Analysis of Novel High-Performance Switch Architectures for Broadband-ISDN

Anil K. Gupta

A THESIS

submitted to the School of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Ottawa-Carleton Institute of Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

University of Ottawa

OTTAWA, ONTARIO, K1N 6N5

ISBN 0-315-83817-5

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Dedicated to my beloved Parents

Abstract

In this thesis, we present an analysis of three novel switch architectures for Broadband-ISDN using the Asynchronous Transfer Mode (ATM). Our focus is on high-performance switch architectures, which have low to medium hardware complexity.

The first switch architecture presented is a speed-up switch with input and output buffers, where a back-pressure mechanism is applied to avoid packet loss at the output buffers. We examine the maximum throughput, mean delay and packet loss rate at the input buffers of this speed-up switch. The switch with a speed-up of 3, and 20 buffers at each output port can achieve a maximum throughput of 90% or more. These results are of a great practical value since a switch with high speed-up is difficult to realize. A simple implementation of the switch is also presented.

The second switch architecture, called Limited Intermediate Buffer (LIB) switch, is based on a crossbar switch fabric. A buffer to store a single packet is provided at each crosspoint of this switch. In addition to this, buffers are provided at the input ports to reduce the packet loss. We propose a new scheduling policy called head-of-line priority selection, which reduces the head-of-line blocking and thus improves the performance of the LIB switch substantially. A 16×16 switch under uniform random traffic can achieve a throughput of 87.5%. A three-stage interconnection network consisting of symmetric and asymmetric LIB switch modules is also presented. The simulation results of the interconnection network prove the efficacy of the LIB switch architecture and the proposed head-of-line priority selection scheme.

Finally, the handling of delay and loss sensitive traffic in ATM networks is discussed. To keep the protocols simple at the ATM layer, we suggest that the handling of these priorities should be de-linked. The performance of two classes of delay-sensitive traffic in an input buffered nonblocking switch architecture is analyzed. The result of the analysis under two different non-preemptive priority schemes suggests that, to reduce the hardware complexity, the packets should not be distinguished based on their priority within the switch fabric. To overcome the throughput limitation of the input buffered switch, a dual plane switch architecture is presented, where each plane is a nonblocking switch with input buffers.

Acknowledgements

I would like to express my sincere gratitude to my thesis supervisor, Dr. N. D. Georganas, for his guidance and encouragement throughout the course of this research. I would also like to express my thanks to Dr. Luis Orozco Barbosa for his invaluable help in some of the simulation and modelling in the course of the thesis.

My special thanks to my beloved wife Alka and son Akshay for their consistent support, without which this work would not have been possible.

I am thankful to all my friends, especially Sudhakar Ganti, who have helped me to bring this work to its present shape.

I am also thankful to the Canadian Commonwealth Scholarship and Fellowship Committee for their financial support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

For some time, the drive of the telecommunications industry is towards the design of a single communications network, which has the potential of providing all services in a unified manner. The advantages of such an integrated network, which can accommodate a variety of diverse services with different bandwidth requirements, are as follows: better utilization of resources, flexibility to support existing as well as future services, and ease of installation and maintenance. Two developments, which motivated the objective of having a unified integrated network are worth mentioning here; these are *the narrowband Integrated Services Digital Network (ISDN)* and *packetized voice* [1].

In the early 1980s, ISDN was defined as a public end-to-end digital telecommunications network for a wide range of user applications, but it was largely in terms of traditional 64 Kb/s TDM channels. A network terminating device on customer premises can accept a wide range of ISDN devices including a telephone with advanced features, a personal computer, alarms, and other devices. The accommodation of high quality video and interactive communication of high-resolution images, on the other hand, may prove difficult with the current ISDN standard [1]. Digitized video requires data rates in the range of 50-100 Mb/s, far exceeding the ISDN rates.

The other development worth mentioning in support of integrated networks is the

integration of packetized voice and data in local/wide area networks. The so-called fast packet networks were conceived as capable of using statistical multiplexing for conserving bandwidth and simultaneously carrying voice, video and a wide range of data traffic.

