Priority Queueing in Asynchronous Transfer Mode Switches

A Thesis

Submitted to the College of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Department of Computational Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Karen L. Reid

August, 1992

Distribution Notice

Priority Queueing in Asynchronous Transfer Mode Switches

Candidate: Karen L. Reid

Supervisor: Richard B. Bunt

Master of Science Thesis

Presented to the College of Graduate Studies and Research

University of Saskatchewan

August, 1992

ABSTRACT

The Broadband Integrated Services Digital Network (BISDN) is a high speed packet switched network designed to support services consisting of various combinations of voice, video, and data. It will accommodate single service bandwidth requirements ranging from a few kilobits per second to more than 150 Megabits per second. The traffic on this type of packet network will differ greatly from the traffic on any of the existing communication networks. This means that new methods of controlling the traffic will be needed in order to use the available bandwidth efficiently and ensure acceptable quality of service for all services supported.

The nature of packet switched networks is such that packets will be lost. It is necessary to engineer switches and associated buffers to minimize packet loss and to satisfy quality of service requirements for all types of traffic. Priority queueing techniques can be used to protect higher priority packets from unacceptable packet loss by absorbing the loss in lower priority traffic.

This thesis investigates several priority queueing techniques with the goal of determining how priority queueing may aid in supporting acceptable quality of service for all classes of traffic. A simulation study of three priority queueing algorithms operating at the output buffer of a switch showed that the performance of these algorithms depends on the shape and type of the traffic. It also showed that priority algorithms, when combined with other congestion control algorithms, can aid in supporting acceptable quality of service for the users of the network.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Rick Bunt, for his guidance and encouragement throughout my research work. I greatly appreciated the time he took to discuss my work and to read my thesis. Without his efforts I would not have been able to complete my thesis in such a short time. I would also like to thank the other members of my committee, Carey Williamson, Mostafa Abd-El-Barr, and Eric Salt, for their helpful suggestions both for the proposal and for the final version of the thesis.

Thanks to the other members of the DISCUS group for listening to my ideas and providing suggestions. Thanks especially go to Darryl Willick for all his help in setting up my simulations, solving systems problems, and finding more disk space.

My parents, Wally and Marg Krause, have always made me believe that I could do anything I put my mind to and have given their total support to each of my endeavors. Without their support and confidence in my abilities I doubt if I would have come this far.

I might still be hunting for a bug in my simulation if not for the help of my husband, Irving. He got me through the bad times and reminded me to take enough time off to have some fun. Irving always challenged me to improve my programming and writing skills and I believe that this thesis is better for it. I greatly value his emotional and intellectual support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The information age encompasses the development of computer and communication technology, and technology continues to struggle to keep pace with the demands of new applications. One aspect of this communication explosion is the development of networks that supply information at high rates to the user. As the demand for more complex communications increases, research in communication networks continues to try to accommodate these and future services as efficiently and flexibly as possible.

Most existing communication networks were designed to support a single application or a single class of applications. For example, the telephone and cable television networks each support a single class of services. Neither network can support other services very efficiently. Although the telephone network is often used for data transfer in the form of modem use or facsimile, it does not usually provide good performance for these applications. Computer networks are engineered to meet the specific demands of data traffic. It is difficult to support services that have strict delay requirements, such as real time voice, on a computer network.

In recent years, the focus in designing communication networks has shifted from providing a specialized network for each new service to providing a more general integrated network that can handle many different types of services. In particular, the main goal in developing the Broadband Integrated Services Digital Network (BISDN) is to support a wide variety of services that can be termed multi-media, multi-rate, and multi-point [Hui, 1989].

The goal of a communications network is to connect geographically distributed sites. A single link provides communication capability between two points and a network is constructed from these links so that all points on the network can communicate with each other. One method of doing this is to provide a communication link between each of the points on the network (see Figure 1.1). It is easy to see that even for a small network the number of links required is large. This is also an inefficient implementation since a single point on the network does not usually need to communicate with all other points at the same time. A better solution is to provide *switches* inside the network that act as intermediate nodes in forming a connection. This means that fewer links are required (see Figure 1.2). For example, consider the telephone network. Each house has a line that connects it to a local switching station. This station acts as an intermediate node for calls within the neighbourhood. Each switching station is connected to other switching stations that become intermediate nodes for long distance calls. A single telephone call may pass through several switching stations.

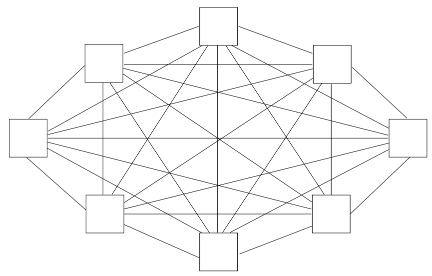

Figure 1.1: A Fully Connected Network (8 nodes, 28 bi-directional links)

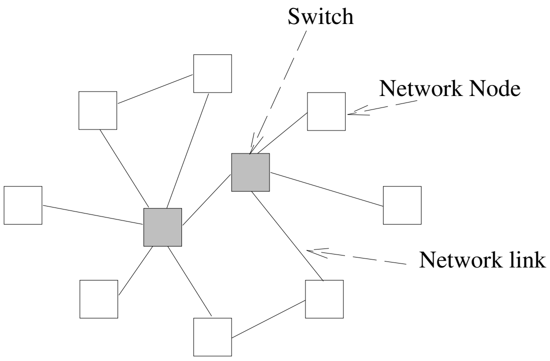

Figure 1.2: A Network with Switches (8 nodes, 2 switches, 11 bi-directional links)

The type of switch used in a network is a key component in defining how the network works. There are two main types of switching techniques: circuit switching, and packet switching. In circuit switched networks, a switch makes a physical connection between two links. The connection remains in place for the duration of the user-level conversation. In packet switched networks, a switch receives incoming information in smaller pieces (called packets or cells) that are individually transmitted across the network. The switch stores each packet temporarily, and forwards it on the appropriate outgoing link. Circuit switched networks are good at providing guaranteed quality of service to applications with a fixed rate of transmission and for connections that are established for longer periods of time (on the order of minutes). Packet switched networks are good at accommodating short bursts of communication.

Packet switched networks provide the most flexibility in delivering a wide variety of services and have been selected as the means to develop high-speed integrated networks [Chen and Dobrowski, 1988]. Switches process packets one at a time using information in a header prepended to the packet to determine its destination. The advent of optical fibre with its low error rate and high transmission rate has further enhanced the position of packet switched networks as the technology for integrated networks since simpler protocols can be developed that will speed up processing at internal nodes [Eklundh *et al.*, 1988]. It is expected, however, that internal switching nodes will continue to be the bottleneck [Petr *et al.*, 1989]. This has led to a vast amount of research into switching techniques that can accommodate the high transmission rate of the network [Ahmadi and Denzel, 1989, Daddis and Torng, 1989].

The statistical nature of packet switched networks means that unless the network is operated at a very low utilization, packets can be dropped. It is possible for the traffic presented at a node in the network to exceed the capacity of the network causing packets to be dropped. This means that congestion control or overload control algorithms are required to maintain the best possible performance for all types of services. Adding buffers at network switches can reduce the overall packet loss, but cannot solve the problem of congestion. If too little memory is used for buffers, the buffers will overflow and packets will be dropped. If a buffer of infinite size is available, packets will be delayed long enough that they will arrive too late to be useful or higher level protocols will already have begun to retransmit delayed packets [Jain, 1990]. Policies based on the priority of each packet determine which packets are added to a buffer and which are dropped. These space priority queueing policies implement a form of congestion management.

A good congestion control policy should be responsive, fair, stable, robust, and should have low overhead [Jain, 1990]. Priority queueing provides immediate response to traffic overload. In one sense it is also stable since the level of throughput does not oscillate during periods of congestion. However, this method is neither fair nor robust; a misbehaving connection will cause service degradation to all connections and priority queueing cannot guarantee acceptable levels of performance to all connections. A good queueing policy will complement other congestion control schemes.

In determining priority classes for an integrated network, the quality of service requirements for each type of application are considered. Service quality has two aspects: accurate transmission of the information and timely transmission of the information. In other words the quality of service depends on how much of the information arrives at the destination without errors and how fast it gets there. Some applications, such as telephone conversations, can afford to have small errors in transmission but require the voice information to arrive at the destination on time. Other applications, such as data transfer, can afford some delay but require that all the information arrives at the destination correctly. Priority queueing schemes can aid in meeting quality of service requirements by reducing the queueing time within the network for delay sensitive traffic. They can also help reduce the number of packets lost in applications that are loss sensitive [Chen and Guérin, 1991]. The main focus of this thesis is in examining priority queueing algorithms that address the problem of packet loss. Introducing loss management algorithms into the systems should not increase the delay for packets traveling through the network.

These issues make it clear that selecting appropriate queueing schemes at internal switching nodes is an important aspect of network performance. The goal of this thesis is to study several priority queueing techniques through simulation to determine if priority queueing is an effective means of ensuring acceptable loss performance to a wide range of traffic types. This thesis combines work done in the areas of traffic modeling, priority queueing, and switching in the context of asynchronous transfer mode (ATM) networks.

1.2 Thesis Outline

Chapter 2 describes the characteristics of BISDN and its underlying transfer (ATM). ATM was chosen because of its flexibility to support many types of services. The decision by the CCITT[1] to support ATM allows research to build on top of this foundation and to tackle issues associated with its use, such as traffic overload.

[1] International Telegraph and Telephone Consultative Committee

Chapters 3 and 4 provide the background for switching techniques and outline previous research in priority queueing and overload control. One of the difficulties in studying components of a high-speed network is that the traffic characteristics are not well understood. Except for the case of voice traffic, little work has been done to develop models that reflect the behaviour of the expected traffic. Chapter 5 describes service requirements and traffic characteristics. It also identifies traffic models that have been used in previous studies.

Chapter 6 describes the models and queueing algorithms used in this research and outlines the methodology for the experiments.

Chapter 7 presents the results of an experiment to determine how each of the traffic models affects the cell loss experienced by the switch and to determine an appropriate load level and buffer size for subsequent experiments.

Chapter 8 analyzes the results obtained for each of the priority queueing algorithms.

Finally, Chapter 9 summarizes the results of the thesis and suggests possible directions for future research.

Chapter 2

Broadband Integrated Networks and ATM

When work first began in designing a network to accommodate many types of services, fairly low bit rates were considered. The CCITT was involved in developing standards for ISDN (Integrated Services Digital Network). The *basic access* interface for ISDN provides fixed rate channels: two 64 kbps channels (B channels), and one 16 kbps channel (D channel). Another type of interface, called the *primary rate access*, offers a gross bit rate of 1.5 Mbps or 2 Mbps, by adding higher speed H channels or mixtures of B and H channels. It was clear that there was a need for services with bit rates greater than 2 Mbps and the CCITT initiated work in establishing standards for Broadband ISDN (BISDN). Two bit rates have been proposed for standardization: 150 Mbps, and 600 Mbps [CCITT, 1989]. After examining many possibilities, the CCITT decided to work towards a standard in which bandwidth could be allocated on demand rather than associated with fixed sized channels. This led to the development of the asynchronous transfer mode.

The asynchronous transfer mode (ATM) is a connection-oriented packet switching and multiplexing technique. ATM provides rules for dividing up the usable capacity of an interface and allocating it to user services. ATM was chosen as the transfer mode because it can provide the basis for supporting many different types of traffic.

Other transfer modes such as synchronous transfer mode (STM) and packet transfer mode (PTM) were considered. The routing information in STM is based on the time slot that the packet is in. This implies that bandwidth may only be assigned in multiples of the basic fixed transfer rate, as was suggested in the early stages of BISDN development. This transfer mode is best suited for services that require fairly constant bit rates. It is less suited to bursty traffic since bandwidth will be wasted if each service is assigned bandwidth equal to its maximum bit rate. PTM differs from ATM only in that it allows variable sized packets. While this is an attractive feature for most types of traffic, it substantially increases the complexity of the switches.

The bandwidth in ATM is divided into small fixed sized cells. A cell consists of a 5-octet (byte) header and 48 octets of data. The header contains the routing and channel identification information. A channel supports a single end to end user connection. Since the cells belonging to a channel are identified by a header field, the channel is termed a virtual channel. It is this concept of a virtual channel that provides the dynamic assignment of bandwidth to each service. In other words, an application is allocated bandwidth on demand. This means that services with highly variable bandwidth requirements need not waste bandwidth by reserving a fixed size channel equal to their maximum transmission rate. For those services that require guaranteed bandwidth, it is still possible to implement in ATM the notion of reserved bandwidth.

There are, however, new problems associated with choosing ATM for BISDN. For example, since bursty services are allowed simultaneous access to the network, instantaneous traffic may exceed the total capacity of the network, causing packets to be dropped or delayed. Because of the high speed at which the network will run, switching elements must be made as simple as possible. In order to reduce the complexity of switches, error recovery and flow control responsibilities are shifted to the network interfaces. At the ATM level, switches and other internal network components must preserve the sequence of cells belonging to each connection. The absence of error recovery within the network is made possible by the low error rate of the fibre-optic medium. Access control and congestion control techniques must be developed to ensure that all services using the network receive acceptable quality of service. Access control algorithms determine if there is sufficient space on the network to add a new connection. Congestion control algorithms prevent established connections from exceeding their proposed rate requirements.

Chapter 3

Switching Techniques

Chapter 1 introduced the concept of switches to reduce the total number of required communication links in a network. In a packet switched network like an ATM network, the switch must be able to extract routing information from the header of each cell and must be able to place that cell on the correct outgoing link.

Two main issues are associated with the switching techniques in any network. They include the internal structure of the switch (fabric) and the placement of buffers at the switches. Note that the switches considered here are switches inside the network. In other words the speed of the network at the inputs is the same as the speed of the network at the outputs.

3.1 Switch Fabrics

The switching techniques and problems encountered in high-speed networks are similar to those found in multiprocessor interconnection networks and consequently the literature is extensive. Two surveys [Ahmadi and Denzel, 1989, Daddis and Torng, 1989] outline the major techniques in constructing switches. Of particular interest are four classes of packet switches:

1. Crossbar-based fabrics
2. Banyan and buffered banyan-based fabrics
3. Sort-banyan-based fabrics
4. Fabrics with disjoint-path topology and output queueing

The following subsections highlight the significant features of each class of switches.

3.1.1 Crossbar-based fabrics

The most obvious way of constructing a switch is to provide a crossbar mesh such that each input is directly connected to each output [Tobagi, 1990]. Figure 3.1 gives an example of a 4 x 4 switch. Note that there is only one path from an input to an output (see the darkened path from A2 to B2 for example) and that the switch is internally non-blocking. A switch is internally non-blocking when cells destined for different outputs cannot collide inside the switch. Like all switching fabrics, the crossbar switch suffers from output port contention where cells arriving at more than one input compete for the same output, and requires buffers to be placed somewhere in the switch to alleviate this problem.

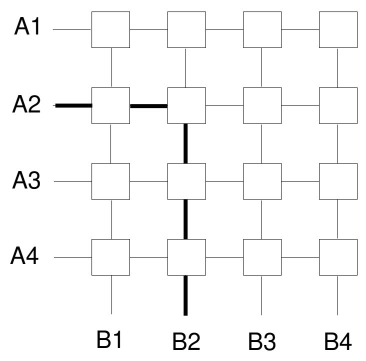

Figure 3.1: A 4 x 4 Crossbar Switch

Crossbar switches are attractive for small sized switches because they are simple, but the number of cross points grows as the square of the number of inputs which makes them uneconomical for large switches.

3.1.2 Banyan and buffered banyan-based fabrics

Originally designed for multiprocessor interconnection networks, banyan networks also have the property that there is exactly one path from each input to each output, but banyan networks are more economical than crossbar switches. The most popular subclass of banyan networks are the self-routing delta networks [Ahmadi and Denzel, 1989]. An N x N delta network is constructed from identical b x b switching nodes in k stages where $N = b^k$. Figure 3.2a shows an 8 x 8 delta switch constructed from 2 x 2 switching nodes. The complexity of banyan networks grow on the order of $N * log_b N$ [Ahmadi and Denzel, 1989]. The smaller growth make delta networks attractive for VLSI implementation. The disadvantage of these switches is that packets can collide within the switch (see Figure 3.2b). When two packets collide, one is transmitted and the other is discarded. This problem can be alleviated by placing buffers at each node inside the switch, or by using a distribution network in front of the banyan network to distribute the load evenly. The latter solution is described in the following paragraph.

3.1.3 Sort-banyan-based fabrics

To modify banyan-based switches so that they are non-blocking a batcher sorting network is added to the front of the banyan network [Hui and Arthurs, 1987]. It has been shown [Ahmadi and Denzel, 1989] that packets with distinct addresses will not collide within the banyan part of the network. Packets with the same address will collide within the banyan network but several implementations of sort-banyan switches overcome this problem. For example, the Starlite switch recycles packets with repeated addresses through the switch [Ahmadi and Denzel, 1989].

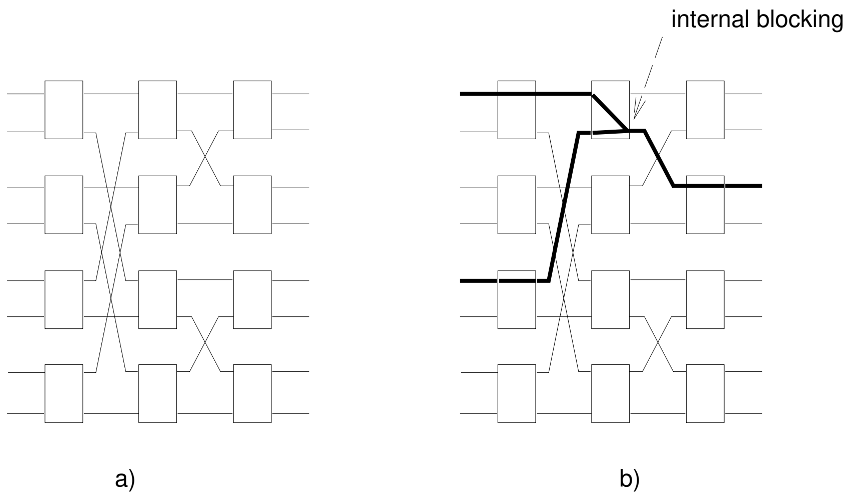

Figure 3.2: An 8 x 8 Delta Switch

3.1.4 Fabrics with disjoint-path topology and output queueing

The knockout switch is the most popular switch in this category [Tobagi, 1990, Ahmadi and Denzel, 1989]. Each input port of the switch is connected to a bus and each output port can receive packets from each bus. Each output port can also receive simultaneous packets and buffer them. This is accomplished as shown in Figure 3.3. It has been shown that for a large N and $L = 8$ a packet loss rate of less than 10^{-6} can be achieved for uniform random traffic [Yoon *et al.*, 1988]. Yoon *et al.* also examine the knockout switch under hot spot traffic and point-to-point traffic. Hot spot traffic refers to case where much of the traffic arriving at all inputs is destined for a single output. They show that for a small N (16) and L in the range of 8 to 16, the packet loss can be maintained at less than 10^{-6}. These values are obtained for hotspot traffic where the fraction of the traffic destined for a single output is 0.31.

This research assumes a non-blocking packet switch. [Ahmadi and Denzel, 1989] observe that when considering small switching fabrics each of the above techniques has

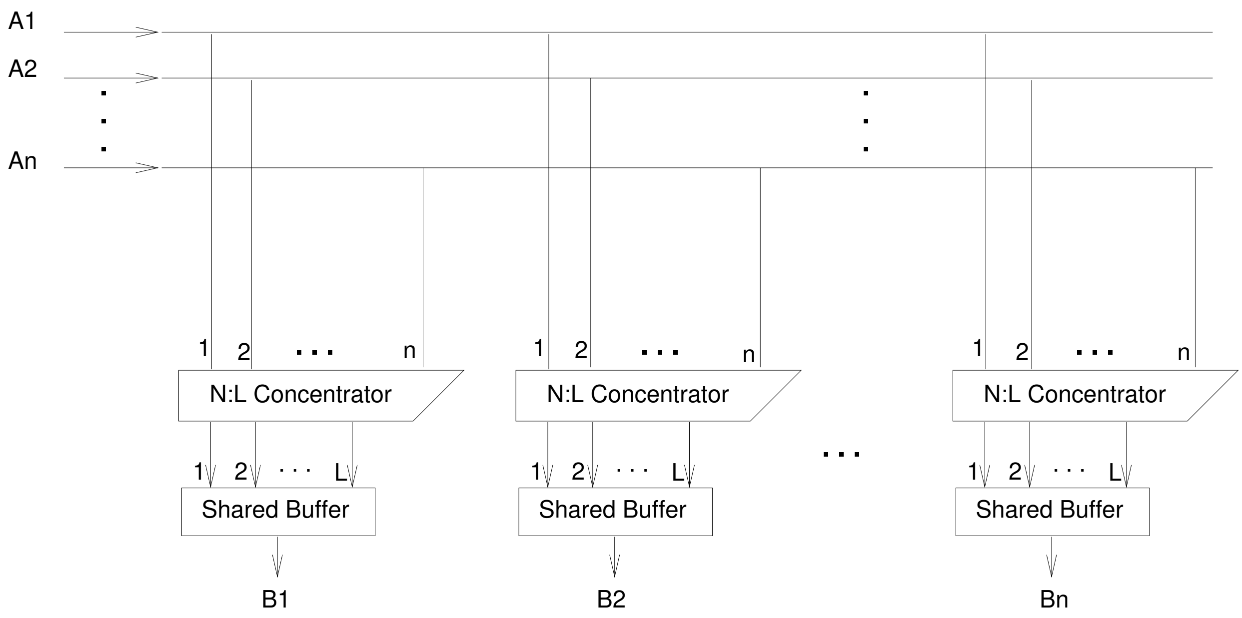

Figure 3.3: The Knockout Switch

similar capabilities and performance. It is also assumed that packets travel through the switching fabric in one switch cycle. In other words it is assumed that the switch does not queue packets internally. Note that a switch cycle may be smaller than a network cycle. Several switch cycles may be necessary to transmit all arriving packets to the output but the time required for the maximum number of switch cycles should be less than one network cycle. The only buffers associated with the switch are at the input ports and/or the output ports.

3.2 Switch Buffers

Buffers are required at packet switches since only one packet may be transmitted from each output per network time slot. More than one input may request the same output so some packets must wait for future time slots to transmit. Assuming a nonblocking packet switch, there are three possibilities for buffer placement outside the switch: at the inputs, at the outputs, or at both.

Consider the case where buffers are placed at the inputs to a switch (Figure 3.4). Each network cycle, a packet from each input may travel to an output provided that output is free. In other words, only one packet per output port may be transmitted through the switch. If more than one input wants to send a packet to the same output, only one may transmit, and the others must hold their packets in the buffer until the next cycle. This leads to the problem called head-of-line blocking [Karol *et al.*, 1987]. Figure 3.4 shows how a cell destined for output B is blocked because the cell ahead of it in the buffer is not able to travel to its destination.

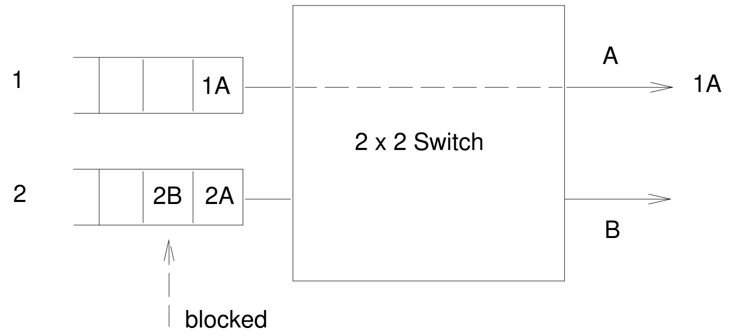

Figure 3.4: Head-of-Line Blocking

It has been shown [Karol *et al.*, 1987] that under uniform traffic where all input lines are saturated, the throughput of a switch with input buffers decreases with N and rapidly reaches a limit of approximately 0.59 as $N \to \infty$. The throughput for $N = 8$ is only 0.62. Algorithms can be used that relax the FIFO requirement of the queue but, as will be shown, the best throughput can be obtained by placing buffers at the outputs of a switch.

Placing buffers at the outputs of a switch requires the assumption that the switching fabric operates at N times the network speed, where N is the dimension of the switch (number of inputs). Each network cycle all inputs transmit their packets to the outputs and one packet per output is transmitted on the network and the rest are buffered. The switch must be N times faster than the network because it is possible that each of the N inputs will want to transmit a packet to the same output. It is assumed that there exists only one path from each input to each output within the switch so the switch must cycle through N times in order for each packet to reach the output buffer in one network cycle. If these assumptions hold true then output buffers achieve the best throughput for the switch (approaching 1).

A combined input and output buffer scheme may be advantageous for several reasons. The first is that if any of the assumptions that are required for the use of output buffers fail then input buffers will be needed to hold the packets that could not get through the switch. Second, most of the research on input versus output buffers assumes randomly distributed traffic. In the presence of hot spot traffic, for example, the total number of buffers required may be reduced if they could be spread across the inputs and the outputs. For example, in Figure 3.5a traffic from all four inputs is destined for output A. If this continued for even a short period of time the output buffer at A would overflow and packets would be lost. By also adding input buffers, the incoming traffic would be spread across all the input buffers rather than concentrating in one output buffer (see Figure 3.5b). Note that this is effective only if the traffic traveling to other output ports is relatively light. The input buffer would require some sort of relaxed FIFO algorithm to prevent head-of-line blocking.

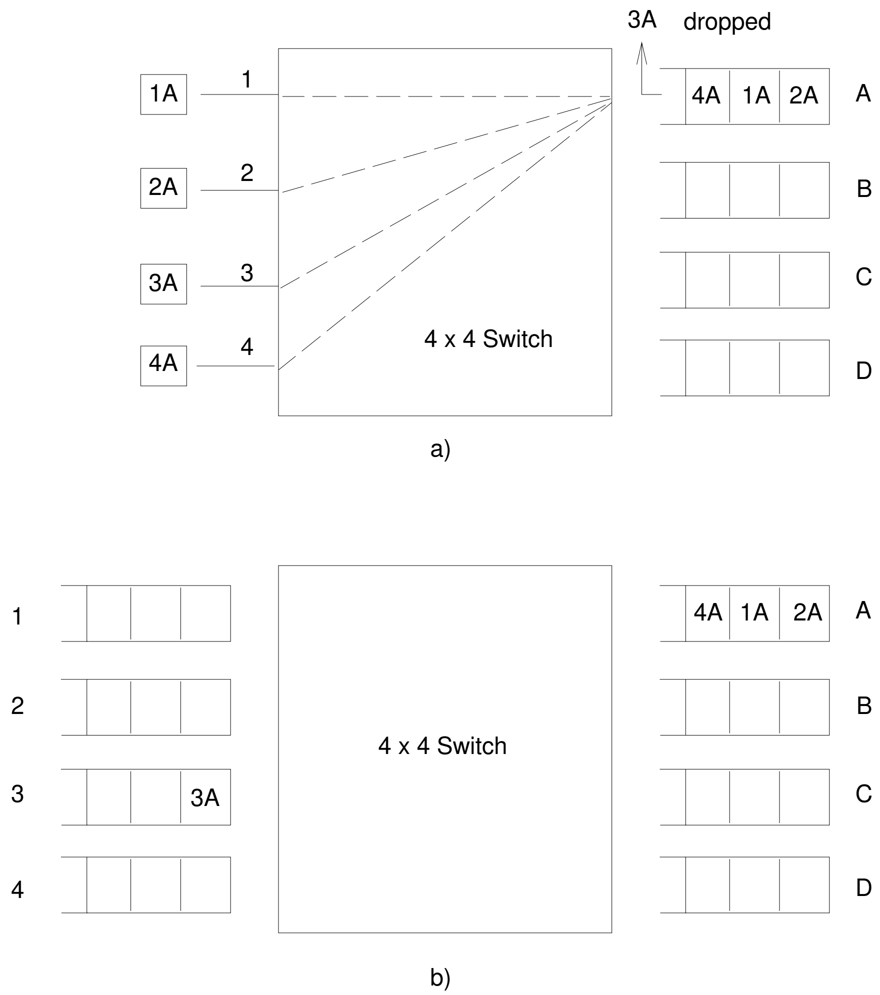

Figure 3.5: Combined Input and Output Buffers

Chapter 4

Priority Queueing and Overload Control

The term priority queueing is used to refer to the method in which packets are added to a buffer based on their priority class. There are two components of priority to consider: space priority and time priority. Space priority refers to the problem of deciding which cell to drop when a buffer fills. This is essentially a form of congestion or overload control. Time priority addresses the problem of determining the order in which cells should gain access to the switch or the order in which they should be transmitted on the network as they leave the switch. This research will consider space priority algorithms since they more directly affect congestion control. Therefore, subsequent use of the term priority queueing will refer specifically to space priority queueing.

Several approaches have been used in studying space priority algorithms on a packet switched network. They can be broadly classified into four areas: push out, buffer sharing, complete partitioning, and selective packet discarding.

The concept of a *work conserving* algorithm is introduced as a means of evaluating buffering approaches. A priority scheme is work conserving if [Lin and Silvester, 1991]:

1. no server is left idle when the buffer is not empty.
2. customers are lost *only* when the buffer is full.
3. all classes of customers have the same service requirements.
4. served customers are immediately removed from the system.
5. no work (i.e. service requirement) is created or destroyed by the particular employed rule.

All output buffering approaches satisfy rules 1, 3, and 4. Rule 1 is satisfied since one cell leaves the buffer each network cycle. Rule 3 is satisfied since the cost of transmitting each cell is the same. If a cell at the head of the buffer in the process of being transmitted on the output link were replaced by a higher priority cell work would be lost. If this rule is added, that the cell at the head of a buffer cannot be replaced, then rule 5 is satisfied. The choice of priority queueing algorithm determines whether rule 2 is satisfied.

Push out algorithms come into effect when the buffer is full. When a low priority cell arrives at a full buffer, it is dropped. When a high priority cell arrives at a full buffer, if there is a lower priority cell in the buffer the lower priority cell is "pushed out" and replaced by the high priority cell. If a high priority cell arrives at a full buffer and there is no lower priority cell in the buffer the arriving cell is dropped. This algorithm may vary the proportion of the buffer that is examined looking for a lower priority cell to replace. [Hebuterne and Gravey, 1990] and [Kröner, 1990] use analytical techniques and Poisson traffic models to study the push out scheme. Both consider two priority classes (referred to as Class 1 and Class 2). [Lin and Silvester, 1991] also use an analytical model to study a multichannel system (each input and each output consist of multiple channels). Lin and Silvester also take into account possible correlation between the two traffic classes. All three studies show that when two priority classes contribute equally to the load, the cell loss for the higher priority traffic (Class 1) is dramatically reduced while the cell loss for the lower priority traffic (Class 2) increases slightly.

