwidth Allocation of Multiple QoS classes in ATM Environment," *Proc IEEE INFOCOM '94*, Toronto, June 1994.

[11] S. Chong, San-Qi Li, and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM," *IEEE J. on Selected Areas in Communication*, Vol.13, No.1, January 1995, pp. 12-23.

[12] D.D. Clark, S. Shenker and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," *Proc. ACM SIGCOMM '92*, August 1992, pp. 14-26.

[13] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang, "Pricing in Computer Networks: Motivation, Formulation, and Example," *IEEE/ACM Transactions on Networking*, Vol. 1, December 1993.

[14] R.L. Cruz, "A Calculus for Network Delay, Part II: Network Analysis," *IEEE Transactions on Information Theory*, Vol. 37, No. 1, January 1991, pp. 132-141.

[15] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm", *Internetworking: Research and Experience*, Vol.1, No.1, September 1990, pp. 3-26.

[16] K.M. Elsayed and H.G. Perros, "The Superposition of Discrete-Time Markov Renewal Processes with an Application to Statistical Multiplexing of Bursty Traffic Sources," *Technical Report TR 94-10*, Dept of Computer Science, North Carolina State University, 1994.

[17] A.I. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks," *IEEE/ACM Trans. Networking*, Vol.1, No.3, June 1993, pp. 329-343.

[18] K.W.Fendick, D. Mitra, I. Mitrani, M.A. Rodrigues, J.B. Seery, and A. Weiss, "An approach to high-performance, high-speed data networks," *IEEE Communications Magazine*, Vol. 29, October 1991, pp. 74-82.

[19] D. Ferrari and D.C. Verma, "A Scheme for Real-time Channel Establishment in Wide-area Networks," *IEEE Journal on Selected Areas in Communications*, Vol.8, No.3, April 1990, pp. 368-379.

[20] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman, 1979.

[21] M. Gerla, H. W. Chan, J. Boisson De Marca, "Routing, Flow Control, and Fairness in Computer Networks," *Proc. IEEE Intl. Conf. on Computer Communications*, May 1984, pp. 1272-1275.

[22] A. Girard, "Routing and Dimensioning in Circuit-Switched Networks", Addison-Wesley Publ. Co., 1990.

[23] S.J. Golestaani, "A Framing Strategy for Congestion Management," *IEEE Journal on Selected Areas in Communications*, Vol.9, No.7, September 1991, pp. 1064-1077.

[24] W. D. Grover, "The Self Healing Network: A Fast Distributed Restoration Technique for Networks Using Digital Cross-Connect Machines," *Proc. IEEE Globecom '87*, December 1987, pp. 1090-1095.

[25] R. Guerin, H. Ahmadi and M. Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks," *IEEE Journal on Selected Areas in Communications*, Vol.9, No.7, September 1991, pp. 968-981.

[26] F. Guillemin and W. Monin, "Management of cell delay variation in ATM Networks, " *Proc IEEE Globecom '92*, Orlando, December 1992, pp. 128-132.

[27] I. Hsu and J. Walrand, "Dynamic Bandwidth Allocation for ATM Switches," Technical Manuscript, University of California, Berkeley, to appear in *Journal of Applied Probability*, September 1996.

[28] R-H. Hwang, J. Kurose and D. Towsley, "The effect of Processing Delay and QoS Requirements in High Speed Networks," *Proc. IEEE INFOCOM '92*, pp. 160-162.

[29] M. Izquieredo and D.S. Reeves, "Statistical Characterization of MPEG VBR Video at the Slice Level," *Proc of the Conference on Multimedia Computing and Networking*, SPIE, San Jose, February 1995.

[30] S. Jamin, P. Danzig, S. Shenker and L. Zhang, "A Measurement-based Admission Control Algorithm for Integrated Services Packet Networks," in *Proc ACM SIGCOMM '95*, Boston, September 1995.