The two developments described here and the growing demand for high performance networks led to the evolution of Broadband Integrated Services Digital Networks (B-ISDN). The rapid development in the area of transmission systems and fiber optics accelerated the evolution of B-ISDN. The combination of laser technology and fibers provides us data transmission in the gigabits-per-second range. During the evolution, CCITT has undertaken the definition of standards as to permit the deployment of a universal multi-services network.

J. P. Coudreuse[2] proposed a multiplexing and switching method called Asynchronous Time-Division, now referred as Asynchronous Transfer Mode (ATM) as the basis for B-ISDN. Since then, ATM has been the focus of CCITT study group on B-ISDN (Study Group XVIII) and has been recommended as the transport technique for broadband networks. ATM is a specific packet-oriented transfer mode. The multiplexed information is organized in fixed size frames called cells of 53 bytes (48 bytes of data and 5 bytes of control information) [3]. Cells are asynchronously multiplexed and switched on the basis of routing information. Most of the network protocol functions, including information error and flow control, can be performed at the edges of the network, allowing it to switch data at very high speed. While the introduction of fiber optics provides the necessary bandwidth for transmission, the implementation of the network that can provide a wide range of services to the users still remains a challenge. Advances in the field of VLSI technology have been supporting new principles in the design and architectures of high performance switching fabrics. Recently, intensive research efforts are being directed towards the development of suitable switching technology required for broadband networks. In this thesis, we look at the switching architectures for Broadband-ISDN which are simple to implement and at the same time achieve high performance.

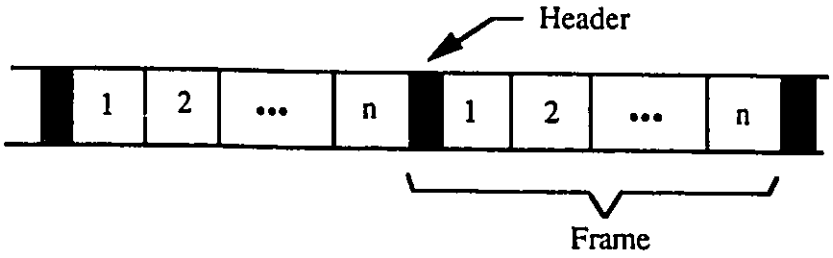


Figure 1.1: STM Frame Structure

1.2 Synchronous Transfer Mode versus Asynchronous Transfer Mode

CCITT Study Group XVIII names the switching and multiplexing aspects as the "Transfer Mode". The Synchronous Transfer Mode (STM) is based on synchronous time division switching and multiplexing. In STM, the time slots are allocated within a recurring structure (frame) to a service for the duration of a call. An STM channel is identified by the position of its time slots within a synchronous frame (Figure 1.1). Once the time slots are assigned for a given service, these time slots are dedicated for the duration of the call. In order to support a variety of services with different information transfer rates, flexible assignment of a set of time slots to a channel for the switched service is possible. In this case, each channel consists of one or more slots per frame. Within a frame, the number of slots required for a service depends upon the peak-rate of the service. But it requires coordination of relatively complex mapping functions by both the user and the network sides of an interface [4].

In order to simplify the mapping function, STM advocates dividing the usable capacity into a limited number of fixed STM-based partitions, called containers. Each container would be permanently assigned a set of time slots. This is called multiple rate STM [4]. In order to increase the flexibility, the containers may be further subdivided. But channels could not span containers. The subdivision of containers can have some undesirable consequences. Multiple rate STM also complicates the switching systems. In terms of utilization of switching bandwidth on a per connection basis, it would be more