Push out algorithms are work conserving; a cell is dropped only when the buffer is full. However, the cost of the algorithm is high. Each time a cell must be dropped the buffer is searched to determine which cell to drop. This is accomplished by maintaining pointers to the list of lower priority cells. Note that the management of this pointer list becomes more complicated when more than two priority classes are considered.

Buffer sharing methods limit the number of buffers that can be shared by all classes of cells. If the number of cells in the buffer exceeds this limit (threshold) then no lower priority cells will be accepted in the buffer but higher priority cells will be accepted as long as there is room in the buffer. [Lin and Silvester, 1991] and [Kröner, 1990] use similar methods as in the push out scheme to study buffer sharing approaches. [Kröner, 1990] shows that partial buffer sharing performs only slightly worse than the push out approach. [Lin and Silvester, 1991] show that show a reduction in cell loss of several orders of magnitude for class 1 traffic comes at the cost of only one or two orders of magnitude increase in cell loss for class 2 traffic.

Partial buffer sharing requires much simpler buffer management than the push out approach, but the performance of such an approach is highly dependent on the choice of the threshold and the proportion of the traffic in each priority class. It is not work conserving since cells may be dropped before the buffer is full.

Complete partitioning provides separate physical paths through the switch for each priority class. This means that all the cells in a virtual connection must be of the same priority, making it impossible to exploit priority encoding techniques. Also, multiple channels or multiple switching planes are required to support this approach. [Lin and Silvester, 1991] study this method by simulation but note its weaknesses. For example, the cell loss rate for each class depends on the traffic mix and the size of buffers allocated to each class. This approach is not work conserving. Complete partitioning is not considered in this thesis because of its serious limitations. [Lin and Silvester, 1991] show that complete partitioning provides performance similar to partial buffer sharing but show that it is possible for the class 2 cell loss probability to be smaller than that of the high priority class.

Selective packet discarding allows the system to choose which packets to discard. In this approach, packets may be discarded before the buffer is full. This approach is based on an understanding of the coding and packetizing methods for video and voice. It has been shown [Li, 1989, Hyman *et al.*, 1991] that consecutive cell loss produces a more noticeable error than losing the same number of cells distributed randomly throughout the cell stream. [Petr and Frost, 1990] describe a packet discarding method in which a decision of which packets to discard is made every switch cycle. A cost function is associated with each priority class. [Yin *et al.*, 1990] use a two-state traffic model where cells are generated with deterministic spacing. They present an analytical model in which voice packets are divided into two priority classes based on the voice coding scheme. They investigate three schemes:

embedded coding In this coding method, the information is grouped into more significant and less significant bits. The more significant bits are placed in high priority cells while the less significant bits occupy the low priority cells.

even/odd samples Even numbered voice samples are given high priority and odd numbered voice samples are given low priority. This means that consecutive cells are less likely to be dropped, allowing voice reconstruction algorithms to rebuild a missing packet by interpolating between the previous and next packet.

speech energy detection thresholds The level of speech energy identifies a sample as a talkspurt (high priority), semisilence (low priority), or silence (not transmitted).

The primary advantage of the selective packet discarding approach is that it employs knowledge about the behaviour of the traffic. It can also prevent small incidences of congestion by ensuring that the buffer does not fill too fast. It can be expensive; a function must compute which cells to drop and the cell sequence must be preserved. Finally, it is not work conserving; cells are dropped before the buffer is full.

Chapter 5

Integrated Network Traffic

5.1 Service Requirements

To begin the process of characterizing BISDN or ATM traffic it is necessary to understand the types of service that the network supports and the basic requirements of these services. It is also important to understand the wide range of requirements so that network components can be engineered to provide acceptable service to all types of traffic. The following service classifications defined by [Minzer, 1989] provide a framework in which to discuss service requirements:

Interactive and distributive. The CCITT used this distinction as the basic division of services [Wright and To, 1990]. Interactive services include telephone conversations and database retrieval. These services generally have a medium to low delay tolerance; the users do not want to wait long for responses. Distribution services include television broadcast and text news services. These services require special routing techniques in order to avoid transmitting multiple copies along the same link [Turner, 1986].

Broadband and narrowband rates. Estimated bit rate requirements range from 16 Kbps for telephone traffic to approximately 140 Mbps for high definition television (HDTV) [Käser and Rao, 1988]. These figures assume high data compression. With lower data compression, a greater bandwidth is required. It is important that all services receive acceptable performance regardless of their bit rate.

Bursty and continuous traffic. Some services will require continuous guaranteed bandwidth, particularly in the early stages of BISDN evolution [Gilbert *et al.*, 1991]. It is expected that most of the services will be bursty [Minzer, 1989], a characteristic that led to the choice of ATM as transfer mode. A difficult problem to solve will be determining the call negotiation parameters for bursty services to obtain maximum utilization of the network and minimum cell loss.

Connection-oriented and connectionless. Connection-oriented services include telephone, teleconferencing, and interactive data transfers. These applications have two phases: connection establishment and information transfer. Connectionless services do not need a separate connection establishment phase. Here the routing information and the data may be contained in the same message. The designation "connectionless" is equivalent to the CCITT concept of "messaging services" [Wright and To, 1990] and includes applications such as electronic mail, updating databases, and facsimile. This type of service is often ignored in studies of BISDN traffic since connectionless services can be provided in a connection-oriented system by establishing short connections.

Network signaling and control. Network control information will also be present on the network and special consideration may be given to this control data to ensure that it is correctly transmitted with low delay [Turner, 1986].

Point-to-point and complex communications. Some services will require a single point-to-point connection while others, such as television broadcast, will require a unidirectional point-to-multipoint connection. Applications such as teleconferencing require a multipoint-to-multipoint connection where parties may be added or deleted from the connection [Batorsky *et al.*, 1988]. Some multimedia applications will require synchronized parallel connections. For example, an application may have live voice that must be synchronized with a previously recorded video sequence.

5.2 Traffic Characteristics

Traffic can be classified according to different sets of characteristics. To get a better understanding of the types of applications that will use the network, services are described in three categories: video, voice, and data.

Video traffic has the highest bit rate and is bursty. The burstiness will depend on the types of scenes, the coding algorithms and the required video quality [Fujii *et al.*, 1988, Joseph *et al.*, 1988]. The bit rate for video ranges from 1.5 Mbps [Maglaris *et al.*, 1988] to 150 Mbps [Verbiest *et al.*, 1988]. An important issue concerning the reconstruction of the video image at the destination is that resynchronization information must be provided so that even when packets are lost the degradation of picture quality will be due only to data loss and the synchronization of subsequent packets will be correct. Perhaps the most important problem in terms of traffic characterization is that video applications are highly delay sensitive. Packets must arrive on time so that they can be "played" in order, at the correct intervals. For "live" video, the delay limit will be on the order of 10's of milliseconds. Video traffic is also loss sensitive. This means it cannot afford to lose many packets, although with prioritized coding techniques and selective cell dropping it may be possible for a video connection to maintain acceptable quality by dropping lower priority packets when necessary.

Voice traffic has a low bit rate. Its traffic characteristics on a packet network are better understood since packet switched voice networks have been feasible for a longer period of time. Like video, voice traffic is bursty and delay sensitive. Although the delay limit for voice is considerably longer (around 100 ms), this delay limit must still be adhered to [Wernik, 1988]. Voice traffic can tolerate packet loss to a certain degree. The coding schemes for voice are well understood and several authors have proposed methods for congestion control that take advantage of this knowledge [Petr and Frost, 1990, Yin *et al.*, 1990].

Data traffic on broadband networks is probably the least studied and the least predictable type of traffic. The behaviour of data traffic will depend largely on the types of applications and flow control procedures. Data traffic has the least stringent service requirements of the three types of traffic. Data applications are typically not delay sensitive but are loss sensitive. All data packets must eventually arrive at their destination for acceptable service to be achieved. Because data traffic is not delay sensitive, retransmission of lost packets is possible but this causes inefficient use of bandwidth and should be minimized.

The distinction between video, voice, and data is not sufficient to build priority classes. For example, voice mail has the same traffic characteristics as other voice applications but is not delay sensitive. Transmission of high quality music or sound will be more loss sensitive than telephone traffic. Also, interactive data traffic may have a limit over which further delay is unacceptable. Furthermore, broadcast quality video has a much lower loss tolerance than teleconferencing video. To classify traffic according to priorities, the distinction between delay tolerance and loss tolerance is important. This distinction is achieved by dividing the traffic into four categories: loss sensitive, delay sensitive, delay and loss sensitive, and best effort. Table 5.1 shows how these categories are related to the categories of voice, video, and data. It is also desirable, as noted in Chapter 4, to allow voice and video connections to divide their cells into high and low priority cells. A possible set of priority classes is described in Section 6.3.

Table 5.1: Traffic Categories

	Delay Sensitive	Delay Tolerant
Loss Sensitive	video	data
Loss Tolerant	voice	?

In order to achieve acceptable quality of service the error rates for each type of service must be kept low. [Yamamoto and Wright, 1989] outline some of the acceptable error characteristics. The acceptable bit error rate for voice is in the approximately 10^{-4}. At this rate the error is noticeable but not annoying. Some video applications require a bit error rate as low as 10^{-10}. They also note that the long term bit error rate is insufficient to guarantee acceptable performance for small time periods, and suggest that several parameters measuring the bit error rate over various time intervals will provide more information. Other authors [Woodruff and Kositpaiboon, 1990, Verbiest *et al.*, 1988] agree with these ranges.

Given this wide range of traffic characteristics and service requirements, it is important to carefully select traffic models that reflect this wide range of characteristics yet are still general enough to be used in studies of network components. The following section describes a number of traffic models that have been used in such studies.

5.3 Traffic Models

A study of any component of a BISDN network requires a workload model to test the performance of designs. Since the actual traffic on a BISDN network does not yet exist and therefore cannot be measured, estimates of the types of traffic and the traffic characteristics of potential services are the only way to obtain a workload or traffic model for use in these studies. Distributions and parameters chosen for modeling each service are justified by examining the behaviour and coding techniques of the service.

For example, the coding techniques used for video frames will have a significant impact on the traffic produced by the transmission of the video frames.

The traffic modeling problem is made more difficult by the knowledge that BISDN traffic will be bursty and correlated. This rules out using simple models such as a Poisson process to describe the traffic characteristics.

It had been thought that a traffic stream formed by multiplexing many voice sources could be approximated by a Poisson process. Although [Friesen, 1991] demonstrates that traffic smoothing occurs as multiplexed traffic streams pass through a switch, the result is far from Poisson. [Heffes and Lucantoni, 1986] also show that the multiplexing of voice lines deviates significantly from a Poisson process. They observe that even low correlation in interarrival times can have a cumulative effect and cause behaviour significantly different from that of a Poisson process. Most authors agree that to achieve more realistic results a more complex model is needed, one that takes into account the burstiness and the correlation factors of the traffic [Friesen, 1991, Eliazov *et al.*, 1990, Fujii *et al.*, 1988, Woodruff and Kositpaiboon, 1990, Gallassi *et al.*, 1989, Filipiak, 1988].

The first thing to note in examining ATM traffic as a whole is that traffic can be defined in terms of multiple layers [Filipiak, 1988, Woodruff and Kositpaiboon, 1990]. Models that take this view vary in complexity by the number of levels that they choose to model. Filipiak constructs a complex model by modeling the behaviour of each of three levels of traffic (Figure 5.1). He justifies his model by noting that "traffic flow tends to be periodic during a duration of a higher layer entity, e.g. packets are usually periodic during a duration of a burst." This phenomenon is caused by the fact that a burst generated by a single source usually consists of a single piece of information (a video frame or a data file). Since the time required to build each packet is approximately constant, the packets in a burst will be evenly spaced. It is also important to note that different granularities of traffic models are required depending on the component of the network being studied. Woodruff argues that complex traffic models are useful only if their parameters can be estimated accurately. Given the current lack of knowledge about traffic sources and total traffic mix, studies of congestion or access control should be performed with traffic models that include only the key traffic parameters that significantly affect performance. For example [Woodruff and Kositpaiboon, 1990] use peak cell rate, average cell rate, and average burst length.

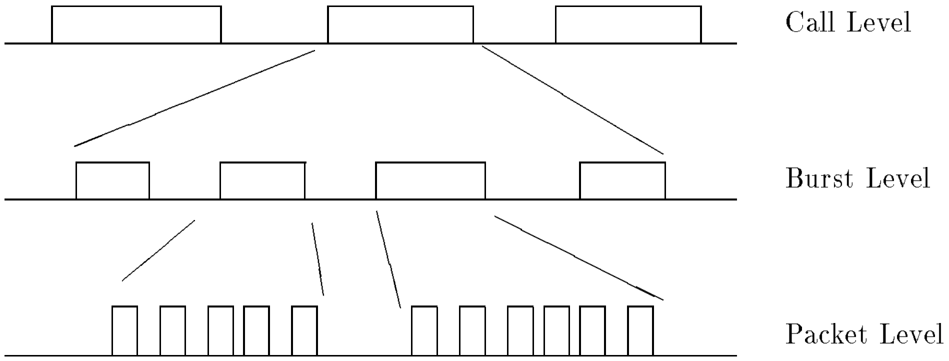

Figure 5.1: ATM Traffic Levels

Perhaps the easiest way to obtain a bursty traffic model is to use a two-state model in which one state represents a burst period. Figure 5.2 gives the parameters of such a model. To fully specify a particular model, distributions are needed for the durations of each state and for the interarrival time pattern in each state. Table 5.2 shows the distributions used in several experiments to describe the parameters in each of the two states. Here 'exp' refers to the exponential distribution, 'geom' to the geometric distribution, 'det' to a deterministic distribution, 'neg binom' to a negative binomial distribution, and 'geom mix' to a mixture of geometric distributions. In Liew's model, cells are transmitted in consecutive slots for the whole burst period. A '0' in the last column indicates that no packets are transmitted in State 2. Until actual traffic is available for measurement, validation of these models will be difficult.

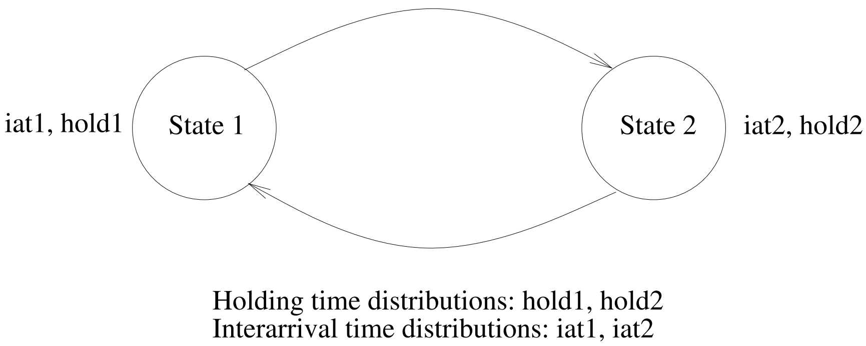

Figure 5.2: Two-state ATM Traffic Models

The choice of a deterministic rate for cell transmission during the active period is motivated by the recognition that bursts usually consist of a single transfer of a larger piece of information and that the packets generated by this transfer will likely be transmitted at a constant rate [Woodruff and Kositpaiboon, 1990, Gallassi *et al.*, 1988]. This is further supported by a measurement study of Ethernet performance by [Ling, 1992]. [Liew and Wu, 1991] use a similar model except that during a burst period, packets are transmitted continuously. Liew assumes that all packets in a burst have the same destination and that there is no correlation between bursts. He also assumes that traffic bursts from different sources will not overlap. This is a dangerous assumption since it implies that the maximum offered load to the network will never exceed the capacity of the network. It seems unlikely, given the high bandwidth of a BISDN network, that a single source will transmit packets continuously to a single destination.

The most commonly used model is the Markov Modulated Poisson Process model (MMPP) [Saito *et al.*, 1991, Yamada and Sumita, 1991, Friesen, 1991, Sriram and Whitt, 1986]. The simplest form of MMPP is a two-state model in which the durations of the two states are exponentially distributed. The interarrival times of packets in each state are also exponentially distributed. This model is used primarily for modeling aggregate traffic streams. For example, [Heffes and Lucantoni, 1986] use

Table 5.2: Two-state Traffic Models

Author	State 1 hold1	State 2 hold2	State 1 iat1	State 2 iat2
[Yamada and Sumita, 1991]	exp	exp	det	0
[Yamada and Sumita, 1991]	exp	exp	exp	exp
[Gallassi *et al.*, 1989]	exp	geom	exp	0
[Gallassi *et al.*, 1988]	exp	exp	det	0
[Saito *et al.*, 1991]	exp	exp	exp	exp
[Liew and Wu, 1991]	geom	geom	consecutive	0
[Eliazov *et al.*, 1990]	neg binom/ geom mix	neg binom/ geom mix	det	0
[Friesen, 1991]	exp	exp	exp	0
[Friesen, 1991]	exp	exp	exp	exp
[Friesen, 1991]	geom	geom	det	0
[Sriram and Whitt, 1986]	geom	exp	det	0
[Heffes and Lucantoni, 1986]	exp	exp	exp	exp

MMPP to model the superposition of voice sources. An advantage of MMPP is that it can be used in an analytical model. A special case of MMPP in which no packets are transmitted during one of the states is called Interrupted Poisson Process (IPP) [Gallassi *et al.*, 1989, Friesen, 1991].

A more complex model is achieved by extending the two-state MMPP model to a k-state MMPP. This allows an aggregate traffic stream to be represented by a model that has a variety of levels of burstiness, producing a correlated traffic stream. This is perhaps more realistic since the combination of several bursty traffic streams will produce an aggregate stream of varying burst levels.

[Eliazov *et al.*, 1990] introduce a significantly different model than the other authors. They maintain the two-state model but use a different method to arrive at the distributions for the parameters. Their approach is based on the realization that a factor that MMPP models do not capture is the periodicity in the cell streams that is inherent in a system with small packets. These periodicities are caused by the fact that it takes a constant amount of time to gather bits to form a cell. As mentioned above, when a larger piece of information is ready to be transmitted it will likely produce a traffic stream of evenly spaced cells. Parameters used to describe this model are the mean of the lengths of the active periods, their squared coefficient of variation (CV), and the cell interarrival time during the active period. The distribution for the lengths of states is determined by the values of the CV. If $CV < 1$, then negative binomial distributions are used. Otherwise, mixtures of geometric distributions are used. They argue that this method is better able to capture the significant features of the traffic, especially the burstiness.

Models for video tend to be more complex and more specific to particular coding methods. Only two models will be described in this study: one by [Hyman *et al.*, 1991] and one by [Maglaris *et al.*, 1988].

The model by [Hyman *et al.*, 1991] is also a two-state model but the method of defining the parameters differs from the models described above. The three parameters, along with the values used by Hyman *et al.* are as follows:

1. Frame duration: $F = 62.5ms$
2. Constant bit rate: $C = 10Mbps$
3. Average active period (burst length): $IE[\Sigma_{active}]$

During the active period, the source emits cells at a rate C/D where D is the cell size (48 bytes). The active period Σ_{active} is a random variable uniformly distributed between $\Sigma_{min} = 10ms$ and $\Sigma_{max} = 40ms$. At the end of the active period, the source stops emitting cells for the duration of the frame ($F - \Sigma_{active}$). The average rate of a source is

$$AR = \frac{IE[\Sigma_{active}]}{F} * C = 4\,Mbps \tag{5.1}$$

Multiplexing K video calls is accomplished by uniformly distributing the frame start time at intervals of F/Kms.

The model by [Maglaris *et al.*, 1988] was developed from measurements of motion video. The bit rate of a single source during the nth frame $\lambda(n)$ is modeled by a first-order autoregressive Markov process generated by the recursive relation

$$\lambda(n) = a\lambda(n-1) + bw(n) \tag{5.2}$$

where $w(n)$ is a sequence of independent Gaussian random variables and a and b are constants. $w(n)$ has a mean η and a variance 1. If $|a| < 1$ then the process achieves steady state with large n. The steady-state average $E(\lambda)$ and discrete autocovariance of bit rate $C(n)$ are given by

$$E(\lambda) = \frac{b}{(1-a)} * \eta \tag{5.3}$$

$$C(n) = \frac{b^2}{1-a^2} * a^n \qquad n \geq 0 \tag{5.4}$$

By matching the above two equations with measured data the values $a \simeq 0.8781$, $b \simeq 0.1108$, and $\eta \simeq 0.572$ are obtained [Maglaris *et al.*, 1988]. Using this model to generate the bit rate, bits were collected into packets that were then transmitted. Note that this model, while easy to simulate, does not lend itself to a queueing analysis.

Chapter 6

Experimental Design

The goal of this research is to study several priority queueing techniques through simulation to determine if priority queueing is an effective means of ensuring acceptable performance to all types of traffic.

There are three aspects of the problem to solve: the switch model, the priority algorithms, and the traffic models. The following three sections present each aspect separately. The fourth section describes the simulation model, and the final section describes the method for obtaining confidence intervals that was used in the experiments that follow in Chapter 7 and Chapter 8.

6.1 Switch Model

An internally nonblocking switch is modeled. The internal structure of the switch can be ignored since it has been shown that for small switches most techniques have comparable capabilities. It is assumed that there are no buffers within the switch. This means that a packet travels from an input to an output in one switch cycle.

A switch size of $N = 8$ is modeled. This size is chosen because it is small enough to allow for reasonable simulation runtimes and large enough to examine realistic problems. This size also conforms to the size of switches studied in other work [Eliazov *et al.*, 1990].

The switch is modeled with output buffers only, so it is assumed that the switch operates at N times the speed of the transmission medium. It is also assumed that buffers are dedicated to each output line and are not shared.

The following events occur in each time slot:

1. Traffic arrives at inputs 1 to N

2. Traffic crosses the switch – the order in which cells cross the switch is randomly distributed.

3. At each output port the queueing algorithm decides whether cells are to be buffered or dropped.

4. At each output port, if a cell arrived in this time slot or there is one or more cells in the buffer, a cell is transmitted.

A cell may arrive at an input and leave from an output in the same time slot if the output buffer is empty. For example, in Figure 6.1, three cells arrive at the switch at time $t = i$ and three cells leave. The cell 3A arrives and leaves in the same time slot since output buffer A is empty. No cells arrive at the inputs at time $t = i + 1$ but cells 1C and 2C have been transmitted through the switch and cell 1C departs.

It is reasonable to model only one output buffer. This simplifies the switch model and the traffic models since only traffic destined to the "tagged" output buffer is modeled.

6.2 Priority Queueing Algorithms

Three types of algorithms are examined: push out, buffer sharing, and selective packet discarding. The case of no priority is simulated with each traffic model to provide a

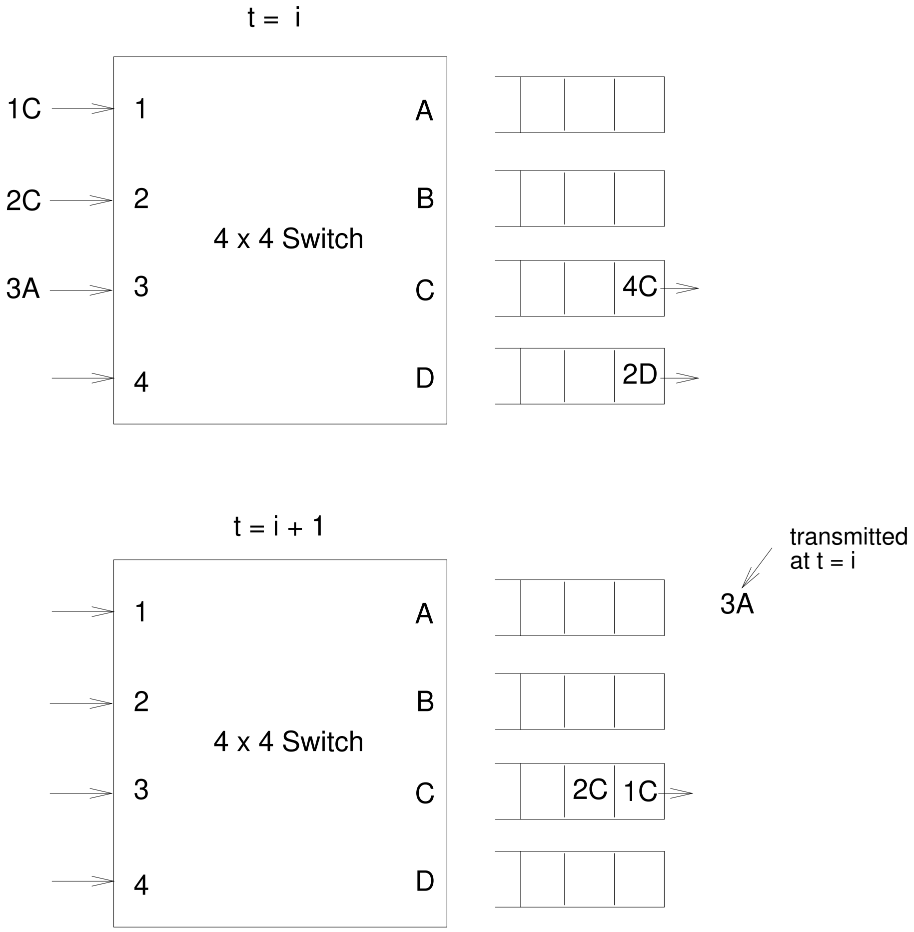

Figure 6.1: Switch Timing

basis for comparison. Complete partitioning is not studied because its performance depends so heavily on the traffic mix and because it does not allow a single connection to transmit packets of different priority classes.

The first variable to consider is the length of the queue. Some preliminary tests were run with the proposed traffic models to determine, for each combination of traffic and level of network load, the range of queue lengths where the crossover from acceptable cell loss to unacceptable cell loss is found. It is in this range where priority queueing algorithms may provide acceptable quality of service for all classes of traffic.

For the push out algorithm, the only variable is the number of cells in the buffer that are tested to find a lower priority cell. For example, only the final cell of the buffer may be tested to see if it has a lower priority than the cell that has just arrived.

The performance of partial buffer sharing algorithms will depend on the portion of the buffer that is reserved for high priority traffic only. This variable is dependent on the proportion of the traffic in each priority class. When four priority classes are considered three parameters identify reserved sections of the buffer. The primary purpose of studying this algorithm is that it requires the least cost for buffer management.

A simple algorithm has been chosen to represent selective packet discarding. Two threshold values (B2, B1) define the overload and underload periods respectively (see Figure 6.2). When the number of cells in the buffer exceeds B2, an overload control mechanism is activated until the buffer length drops below B1. The overload control mechanism decides which cells to drop based on the probability of dropping a packet of each priority level. These probabilities are related to quality of service requirements and are specified per switch. This means that the probability of dropping a cell at the switch must be lower than the end to end probability of cell loss. It is assumed that the probability of dropping a low priority cell is greater than the probability of dropping a high priority cell. Note that the percentage of packets dropped depends on the traffic mix.

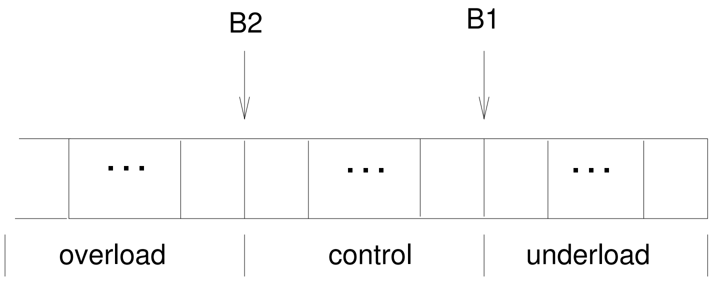

Figure 6.2: Selective Packet Discarding

6.3 Traffic Models

One of the first problems to solve in developing traffic models for this experiment is how to divide the traffic into priority classes. Two methods are proposed: one with two priority classes, and with for four priority classes. The decision to model two priority classes was based on the recommendation by the CCITT that one bit in the ATM header be used for loss priority [Eckberg *et al.*, 1991]. The decision to model four priority classes is based on the realization that two priority classes may not adequately represent the differences between varying degrees of loss sensitive and delay sensitive traffic. One of the goals of this thesis will be to determine if the number of priority classes influences the quality of service for each traffic type.

When restricted to two priority classes only the distinction between loss sensitive traffic and loss tolerant traffic can be made. Table 6.1 shows which types of traffic fit into each priority class.

Table 6.1: Two Priority Classes

Class		Traffic types
1	delay and/or loss sensitive	Data High priority video High priority voice
2	best effort	Low priority video Low priority voice

Using four priority classes allows a finer distinction between service requirements. Table 6.2 gives the definitions and traffic types for each class.

The second major problem to solve is associating traffic models with the priority classes mentioned above. For this research a model is provided for each type of traffic: data, voice, and video. The CCITT Recommendation I.311 [Rathgeb *et al.*, 1988] proposes that three parameters be provided by a connection as descriptors of its behaviour: average cell rate, peak cell rate, and mean burst duration. These

Table 6.2: Four Priority Classes

Class		Traffic types
1	delay and loss sensitive	High priority video
2	loss sensitive	Data
3	delay sensitive	High priority voice
4	best effort	Low priority video Low priority voice

descriptors will be used in call acceptance and enforcement algorithms. They are also useful in establishing parameters for traffic models. The three traffic descriptors are used to define the parameters for the data model and the voice model.

The following subsections describe each of the traffic models.

6.3.1 Data Model – Geom/Exp/Det

A simple model for data traffic does not exist in the literature but it should be sufficient to use a two-state model where packets arriving during a burst have deterministic arrival times. This mimics the behaviour of larger file transfers where the source requires approximately the same amount of time to build each packet.

The model chosen has an exponentially distributed idle period with mean $1/\alpha$ and the number of packets in a burst is geometrically distributed with mean λ. During a burst packets are transmitted at the peak rate. Given the peak bit rate (Bp), mean bit rate (Bm), and mean burst length ($1/\beta$), the parameters of the model ($1/\alpha$ and λ) are calculated as follows

$$\lambda = \frac{Bp \times (1/\beta)}{C}, \qquad \frac{1}{\alpha} = \frac{\lambda \times C}{Bm} - \frac{1}{\beta} \tag{6.1}$$

where C is the size of a cell.