[31] Y.H. Jeon and I. Viniotis, "Achievable Loss Probabilities and Buffer Allocation Policies in ATM Nodes with Correlated Arrivals," *Proc. IEEE Intl. Conference on Communications*, 1993, pp. 365-369.

[32] C.R. Kalmanek, H. Kanakia and S. Keshav, "Rate Controlled Servers for Very High-speed Networks," *Proc. IEEE Globecom '90*, San Diego, December 1990, pp. 12-20.

[33] D.D. Kandlur, K.G. Shin and D. Ferrari, "Real-time Communication in Multi-hop Networks," *Proc. IEEE Intl. Conf. on Distributed Computing*, 1991, pp. 300-307.

[34] E. Knightly and H. Zhang, "Traffic Characterization and Switch Utilization Using a Deterministic Bounding Interval Dependent Traffic Model," *Proc IEEE INFOCOM '95*, April 1995, Boston, pp. 1137-1145.

[35] V. Kompella, J. Pasquale and G. Polyzos, "Multicast Routing for Multimedia Communications," *ACM/ IEEE Transactions on Networking*, Vol.1, No.3, June 1993, pp. 286-292.

[36] J. Kurose, "Open issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks," *ACM Computer Communication Review*, Jan 1993, pp. 6-15.

[37] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson and J. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Trans. Communications*, Vol.36, No.7, July 1988, pp. 834-844.

[38] N.M. Mitrou, K.P. Kontovasilis, H. Kroner and V.B. Iversen, "Statistical Multiplexing, Bandwidth Allocation Strategies and Connection Admission Control in ATM Networks," *European Transactions on Telecommunications*, Sp. issue on "Teletraffic Research for B-ISDN in the RACE Programme," February 1994.

[39] P.M. Morse, *Queues, Inventories and Maintenance : The Analysis of Operation Systems with Variable Supply and Demand*, Wiley, New York, 1958.

[40] K. Murakami, "Near-Optimal Virtual Path Routing for Survivable ATM Networks," *Proc. IEEE INFOCOM '94*, June 1994, pp. 208-215.

[41] R. Nagarajan, J. Kurose and D. Towsley, "Local Allocation of End-to-end Quality-of-Service Resources in High-Speed Networks ," in *Proc. IFIP Workshop on Performance Analysis of ATM Systems*, Martinique, January 1993.

[42] R. Nagarajan, J. Kurose and D. Towsley, "Finite-horizon Statistical Quality-of-Service Measures for High-Speed Networks," Technical Manuscript, University of Massachusetts at Amherst, to appear in *Journal of High Speed Networks*.

[43] R.O. Onvural and Y.C. Liu, "On the Amount of Bandwidth Allocated to Virtual Paths in ATM Networks," *Proc IEEE Globecom '92*, 1992, pp. 1460-1464.

[44] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks-The Multiple Node Case," *Proc. IEEE INFOCOM '93*, pp. 521-530.

[45] C.J. Paris and D. Ferrari, "A Dynamic Connection Management Scheme for Guaranteed Performance Services in Packet-Switching Integrated Services Networks," Tenet Technical Report TR-93-005, Computer Science Division, University of California at Berkeley, 1993.

[46] C. Partridge, *Gigabit Networking*, Addison-Wesley, 1993.

[47] R. Perlman, "Interconnections: Bridges and Routers", Addison-Wesley, 1992.

[48] H. Schulzrinne, "IPv6 - The New Internet Protocol, " *PIK (Praktische Informatik und Kommunikation)*, vol. 18, June 1995, (anonymous ftp ftp://gaia.cs.umass.edu/pub/Schu9506:Ipv6.psVgz).

[49] N. Shroff and M. Schwartz, "Video Modeling in ATM Networks using Deterministic Smoothing at the Source, " in *Proc. IEEE INFOCOM '94*, Toronto, June, 1994, pp. 342-349.

[50] K. Sriram, "Methodologies for Bandwidth Allocation, Transmission Scheduling, and Congestion Avoidance in Broad band ATM networks," *Computer Networks and ISDN Systems*, Vol. 26, 1993, pp. 43-59.