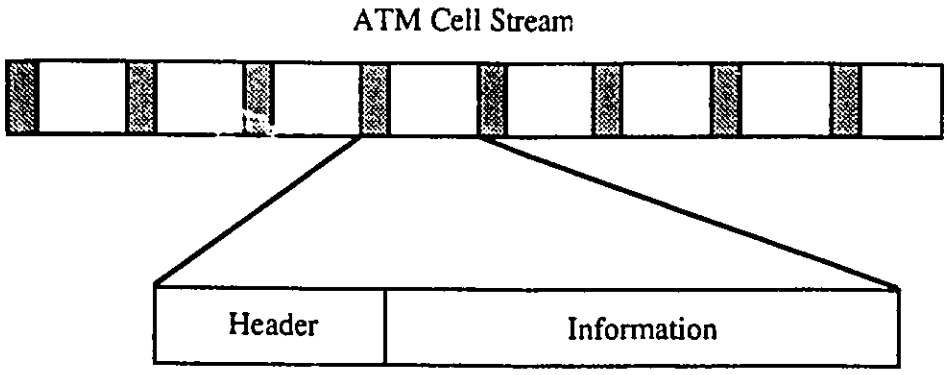


Figure 1.2: ATM Cell Stream

efficient to deploy a separate switching fabric for each channel rate. However, deployment of multiple fabrics is undesirable from the perspective of the network as a whole, complicating management, provisioning and maintenance.

STM is best suited for continuous bit oriented services. B-ISDN, however, needs to support different types of traffic including bursty traffic. Peak rate allocation for each service results in very inefficient bandwidth utilization for the bursty services.

The Asynchronous Transfer Mode (ATM) technique attempts to eliminate these limitations [4]. Usable capacity can be dynamically assigned on demand. A single fabric could conceivably be used to switch all services. The ATM-based networks may be engineered to take advantage of statistical gain among bursty services while guaranteeing acceptable performance for continuous-bit-rate services.

In ATM, specific periodic time slots are not assigned to a channel. Usable capacity (bandwidth) is segmented into fixed-size units called "cells". Each cell consists of a header containing network routing information, and an information field containing user data (Figure 1.2). These cells can be allocated to various services on demand. The header contains a Virtual Channel Identifier (VCI) field. The VCI is a label that is used like the STM time slot position for channel identification. Like a time slot number, a VCI value is locally significant to an interface and a cell may undergo VCI translation, analogous to the time slot interchange in STM, before it is transported over another interface.

ATM can be considered as a compromise between circuit switching and packet

switching. It is intended to achieve time transparency for real-time applications as provided by the circuit switching technique while maintaining the bandwidth sharing flexibility of packet switching. Like circuit switching, ATM is basically a connection-oriented technique although it supports both connection-oriented and connectionless services. Like packet switching, an ATM cell is routed based on the routing information provided by the header.

The main problems apparent with the ATM networks are the variable delay associated with a cell spent in the network, the measures in the network to ensure low cell loss rate and the expense of interworking with existing networks[5]. Some of the issues regarding the universal application of the ATM networks have been recently raised in [6].

Although STM was initially assumed to be the appropriate transfer mode for B-ISDN, the focus gradually shifted in the direction of ATM. For some time, hybrid structures were also considered as an option. CCITT Recommendation I.121 [7], a guideline for future B-ISDN standardization, designates ATM as the "target transfer mode solution for implementing B-ISDN". The advantages of ATM comes in saving bandwidth when sources generate cells at statistically varying rate. ATM can support continuous bit rate as well as bursty traffic, and appears to provide nearly ideal means of transport for the wide range of services to be supported by B-ISDN.

The term "Asynchronous" in ATM does not mean asynchronous transmission systems. The asynchronous here implies that there is no periodicity in the order of cells from different services which make up a multiplexed ATM channel. Fiber transmission standards such as Synchronous Optical Network (SONET) can provide the reliable transmission structure to transport the ATM cells.

1.3 ATM Switching

The main functions of a packet switch are routing and buffering. From the functionality requirement point of view, the ATM switches and conventional packet switches are quite the same, except that the ATM switches are required to switch packets at very high

speeds. The switching capacity of packet switches used in current computer networks is 1 to 4 thousand packets per second with average nodal delays of $20 - 50$ ms [8]. The fast packet switches are expected to handle up to one million or more packets per second per input line with transfer delay of less than half a millisecond [9]. To achieve this objective, the ATM network protocols have to be very simple and to be implemented in dedicated hardware. Since fiber optic communication is virtually error free, new protocols are needed for ATM based networks. Error checking and flow control are pushed to higher layers so as to achieve fast switching and high throughput by simple protocols at the lower layers.