6.3.2 Voice Model – MMPP

MMPP is the most widely accepted model for aggregate voice traffic. The number of calls represented can be varied by changing the parameters to the model. Recall that MMPP is a two-state model where the holding times in state 1 and state 2 are exponentially distributed with means $1/\alpha$ and $1/\beta$, respectively. The interarrival times in each state are also exponentially distributed with means $1/\lambda_1$ and $1/\lambda_2$. For this model the three traffic descriptors are not enough to define the parameters of the model so an additional parameter, p_1, is introduced. p_1 denotes the proportion of arrivals that occur in state 2. [Friesen, 1991] provides the following equations to calculate the MMPP parameters, where π_2 is the probability of being in state 2 and $Bp = \lambda_2$, Bm, $1/\beta$, and p_1, are given:

$$\pi_2 = \frac{p_1 \times Bm}{Bp}, \quad \lambda_1 = \frac{(Bm - Bp \times \pi_2)}{(1-\pi_2)}, \quad \frac{1}{\alpha} = \left(\frac{1}{\pi_2} - 1\right) \times \frac{1}{\beta} \tag{6.2}$$

This model had to be modified to enable it to label cells as high or low priority. Chapter 4 described voice encoding methods studied by [Yin *et al.*, 1990]. In each of the three encoding methods the low priority packets were evenly distributed throughout the traffic stream. Also, as many as half of the packets were low priority packets. Because MMPP models aggregate voice traffic low priority cells are randomly distributed throughout the traffic stream and the proportion of low priority cells to high priority cells is determined by a parameter to the model.

6.3.3 Video Model – Maglaris

The video model developed by [Maglaris *et al.*, 1988] and described in Section 5.3 will be used for the video model. The model parameters are modified to allow sufficient variation in a shorter simulation run and to vary the load presented by a video connection. [Maglaris *et al.*, 1988] considered a low bandwidth video source where the mean bit rate is on the order of 1.5 Mbps. They note that the basis of their model, the assumption that autocovariance for bits per pixel can be approximated by an exponential curve, has been shown to hold for a wide variety of video sources. Recall that bits per pixel is another way of expressing bit rate given the number of pixels per frame and the number of frames per second. Because the parameters to this model are defined in terms of fitting the covariance function to an exponential curve, it was not possible to define the parameters in terms of peak bit rate, mean bit rate, and burst length. Note, however, that the burst length is defined as the inverse of the number of frames per second.

It is expected that video traffic will have a long burst length. For example, [Maglaris *et al.*, 1988] assume that a video source generates 30 frames per second, which corresponds to a burst length of approximately 35 ms. [Verbiest *et al.*, 1988] assume a burst length of 40 ms. However, burst lengths in this range would require infeasible simulation lengths to achieve sufficient variation and to obtain reasonable confidence intervals so a much shorter burst length (on the order of 1 or 2 ms) was used. This burst length corresponds to 600 frames per second (1.7 ms) in the Maglaris model. Longer burst lengths lead to smoother traffic at the switches and if call acceptance algorithms ensure that the total traffic presented to the network does not exceed the network's capacity, this should lead to lower cell loss.

Recall that this model computes the bits per pixel in the nth frame based on the formula shown in Equation 5.3 in Section 5.3. For this experiment the same values for the constants $a \simeq 0.8781$ and $b \simeq 0.1108$ were used. Likewise the same value for the mean of the Gaussian random variable was used: $\eta \simeq 0.572$. These values control the shape of the traffic. Since the model computes the number of bits per pixel in a frame, a further step is required to generate the number of cells per frame, or equivalently, the number of cells in a burst. This means that a single parameter, the number of pixels per frame, can be used to vary the mean cell rate of the model.

The video model was modified to label cells as high or low priority. The method used is identical to that used for the voice model. A parameter is added to the model to specify the proportion of high and low priority cells.

6.4 Simulation Model

The models described in the previous sections are combined in the simulation as shown in Figures 6.3 and 6.4. Eight traffic sources (denoted by T1 to T8 in Figure 6.3) inject traffic into a switch with eight input ports and eight output ports. Only one output port is modeled and all the generated traffic is transmitted to this output port. A buffer is associated with the output port. A cell is described by the time it arrives at the switch and its priority level. Arriving cells pass through the switch model as defined in Section 6.1. A priority queueing algorithm checks the cells as they enter the output buffer to determine which cells should be dropped. Note that the priority queueing algorithm operates on one cell at a time, so if multiple cells arrive in the same time period they are processed one at a time in the order that they arrive at the output buffer. When the buffer is occupied, a cell leaves the buffer every time slot to be transmitted to the rest of the network. Queueing delay is measured when cells leave the output buffer.

Each input traffic stream is made up of three separate traffic streams generated by the models described in Section 6.3. The output from each traffic model is merged to produce the traffic stream Ti (see Figure 6.4).

Figure 6.5 shows the general shape of the output from each of the traffic models, the parameters of which are given in Table 6.3. The mean bit rate for each traffic model is 15 Mbps but the peak bit rates vary. Note the different behaviour of each model. The data traffic rate drops to 0 frequently. The variation in arrival rates is much greater for the data model than the other two models. The video model shows the least variation in the traffic rate, and this variation occurs over longer periods of

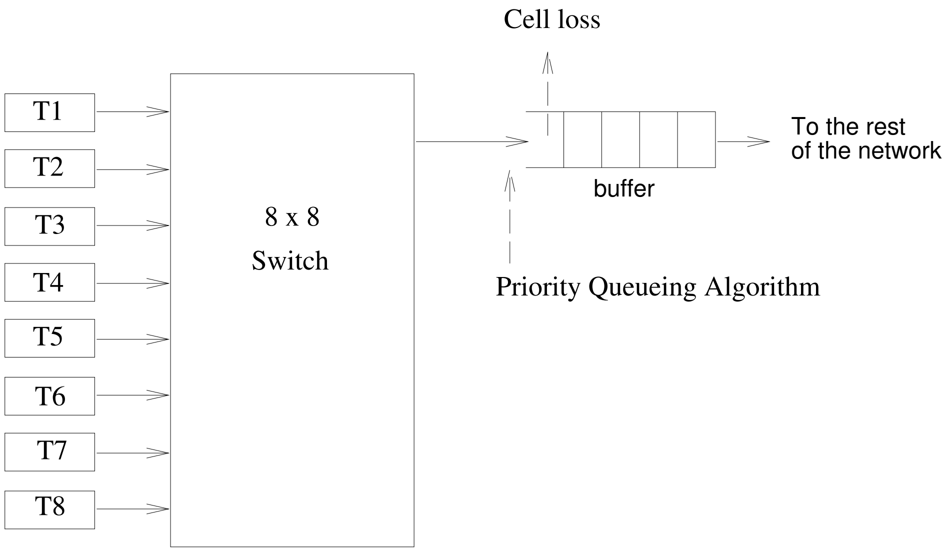

Figure 6.3: Simulation Model

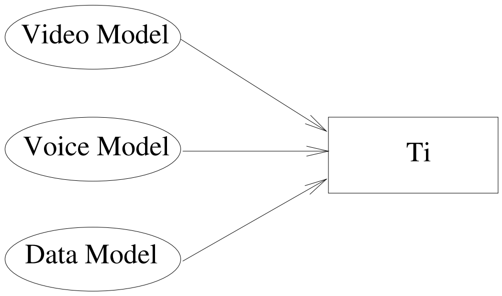

Figure 6.4: Input Traffic Models

time than in the other two models. Both video and voice show a sustained level of traffic, meaning that the arrival rate rarely drops to 0.

6.5 Confidence Intervals

Statistically significant results were obtained using the method of batch means to determine simulation run lengths. Since the input traffic is not independent it was not sufficient to compute confidence intervals for cell loss based on the variance of the whole sample. The output of each simulation was divided into m subsamples with

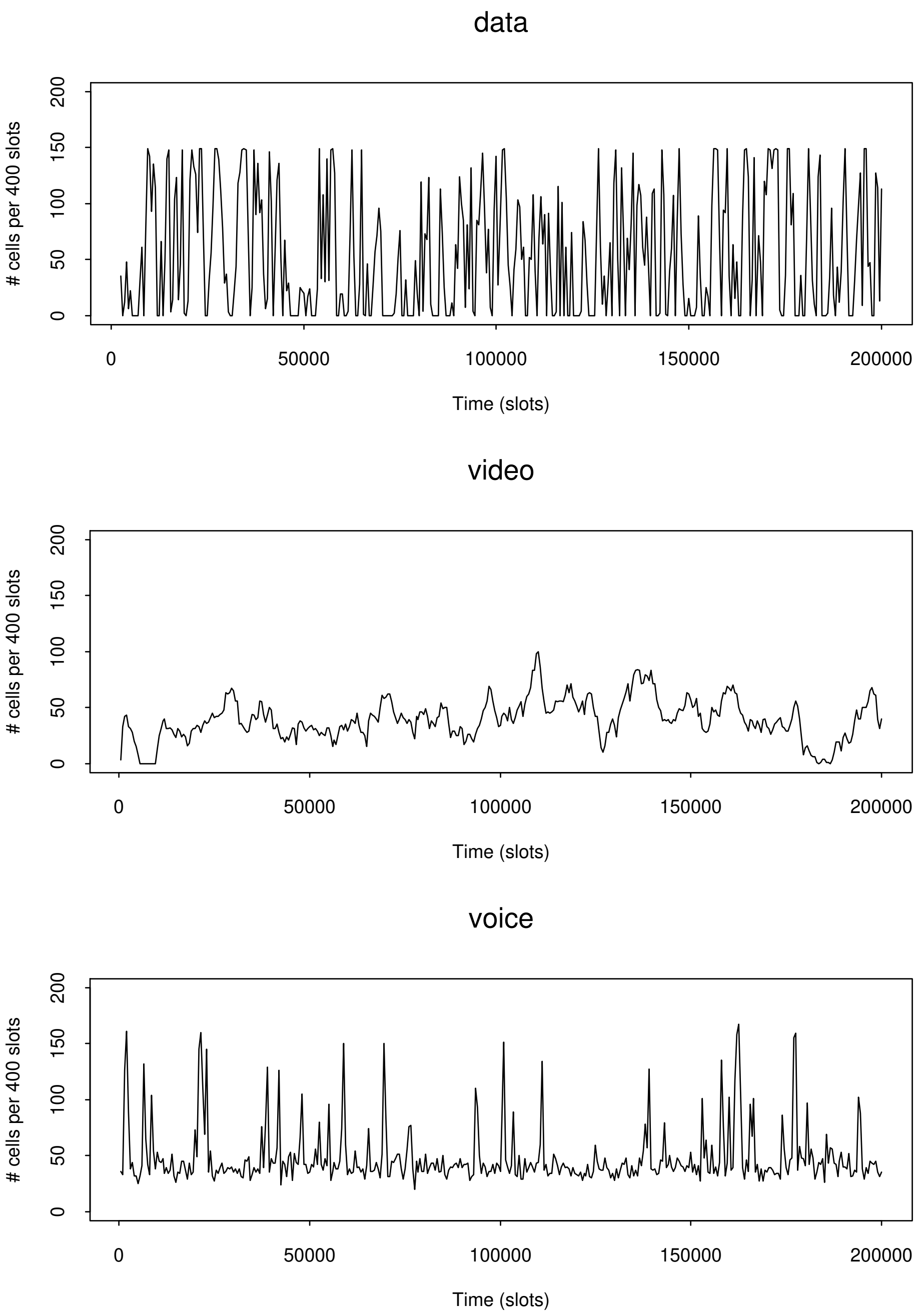

Figure 6.5: Shape of Three Traffic Models (load = 0.1)

Table 6.3: Model Parameters and Measurements

Model	Parameters				Measured		
	Bm (Mbps)	Bp (Mbps)	$1/\beta$ (slots)	p_1	Bp (Mbps)	Bm (Mbps)	$load$
Data	15.0	45.0	357	–	15.6	45.1	0.10
Voice	15.0	45.0	357	0.3	15.0	55.4	0.10
Video	pixels/frame = 52000				15.2	38.2	0.10

n samples each. The total number of samples is $N = m \times n$. The mean for each subsample is

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \quad i = 1, 2, \ldots, m \tag{6.3}$$

The overall mean is

$$\bar{\bar{x}} = \frac{1}{m}\sum_{i=1}^{m} \bar{x}_i \tag{6.4}$$

If the subsamples are long enough they can be assumed to be independent. Then the variance of the means of each subsample is used to calculate the confidence intervals.

$$Var(\bar{x}) = \frac{1}{m-1}\sum_{i=1}^{m}(\bar{x}_i - \bar{\bar{x}})^2 \tag{6.5}$$

The length of each subsample is long enough if the covariance of the subsample means is small with respect to the variance. If $m > 30$ then the confidence level $(1-\alpha)100\%$ is

$$\bar{\bar{x}} \pm z_{1-\alpha/2}\sqrt{\frac{Var(\bar{x})}{m}} \tag{6.6}$$

Preliminary tests of the simulation showed that a sample size of $n = 30000$ and $m = 100$ subsamples provided reasonable confidence intervals. These intervals were as high as $\pm 10\%$ but were small with respect to the differences between results.

Measures for average buffer occupancy and average queueing delay for each priority class were obtained using the same method. Note that these two measures are related.

Average buffer size is likely to be smaller than the average queueing delay since when the buffer is empty for a few cycles it is reflected in the average buffer size but this will not be reflected in the average delay since delay is measured only when a cell leaves the buffer. Since the two measures are so closely related only queueing delay will be reported.

Chapter 7

Buffer Size Experiment

Choosing the buffer size for an ATM switch is a difficult problem. The cell loss rate for a particular buffer size depends on load, type of traffic and level of burstiness. Increasing the buffer size to reduce cell loss eventually leads to increasing the delay within the network beyond acceptable levels for real time traffic.

To examine how each traffic model contributes to the cell loss at the buffer, experiments were performed where each model generated all the traffic. This set of experiments was also used to select a buffer size for subsequent experiments.

7.1 Data Traffic

Table 7.1 gives the parameters used to generate the load levels shown in Figure 7.1. The burst length was set to 357 slots which is equivalent to 1 ms. Cell loss is measured as

$$\frac{\#\text{ of cells lost}}{\text{total cells generated}} \tag{7.1}$$

Because the data traffic model generates packets at the peak bit rate with idle periods in between, the period over which the traffic exceeds the capacity of the output buffer is relatively short and adding more buffers reduces the total cell loss. However, the

Table 7.1: Data Traffic Parameters

Load	Input Ports	Parameters
low	1–4	Bp = 35 Mbps, Bm = 10Mbps, load = 0.066
	5–8	Bp = 45 Mbps, Bm = 15Mbps, load = 0.100
medium	1–8	Bp = 45 Mbps, Bm = 15Mbps, load = 0.100
high	1–4	Bp = 55 Mbps, Bm = 25 Mbps, load = 0.164
	5–8	Bp = 45 Mbps, Bm = 15 Mbps, load = 0.100

slope of the lines is not constant, indicating that the benefit of adding more buffers diminishes. Note that the increase in cell loss from the load level 0.80 to the load level 1.06 for each buffer size is approximately 0.1. This corresponds to an increase in cell loss at a buffer length of 10 of

$$\frac{(0.18 - 0.08)}{0.08} \times 100\% = 125\% \tag{7.2}$$

or a 250% increase at a buffer length of 250. At such a high load the network begins to saturate and cell loss increases dramatically with small increases in load level.

7.2 Voice Traffic

Although the parameters used for voice and data traffic models are identical and yield the same level of burstiness their performance differs greatly. Table 7.2 gives the parameters used in this experiment. Again, the burst length was set to 357 cells. Figure 7.2 shows that the total cell loss is much lower for voice traffic than for the data traffic. This is due to the distribution of cell arrivals through both states of the voice model. In the data model, cell arrivals are concentrated in the burst period but in the voice model only 70% of the cells arrive in the burst period (State 2). Note that the curves begin to flatten out indicating that adding more buffers will not significantly reduce the cell loss.

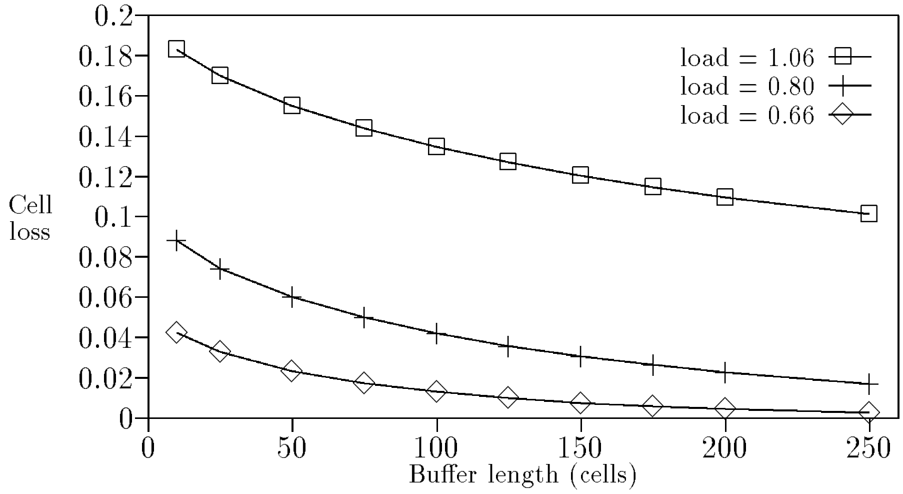

Figure 7.1: Cell Loss for Data Traffic

7.3 Video Traffic

Unlike the models for voice and data, the parameters for the video model do not directly give peak and mean bit rates. Table 7.3 shows the parameters used for each input traffic stream as well as the measured peak and mean bit rates.

Figure 7.3 shows that increasing the buffer size does not significantly reduce the cell loss experienced by this type of traffic. Recall from Figure 6.5 that the video

Table 7.2: Voice Traffic Parameters

Load	Input Ports	Parameters
low	1–4	Bp = 35 Mbps, Bm = 10Mbps, load = 0.066
	5–8	Bp = 45 Mbps, Bm = 15Mbps, load = 0.100
medium	1–8	Bp = 45 Mbps, Bm = 15Mbps, load = 0.100
high	1–4	Bp = 55 Mbps, Bm = 25 Mbps, load = 0.165
	5–8	Bp = 45 Mbps, Bm = 15 Mbps, load = 0.100

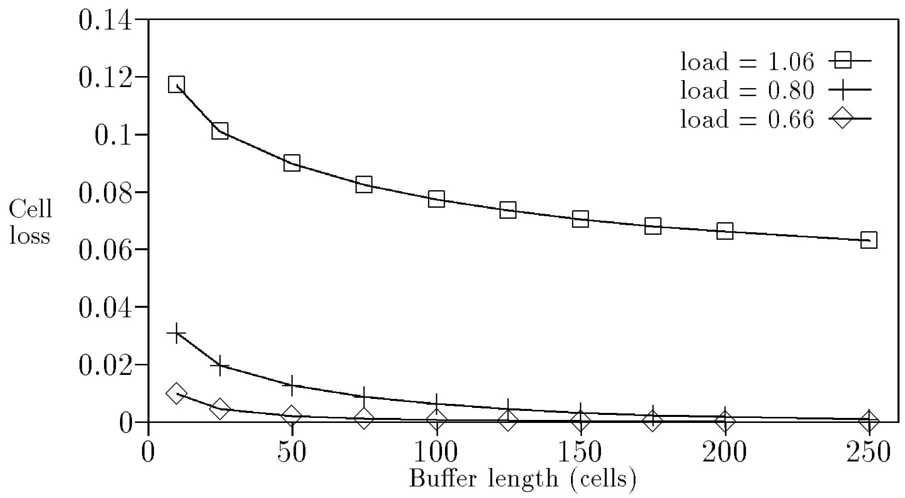

Figure 7.2: Cell Loss for Voice Traffic

traffic had a higher sustained rate. This means that although the mean bit rate is the same, there is much less variation in bit rate in the video model than in the other two models. This means that the number of occupied cells in the buffer increases and decreases more slowly. The total cell loss is lower than in the other two traffic models due to a smoother traffic stream. In fact, at a load of 0.68 and a buffer size greater than 25, no cells were lost over the time period simulated.

Table 7.3: Video Traffic Parameters

Load	Input Ports	Pixels/Frame	Bm (Mbps)	Bp (Mbps)	Load
low	1–8	40000	12.7	34.0	0.084
medium	1–3	60000	19.0	48.2	0.125
	4–8	40000	12.7	34.0	0.084
high	1–2	76000	24.4	50.6	0.161
	5–8	60000	19.0	48.2	0.125

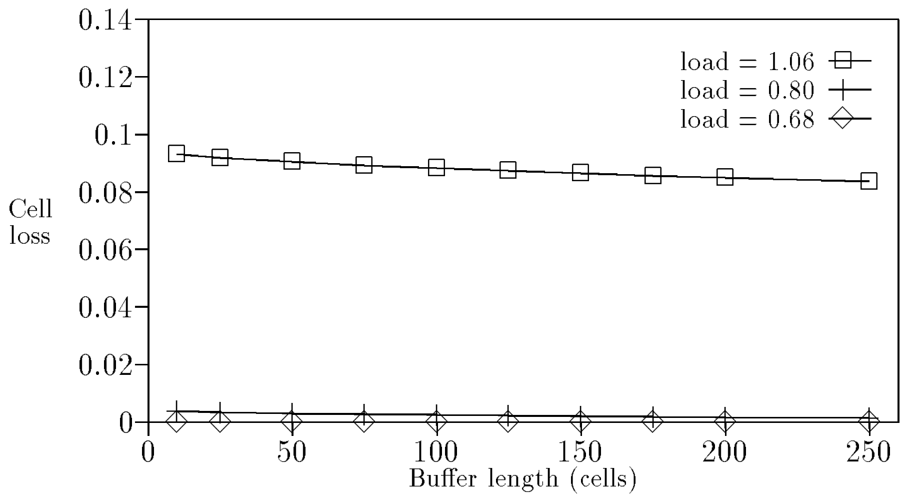

Figure 7.3: Cell Loss for Video Traffic

7.4 Summary

Studying the effects of each traffic model on a switch buffer provides some insight into the problem of buffer management. The data model showed the most cell loss at each load level. This can be attributed to the concentration of cell arrivals in the burst periods. The results from both the data and the voice models indicate that adding buffers decreases cell loss, but that as the buffer size grows the additional benefit decreases. The video model showed that at the load levels considered, adding buffers does not significantly decrease the cell loss. This is because the video model generates a smoother traffic stream than the other two models so cell loss occurs over an extended period of time.

A buffer size of 50 was chosen for the subsequent experiments on the priority queueing algorithms. This choice was based on the results from this section; the cell loss from the voice and data traffic began to flatten out at a buffer size of approximately 50 when the load is 0.80. This buffer size was also chosen to contain the length of the simulations while providing sufficient variation in parameter ranges to demonstrate the behaviour of the algorithms. Note that a buffer size of 50 corresponds to a maximum queueing delay of 140 μs per switch.

The results from this section show that for each traffic model the following cell loss is expected (at load = 0.80): 0.003 for video, 0.013 for voice, and 0.06 for data. Although this level of cell loss is clearly unacceptable in a real network, the results shown in the next chapter should scale to systems with larger buffers and therefore lower cell loss. The following chapter reports on experiments to study priority queueing algorithms to determine their effect on cell loss for each of the priority classes. Five combinations of the traffic models described in this section were used to generate the traffic. Each of the combinations produced a total load of approximately 0.87. Although this load level is at the high end of the operating region, it provides a good environment in which to exercise the priority queueing algorithms. It is expected that the priority queueing algorithms studied in this thesis would be even more effective in protecting high priority traffic from cell loss at lower load levels.

Chapter 8

Priority Queueing Experiments

8.1 Outline of Experiments

To obtain an experimental set of a reasonable size, five traffic mixes were chosen, each providing a total aggregate load of approximately 0.87. Tables 8.1 and 8.2 show the composition of each of the traffic mixes. Class 1 and Class 2 represent high and low priority classes respectively. In the first mix, each model contributes equally to the total load. In mixes 2–4, the traffic from one of the models dominates. The fifth mix was included to provide an example where the lowest priority class has a higher proportion of cells. The only difference between mix4 and mix5 is the proportion of cells in each class. The purpose of structuring the traffic mixes in this way was to provide a sufficiently wide range of traffic shapes. Note that the proportion of the traffic belonging to each priority class is governed by the traffic mix and by the functions determining the priority classes in video and voice. Therefore, this choice of traffic mixes limits the amount of traffic in each priority class.

The cell loss for various buffer sizes was measured using each of the five traffic mixes as input traffic (see Figure 8.1) to compare the results obtained in the previous chapter with the traffic mixes presented here. As expected, the traffic mixes in which

Table 8.1: Composition of Traffic Mixes (traffic types)

Name	% video	% voice	% data	Load
mix1	33.3	33.3	33.3	0.87
mix2	50	25	25	0.88
mix3	25	25	50	0.87
mix4	25	50	25	0.87
mix5	25	50	25	0.87

Table 8.2: Composition of Traffic Mixes (priority classes)

Name	Two Classes		Four Classes			
	% Class 1	% Class 2	% Class 1	% Class 2	% Class 3	% Class 4
mix1	76	24	27.7	29.9	18.3	24.1
mix2	75	25	36.3	24.0	14.5	25.2
mix3	82	18	19.0	48.3	14.7	18.0
mix4	72	28	18.9	24.3	29.2	27.6
mix5	57	43	13.5	24.3	19.5	42.8

voice and video dominate show the least cell loss. Remember that mix4 and mix5 differ only in the distribution of traffic among the priority classes so their total cell loss is the same. Once again the curves flatten out as the buffer size increases indicating that the benefits of increasing the buffer size diminish as more buffers are added.

The following sections describe how each of the priority queueing algorithms affects the cell loss of each traffic mix when two or four priority classes are used. They show how the parameters to each algorithm change the distribution of cell loss among the different priority classes. A buffer size of 50 was used for all of the remaining experiments.

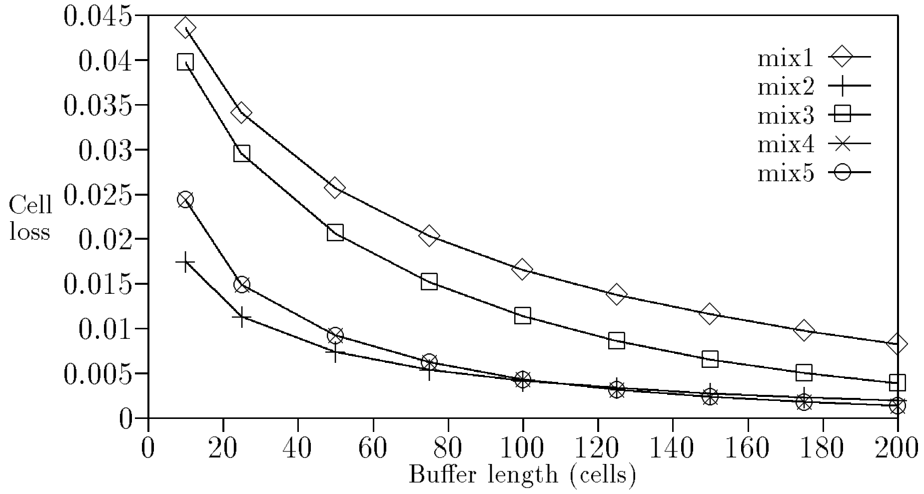

Figure 8.1: Cell Loss for Five Traffic Mixes

The results that follow present the cell loss of a priority class i in two ways: as a percentage of the total cells generated (T_i) and as a percentage of the number of cells in the class i (C_i).

$$T_i = \frac{\text{\# of Class } i \text{ cells lost}}{\text{total cells generated}} \tag{8.1}$$

$$C_i = \frac{\text{\# of Class } i \text{ cells lost}}{\text{\# of Class } i \text{ cell generated}} \tag{8.2}$$

The first method provides information about how the total cell loss is divided among the priority classes while the second method shows how cell loss affects each class. Average queueing delay is also measured to show how the priority queueing algorithms affect delay.

8.2 Two Priority Classes

This section describes the performance of each of the priority algorithms when two priority classes are used. The CCITT allows a single bit in the ATM header for loss priority so examining the algorithms using two priority classes is a natural place to start. The effects of each of the algorithms are also more clearly demonstrated when only two priority classes are used. Section 8.3 presents the results of using four priority classes with these algorithms.

8.2.1 Push Out Algorithm

Figures 8.2–8.6 give the results of the push out algorithm where cell loss for each priority class is presented in terms of the total number of cells arriving at the switch (T_i). This means that the sum of the loss for each of the priority classes equals the total cell loss.

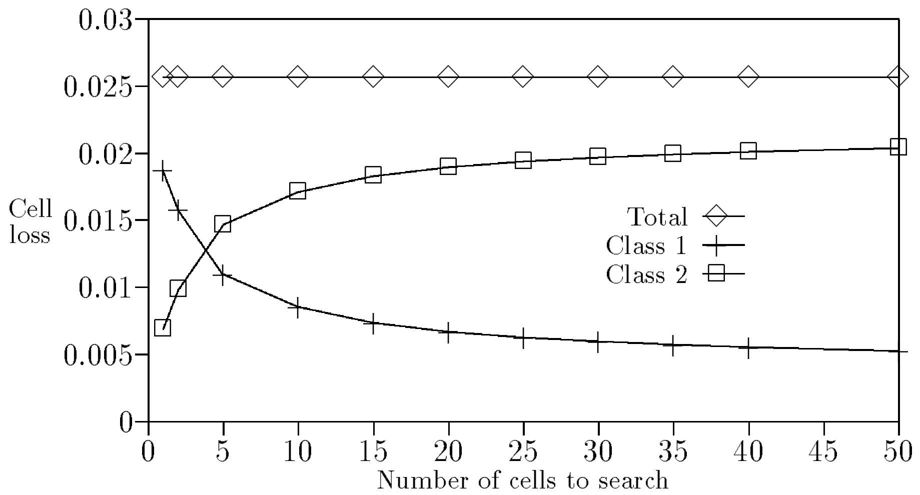

Figure 8.2: Push Out (mix1, two classes, fraction of total)

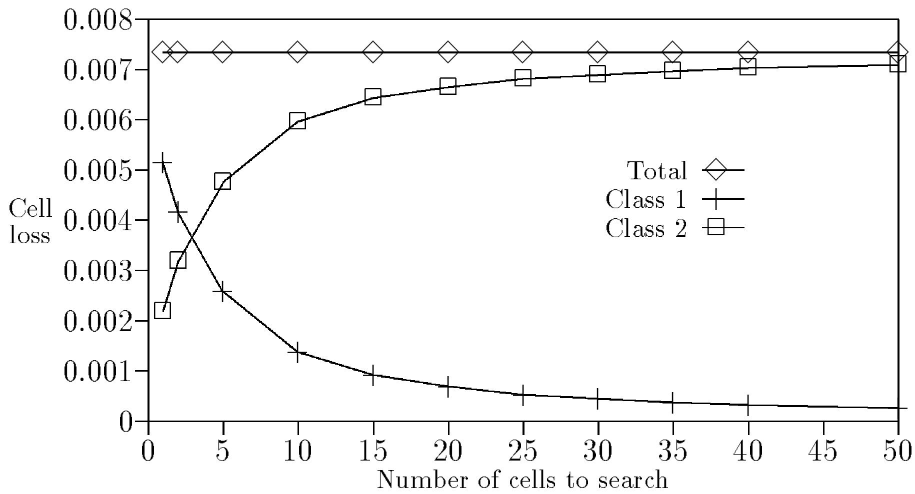

Figure 8.3: Push Out (mix2, two classes, fraction of total)

The total cell loss for all of the figures in this section is the same regardless of how many cells are searched. This is a function of the work conserving property of the push out algorithm. Since cells are dropped only when the buffer is full, the total cell loss will always be the same for a given traffic workload.