[51] G. de Veciana, C. Courcoubetis, and J. Walrand, "Decoupling bandwidths for networks: A Decomposition approach to Resource Management," in *Proc IEEE INFOCOM '94*, Toronto, June 1994, pp. 466-473.

[52] Q. Wang and V. Frost, "Efficient Estimation of Cell Blocking Probability for ATM Systems, *IEEE/ACM Transactions on Networking*, Vol. 1, April 1993, pp. 230-235.

[53] C.-Y. Wang, D. Logothetis, K.S. Trivedi and I. Viniotis, "Transient Behavior of ATM Networks under Overloads," *Technical Manuscript*, Dept of Electrical Engg., Duke University, April 1995.

[54] R.W. Wolff, "Stochastic Modeling and the Theory of Queues," Prentice Hall, 1989.

[55] D. Yates, J. Kurose, D. Towsley and M. Hluchyj, "On Per-session End-to-end Delay Distributions and the Call-admission Problem for Real-time Applications with QoS Requirements," in *Proc ACM SIGCOMM '93*, September 1993, San Francisco, pp. 2-12.

[56] Hui Zhang and Domenico Ferrari, "Rate-controlled Static Priority Queuing," in *Proc IEEE INFOCOM '93*, March 1993, pp. 227-236.

Appendices

Appendix A

Monotonicity of Cell Loss with Service Rate

The first part of Theorem 1 is proved here. We show that for an arbitrarily fixed sequence of arrivals all of the same length to a finite capacity queue, the total number of losses increases monotonically as the service rate of the queue is increased.

To prove the above statement, two queues Q_1 and Q_2 will be compared, each with a buffer of size B and service rates μ_1 and μ_2 respectively ($\mu_1 < \mu_2$). It will be shown that for any arbitrarily specified sequence of cell arrivals, the total number of losses due to buffer overflow at Q_1 can not be less than at Q_2.

The length of each cell represents the amount of work needed to service (transmit) this cell. Let this fixed amount of work be l. The total amount of remaining work at a queue at some instant is the sum of the lengths of waiting cells and the length of the untransmitted portion of the cell in service at that instant.

The following notation is used.

t_i : The arrival instant of the ith cell in the sequence.

a_i : The inter-arrival time between cell i and cell $i+1$.

W_j^t : The total remaining work at queue j, $(j = 1, 2)$ at time t.

Q_j^t : The length of queue j, $(j = 1, 2)$, at instant t.

$W_j^{t_i}$ is hence, the total remaining work at queue j seen by cell i upon arrival (not including the work corresponding to the ith cell itself) and $Q_j^{t_i}$ is the number of cells in queue j seen by cell i upon arrival (not including itself).

The total number of losses at each queue will be examined over time, starting with empty queues at $t = 0$. Whenever an arriving cell is lost at one queue but not at the other,

the difference in the total losses seen so far at the two queues changes. Such a cell loss will be denoted as a unique cell loss (or UCL). It is shown that the total number of UCLs at Q_1 is at least as much as the number at Q_2 for any sequence of arrivals, thereby proving the result. This will be shown as follows. First it is shown that starting with empty queues at $t = 0$, Q_1 is the first queue to see a UCL. Next it is shown that whenever a UCL at Q_2 does occur, the following UCL must be at Q_1. In contrast after a UCL at Q_1 the next UCL may still be at Q_1. Hence, starting with empty queues at $t = 0$, at no time will the number of UCLs at Q_1 be less than that at Q_2. Thus the total number of losses at Q_1 can not be less than at Q_2.