ATM is a fixed block size (cell) based fast packet switching. Fixed size cells allow the design of simpler multiplexers and switches and solves some potential network problems of long packets, delaying the delivery of short packets. To satisfy the requirements of various services, the ATM network should be capable to identify cells depending upon the information they contain. The cells are to be distinguished based on their delay and loss sensitivity. For example, packet video cells require low end-to-end delay as well as low cell loss rate.

An $N \times N$ switch has N input and N output ports (Figure 1.3). The function of the switch is to route the packets (cells) arriving at its input ports to the appropriate outputs as indicated in the header of the arriving packets. An ATM switch has to operate on fixed length packets in a synchronous fashion. Due to the statistical nature of the traffic offered to the switch ports, packets of several inputs may be destined to the same output port simultaneously. This is called output port contention. To resolve this output port contention problem, buffers have to be provided to store packets until they are delivered to the desired output links. The position of the buffers in a switch architecture greatly influences the switch performance. A switch may have buffers at input ports, at output ports, within the switch fabric or a combination of them.

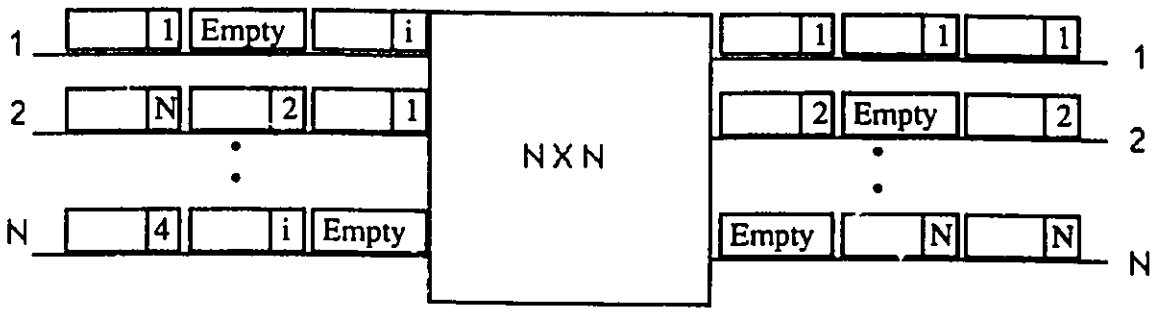


Figure 1.3: $N \times N$ Packet Switch

1.4 Objectives

The objectives of the thesis are to analyze the performance of three different high-performance switch architectures for Broadband-ISDN. The high-performance switch architectures, in general, require that the buffers be placed at the output ports. These output buffered switches tend to have large hardware complexity or require high speed-up in their operation. Our interest is in high-performance switch architectures, which have low to medium hardware complexity.

1. Switch with Speed and Output Buffer Constraints: In this switch, the fabric is speeded-up by a small factor, say 2 to 4 times, with respect to the speed of input-output ports. This is called speed constraint. In addition to the speed constraint, there is an output buffer constraint, i.e., only a limited amount of buffer per output port (e.g., 10 to 20 buffers) is provided. Whenever the output buffer is full, a back-pressure is applied to avoid packet loss at the output buffers.

2. Limited Intermediate Buffer (LIB) Switch: The LIB switch is basically a crossbar switch fabric with a single buffer at each crosspoint. In addition to this, buffers are provided at the input ports to reduce the packet loss. The switch fabric operates at the same speed as that of input-output ports.

3. Input Buffered Nonblocking Switches: An input buffered nonblocking switch is analyzed under delay sensitive priority traffic. As an input buffered switch has severe

throughput limitation, a dual plane switch is also analyzed, where each plane is a nonblocking switch with input buffers.