Figures 8.2–8.5 show that when fewer than 5 cells (10 in the case of mix3) are checked to see if they contain lower priority cells that can be replaced, Class 1 cells make up the majority of the total cells lost. This is because Class 1 cells make up such a high proportion of the total cells in each of these traffic mixes (see Table 8.2). Figure 8.6 (mix5) confirms this statement; when only 1 cell is checked to see if it has a lower priority, Class 2 accounts for more of the total cell loss than Class 1. Increasing the number of cells searched has the effect that the share of the total cell loss diminishes for Class 1 and increases for Class 2.

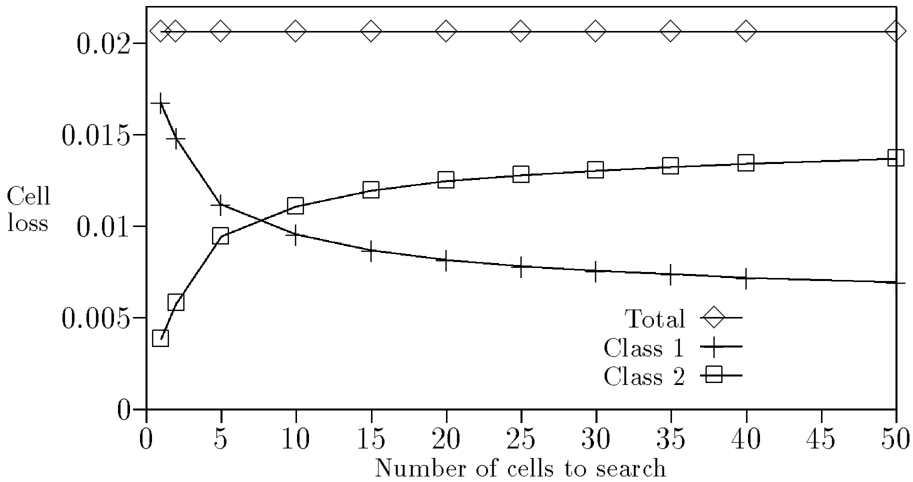

Figure 8.4: Push Out (mix3, two classes, fraction of total)

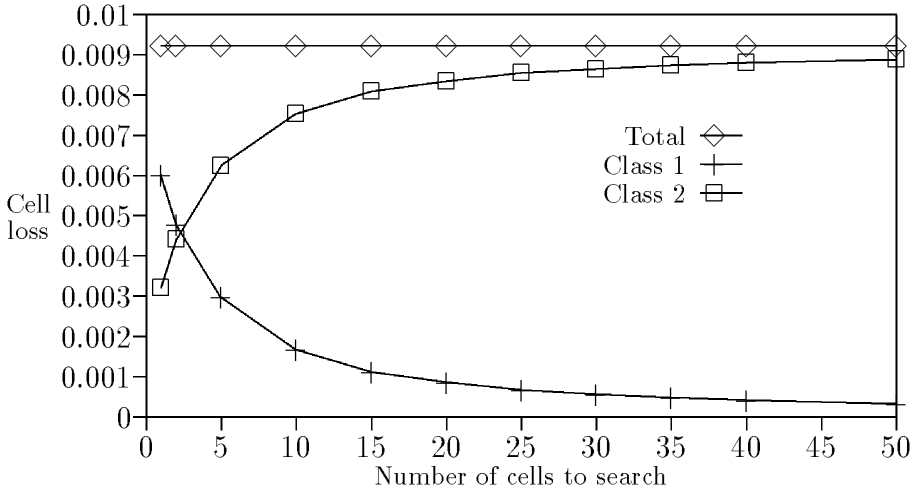

Figure 8.5: Push Out (mix4, two classes, fraction of total)

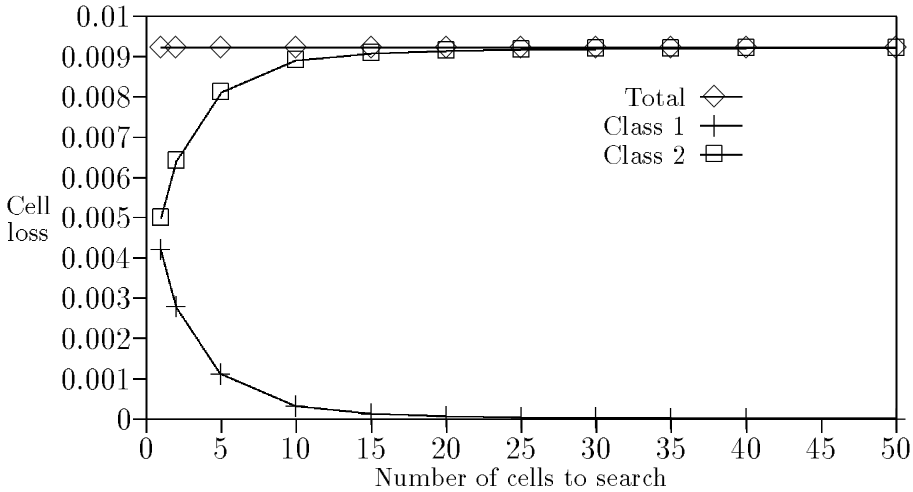

Figure 8.6: Push Out (mix5, two classes, fraction of total)

The results show that it is not necessary to search the whole buffer. If more than 15 cells are searched, the cell loss for Class 1 traffic does not decrease very much nor does the cell loss for Class 2 traffic increase much.

Figures 8.7–8.11 present the cell loss for each priority class in terms of the fraction of cells of each priority class that are lost. This gives an indication of whether or not each class is meeting loss requirements. Mix1 and mix3 show the highest total cell loss and the highest cell loss for Class 1 traffic. When 25 cells are checked, both mixes 1 and 3 still lose 0.01 Class 1 cells. These figures indicate that the protection of Class 1 traffic provided by the push out algorithm with two priority classes depends on the proportion of each type of traffic in the mix and the proportion of each priority class.

Figures 8.10 and 8.11 illustrate what happens when the traffic mix shifts from a high proportion of Class 1 cells to a high proportion of Class 2 cells. When only half the buffer is searched, Class 2 cells account for almost all of the cell loss when the proportion of Class 2 traffic in the network is high.

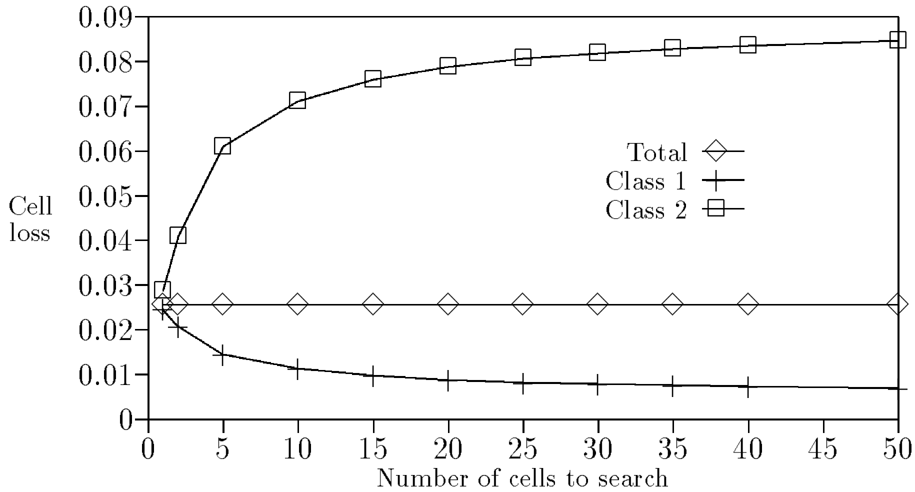

Figure 8.7: Push Out (mix1, two classes, fraction of class)

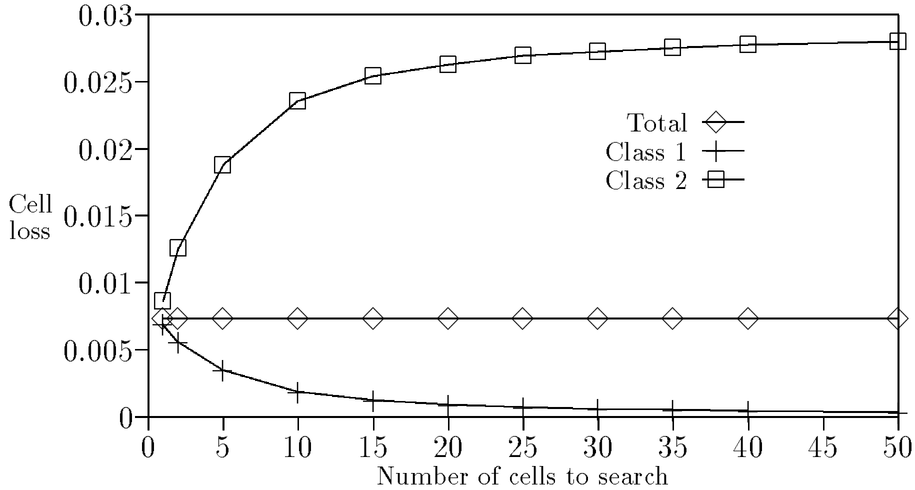

Figure 8.8: Push Out (mix2, two classes, fraction of class)

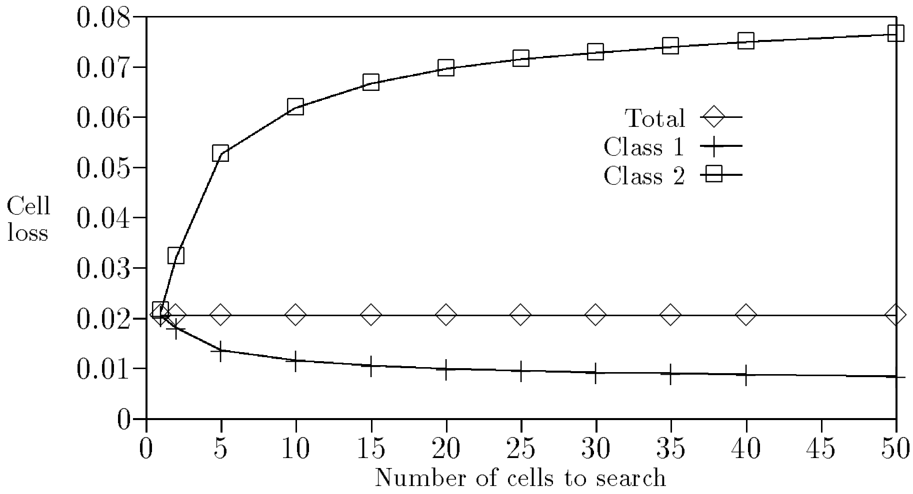

Figure 8.9: Push Out (mix3, two classes, fraction of class)

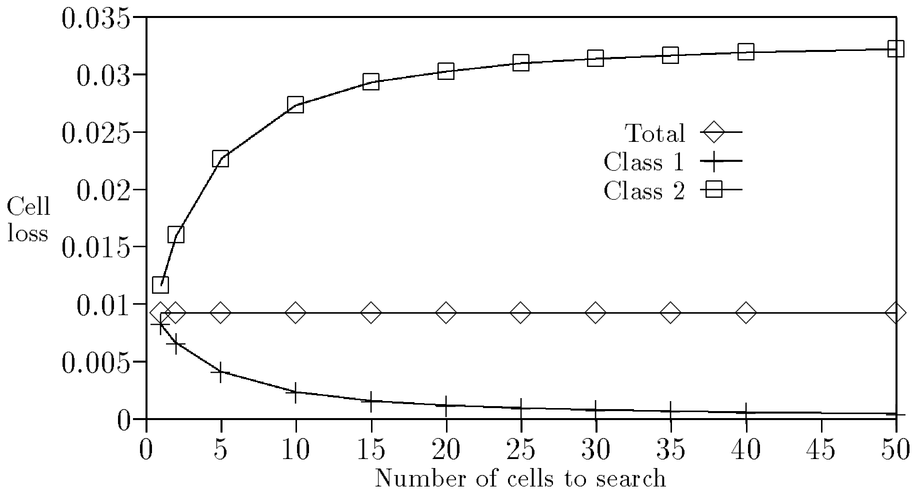

Figure 8.10: Push Out (mix4, two classes, fraction of class)

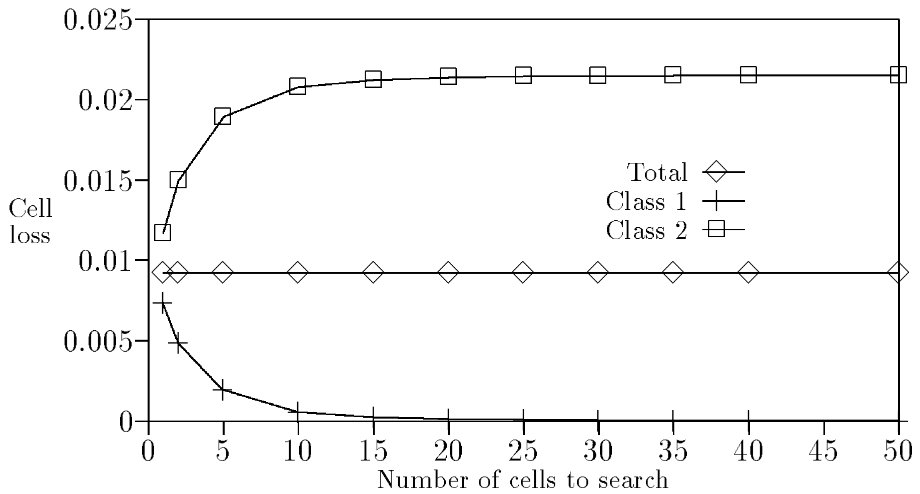

Figure 8.11: Push Out (mix5, two classes, fraction of class)

The queueing delay measured in slots for each traffic mix is displayed in Table 8.3. The delay shown for the push out algorithm was measured when the whole buffer was checked to find a lower priority cell to replace. When only one cell was checked, the queueing delay was statistically indistinguishable from the delay when no priority queueing algorithm was used. Except for the queueing delay experienced by Class 2 cells in mix1 and mix3, the queueing delay did not vary significantly with the change in number of cells to search. Even in mix1 and mix3 the Class 2 delay varied by only 2 slots.

The delay for the two classes is very similar for mix2, mix4, and mix5 when no priority algorithm is used. The difference between the delay in mix1 and mix3 is caused by the burstier nature of these traffic mixes. Because Class 1 cells make up a larger fraction of the total traffic, the buffer is likely to contain a larger number of Class 1 cells when it fills. This means that there are more Class 1 cells with larger delays than Class 2 cells, contributing to a larger average delay for Class 1 traffic.

Table 8.3: Queueing Delay for the Push Out Algorithm

	No Priority Queueing		Push Out (50 cells searched)	
	Class 1	Class 2	Class 1	Class 2
mix1	14.97 ± 0.67	12.62 ± 0.63	15.47 ± 0.69	10.27 ± 0.49
mix2	10.35 ± 0.92	9.38 ± 0.88	10.55 ± 0.95	8.61 ± 0.76
mix3	14.55 ± 0.50	11.62 ± 0.45	14.88 ± 0.52	9.33 ± 0.32
mix4	10.65 ± 0.56	9.97 ± 0.54	10.91 ± 0.58	9.10 ± 0.46
mix5	10.59 ± 0.56	10.09 ± 0.54	10.89 ± 0.59	9.62 ± 0.49

A similar phenomenon is observed when the push out algorithm is used; Class 2 traffic has a shorter average delay than Class 1. This is caused by the fact that when the buffer overloads, Class 2 cells will be dropped and will not be counted in the average delay.

8.2.2 Buffer Sharing Algorithm

A buffer sharing limit of B means that when the buffer length is less than B, cells from both priority classes will be added to the buffer. When the buffer length exceeds B only Class 1 cells will be added. Therefore when B is small, Class 2 cells may be dropped when there are few cells in the buffer. This results in a higher total cell loss and a much higher cell loss for Class 2 traffic. The total cell loss and the Class 2 cell loss decrease as the number of cells that both classes can occupy is increased.

Figures 8.12–8.16 present the cell loss for each priority class as a fraction of the total cells and show that Class 1 cell loss begins to increase only when the buffer sharing limit approaches the size of the buffer. This means that only a few cells may need to be reserved to provide acceptable performance for Class 1.

It is important to note that when the sharing limit is small the total cell loss is higher than when the sharing limit is bigger. This is caused by the fact that the buffer sharing algorithm is not work conserving. Therefore, when the sharing limit is

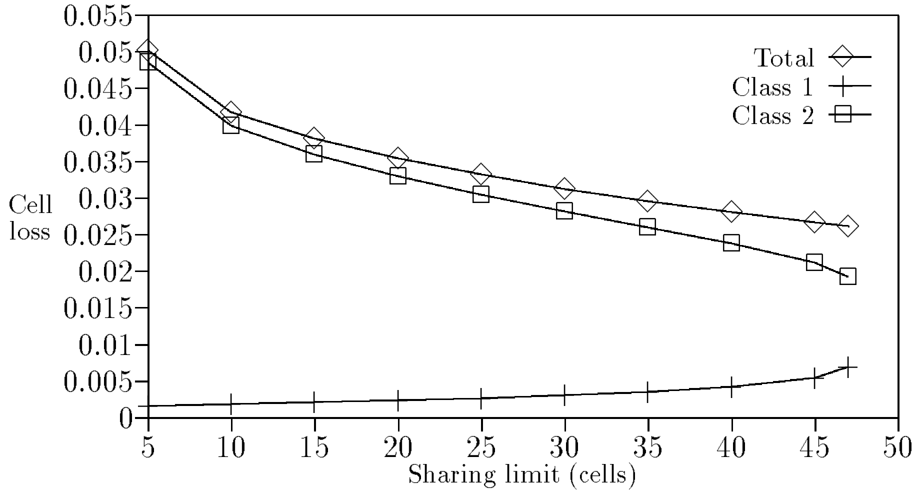

Figure 8.12: Buffer Sharing (mix1, two classes, fraction of total)

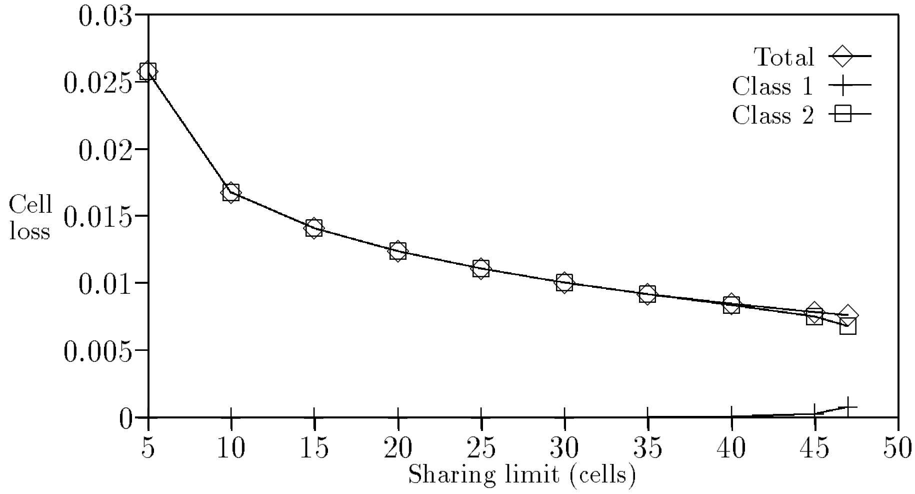

Figure 8.13: Buffer Sharing (mix2, two classes, fraction of total)

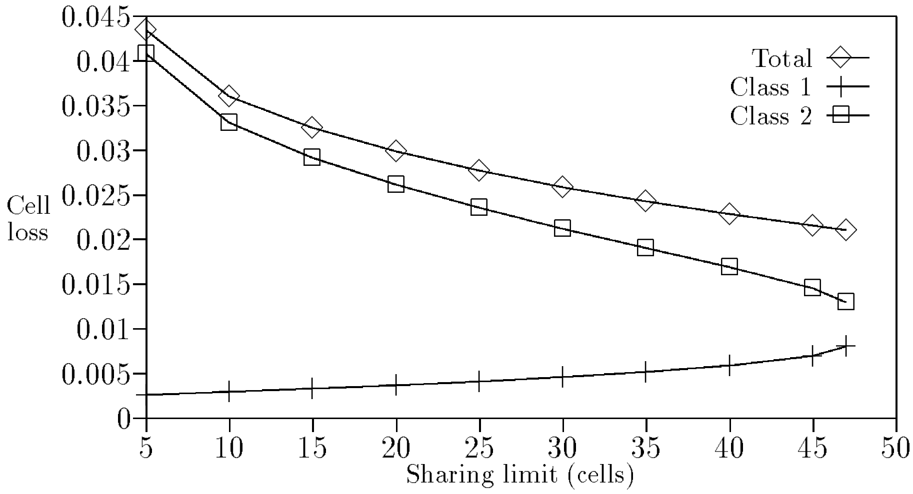

Figure 8.14: Buffer Sharing (mix3, two classes, fraction of total)

lower it is more likely that the buffer space which both classes are allowed to occupy will fill, causing Class 2 cells to be dropped.

Figures 8.13, 8.15, and 8.16 display a very low cell loss for Class 1 traffic. The confidence intervals show that Class 2 cell loss is indistinguishable from the total cell loss and Class 1 cell loss is indistinguishable from 0. This means that Class 2 traffic absorbs almost all the cell loss. Note also, that mix4 and mix5 show very similar results (see Figures 8.15 and 8.16). This indicates that the low cell loss experienced by Class 1 traffic is not the result of a higher (or lower) proportion of Class 1 traffic.

Class 1 cell loss increases slightly when the buffer sharing size approaches the size of the buffer. When only a few cells are reserved for Class 1 cells, the buffer will overflow more often resulting in a higher loss for Class 1 cells.

Figures 8.17–8.21 show how much the performance of Class 2 traffic degrades when cell loss for Class 1 traffic is achieved. The cell loss for Class 2 traffic is consistently higher than the total cell loss.

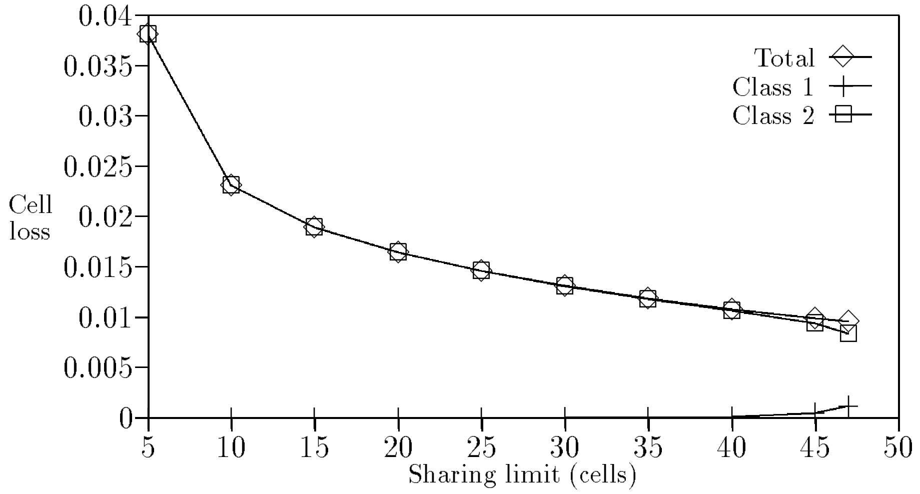

Figure 8.15: Buffer Sharing (mix4, two classes, fraction of total)

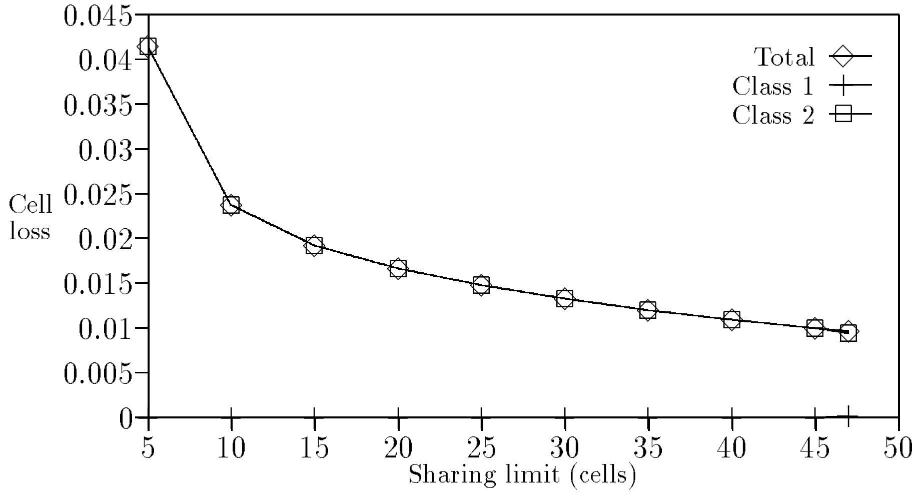

Figure 8.16: Buffer Sharing (mix5, two classes, fraction of total)

Note the sharp decrease in cell loss from a sharing limit of 5 to a sharing limit of 10 in Figures 8.18, 8.20, and 8.21. When Class 2 cells are allowed to occupy only one tenth of the buffer, a high percentage of Class 2 cells must be dropped.

The difference between Figure 8.20 and Figure 8.21 shows that the proportion of cells in each class does affect the performance. Class 2 cell loss as a fraction of the number of Class 2 cells generated is smaller in Figure 8.21 (mix5) where there is a higher proportion of Class 2 traffic. Although the total cell loss and the fraction of the total cell loss attributed to Class 2 is higher in mix5, cells in this class will have more opportunity to occupy the shared buffer space since Class 1 traffic makes up a smaller portion of the total traffic. At higher levels of buffer sharing, Class 1 cell loss is greater in Figure 8.20 (mix4). At a buffer sharing limit of 35 the cell loss for Class 1 in mix4 is 10^{-4}. At the same limit with mix5 traffic the Class 1 cell loss is still 0. Cell loss remains 0 in mix5 until a sharing limit of 45 is reached where the cell loss is 5×10^{-5}. Because there is more Class 1 traffic in mix4, there is a higher probability of dropping Class 1 cells when the buffer fills.

The queueing delay for the buffer sharing algorithm does not remain constant. When the sharing limit is 5, the Class 2 delay for all mixes is approximately 2 slots. The Class 2 delay for all mixes differs by at most 1.7 slots, and most often differs by less than 1 slot. Figure 8.22 gives the delay for Class 1 cells and includes the delay for Class 2 cells from mix4 as a comparison. Note how similar the delay for mixes 2, 4, and 5 is. Mix1 and mix3 are also similar and have the highest delay.

The delay for Class 2 cells increases as the buffer sharing limit increases because they can occupy more positions in the buffer. The delay for Class 1 cells increases as the sharing limit increases because fewer Class 2 cells are dropped, so the average size of the occupied space in the buffer increases.