First a basic result is proven, which shows that if at any arrival instant the remaining work at Q_2 is observed to be no more than that at Q_1, then the next UCL to occur will be at Q_1. First consider a sequence of arrivals which are not lost at either queue. Lindley's recursion for queue evolution [54] can be re-written here as

$$W_j^{t_{i+1}} = [W_j^{t_i} + l - a_i \mu_j l] \qquad i = 2, \ldots n \qquad j = 1, 2 \tag{A.1}$$
$$[X] = max(0, X)$$

Let, at the arrival instant of some cell i, the remaining work at Q_1 be at least as much as that at Q_2 (i.e. $W_1^{t_i} \geq W_2^{t_i}$). Since, $\mu_1 < \mu_2$, from the above relation (Eqn A.1) we get $W_1^{t_{i+1}} \geq W_2^{t_{i+1}}$. The very first cell arrival sees no outstanding work at either queue. Hence $W_1^{t_1} \geq W_2^{t_1}$ is trivially true. Consequently, by induction, $W_1^{t_i} \geq W_2^{t_i}$ for all i as long as no cells are lost. Hence, if at some arrival instant, the remaining work is less at Q_2, then the remaining work at Q_2 continues to be less than at Q_1 for future arrivals as long as no cells are lost. Since $Q_j^{t_i} = \lceil (W_j^{t_i}/l) \rceil, j = 1, 2$, we also have $Q_1^{t_i} \geq Q_2^{t_i}$ for all cells i as long as no cells are lost. This means that starting from any time when the remaining work at the faster queue is no more than at the slower queue, the queue length seen by an arriving cell at the slower queue (Q_1) will always be at least as much as that at the faster queue (Q_2) as long as there are no losses. Hence, an arriving cell which sees a full buffer at Q_2 must also see a full buffer at Q_1 (but not necessarily vice versa). Hence, after any instant at which the remaining work at Q_1 is at least as much as that at Q_2, the first UCL to occur must be at Q_1.

A consequence of this basic result is that starting with both queues empty, the first UCL must occur at Q_1 (the slower queue) (since $W_1^{t_1} \geq W_2^{t_1}$ is trivially true).

We are interested in UCLs at Q_2 since this is the only way that the total number of losses at Q_2 can exceed that at Q_1. We now show that if at all a UCL occurs at Q_2, the next successive UCL must be at Q_1.

If the first UCL (at Q_1) occurs at time t, and at time t^+ the queue length at Q_2 is less than that at Q_1 (i.e. $Q_2^{t^+} \leq Q_1^{t^+} - 1$), then the remaining work at queue Q_2 at time t^+ will also be less than at Q_1. (This is because $W_2^{t^+} \leq l * Q_2^{t^+}$ while $W_1 > l * (Q_1^{t^+} - 1)$. In this case, from the above result, the next UCL will again be at Q_1.

Hence, the only way a UCL at Q_1 can cause a UCL at Q_2 later, is if at t^+, both queues have B cells. At t^+, the remaining work at Q_2 can exceed that at Q_1 by at most the length of one cell i.e. l, (depending on the difference in the amount of service so far received by the cells in service at Q_1 and Q_2). Since Q_2 has a lower service time, at most one cell can be transmitted by Q_1 before a cell leaves Q_2. In other words, at most one cell can be accepted at Q_1, while Q_2 is still full, resulting in a UCL at Q_2. After this UCL however, the next departure from either queue must be from Q_2 (since the service rate of Q_2 is higher, we cannot see two successive departures from Q_1 before a departure from Q_2). After this departure from Q_2, the queue length at Q_2 will become strictly less than that at Q_1 (which will be full). If $Q_2^t < Q_1^t$, we must have $W_2^t < W_1^t$. Hence, the remaining work at Q_2 will again be less than at Q_1. (A consequence of the relation $Q_j^t = \lceil W_j^t / l \rceil$). Since the remaining work decreases faster for the faster queue, even at the arrival instant of the next cell, the remaining work at Q_2 will be less than that at Q_1. From the above arguments hence, after a UCL at Q_2, the next UCL will again be at Q_1.

The above arguments have shown that starting with empty queues, the first UCL to occur will be at Q_1. Further, after any UCL at Q_2, the next UCL must be at Q_1. By induction over time hence, the total number of UCLs at Q_1 is never less than the number at Q_2. Hence, the total number of losses must also be greater at Q_1, the "slower" queue. □

Appendix B

Monotonicity of Cell Loss with Buffer Size

The first part of Theorem 2 is proved here. It is proved that the total number of losses experienced by an arbitrarily fixed sequence of arrivals of cells of equal length to a finite capacity queue increases monotonically as the buffer size of the queue is decreased.

To prove the above statement, two queues Q_1 and Q_2 shall be compared, each with the same service rate (μ) and buffer sizes B^1 and B^2 respectively ($B^1 < B^2$). It will be shown that for any arbitrarily specified sequence of cell arrivals, the total number of losses due to buffer overflow at Q_1 can not be less than at Q_2. The buffer size of a queue refers to the total amount of buffering which is the sum of the queuing buffer space and the unit buffer required for the cell in service.

The same (arbitrarily specified) arrival process is input simultaneously to both queues. It is shown that at any instant, starting with both queues empty, the total number of losses at Q_1 is at least as much as that at Q_2. First the result is proved for $B^2 = B^1 + 1$. Using simple induction, this result will then be shown to hold for any $B^2 > B^1$.

Let l be the amount of work needed to service a single cell (the length of a cell is representative of the work per cell). Denote W_j^t be the total amount of remaining work in queue j at time t, which is defined as the sum of work corresponding to each waiting cell and the unfinished work corresponding to the cell in service. Hence the number of cells in queue j at any time t, is $\lceil W_j^t/l \rceil$.

Let $B^2 = B^1 + 1$. Starting with empty queues, the given arrival sequence is fed to both queues. As long as there are no more than B_1 cells in each queue, both queues behave identically since the service rate and initial conditions are the same. Hence the first cell to

be lost (say cell number i) must be at Q_1 at which time both queues have B^1 cells but only Q_1 is full since Q_2 has an extra buffer slot. Hence cell i gets accepted at Q_2. Now, tag this cell as being "extra". Consequently, from $t = 0$ up to the first cell loss, the number of cells lost at Q_2 is never more than at Q_1.

Acceptance of this "extra" cell increases the amount of unfinished work at Q_2 by l relative to the unfinished work at Q_1. A difference in unfinished work of l implies a difference in the number of cells of 1. However, since an extra buffer slot is anyway available in Q_2, the number of empty slots available for newly arriving cells is never more in Q_1. As a result, a cell is accepted at Q_1 only if it is also accepted at Q_2. Hence, as long as both queues do not become empty again, the difference in the unfinished work can not increase due to Q_2 taking on excess work relative to Q_1. Further, since the two servers operate at the same rate, the unfinished work decreases at the same rate in both queues. Hence the difference in the unfinished work can not increase even because of departures.

Hence as long as both queues do not become empty again, the unfinished work at Q_2 can not exceed that at Q_1 by more than l. Hence, the number of cells in Q_2 can not exceed that in Q_1 by more than one. Since an extra buffer slot is available to Q_2, a cell can be lost at Q_2 only if it is also lost at Q_1. Hence Q_1 can not experience fewer losses than Q_2. When both queues become empty, similar to earlier arguments, Q_1 is the first to experience a loss. Hence at all times, the total number of losses at Q_1 can not be less than at Q_2. This proves the result for $B^2 = B^1 + 1$.

Using the same arguments, the number of losses in a system with $B^1 + 2$ cells can not be more than that at one with $B^1 + 1$ cells, and hence not more than one with B^1 cells. By transitivity of the '$\leq$' relation, this result is valid for any difference in the buffer sizes. Thus, if buffer size of a queue is decreased by any amount, the total number of losses for any given (arbitrary) arrival sequence can only increase. □

Appendix C

Optimality of MGF policy in the Variable Buffer Allocation Case

Theorem 4 is proved here. The theorem is restated first for convenience.