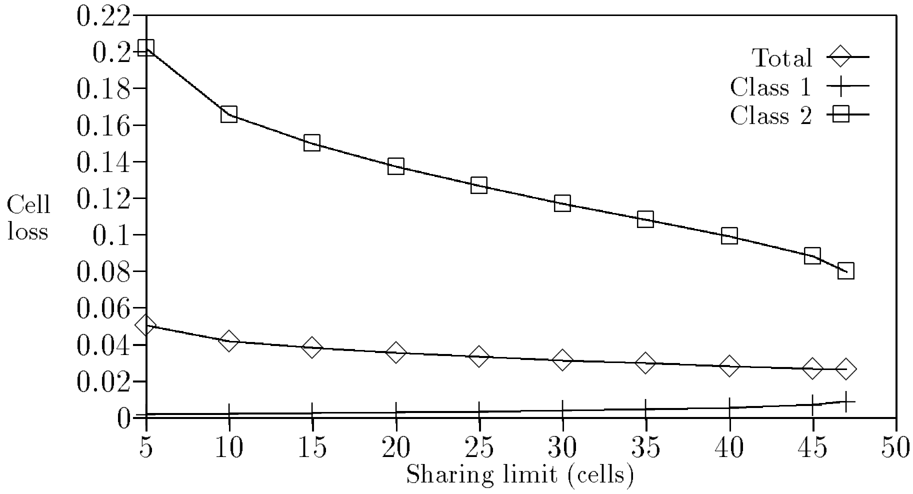

Figure 8.17: Buffer Sharing (mix1, two classes, fraction of class)

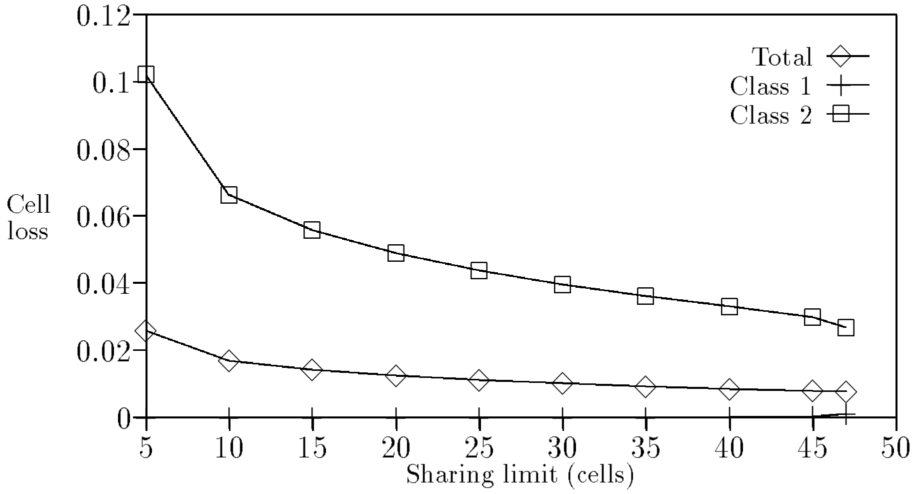

Figure 8.18: Buffer Sharing (mix2, two classes, fraction of class)

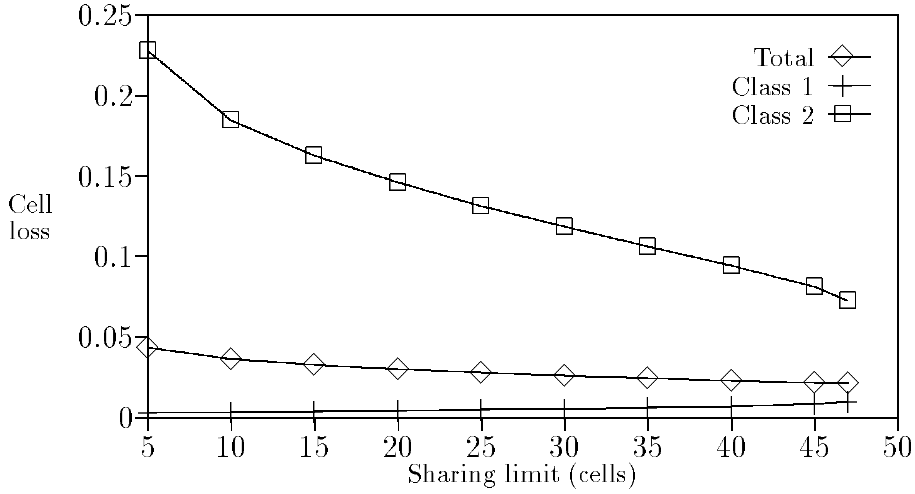

Figure 8.19: Buffer Sharing (mix3, two classes, fraction of class)

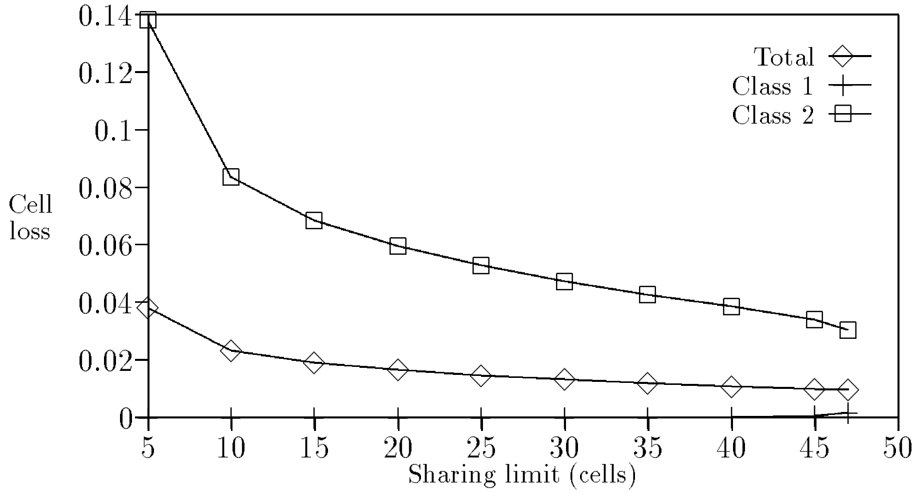

Figure 8.20: Buffer Sharing (mix4, two classes, fraction of class)

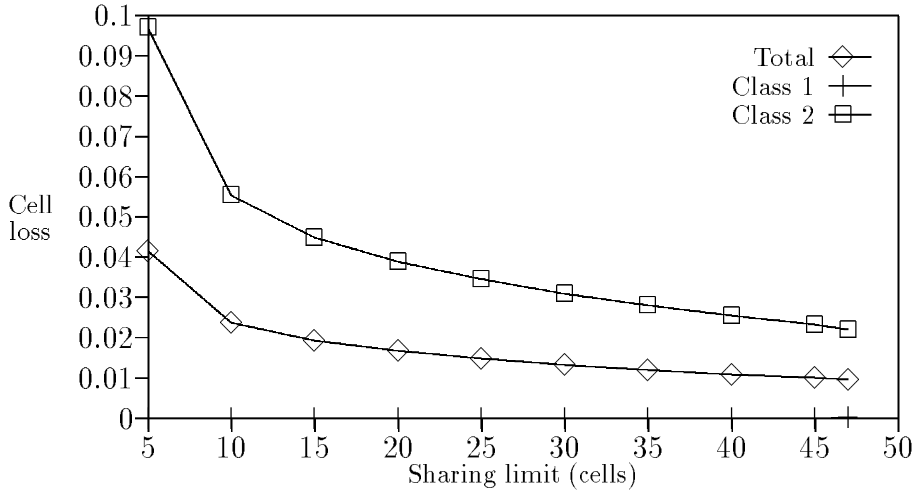

Figure 8.21: Buffer Sharing (mix5, two classes, fraction of class)

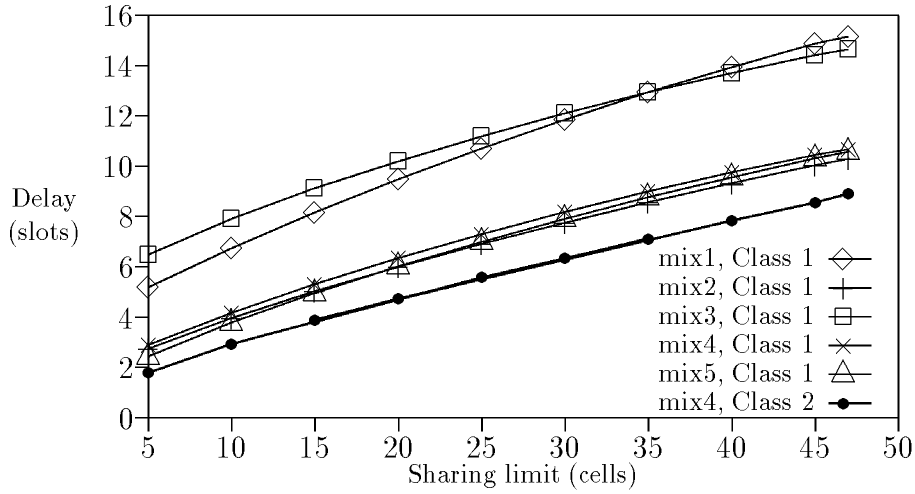

Figure 8.22: Queueing Delay for the Buffer Sharing Algorithm

8.2.3 Selective Packet Discarding

Selective packet discarding is the most complex of the three priority queueing algorithms studied in this thesis. Two sets of parameters govern its behaviour: the threshold parameters (B1 and B2) and the function that decides which packets to discard when the buffer is in the control state.

The control function used for this research is simply a pair of parameters (P1 and P2) that determine with what probability a cell from each priority class will be dropped. This function only operates during the control period as described in Section 6.2.

The first test of the algorithm involved fixing the probability parameters, (P1 = 0.001, P2 = 0.1) and varying the threshold parameters, B1 and B2. The values for B1 and B2 are found in Table 8.4. The threshold parameter sets are labeled b1 to b10. The results show that threshold parameters have little influence on the cell loss rate.

Table 8.4: Threshold Parameters for Selective Packet Discarding

	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
B1	10	10	10	20	20	30	30	30	35	35
B2	20	30	40	30	45	35	40	45	40	45

The total cell loss varied by less than 0.005 in all traffic mixes. In all cases, parameter sets b8 and b9 showed the lowest total cell loss and the lowest cell loss for Class 2 cells. Cell loss for Class 1 traffic did not vary significantly in any of the traffic mixes. These results suggest that the threshold parameters do not strongly influence the cell loss and that placing the thresholds closer to the end of the buffer gives the lowest cell loss. Tables 8.5 and 8.6 provide the results for parameter set b8.

The queueing delay experienced by each class was statistically indistinguishable from the delay for the traffic when no priority algorithm was used.

Table 8.5: Cell Loss for Selective Discarding (b8, P1 = 0.001, P2 = 0.1)

	Fraction of Total		
	Total	Class 1	Class 2
mix1	0.0288 ± 0.0022	0.0183 ± 0.0016	0.0083 ± 0.0007
mix2	0.0079 ± 0.0015	0.0046 ± 0.0009	0.0033 ± 0.0006
mix3	0.0213 ± 0.0018	0.0163 ± 0.0014	0.0051 ± 0.0004
mix4	0.0099 ± 0.0013	0.0056 ± 0.0008	0.0043 ± 0.0005
mix5	0.0101 ± 0.0013	0.0040 ± 0.0006	0.0062 ± 0.0007

Table 8.6: Cell Loss for Selective Discarding (b8, P1 = 0.001, P2 = 0.1)

	Fraction of Each Class	
	Class 1	Class 2
mix1	0.0240 ± 0.0021	0.0346 ± 0.0029
mix2	0.0061 ± 0.0012	0.0132 ± 0.0023
mix3	0.0199 ± 0.0017	0.0283 ± 0.0023
mix4	0.0077 ± 0.0011	0.0157 ± 0.0019
mix5	0.0069 ± 0.0010	0.0144 ± 0.0017

This set of probabilities for dropping cells used in combination with a wide range of threshold parameters was unable to significantly reduce the cell loss of Class 1 traffic with only two priority classes. In each case, the difference between Class 1 cell loss (Table 8.6) and the total cell loss (Table 8.5) was less than 0.005. At these parameter settings mix4 and mix5 showed very similar results; the total cell loss was statistically identical and the only difference was that the cell loss in each class was slightly higher for mix4. This suggests that the performance of the algorithm is not very dependent on the proportion of each priority class. This is desirable in a setting where, if Class 1 cells make up most of the traffic, the performance of lower priority classes should not completely deteriorate. If the desired performance of the system is that Class 1 traffic should meet quality of service requirements regardless of the cost to other classes of traffic then the probability parameters should be set so that low priority cells have a much higher probability of being dropped.

The second part of the experiment using the selective packet discarding algorithm involved fixing the threshold parameters and varying the probability parameters (see Table 8.7). The threshold parameters chosen were those from b8 (B1 = 30, B2 = 45) in the previous experiment.

Table 8.7: Probability Parameters for Selective Packet Discarding (b8)

	p1	p2	p3	p4	p5	p6
P1	0.001	0.01	0.1	0.001	0.001	0.0001
P2	0.5	0.5	0.5	0.2	0.1	0.1

	p7	p8	p9	p10	p11	p12
P1	0.00001	0.0001	0.0001	0.00001	0.0001	0.1
P2	0.1	0.05	0.01	0.01	0.001	0.2

Figures 8.23–8.27 show that the probability parameters have a greater influence on the performance of the selective discarding algorithm than the threshold parameters. If the probabilities are too high, more packets than necessary are dropped and the cell loss for the class with high probability of dropping packets is high. If the probabilities are too low then the buffer fills and packets of both priority classes are discarded.

Figures 8.23–8.27 also show that the parameter P2 has the strongest influence on the cell loss of both classes. For example, parameter sets p5, p6, and p7 all show similar results despite the fact that the parameter P1 differs by three orders of magnitude. In each of these parameters sets P2 is the same. This is true for all traffic mixes which indicates that this effect is independent of the proportion of traffic of each class.

Unlike the push out and buffer sharing algorithms, Class 1 traffic makes up a significantly larger fraction of the total cell loss in many of the parameter settings as

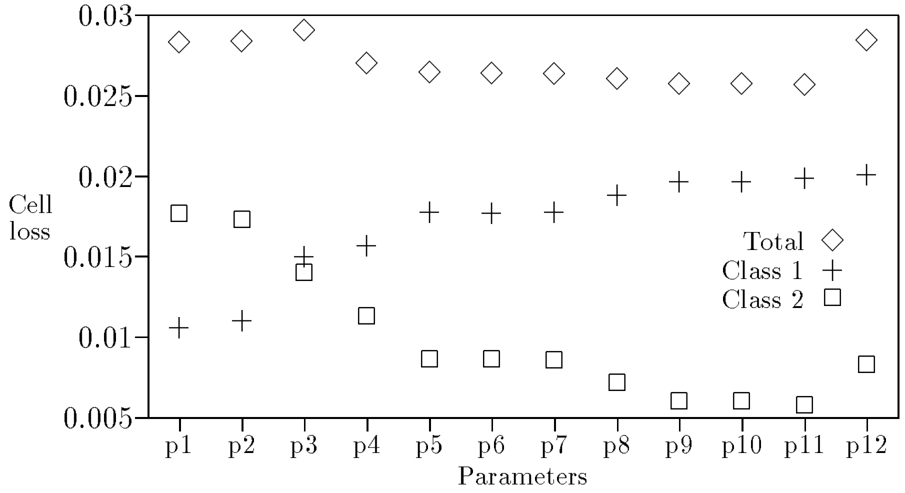

Figure 8.23: Selective Discarding (mix1, two classes, fraction of total)

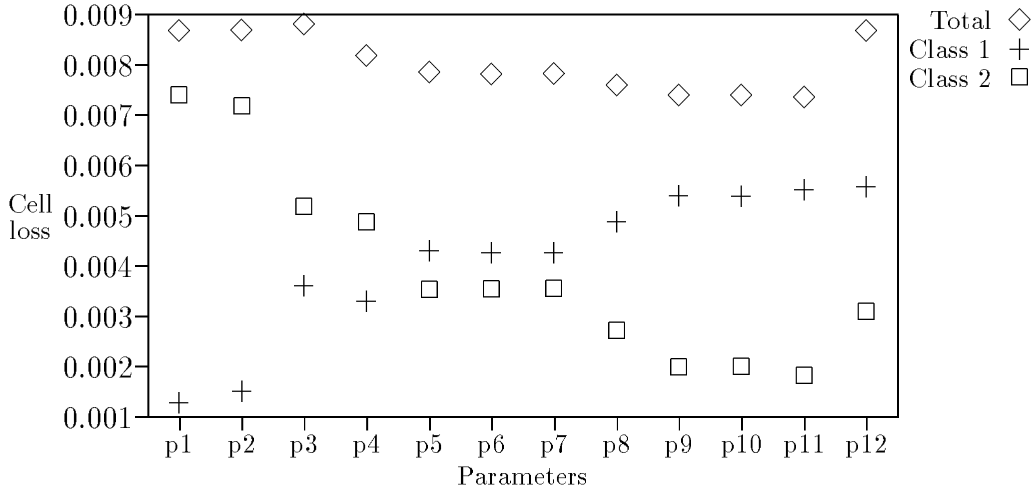

Figure 8.24: Selective Discarding (mix2, two classes, fraction of total)

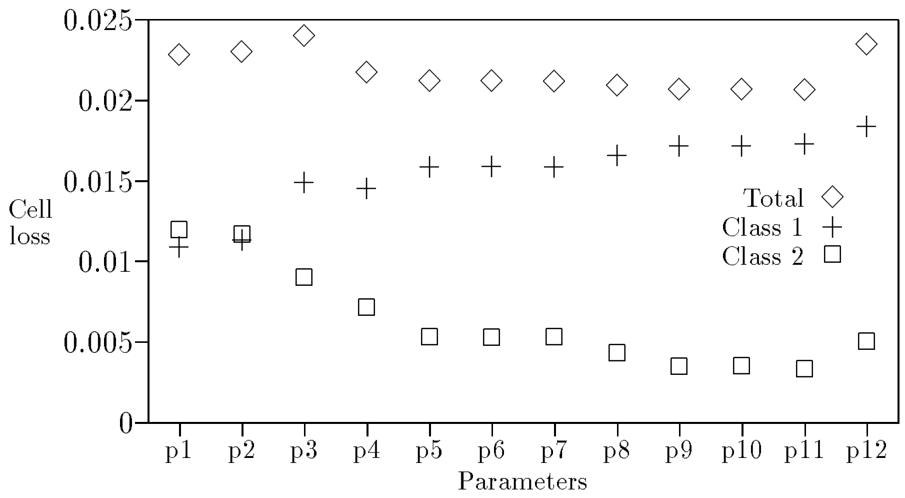

Figure 8.25: Selective Discarding (mix3, two classes, fraction of total)

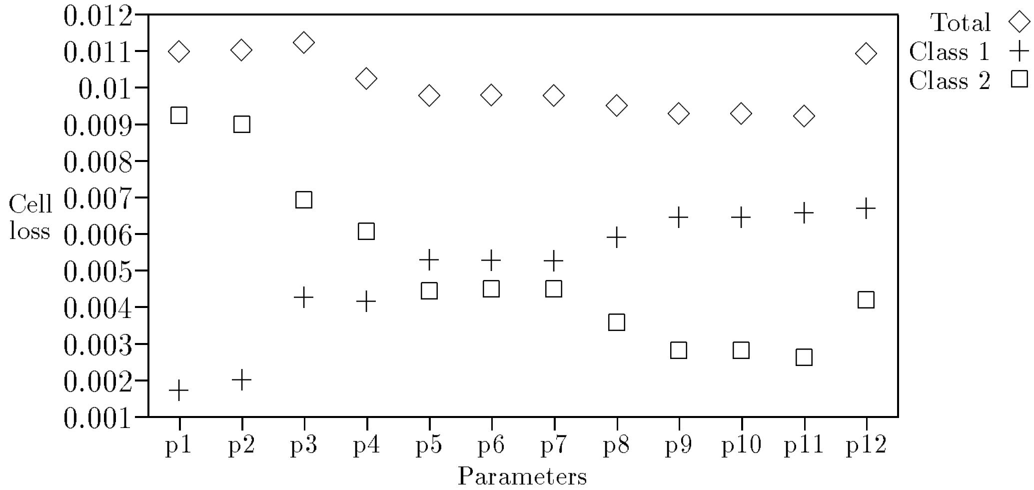

Figure 8.26: Selective Discarding (mix4, two classes, fraction of total)

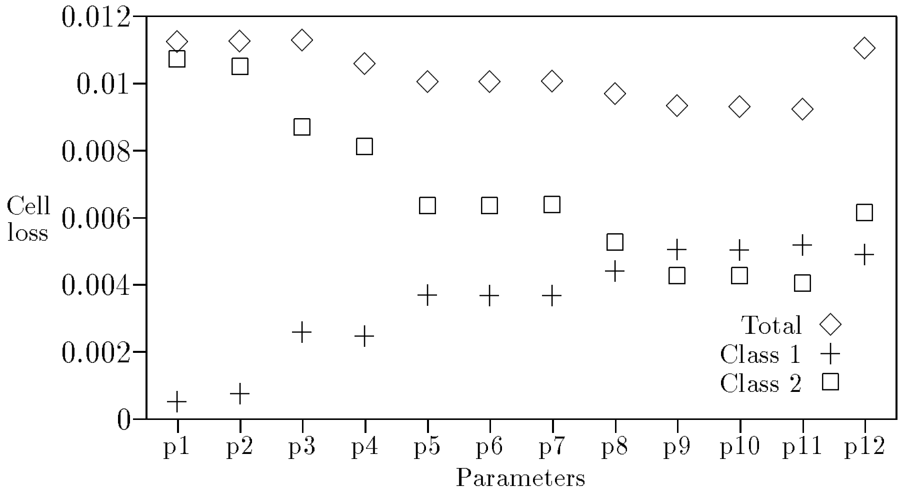

Figure 8.27: Selective Discarding (mix5, two classes, fraction of total)

shown in Figures 8.23–8.27. This occurs because the process that decides when to drop a packet is not influenced by the division of the traffic among the classes. Note that this effect is not necessarily bad, as long as the fraction of Class 1 cells dropped is lower than the fraction of Class 2 cells dropped. Figures 8.28–8.32 show that this holds true for most parameter settings.

Even when the probability parameters vary widely the queueing delay does not change much for either class. In each traffic mix, Class 1 delay for each parameter set was statistically identical and was also the same as the Class 1 delay when no priority algorithm was used (see Table 8.3). Class 2 delay ranged from the values for no priority algorithm up to two slot times less than those values.

8.2.4 Summary

The results of the three priority algorithms with two priority classes show that protection of Class 1 traffic from excessive cell loss can be achieved to some degree. The

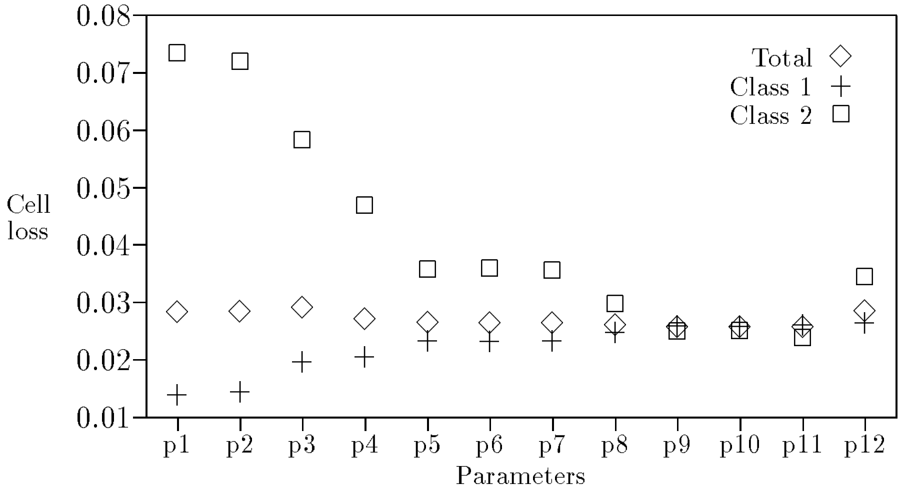

Figure 8.28: Selective Discarding (mix1, two classes, fraction of class)

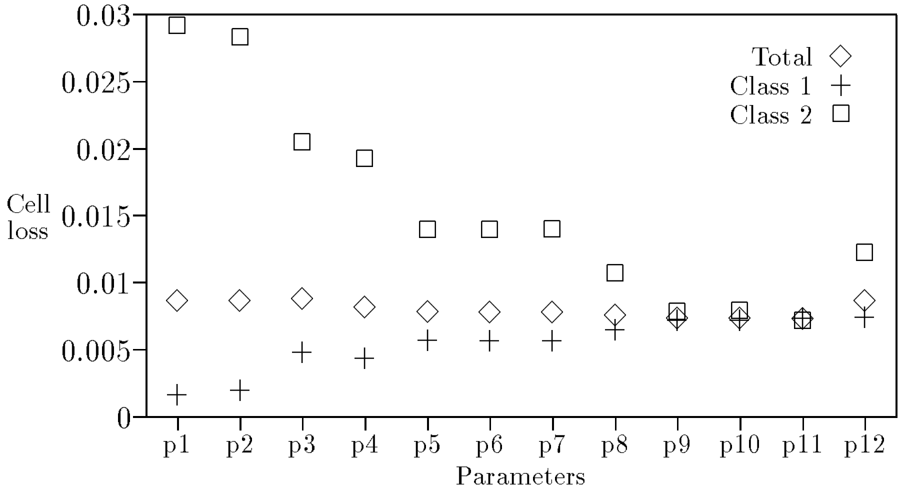

Figure 8.29: Selective Discarding (mix2, two classes, fraction of class)

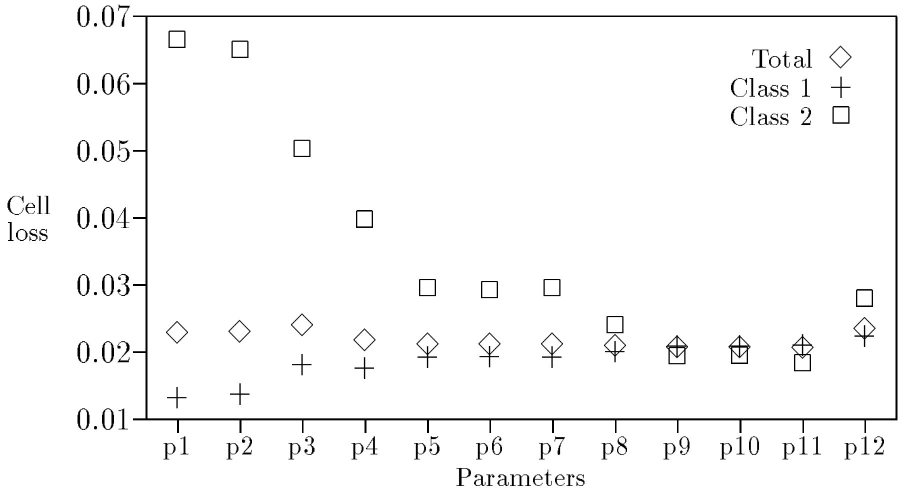

Figure 8.30: Selective Discarding (mix3, two classes, fraction of class)

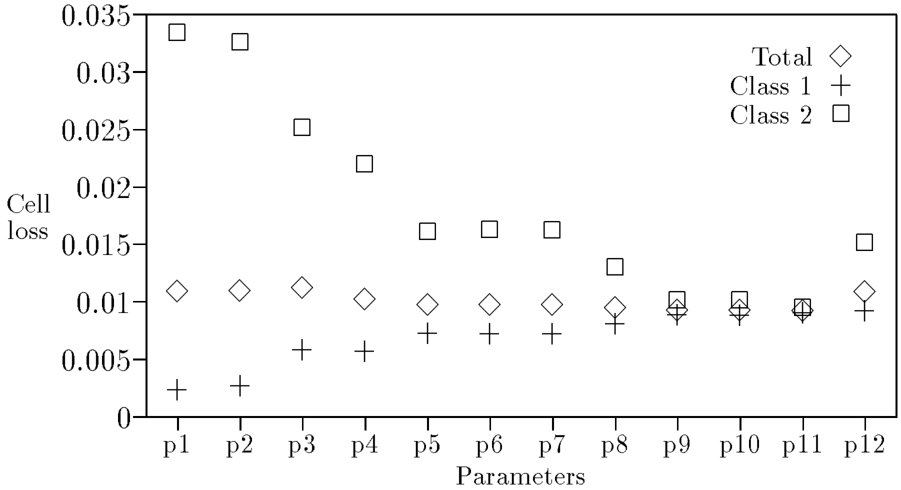

Figure 8.31: Selective Discarding (mix4, two classes, fraction of class)

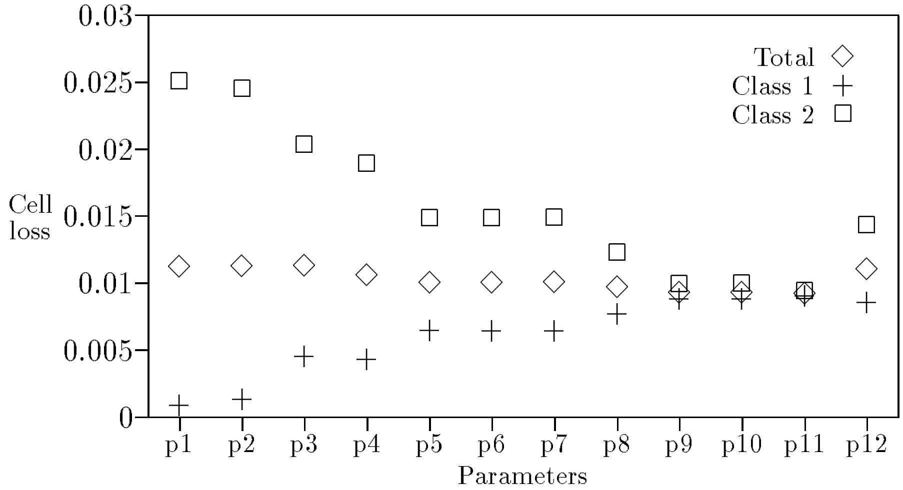

Figure 8.32: Selective Discarding (mix5, two classes, fraction of class)

performance of Class 2 traffic necessarily degrades, absorbing cell loss from Class 1. The goal of priority queueing algorithms is to ensure that the higher priority traffic does not suffer too much cell loss while the performance of the lower priority class degrades as little as possible.

The method of dividing traffic into two priority classes that was used here allows applications to identify which of their packets they can afford to lose. The function of the priority queueing algorithm, then, is to ensure that the less important packets get dropped first. The buffer sharing algorithm most closely adheres to this philosophy, dropping lower priority cells before the buffer is even full. The push out algorithm applies this philosophy more conservatively; it allows the buffer to fill before dropping lower priority packets. The selective discarding algorithm follows a slightly different philosophy. It assumes that all types of traffic can tolerate some cell loss so it tries to control the cell loss for each class by dropping cells before the buffer fills.

If the goal of dividing traffic into two classes is solely to allow applications to identify which cells to drop first, then it is expected that the proportion of higher priority traffic will be large with respect to lower priority traffic. Few applications will be able to give low priority to the majority of their packets. In the traffic mixes where the proportion of Class 1 traffic was high the push out algorithm and the selective packet discarding algorithms were limited in how much they could protect Class 1 against loss. Buffer sharing was able to provide the most protection for Class 1 traffic, but these gains were costly; Class 2 loss and the total cell loss was much higher than for the work conserving algorithm.

The queueing delay results show that these algorithms have little effect, either positive or negative, on the queueing delays for any of the classes. This means that it is safe to use loss priority algorithms on delay sensitive traffic. It also means that efforts to protect delay sensitive traffic from excessive delay will not be hindered by the presence of a loss priority algorithm.

The next section presents the results of experiments with the same three algorithms using four priority classes. By dividing the total traffic into more traffic classes, it may be possible to almost completely shield Class 1 traffic from cell loss without causing the performance of the lower classes to degrade too much.

8.3 Four Priority Classes

Section 8.2 has shown that the extent to which Class 1 cell loss may be reduced is limited when two priority classes are used. This section examines the same priority queueing algorithms for four priority classes. The goal in dividing the traffic into more classes is not only to achieve lower cell loss for Class 1 but also to ensure that the cell loss for the other classes does not become too high.

Many of the results from the experiments with four priority classes are very similar to the results with two priority classes. Therefore this section will comment on these similarities and discuss in detail only the cases where the performance with four priority classes changed. The complete set of figures for the experiments with four priority classes can be found in Appendix A.

8.3.1 Push Out

When the traffic is divided into four priority classes, the cell loss for Class 1 traffic is greatly reduced, becoming 0 when only 10 cells are checked. This is a result of the fact that a Class 1 cell is more likely to find a Class 2, 3, or 4 cell to replace. When fewer buffers are searched, the loss experienced by the top three priority classes is spread over a wider range but as the number of buffers searched approaches the size of the buffer, Class 4 accounts for almost all of the cell loss. Figures 8.33 and 8.34 serve to illustrate the behaviour of the push out algorithm when four priority classes are used.

Class 4 traffic exhibits the same cell loss here as Class 2 traffic did in the two priority class example since the traffic that makes up theses two classes is the same (see Figure 8.5). This means that the Class 1 traffic in the two priority class example is divided into three priority classes here.

As in the study of two priority classes, at least 5 buffer slots must be searched before the operation of the algorithm overcomes the distribution of cell loss caused by the proportion of traffic in each class.

Note that in mix5, a smaller percentage of Class 2 cells are lost than in mix4. This is again a result of the different distribution of cells among priority classes. Because more of the cells belong to Class 2, more of the successfully transmitted cells will belong to Class 2.

Figures 8.35–8.36 show that unlike the two priority class example, it may be more worthwhile to search more of the buffer for cells to replace. This is particularly true if the performance of Class 3 traffic is a real concern. In Section 8.2.1 it was noted

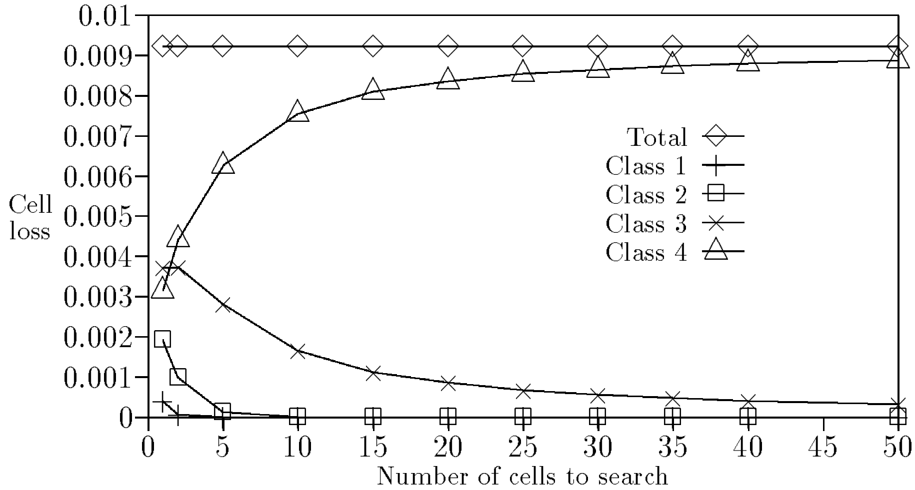

Figure 8.33: Push Out (mix4, four classes, fraction of total)

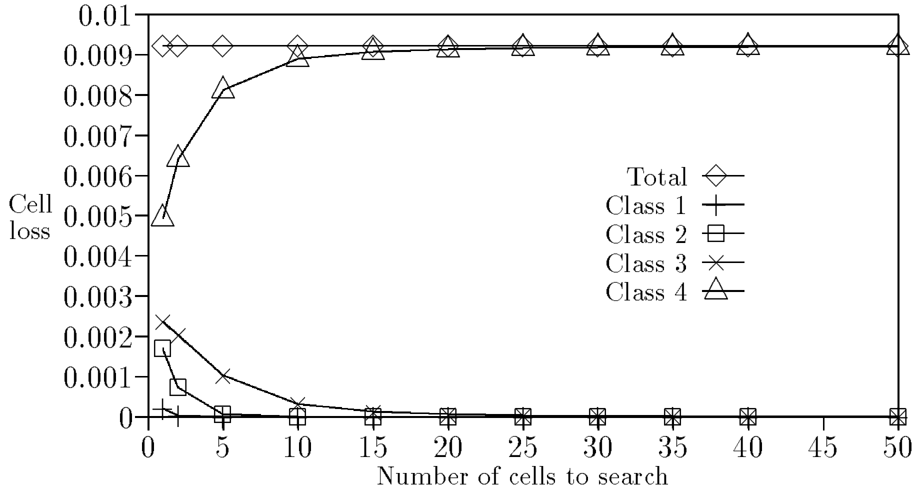

Figure 8.34: Push Out (mix5, four classes, fraction of total)

that in mixes 1 and 3, Class 1 has a cell loss of approximately 0.01. When the traffic is divided into four priority classes, Class 1 loss quickly drops to 0.

Alternatively, if the goal is to keep the percentage of cell loss in each class as low as possible, it would be better to keep the number of cells to search small, in the region before the cell loss curves begin to flatten.

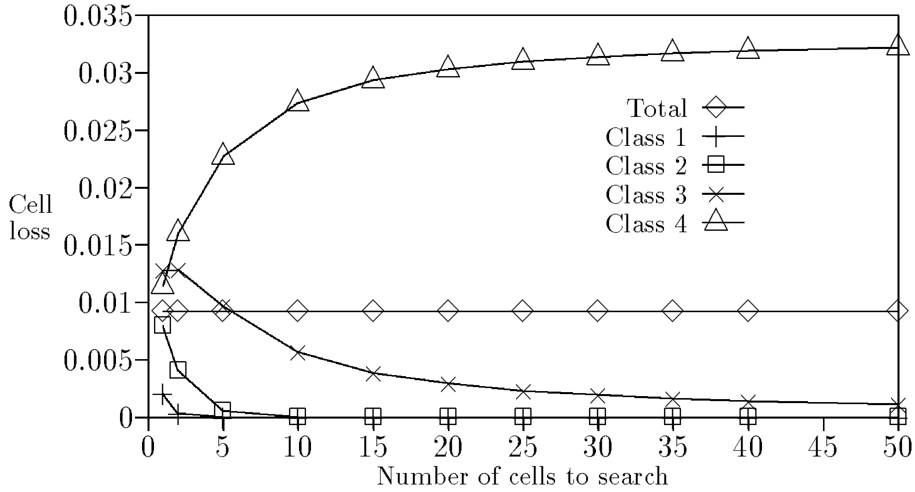

Figure 8.35: Push Out (mix4, four classes, fraction of class)

The queueing delay for each class does not vary significantly when the parameters change. In fact, the queueing delay is not significantly different than when no priority queueing algorithm is in effect. The queueing delays when no priority algorithm is used are displayed in Table 8.8.

8.3.2 Buffer Sharing

When a buffer sharing algorithm is used with four priority classes, more parameters are necessary to specify each buffer sharing limit. The parameters used in this experiment are given in Table 8.9.

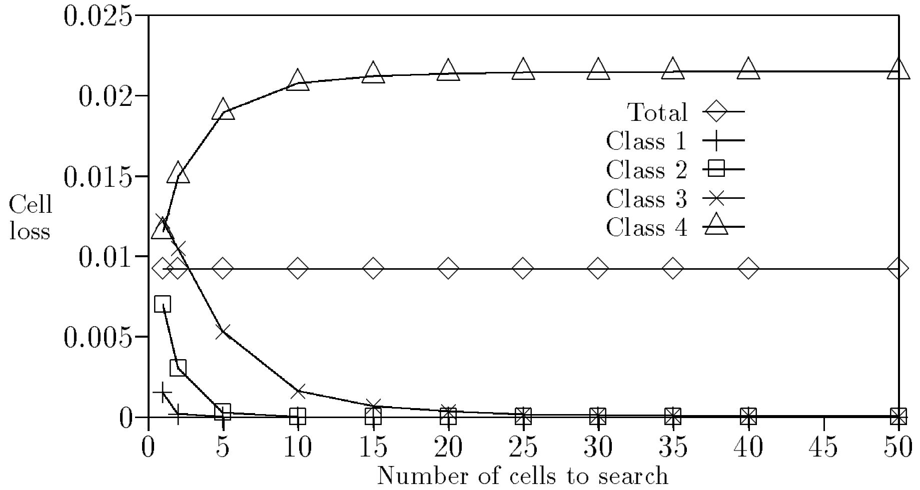

Figure 8.36: Push Out (mix5, four classes, fraction of class)

Perhaps the most significant feature of the results for four priority classes is the similarity to the results for two priority classes. The cell loss for Classes 1–3 is statistically indistinguishable. Class 4 absorbs almost all of the cell loss.

Figure 8.37 shows that parameter B1 most strongly influences the total cell loss and the cell loss for Class 4. Again, the only difference in performance between mix4 and mix5 is that the Class 4 cell loss and the total cell loss is higher in mix4.

Table 8.8: Queueing Delay for No Priority Queueing (slots)

	Class 1	Class 2	Class 3	Class 4
mix1	12.00 ± 0.62	17.57 ± 0.77	13.26 ± 0.65	12.62 ± 0.63
mix2	9.42 ± 0.90	10.90 ± 1.01	9.42 ± 0.89	9.38 ± 0.88
mix3	11.39 ± 0.45	15.85 ± 0.55	11.84 ± 0.45	11.62 ± 0.45
mix4	9.00 ± 0.50	10.87 ± 0.62	10.38 ± 0.55	9.97 ± 0.54
mix5	8.96 ± 0.50	10.87 ± 0.62	10.28 ± 0.55	10.09 ± 0.54

Table 8.9: Parameters for Buffer Sharing (four priority classes)

	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
B1	5	10	10	10	13	20	25	25	30	35
B2	10	15	20	25	25	30	35	30	35	40
B3	15	20	30	45	38	40	41	35	40	45

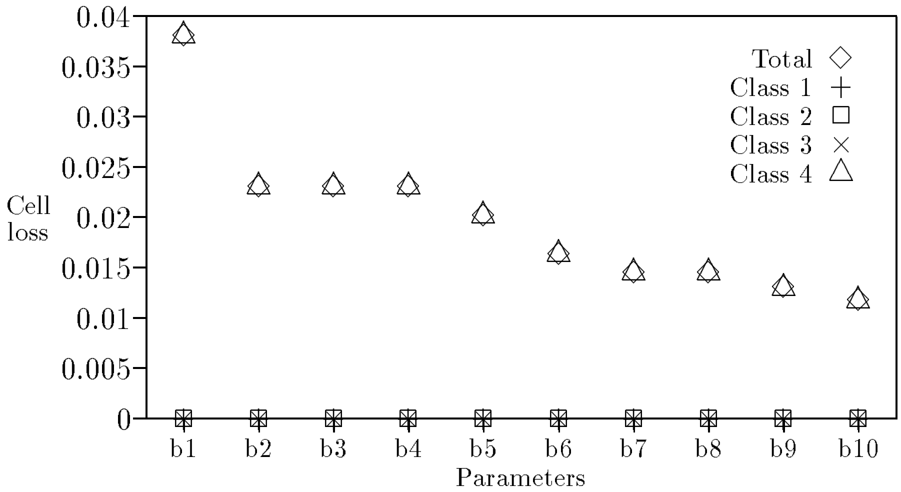

Figure 8.37: Buffer Sharing (mix4, four classes, fraction of total)

The queueing delays for each priority class are comparable to the queueing delays for two priority classes. Here, Class 4 delay is very similar (± 1 slot) to the Class 2 delay in the two priority case. Delay for Classes 1, 2, and 3 is very similar to the Class 1 delay when two priority classes are considered. Figure 8.38 presents the delay for each class in mix1. The other traffic mixes showed similar results; the delay for each class in mixes 2, 4, and 5 was smaller in magnitude but the shape of the curves was the same.

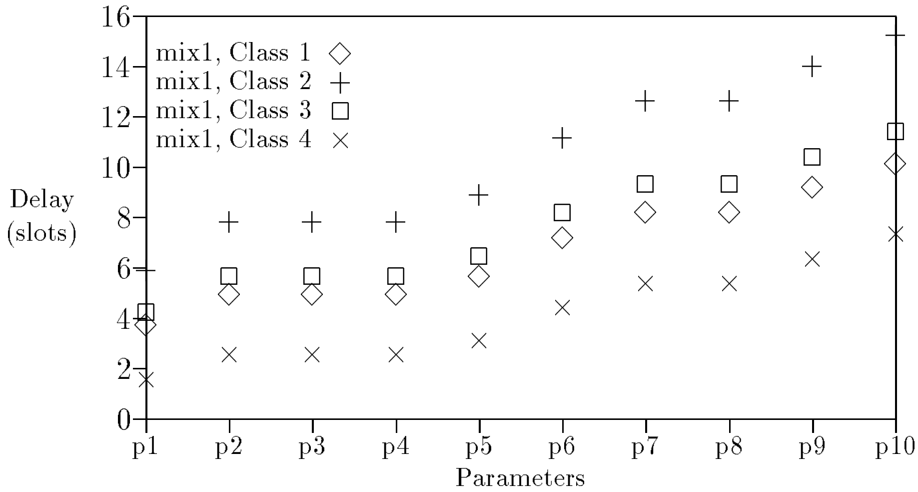

Figure 8.38: Queueing Delay for the Buffer Sharing Algorithm (mix1)

8.3.3 Selective Packet Discarding

The selective packet discarding algorithm requires more parameters when four priority classes are considered. These parameters are given in Tables 8.10.

Table 8.10: Parameters for Selective Packet Discarding

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
P1	0.1	0.001	0.01	0.01	0.001	0.0001	0.00001	0.0001	0.0001	0.00001
P2	0.2	0.01	0.1	0.05	0.05	0.01	0.001	0.005	0.001	0.0001
P3	0.5	0.2	0.2	0.1	0.1	0.1	0.1	0.01	0.01	0.001
P4	0.7	0.5	0.3	0.2	0.2	0.2	0.2	0.1	0.1	0.1

Here, the portion of total cell loss attributed to each class is not strictly ordered by class. In mixes 1 and 2, Class 3 cell loss is a smaller portion of the total cell loss than Class 2. In mix3, Class 2 traffic forms a larger portion of the total cell loss than Class 3 or Class 4. As explained in Section 8.2.3, it is more important that the fraction of cells lost in each class be ordered according to class. In other words, the fraction of cells lost by Class 1 traffic should be the smallest and the fraction of cells lost by Class 2 traffic should be the largest.

Four priority classes make it possible to shield Class 1 from cell loss more effectively than two priority classes could. The cell loss is absorbed by the three lower classes in varying amounts. Controlling the cell loss of four classes makes it possible to subdivide the traffic according to a wider range of quality of service requirements. This means that a class of traffic (A) that has less stringent loss requirements may absorb some of the loss from a class (B) with a tight loss requirement. With fewer priority classes, the traffic type A would be lumped together with traffic type B forcing other traffic types to absorb cell loss that traffic type A could have otherwise absorbed.

Figure 8.39 serves as an example to show how sensitive the cell loss results are to changes in parameter settings. For example, note the difference between the loss for p1 and p2. A larger spread in the probabilities in p2 allows slightly better performance for Classes 1, 2, and 3 but at the cost of a large increase in the loss for Class 4. Note that decreasing the probability that a Class 1 cell will be lost without changing other probabilities cannot improve the Class 1 cell loss. The results for p8, p9, and p10 show that when the probabilities for the higher classes become too small, the cell loss for these classes actually increases. This occurs because the buffer is more likely to overflow since fewer packets are being dropped before it fills. All packets arriving at a full buffer are dropped regardless of their class.

Except for the queueing delay for parameter set p1, Class 1 and Class 2 delay was almost identical to the delay when no priority class is considered (see Table 8.8). The delay for p1 was approximately 1 slot less than the others.

The delay for the other two classes was very similar for all parameter sets. In each traffic mix, the delay ranged over 2 slots where the upper limit was the value for the no priority case. This decrease in delay is accompanied by an increase in cell loss.

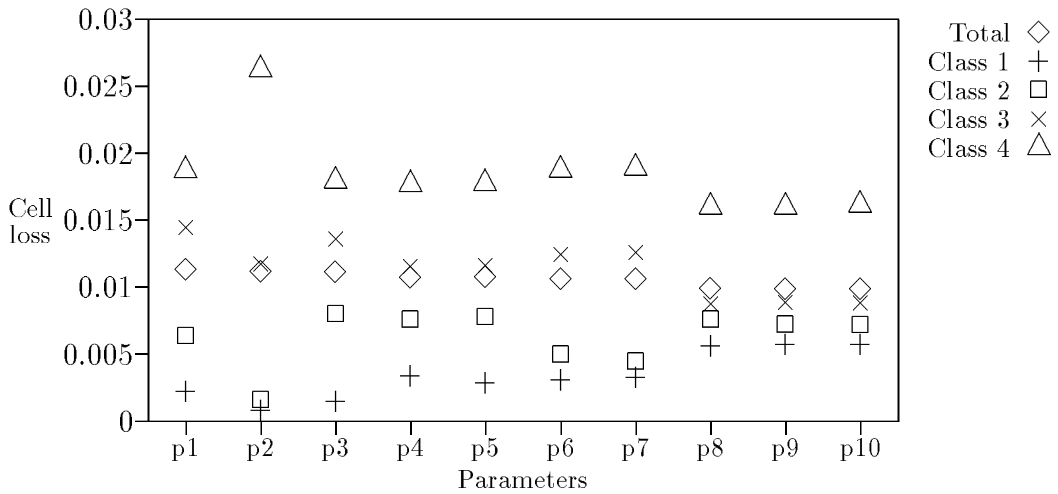

Figure 8.39: Selective Discarding (mix4, four classes, fraction of class)

8.3.4 Summary

The results of the three priority queueing algorithms for four priority classes were quite similar to the results for two priority classes.

The philosophy behind using four priority classes fits well with the concept that applications will have a wide range of service requirements. By providing four priority classes, applications that can tolerate some traffic loss can absorb some of the loss from loss sensitive traffic without suffering more degradation than their quality of service requirements will allow. The difficulty in implementing a policy based on this philosophy is in determining into which priority class applications should fall. All of the cells from a connection may be at a lower priority or the cells from a connection may be divided among two or more priority classes.

It was shown that the buffer sharing algorithm gained no benefits from the extra priority classes. Adding more priority classes to the push out algorithm made it possible to protect Class 1 traffic from loss more thoroughly than with only two priority classes without increasing Class 2 cell loss any more than when two priority classes are used. The selective discarding algorithm results are varied. They show that the best performance for all classes is achieved when the probability parameters are fairly high.

8.4 Consecutive Packet Loss

The number of packets lost is not the only aspect of quality of service degradation; the distribution of the dropped cells throughout the traffic stream is also important.

Some traffic types, such as video and voice, use error correction methods to reconstruct the information from lost packets. The error correction algorithms are effective only when each chunk of lost information is not too large. If equal numbers of cells are lost, the performance of these error correction algorithms is better if the lost cells are spread throughout the traffic stream than if they are clustered together.

Other applications, such as data file transfer, upon losing a cell, must retransmit a larger piece of information. This is often the result of higher level protocols which operate on larger pieces of information that are divided into cells by lower level protocols. In this case, consecutive cell loss is more desirable since may it keep the total number of retransmitted cells smaller.

Note that the notion of consecutive cell loss applies only at the call level of the network. In other words, consecutive cell loss at the user level means that consecutive cells in a single call are lost. Since the traffic on a network will be composed of many overlapping calls, monitoring consecutive cell loss at the switch layer may not be feasible.

The distribution of dropped cells throughout the traffic stream is affected by each of the priority queueing algorithms. As an example, Figures 8.40 and 8.41 present measurements of consecutive cell loss. The other traffic mixes show similar results with varying degrees of magnitude. Much longer simulations would be needed to make quantitative comparisons between the consecutive cell loss for different priority algorithms. This discussion focuses on the shape of the curves and the relative distribution of consecutive cell loss among the priority classes.

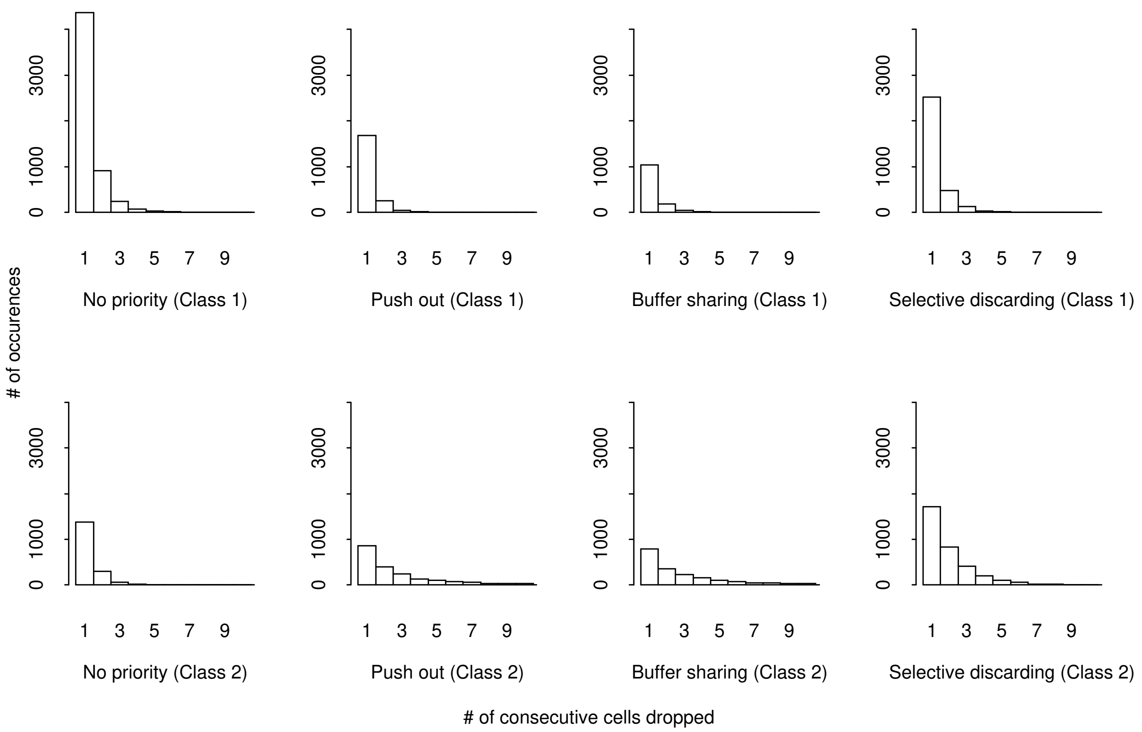

Figure 8.40: Consecutive Cell Loss (mix1, two classes)

Cells were identified as consecutive if they came from the same input port and were of the same priority class. Since the same traffic mix and load level were used at each input port, the results are very similar across the input ports. Consequently, only the results for one input port are displayed. A single parameter set was chosen for each algorithm: a search parameter of 25 for the push out algorithm, parameter set p1 for the selective discarding algorithm, a sharing limit of 40 for the two priority class case of buffer sharing, and parameter set b9 for the four priority class case of buffer sharing. These parameter sets were chosen because they fall into the region where higher priority cells are best protected from loss.

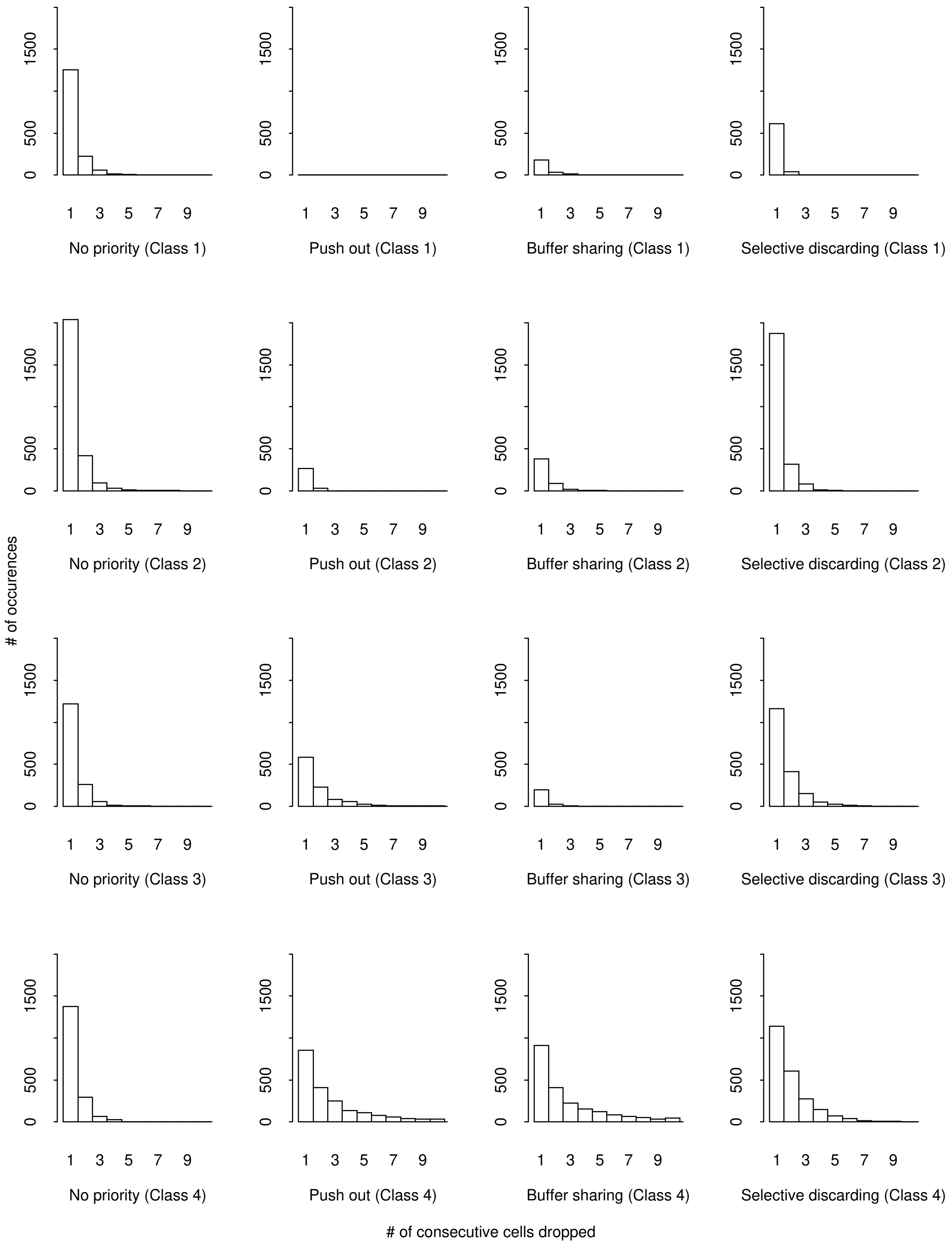

Figure 8.41: Consecutive Cell Loss (mix1, four classes)

The bar graphs help to illustrate how the cell loss experienced by Class 1 traffic when no priority algorithm is used is pushed into lower priority classes when one of the priority algorithms is used. The lowest priority class shows a higher degree of consecutive cell loss when priority queueing algorithms are used.

The results from all traffic mixes show that consecutive cell loss is only an issue for the lowest level priority class. The selective discarding algorithm seems to be slightly less susceptible to consecutive cell loss since it shows a greater number of occurrences of a single cell dropped in between transmitted cells and the curve for consecutive cell loss falls to 0 more quickly than the others.

If consecutive cell losses of more than 10 are rare, they will probably have little effect on the overall performance of applications since the cells in each set of consecutive dropped cells may belong to more than one connection. This means that at the user level, the effective consecutive cell loss could be much lower than the consecutive cell loss shown at the switch. If, however, further research in characterizing traffic on ATM networks shows that bursts tend to contain cells from a single connection, the increase in consecutive cell loss caused by priority queueing algorithms will be of greater concern.

8.5 Performance Comparisons

Of the three algorithms studied, the push out algorithm provides the smallest total cell loss. This result is due to the work conserving nature of the push out algorithm. In general, the total cell loss for the selective packet discarding algorithm was lower than the buffer sharing algorithm. In fact, the total cell loss for selective packet discarding was not much higher than for the push out algorithm.

When two priority classes are considered, the buffer sharing algorithm provides the least cell loss for Class 1 traffic, but also has the highest cell loss for Class 2. All of the algorithms show similar behaviour across the five traffic mixes, differing only in the magnitude of the cell loss.

Four priority classes were not any more effective than two for the buffer sharing algorithm, but substantial gains are realized in the other two algorithms. Under four priority classes, the push out algorithm reduced Class 1 cell loss to 0 without increasing Class 4 loss any more than when two priority classes are used. Class 2 and Class 3 absorbed the cell loss but even these classes showed acceptable performance. The performance of the selective packet discarding algorithm remains fairly sensitive to the parameter settings.

The overhead of each algorithm should also be taken into consideration. The buffer sharing algorithm is the least expensive since it requires only one or three counters, depending on the number of priority classes. The push out algorithm requires pointers to keep track of lower priority cells in the buffer. The number of pointers needed is on the order of the number of cells that are searched. The selective packet discarding algorithm requires a counter, a state variable, and a processor that can compute the function that determines whether or not to drop a cell.

The results of this research show that a priority queueing algorithm can be effective in protecting higher priority cells from excessive loss. More work needs to be done before an algorithm can be chosen. Based on the results in this thesis, the push out algorithm may be the best choice if four priority classes are allowed. It is difficult to know how these algorithms will behave under different traffic conditions. For example, if the burst periods are much longer than the length of the buffer then the push out algorithm may drop most of the lower priority cells that arrive.

Chapter 9

Conclusions and Future Research

The goal of this research was to examine priority queueing algorithms for use at ATM switch buffers as a means of supporting quality of service requirements within the network. The work involved simulating a switch and three priority queueing algorithms using two and four priority classes.

The results show that space priority queueing algorithms can be effective in protecting loss sensitive traffic from excessive cell loss by moving the cell loss to a lower priority class. However, priority queueing cannot stand alone as a congestion control mechanism. It depends on admission control algorithms and traffic shaping to keep the load at a manageable level. If the load on a switch becomes high for an extended period of time, the priority queueing algorithms will be ineffective since the buffer will continuously be full. Also, if a misbehaving connection consumes more bandwidth than it is allowed it may overwhelm the buffer with cells of a single priority class.

This thesis highlights the relative effects of the priority queueing algorithms on cell loss. Only the results for one buffer size and one switch size were presented. It is expected that the algorithms in this study would exhibit similar performance for different buffer sizes. Also, it is unlikely that a larger switch would have a significant impact on the performance of these algorithms. The only difference would be that more packets could arrive at the output at the same time requiring the decision algorithm to operate at a higher speed.