"Given the total amount of end-to-end waiting buffer space W and the service rate at the last hop H (C^H), an MGF type configuration with $B^1 = W + 1$, $B^2 = B^3 = \cdots = B^H = 1$ and $C^1 = C^2 = \cdots = C^H$ results in the minimal total number of losses for any arbitrarily given sample path of arrivals as compared to any other configuration of rates and buffer sizes under the same constraints".

First, this is proved for the case of $H = 2$ in the following lemma. Theorem 4 is then proved using this lemma.

Lemma 1 *Given a two hop queuing network, the total waiting buffer space W and the service rate at the second hop (C^2), an MGF configuration with equal service rates and all waiting buffer at hop 1 results in minimal losses for any arbitrarily given sequence of arrivals of constant length as compared to any other configuration of rates and buffer sizes under the same constraints.*

Proof: As explained in section 3.4, the values of the propagation delays do not affect the total number of losses. Hence without loss of generality, all propagation delays are considered to be zero.

Two queuing networks will be compared with the same service rate at hop 2 and total waiting buffer space. Network N_1 consists of queues $Q_{1,1}$ and $Q_{1,2}$ with buffer sizes (including the unit buffer for the cell in service) of $B^{1,1}$ and $B^{1,2}$ and service rates $C^{1,1}$ and $C^{1,2}$

respectively. The total waiting buffer space in the network is W, so that $B^{1,1}+B^{1,2} = W+2$. Network N_2 consists of a single queue $Q_{2,1}$ with buffer size (including the cell in service) of $B^{2,1} = W + 1$ and service rate $C^{2,1} = C^{1,2}$. The service rates are assumed to be expressed in units of cells served in unit time since each cell requires the same amount of service. Network N_2 may be seen as the first queue in an MGF configuration with the same constraints as N_1. Since no loss occurs at downstream queues in an MGF configuration, only the first queue needs to be analyzed for determining the total end-to-end losses. The downstream queue in the MGF configuration is not considered in the analysis (and is hence shown in the figure with dotted lines). It will be shown that if an arbitrary arrival sequence is input to both networks, the total number of losses in N_2 will not exceed those in N_1 for any such configuration N_1 and any such arrival sequence.

Let the $W+2$ available buffer slots in N_1 be numbered $1, 2, \cdots W+2$ with slot 1 denoting the slot for the cell in service in $Q_{1,2}$ and slot $W + 2$ denoting the slot farthest from the server in $Q_{1,1}$. Let the $W + 1$ available buffer slots in N_2 be numbered $1, 2, \cdots W + 1$ with slot 1 denoting the slot for the cell in service at $Q_{2,1}$ and slot $W + 1$ denoting the slot at the tail of $Q_{2,1}$. The two networks are illustrated in Figure C.1.

The same sample path of arrivals is fed simultaneously to both networks as shown. A cell arriving to N_1 is first served at $Q_{1,1}$ and then at $Q_{1,2}$ even if there are no other cells in the network. In contrast, a cell arriving at N_2 instantaneously moves to the buffer slot with the lowest possible index i.e. closest to server of $Q_{2,1}$.

Consider the following "renaming" strategy for cells in N_2. Whenever some cell (say the ith in the arrival sequence) is dropped in N_1 due to overflow at $Q_{1,2}$, if cell i is present in N_2, rename cell i as $i + 1$, cell $i + 1$ as cell $i + 2$ and so on. Finally, rename the cell to arrive last to N_2 as "extra". This renaming essentially emulates the loss of cell i even in network N_2 and all the additional cells in N_2 are tagged "extra" (these denote "extra" cells which are lost in N_1 but not in N_2). Such a renaming does not change the total number of losses but only the identities of lost cells. This renaming is also carried out at each arrival instant, in such a way that any cells tagged "extra" always occupy the buffer slots farthest from the server at $Q_{2,1}$. Hence it can be assumed that any "extra" cells are transparent to arrivals at N_2. If an arriving cell finds $Q_{2,1}$ full, it can push out (again by renaming) any cells tagged "extra" (it will thus be assumed that the "extra" cells were lost, not the new