Although the focus of this research was on space priority algorithms, the results indicate that the effect on the queueing delay for all classes was negligible. The issue of guaranteeing acceptable delay limits was not addressed in this research but a similar study of delay priority algorithms, or combined loss and delay priority algorithms, could be performed. Since there is a restriction that cells from a single connection may not be reordered, managing delay within the network is a difficult problem. A delay priority algorithm applied at switch buffers needs to know both the priority class and the connection to which each cell belongs.

Although the CCITT has already determined that only two loss priority classes will be allowed, the possibility of extending this to four priority classes should not be ignored. Four priority classes allow a finer control over the level of performance that different service types can expect. To guarantee very low cell loss to some services when only two priority classes are allowed, only a small subset of connections can use high priority cells. This forces other services that can afford to lose some cells to be treated the same as services that can afford much higher cell loss. On the other hand, if the services with medium loss requirements are placed in the high priority class, the total performance of the higher priority class may suffer since a higher proportion of the cells are in that class, or the low priority class will absorb more cell loss than is really necessary. By placing these services in different priority classes, it may be easier to accommodate the quality of service requirements for a wide variety of services.

The traffic mixes used in this research demonstrate that the choice of a buffer size and the behaviour of the priority queueing algorithms depend to a large extent on the shape of the traffic. This makes it difficult to establish appropriate algorithms for priority queueing until the traffic on the network can be measured and characterized. It is possible that the traffic found on individual ATM networks and on segments of larger ATM networks will vary dramatically according to the applications being run on each segment. To obtain good performance on each network or network segment, parameters such as buffer size and choice of queueing algorithm and its parameters should be tunable.

A better understanding of quality of service requirements and how access control algorithms and congestion control algorithms can satisfy these requirements is needed. It is difficult to determine to what extent higher priority traffic should be able to cause poorer performance to lower priority algorithms. The priority queueing algorithms discussed here vary in their ability to cause lower priority traffic to absorb loss from higher priority classes. The push out algorithm and the buffer sharing algorithm protect higher priority cells at all cost. The selective packet discarding algorithm is the only one where higher priority cells are dropped before the buffer is full.

A more thorough study of consecutive cell loss would provide more insight on how consecutive cell loss at the switch level affects user level performance. If it is a serious problem priority queueing algorithms that avoid consecutive cell loss could be developed.

Much more research is required before priority queueing algorithms can be effectively used in switches. These algorithms should be studied under traffic models that more closely approximate "real" traffic. The end-to-end cell loss through a series of switches should also be studied to determine the parameter settings that should be used when a switch is part of a larger network.

The traffic mixes studied did not vary much in level of burstiness. These queueing algorithms should be tested under other traffic models that show a higher degree of burstiness. For example, it is expected that the selective discarding algorithm would perform well under short bursts. It would be able to spread the loss over the length of the burst instead of clustering the cell loss in the burst.

This research considered switches that have separate buffers at each output. Priority queueing could be applied to buffers that are shared among the outputs.

Different algorithms for the selective packet discarding approach should be considered. For example, the performance of higher priority classes would improve if lower priority cells in the buffer could be replaced.

High demands will be placed on future integrated services networks and the challenge for network designers will be to provide the mechanisms to ensure good performance for all of the applications using the network. This research has shown that priority queueing in combination with congestion control algorithms and access control algorithms can play a significant role in supporting quality of service requirements and should be considered for use in high speed packet switches.

Appendix A

Results for Four Priority Classes

The complete set of results for priority queueing algorithms under four priority classes are presented in this appendix. Sections A.1, A.2, and A.3 display the results for the push out algorithm, buffer sharing algorithm and the selective discarding algorithm, respectively. These figures parallel the figures on two priority classes presented in Section 8.2.

A.1 Push Out

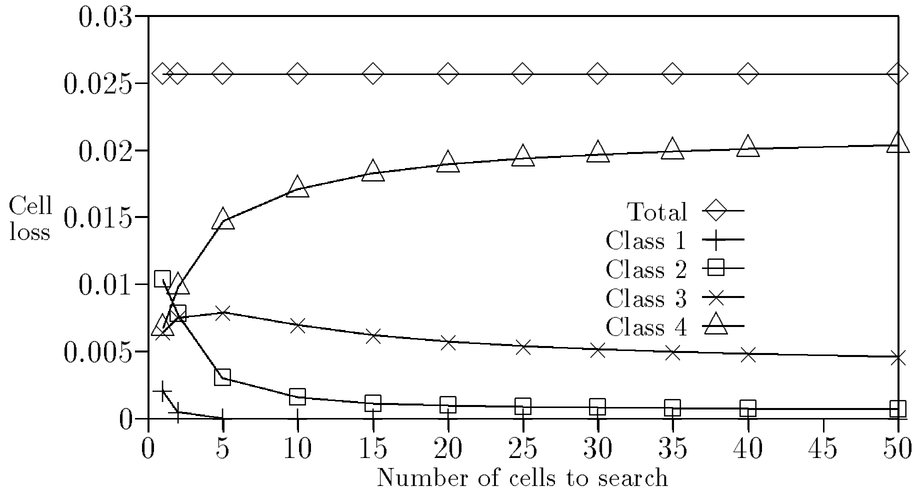

Figure A.1: Push Out (mix1, four classes, fraction of total)

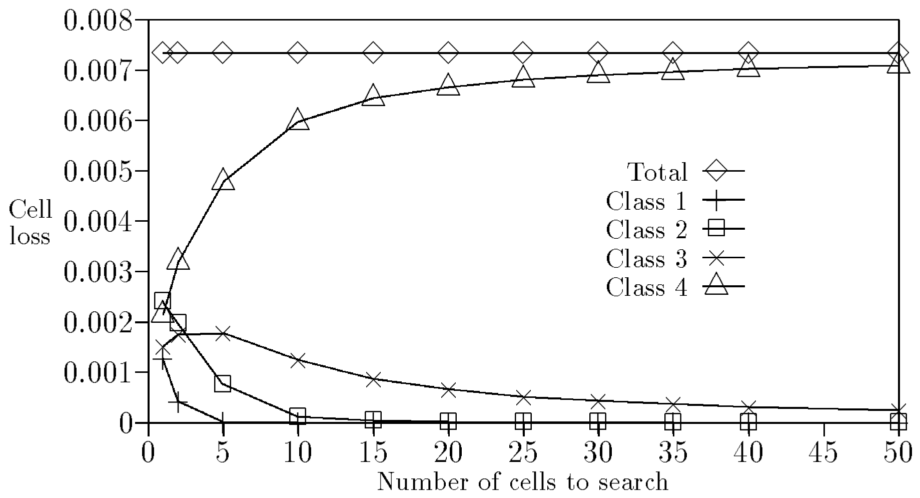

Figure A.2: Push Out (mix2, four classes, fraction of total)

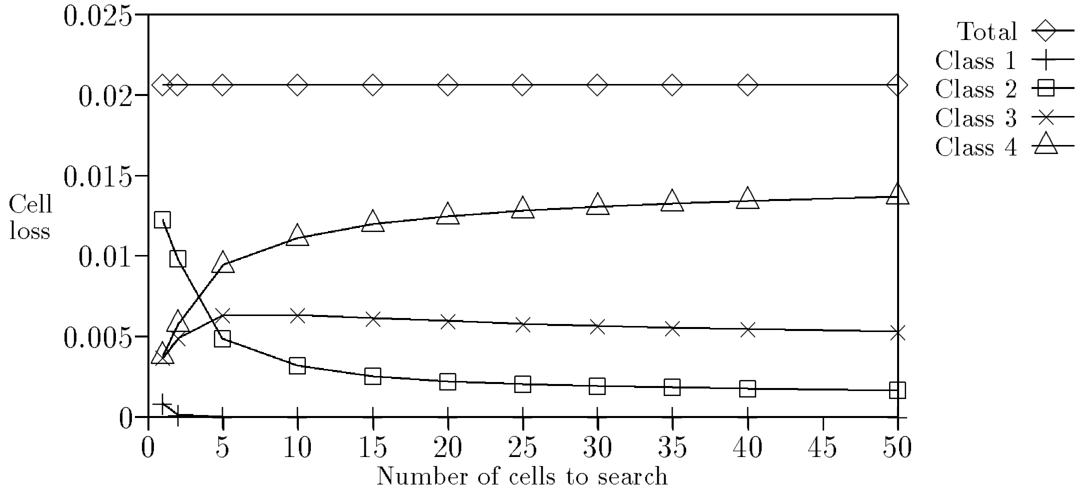

Figure A.3: Push Out (mix3, four classes, fraction of total)

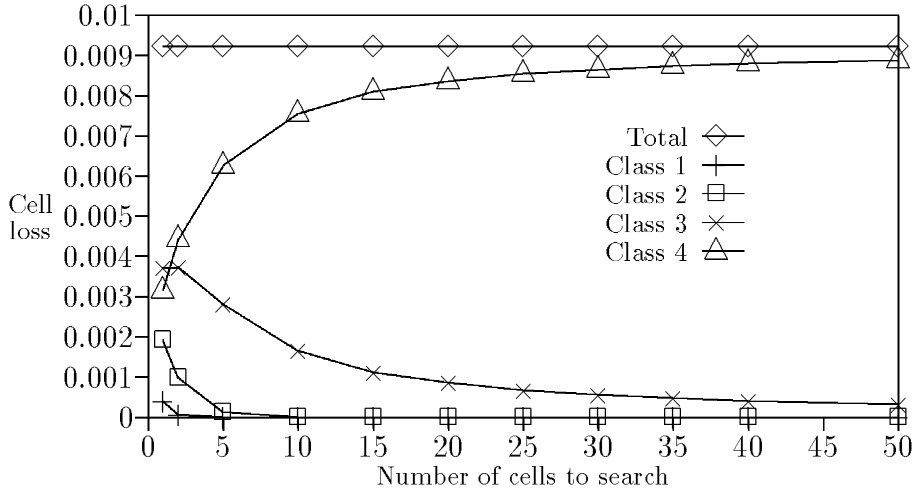

Figure A.4: Push Out (mix4, four classes, fraction of total)

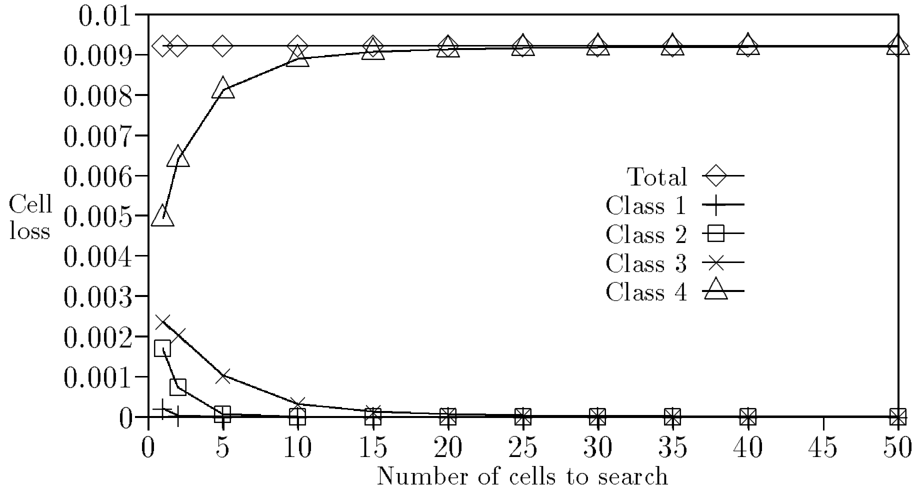

Figure A.5: Push Out (mix5, four classes, fraction of total)

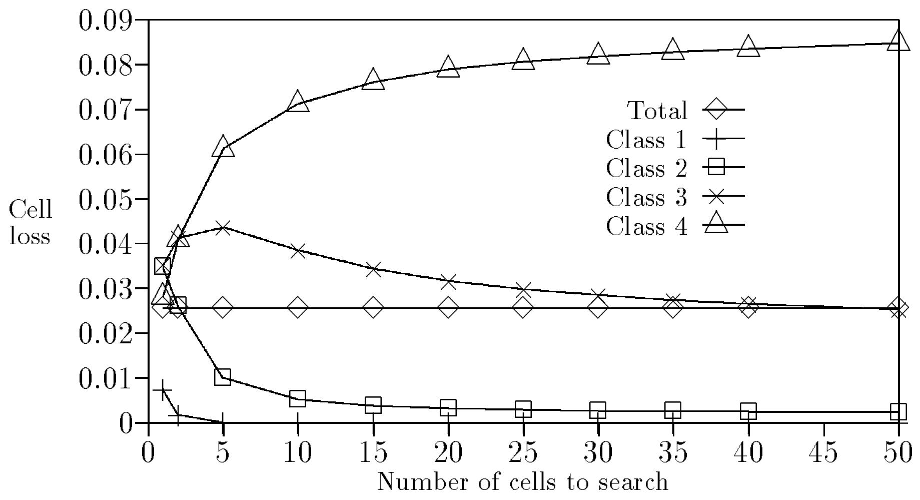

Figure A.6: Push Out (mix1, four classes, fraction of class)

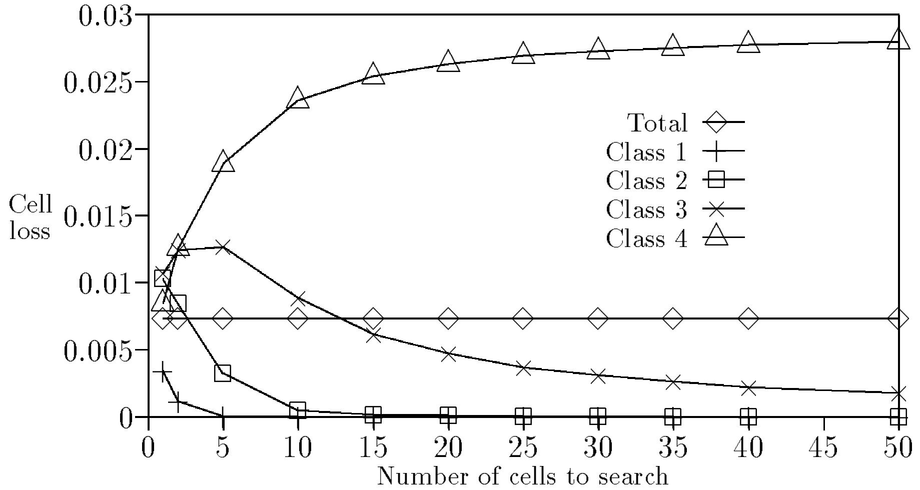

Figure A.7: Push Out (mix2, four classes, fraction of class)

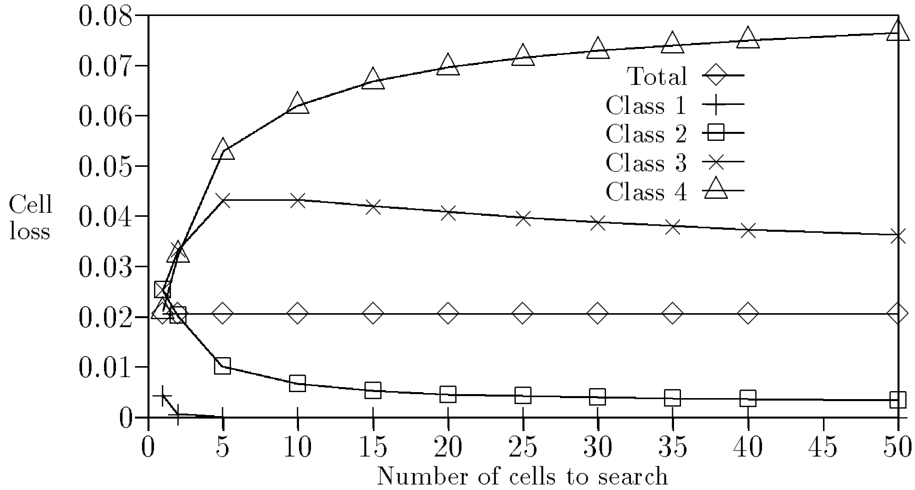

Figure A.8: Push Out (mix3, four classes, fraction of class)

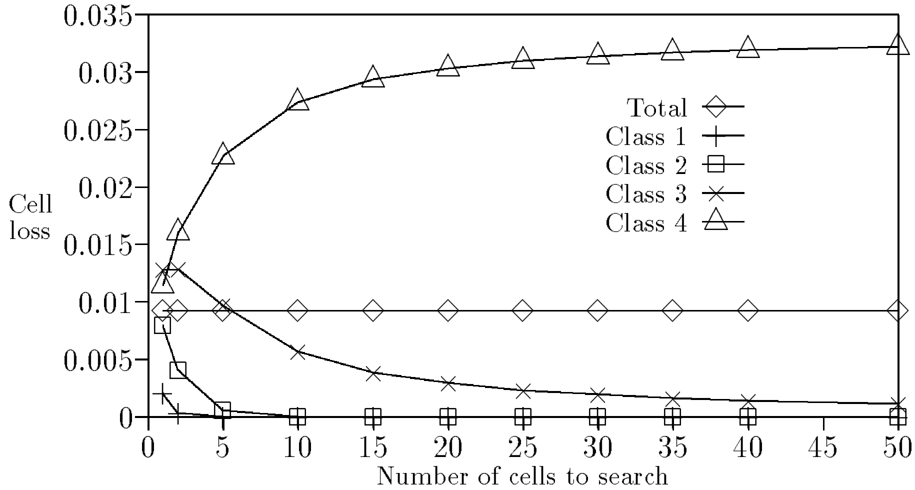

Figure A.9: Push Out (mix4, four classes, fraction of class)

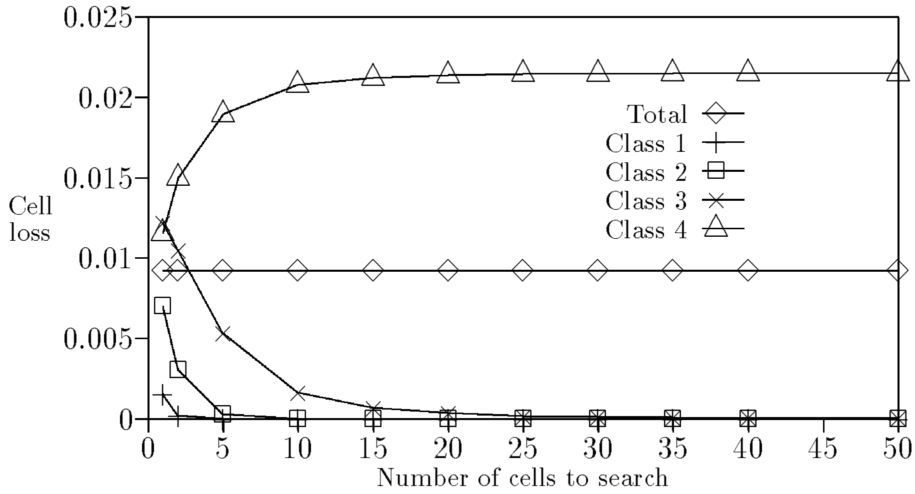

Figure A.10: Push Out (mix5, four classes, fraction of class)

A.2 Buffer Sharing

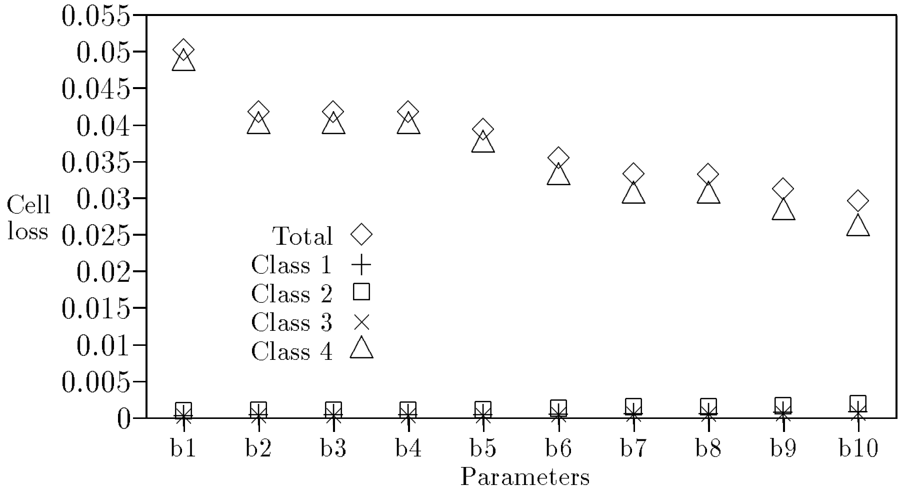

Figure A.11: Buffer Sharing (mix1, four classes, fraction of total)

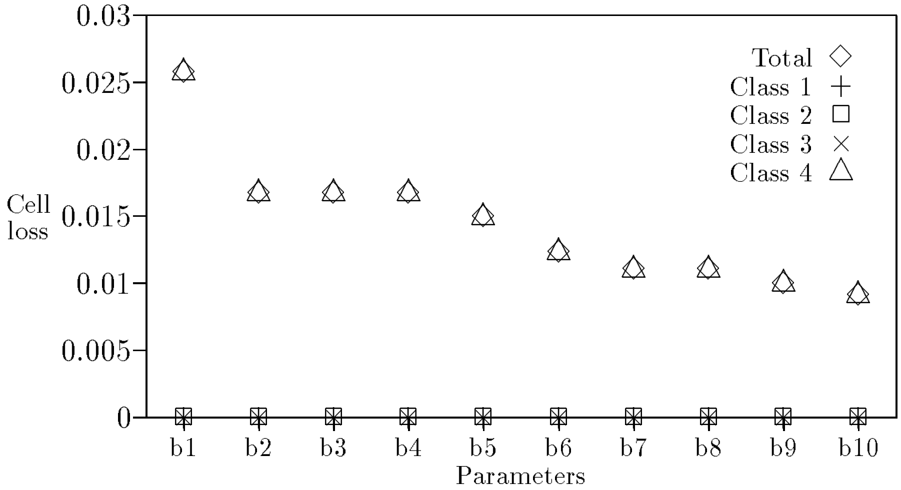

Figure A.12: Buffer Sharing (mix2, four classes, fraction of total)

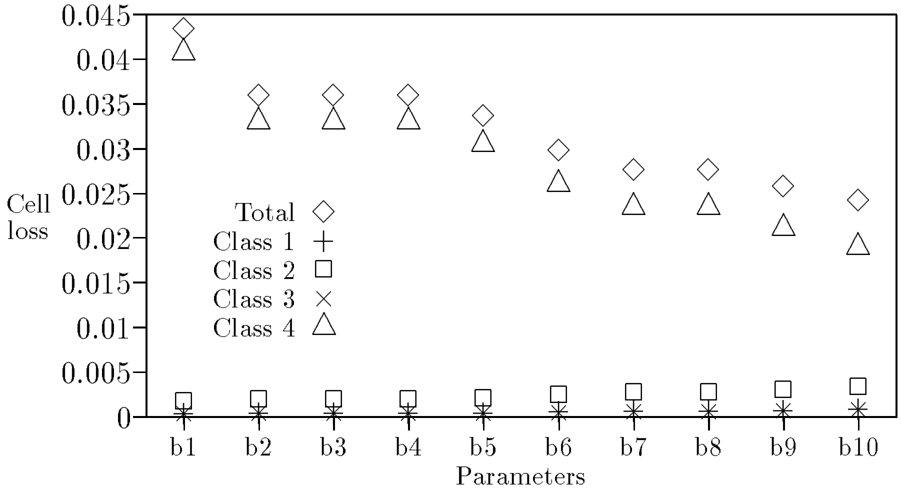

Figure A.13: Buffer Sharing (mix3, four classes, fraction of total)

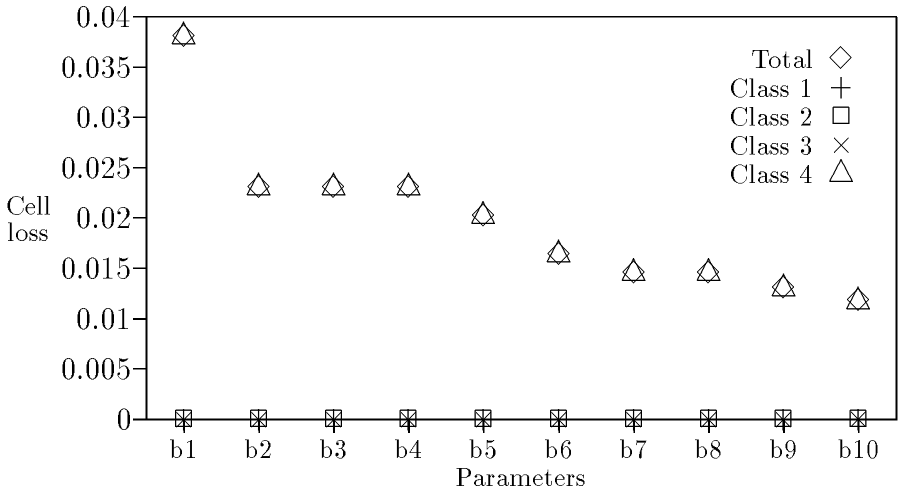

Figure A.14: Buffer Sharing (mix4, four classes, fraction of total)

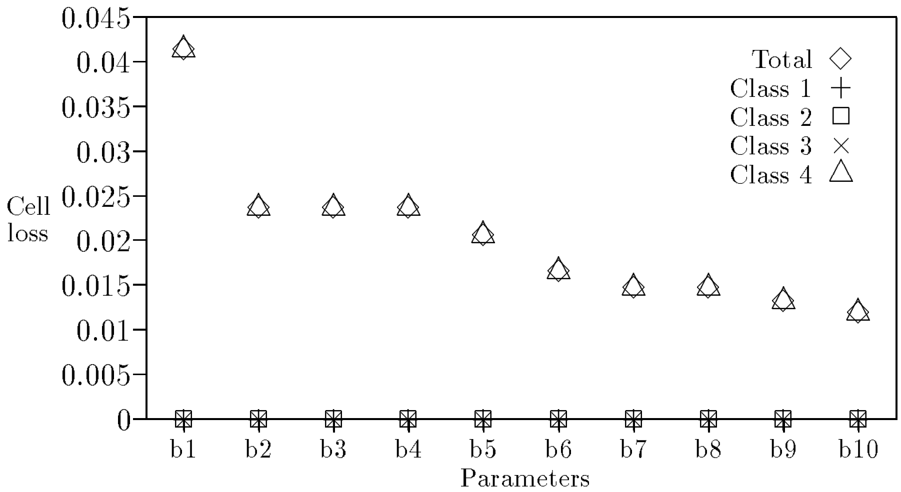

Figure A.15: Buffer Sharing (mix5, four classes, fraction of total)

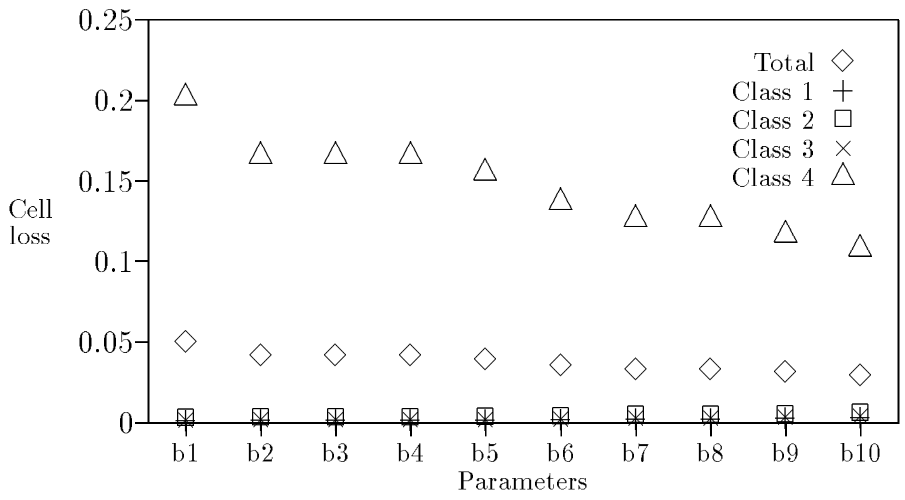

Figure A.16: Buffer Sharing (mix1, four classes, fraction of class)

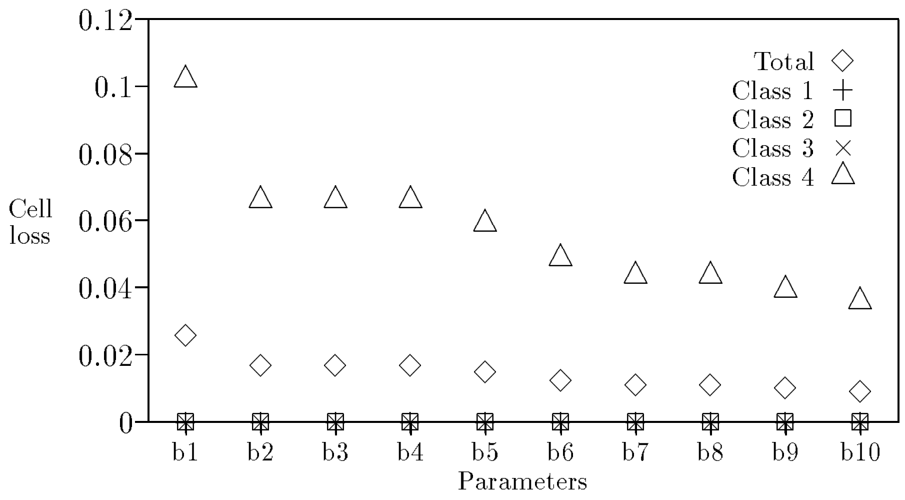

Figure A.17: Buffer Sharing (mix2, four classes, fraction of class)

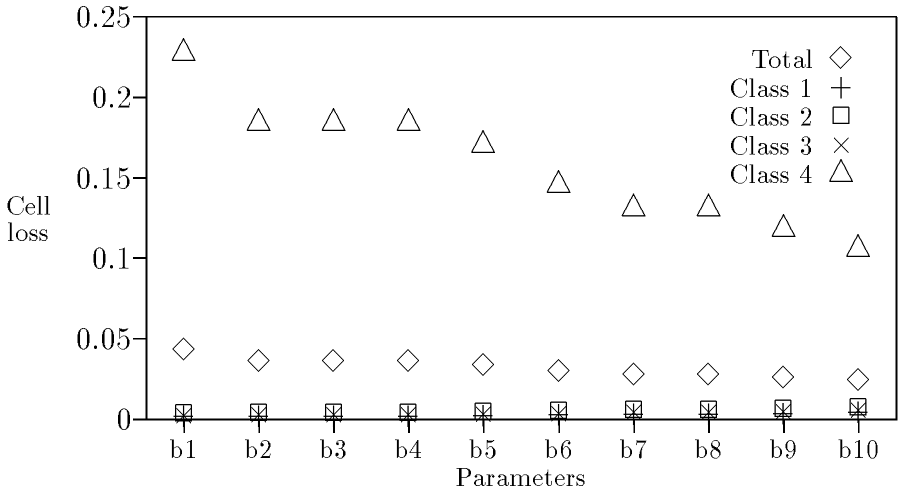

Figure A.18: Buffer Sharing (mix3, four classes, fraction of class)

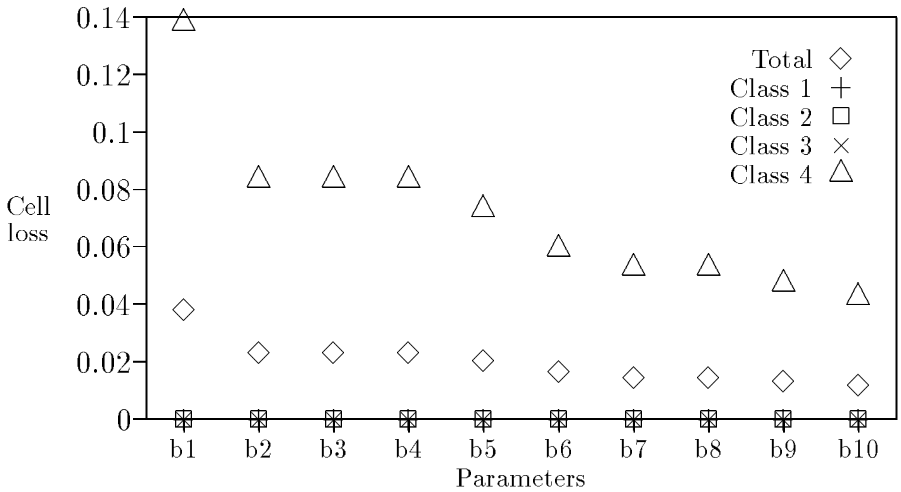

Figure A.19: Buffer Sharing (mix4, four classes, fraction of class)

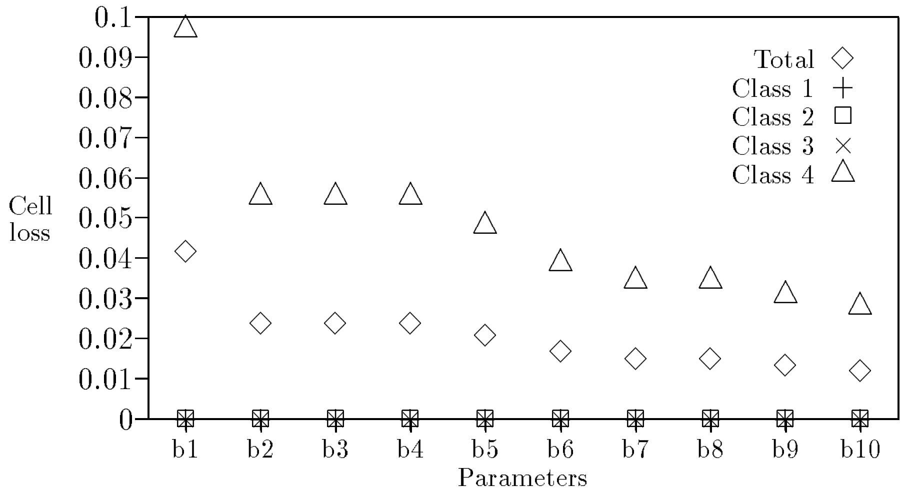

Figure A.20: Buffer Sharing (mix5, four classes, fraction of class)

A.3 Selective Packet Discarding

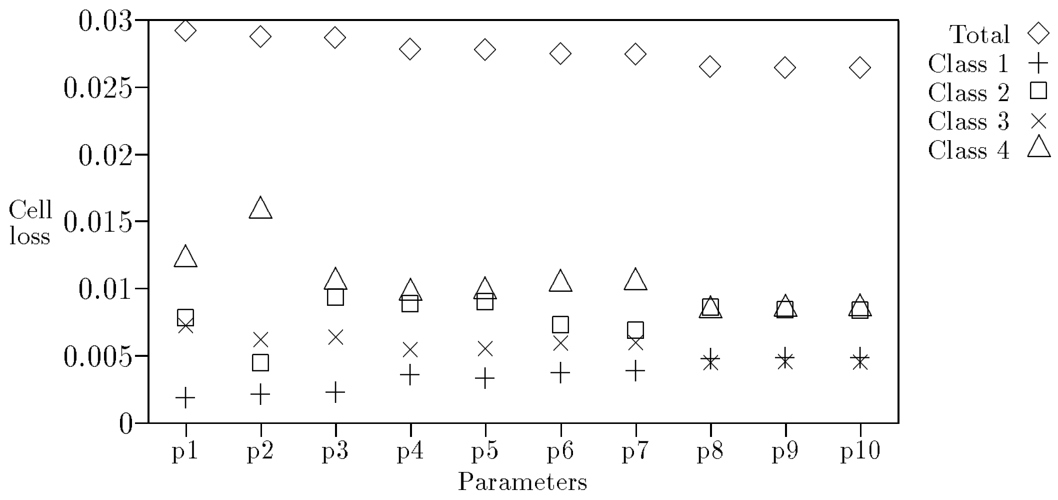

Figure A.21: Selective Discarding (mix1, four classes, fraction of total)

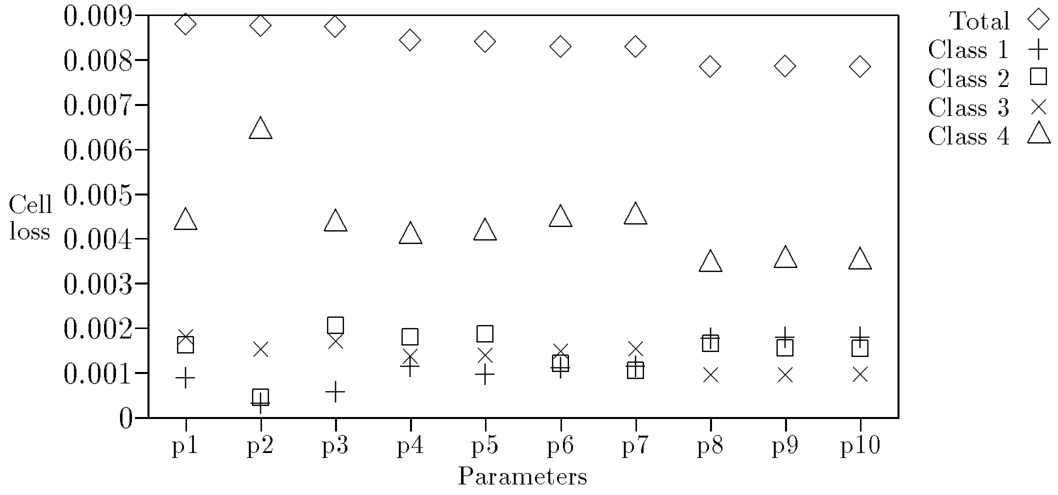

Figure A.22: Selective Discarding (mix2, four classes, fraction of total)

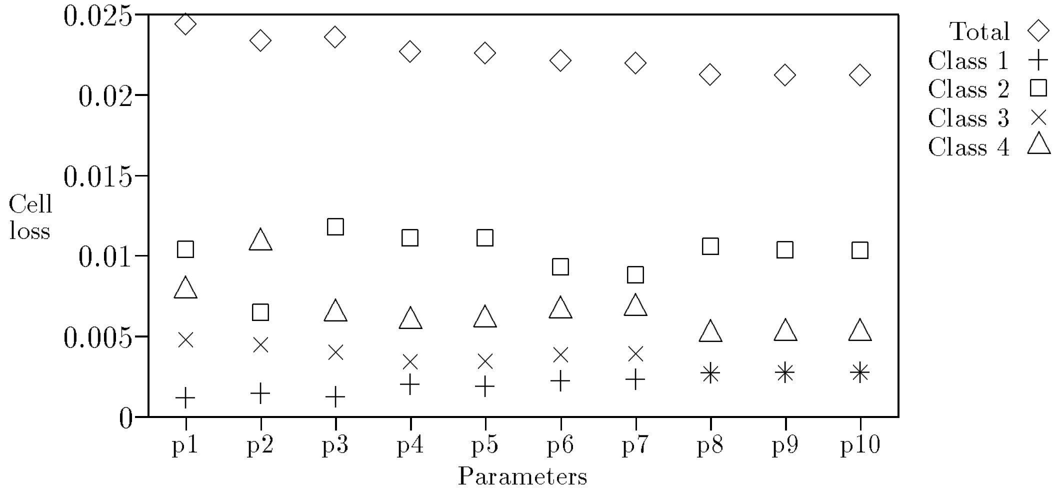

Figure A.23: Selective Discarding (mix3, four classes, fraction of total)

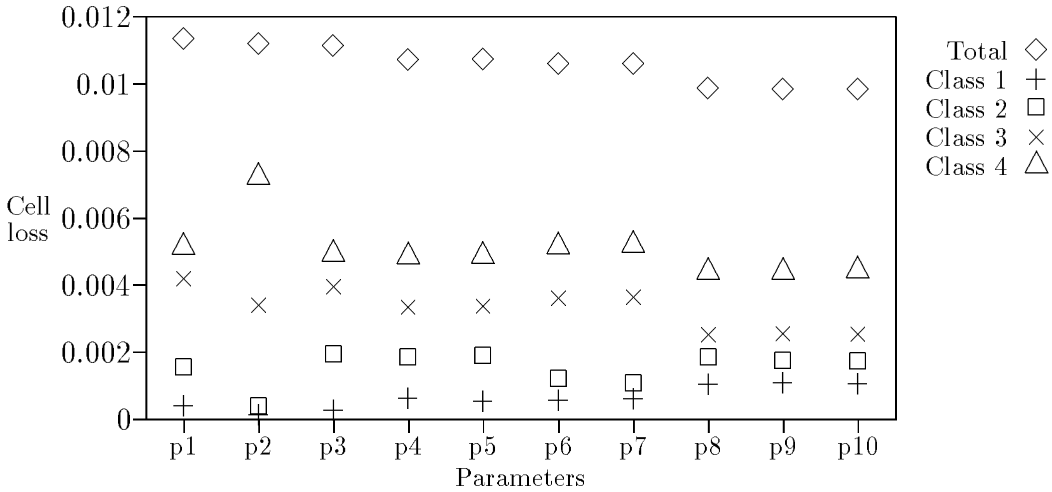

Figure A.24: Selective Discarding (mix4, four classes, fraction of total)

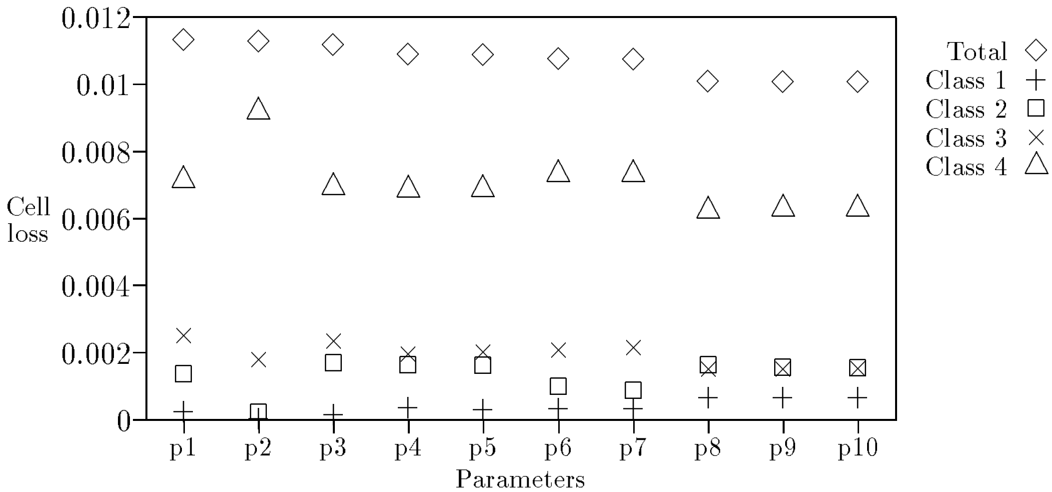

Figure A.25: Selective Discarding (mix5, four classes, fraction of total)

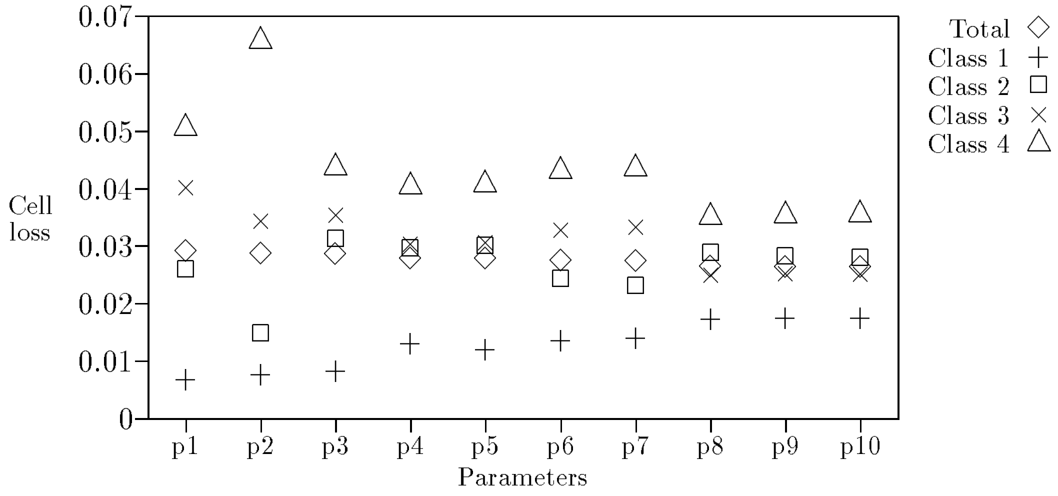

Figure A.26: Selective Discarding (mix1, four classes, fraction of class)

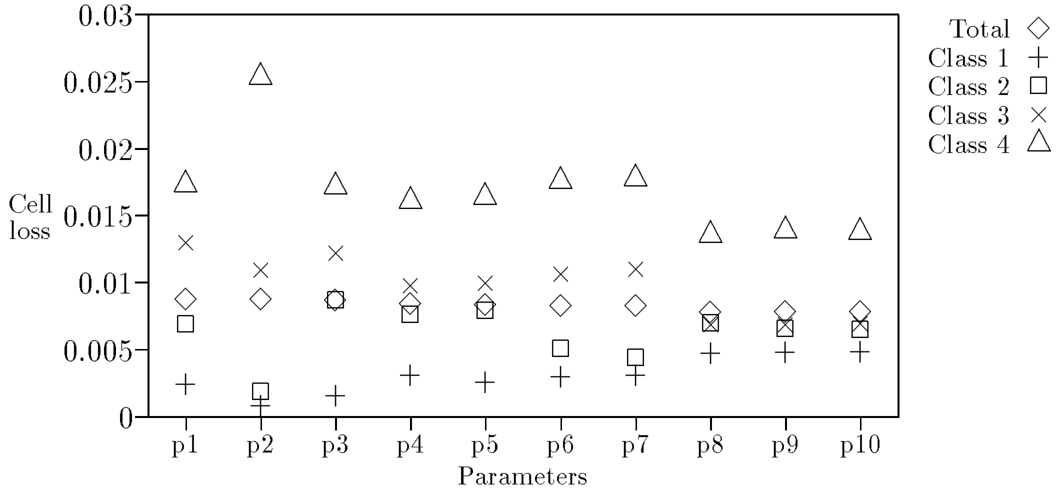

Figure A.27: Selective Discarding (mix2, four classes, fraction of class)

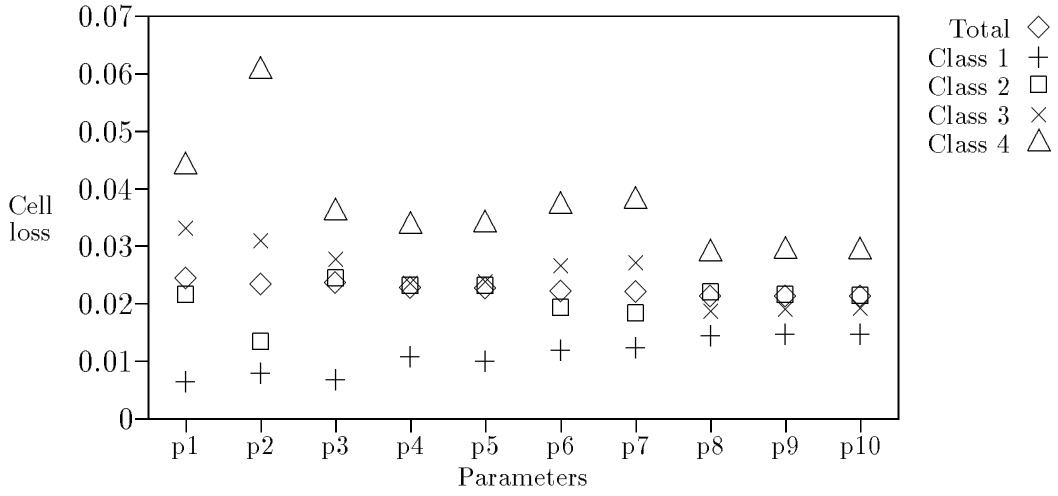

Figure A.28: Selective Discarding (mix3, four classes, fraction of class)

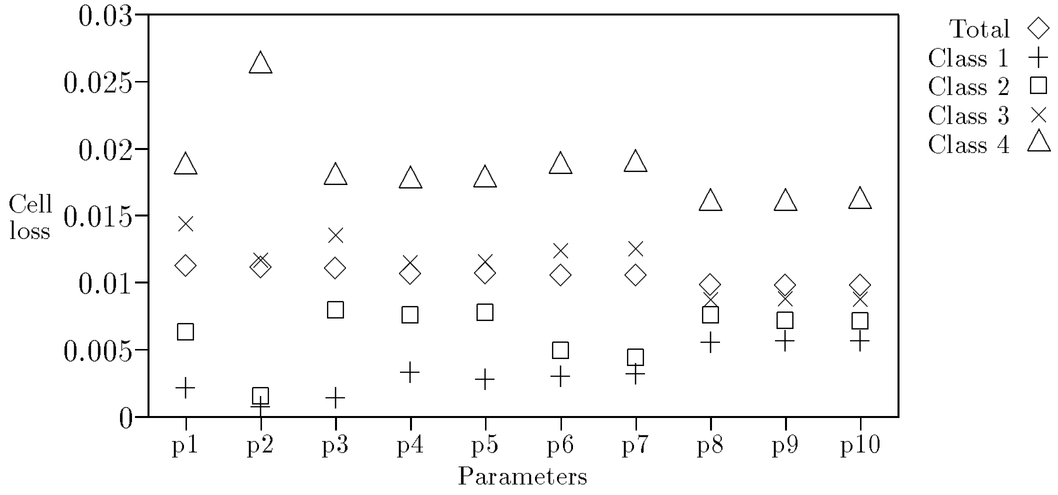

Figure A.29: Selective Discarding (mix4, four classes, fraction of class)

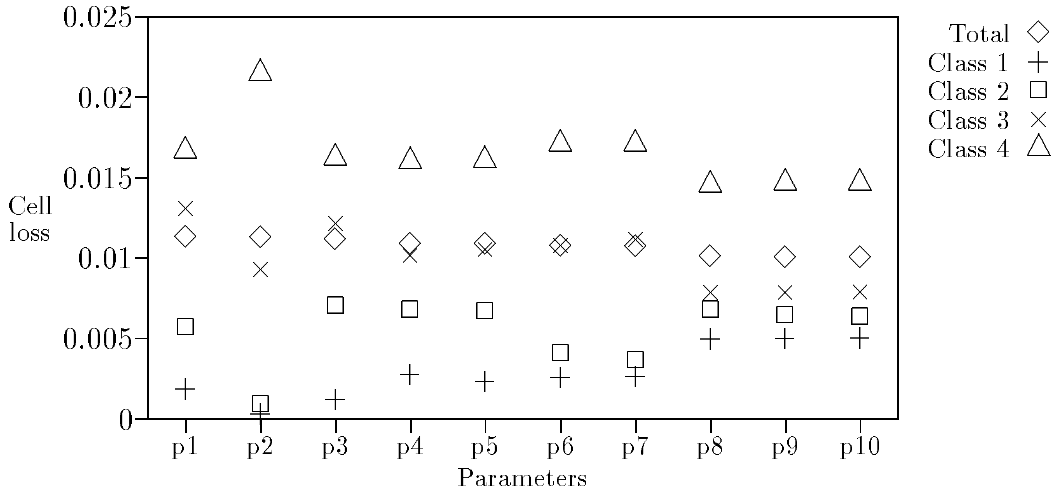

Figure A.30: Selective Discarding (mix5, four classes, fraction of class)

References

[Ahmadi and Denzel, 1989]

Hamid Ahmadi and Wolfgang E. Denzel. A survey of modern high-performance switching techniques. *IEEE Journal on Selected Areas in Communications*, 7(7):1091–1102, September 1989.

[Batorsky *et al.*, 1988]

D. V. Batorsky, D. R. Spears, and A. R. Tedesco. The evolution of broadband network architectures. *Proceedings GLOBECOM 88*, pages 12.1.1–7, 1988.

[CCITT, 1989]

CCITT Recommendation I.121, Broadband Aspects of ISDN. Blue Book, Geneva, Switzerland, 1989.

[Chen and Dobrowski, 1988]

Ben-Ren Chen and George H. Dobrowski. The use of high-speed packet switching technology for supporting switched multi-megabit data service. *Proceedings GLOBECOM 88*, pages 20.5.1–5, 1988.

[Chen and Guérin, 1991]

Jeane S. C. Chen and Roch Guérin. Performance study of an input queueing packet switch with two priority classes. *IEEE Transactions on Communications*, 39(1):117–126, January 1991.

[Daddis and Torng, 1989]

George E. Daddis and H. C. Torng. A taxonomy of broadband integrated switching architectures. *IEEE Communications Magazine*, 27(5):32–42, May 1989.

[Eckberg *et al.*, 1991]

Adrian E. Eckberg, Bharat T. Doshi, and Richard Zoccolillo. Controlling congestion in B-ISDN/ATM: Issues and strategies. *IEEE Communications Magazine*, 29(9):64–70, September 1991.

[Eklundh *et al.*, 1988]

Berth Eklundh, Ingemar Gard, and Gosta Leijonhufvud. A layered architecture for ATM networks. *Proceedings GLOBECOM 88*, pages 13.1.1–5, 1988.

[Eliazov *et al.*, 1990]

T. E. Eliazov, V. Ramaswami, W. Willinger, and G. Latouche. Performance of an ATM switch. *Proceedings IEEE INFOCOM '90*, pages 644–659, 1990.

[Filipiak, 1988]

Janusz Filipiak. Structure of traffic flow in multiservice networks. *Proceedings GLOBECOM 88*, pages 13.4.1–5, 1988.

[Friesen, 1991]

Verna J. Friesen. *A Performance Study of Broadband Networks*. M.Math. thesis, Department of Computer Science, University of Warterloo, 1991.

[Fujii *et al.*, 1988]

Tetsurou Fujii, Mitsuru Nomura, and Naohisa Ohta. Characterization of variable rate inter-frame video coding for ATM-based networks. *Proceedings GLOBECOM 88*, pages 33.1–5, 1988.

[Gallassi *et al.*, 1988]

G. Gallassi, G. Cappellini, L. Fratta, G. Rigolio, and F. Rossi. Performance evaluation of flow control mechanisms in multi-throughput class networks. *Proceedings GLOBECOM 88*, pages 16.3.1–6, 1988.

[Gallassi *et al.*, 1989]

G. Gallassi, G. Rigolio, and L. Fratta. ATM: Bandwidth assignment and bandwidth enforcement policies. *Proceedings GLOBECOM 89*, pages 49.6.1–5, 1989.

[Gilbert *et al.*, 1991]

Henry Gilbert, Osama Aboul-Magd, and Van Phung. Developing a cohesive traffic management strategy for ATM networks. *IEEE Communications Magazine*, 29(10):36–45, October 1991.

[Hebuterne and Gravey, 1990]

Gérard Hebuterne and Annie Gravey. A space priority queueing mechanism for multiplexing ATM channels. *Computer Networks and ISDN Systems*, 20(1-5):37 –44, December 1990.

[Heffes and Lucantoni, 1986]

Harry Heffes and David M. Lucantoni. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):856–868, September 1986.

[Hui and Arthurs, 1987]

Joseph Y. Hui and Edward Arthurs. A broadband packet switch for integrated transport. *IEEE Journal on Selected Areas in Communications*, SAC-5(8):1264–1273, October 1987.

[Hui, 1989]

Joseph Y. Hui. Network, transport, and switching integration for broadband communications. *IEEE Network*, pages 40–51, March 1989.

[Hyman *et al.*, 1991]

Jay M. Hyman, Aurel A. Lazar, and Giovanni Pacifici. Real-time scheduling with quality of service constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, September 1991.

[Jain, 1990]

Raj Jain. Congestion control in computer networks: Issues and trends. *IEEE Network Magazine*, pages 24–30, May 1990.

[Joseph *et al.*, 1988]

K. Joseph, D. Raychaudhuri, and J. Zdepski. Packet video transmission over a broadband implicit token passing LAN. *Proceedings GLOBECOM 88*, pages 20.2.1–7, 1988.

[Karol *et al.*, 1987]

Mark J. Karol, Michael G. Hluchyj, and Samuel P. Morgan. Input versus output queueing on a space-division packet switch. *IEEE Transactions on Communications*, COM-35(12):1347–1356, December 1987.

[Käser and Rao, 1988]

Andre Käser and S. Rao. International standards for broadband aspects of ISDN. *Proceedings GLOBECOM 88*, pages 4.1.1–4, 1988.

[Kröner, 1990]

Hans Kröner. Comparative study of space priority mechanisms for ATM networks. *Proceedings IEEE INFOCOM '90*, pages 1136–1143, 1990.

[Li, 1989]

San-Qi Li. Overload control in a finite message storage buffer. *IEEE Transactions on Communications*, 37(12):1330–1338, December 1989.

[Liew and Wu, 1991]

Soung C. Liew and Kevin S. Wu. Comparision of buffering strategies for asymmetric packet switch modules. *IEEE Journal on Selected Areas in Communications*, 9(3):428–438, April 1991.

[Lin and Silvester, 1991]

Arthur Y. M. Lin and John A. Silvester. Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integrated broadband switching system. *IEEE Journal on Selected Areas in Communications*, 9(9):1524–1536, December 1991.

[Ling, 1992]

S. H. Ling. *Characterizing and Modelling the Workload of Ethernet Local Area Networks.* M.Sc. thesis, Department of Computational Science, University of Saskatchewan, 1992.

[Maglaris *et al.*, 1988]

Basil Maglaris, Dimitris Anastassiou, Prodip Sen, Gunnar Karlsson, and John D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Transactions on Communications*, 36(7):834–843, July 1988.

[Minzer, 1989]

Steven E. Minzer. Broadband ISDN and asynchronous transfer mode (ATM). *IEEE Communications Magazine*, 27(9):17–24, September 1989.

[Petr and Frost, 1990]

David W. Petr and Victor S. Frost. Optimal packet discarding: An ATM-oriented analysis model and initial results. *Proceedings IEEE INFOCOM '90*, pages 537 – 542, 1990.

[Petr *et al.*, 1989]

David W. Petr, Luiz A. DaSilva, Jr., and Victor S. Frost. Priority discarding of speech in integrated packet networks. *IEEE Journal on Selected Areas in Communications*, 7(5):644–656, June 1989.

[Rathgeb *et al.*, 1988]

Erwin P. Rathgeb, Thomas H. Theimer, and Manfred N. Huber. Buffering concepts for ATM switching networks. *Proceedings GLOBECOM 88*, pages 39.3.1–5, 1988.

[Saito *et al.*, 1991]

Hiroshi Saito, Masatoshi Kawarasaki, and Hiroshi Yamada. An analysis of statistical multiplexing in an ATM transport network. *IEEE Journal on Selected Areas in Communications*, 9(3):359–367, April 1991.

[Sriram and Whitt, 1986]

Kotikalapudi Sriram and Ward Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):833–846, September 1986.

[Tobagi, 1990]

Fouad A. Tobagi. Fast packet switch architectures for broadband integrated services digital networks. *Proceedings of the IEEE*, 78(1):133–167, January 1990.

[Turner, 1986]

Jonathon S. Turner. New directions in communications. *IEEE Communications Magazine*, 24(10):8–15, October 1986.

[Verbiest *et al.*, 1988]

Willem Verbiest, Luc Pinnoo, and Bart Voeten. The impact of the ATM concept on video coding. *IEEE Journal on Selected Areas in Communications*, 6(9):1623–1632, December 1988.

[Wernik, 1988]

Marek R. Wernik. Architecture and technology considerations for multimedia broadband communications. *Proceedings GLOBECOM 88*, pages 21.2.1–5, 1988.

[Woodruff and Kositpaiboon, 1990]

Gillian Woodruff and Rungroj Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE Journal on Selected Areas in Communications*, 8(3):437–446, April 1990.

[Wright and To, 1990]

David J. Wright and Michael To. Telecommunication applications of the 1990s and their transport requirements. *IEEE Network Magazine*, pages 34–40, March 1990.

[Yamada and Sumita, 1991]

Hiroshi Yamada and Shucchi Sumita. A traffic measurement method and its application for cell loss probability estimation in ATM networks. *IEEE Journal on Selected Areas in Communications*, 9(3):315–323, April 1991.

[Yamamoto and Wright, 1989]

Yutaka Yamamoto and Tim Wright. Error performance in evolving digital networks including ISDNs. *IEEE Communications Magazine*, 27(4):12–18, April 1989.

[Yin *et al.*, 1990]

Nanying Yin, San-Qi Li, and Thomas E. Stern. Congestion control for packet voice by selective packet discarding. *IEEE Transactions on Communications*, 38(5):674–683, May 1990.

[Yoon *et al.*, 1988]

Hyunsoo Yoon, Ming. T. Liu, and Kyungsook Y. Lee. The knockout switch under nonuniform traffic. *Proceedings GLOBECOM 88*, pages 49.5.1–7, 1988.