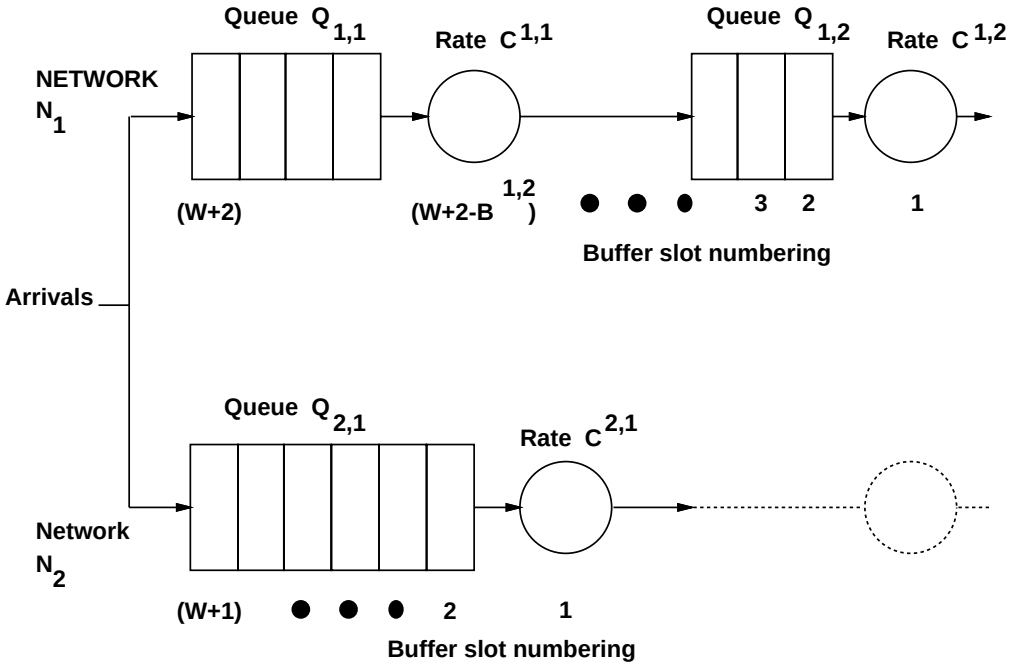

Figure C.1: The two networks being compared for proof of optimality of the MGF policy

arrivals).

Using this renaming strategy, any cell i always reaches server $Q_{2,1}$ in N_1 before it reaches $Q_{1,2}$ in N_2. This is because queues $Q_{1,2}$ and $Q_{2,1}$ operate at the same rate so cells cannot leave $Q_{1,2}$ earlier than $Q_{2,1}$. Additionally, in N_1, cells have to be served at $Q_{1,1}$ before reaching $Q_{1,2}$ while in N_2 cells instantaneously move to the farthest downstream position possible. The final way in which a cell can get ahead in N_1 as compared to N_2 is if some cells ahead of it get dropped due to over flow causing it to move ahead. This is taken care of by the renaming procedure, which emulates the same losses even in N_2.

Further, each such cell i will start service at $Q_{2,1}$ at least $1/C^{1,1}$ units of time after it starts service at $Q_{2,1}$. This can be shown by induction. The very first cell experiences a delay of $1/C^{1,1}$ before starting service at $Q_{1,2}$, while it is immediately served upon arrival at $Q_{2,1}$. Hence the above statement is true for the first cell. Further, let this be true for cell i (i.e. cell i starts getting served at $Q_{1,2}$ at least $1/C^{1,1}$ time units after it starts at $Q_{2,1}$). As a consequence, the completion time of cell i in $Q_{1,2}$ also occurs at least $1/C^{1,1}$ time units after the completion at $Q_{2,1}$. Also, cell $i+1$ will reach the server in $Q_{2,1}$ no later

than in $Q_{1,2}$. Hence, the result must be valid even for cell $i+1$. By induction, every cell starts service at $Q_{1,2}$ at least $1/C^{1,1}$ time units after it starts service at $Q_{2,1}$.

The above result is now used to prove the main result. Consider first, sample paths such that all $W+2$ available buffer slots in N_1 are never simultaneously occupied. It is first shown that in such a case, the number of losses in N_2 is never more than in N_1. This is because N_2 may be seen as being made up of two queues in tandem, with the downstream server identical to $Q_{2,1}$ and the upstream server to be an infinite rate server with blocking with a total waiting buffer space of $W-1$ (using standard terminology from queuing systems literature [54]) An infinite rate server upstream with blocking makes the best use of end-to-end buffer space since cells are transmitted instantaneously downstream as long as the downstream buffer has any free slots, but no cells are transmitted when the downstream buffer is full since any transmitted cells will be lost anyway. Hence for such arrival sequences, the total number of losses in N_2 is no more than in N_1.

The advantage that N_1 has over N_2 is that one additional buffer slot is available in the end-to-end path, which may be used to save an additional cell which is lost at N_2. It is now shown that whenever a cell is saved at N_1 due to this extra buffer slot which is lost at N_2, another cell must get lost at N_1 which is not lost at N_2 thereby nullifying this gain. In other words, network N_1 is never able to take advantage of this extra buffer slot.

Let at some instant, there be $W+2$ cells in N_1 occupying all $W+2$ slots available in the end-to-end path. Let the first $W+1$ cells occupy the available buffer slots in N_2 and the last cell be dropped due to overflow of $Q_{2,1}$. This cell was saved at N_1 due to the extra buffer slot available. At the instant of the loss in N_2, if the index of the cell in service at $Q_{1,2}$ is i, then that of the cell in service at $Q_{2,1}$ must also be i. (From the above arguments, the index of cell in service at $Q_{2,1}$ can not be less than i while if it was greater than i, then there would be less than $W+1$ cells in N_2 and this cell loss would not have occurred). Hence the remaining service time of cell i at $Q_{1,2}$ must be strictly greater than $1/C^{1,1}$ as per the earlier result. Since the service time of the cell at $Q_{1,1}$ is $1/C^{1,1}$, the cell in service at $Q_{1,1}$ will arrive at $Q_{1,2}$ before any cell leaves $Q_{1,2}$ and will be lost due to overflow. This cell is not lost at N_2 since it has already been accepted at $Q_{2,1}$.

To summarize, whenever an extra cell is saved at N_1 to the extra buffer slot in its end-to-end path, another cell will be lost which compensates this saving. N_1 is never able to

exploit its $W+2$ available buffers and N_2 makes the best possible use of its $W+1$ buffers. Hence, for any arrival sequence, the number of losses in N_1 will be at least as much as that at N_2. □

Theorem 4 is now proved for any length (H) of the path using the above lemma.

Proof of Theorem 4: For some value of $H > 2$, let if possible, some configuration of rates and buffer sizes (other than the MGF configuration) which satisfies the given constraints (on total waiting buffer space and rate at the last hop), achieve the minimum number of losses for all arrival sequences. Now consider the last two hops of this configuration. From Lemma 1, by setting the rate at hop $H-1$ to equal that at hop H and moving all the waiting buffer over these two hops to hop $H-1$, the number of losses at the last two hops can only decrease. Further the losses at upstream nodes will be unaffected since the network is unchanged up to the last two hops. Hence the total number of losses in the network can only decrease.

Starting from the given supposedly optimal configuration, we now have another configuration with no greater total number of losses. Now, repeat this process at hops $H-2$ and $H-1$, making the rate at $H-2$ equal to that at $H-1$ (and hence the same as that at hop H) and moving all waiting buffer space to hop $H-2$). Again this can only decrease the total number of losses. This process can be repeated until the entire network is transformed into an MGF type configuration without increasing the number of losses, thereby proving the optimality of the MGF policy for all values of H. □