1.5 Outline of the Thesis

Many switch architectures have been proposed in the literature for high speed packet switching networks. The performance of a switch is primarily governed by its ability to cope with different types of blockings. Although the switches are mainly classified based on internally blocking/ nonblocking features, output blocking and Head Of Line (HOL) blocking also considerably affect the performance of a switch. In the next chapter, we first discuss the various types of blockings and then we present an overview of the fast packet switch architectures. Blocking switches are generally based on Banyan or topologically equivalent structures. Several internally nonblocking switch architectures have also been proposed in the recent past. We discuss some of the blocking and nonblocking switch architectures. In Chapter 3, we summarize the performance of different switch structures.

In Chapter 4, we present a detailed analysis of the switch with speed and output buffer constraints. The analysis is divided into two parts. In the first part, we consider the case when a speed-up factor (defined as the number of packets that can be switched to a given output port in a time slot) is higher than the amount of buffer per output port. The second part of the analysis corresponds to the case when the speed-up factor is equal to or less than the amount of output buffer. In both parts, we study three performance parameters: the maximum throughput, mean delay and packet loss rate at the input buffers. We also present a simple implementation of such a switch.

Our second approach to achieve a high-performance packet switch, called Limited Intermediate Buffer Switch is based on a crossbar switch fabric. This part is presented in Chapter 5. We propose to have a single buffer at each crosspoint of the switching array. This reduces the output blocking considerably. The most important feature is that the switch fabric operates at the speed of the input-output port. A novel scheduling scheme based on HOL blocking is proposed, which improves the performance significantly. To

build the large switching system, a multistage interconnection network is used, which meets the demands of large scale ATM switch design.

An ATM network should be capable of identifying cells based on their loss and delay priorities. In Chapter 6, we analyze the performance of two classes of delay-sensitive traffic in an input buffered nonblocking switch architecture. To overcome the throughput limitation of the input buffered switch, we examine a dual plane switch architecture, where each plane is a nonblocking switch with input buffers. The two planes are connected in parallel to form a load sharing arrangement. We consider that the switch is carrying three classes of delay sensitive traffic, which is shared by the two planes. The highest priority traffic is carried by the second switching plane and the traffic of other two classes is served by the first plane. A buffer is provided at each output of the first plane to queue the packets, whenever the output port is busy transmitting packets from the second plane. We analyze the mean delay in this output buffer using a conservation law.

Finally, in Chapter 7 we conclude by summarizing the contributions of this thesis and suggest directions for future research.

1.6 List of Publications

Publications that have resulted from this research are:

1 A. K. Gupta, L. Orozco Barbosa and N. D. Georganas, "Switching Modules for ATM Switching Systems and their Interconnection Networks," *International Journal on Computer Networks and ISDN Systems*, (to appear).

2 A. K. Gupta and N. D. Georganas, "Analysis of a Packet Switch with Input and Output Buffers and Speed Constraints," *Proc. IEEE INFOCOM'91*, pp. 694-700, Miami, Fl., April 1991.

3 A. K. Gupta and N. D. Georganas, "Buffer Allocation in an ATM Switch with Output Buffer and Speed Constraints," *Proc. Canadian Conf. on Elec. and Comp. Eng.'91*, Québec, Canada, 42.1, Sept. 1991.

4 A. K. Gupta, L. Orozco Barbosa and N. D. Georganas, "16 × 16 Limited Intermediate Buffer Switch Module for ATM Networks," *Proc. IEEE GLOBECOM'91*, 27.5, Phoenix, Az., Dec. 1991.

5 A. K. Gupta and N. D. Georganas,"Priority Performance of ATM Packet Switches," *Proc. IEEE INFOCOM'92*, pp. 727-733, Florence, Italy, May 1992.

6 A. K. Gupta, L. Orozco Barbosa and N. D. Georganas, "Limited Intermediate Buffer Switch Modules and their Interconnection Networks for B-ISDN," *Proc. IEEE ICC'92*, 354.7, Chicago, June 1992.

Chapter 2

Review of Fast Packet Switch Architectures for B-ISDN

2.1 Introduction

In this chapter, we present a survey of fast packet switch architectures for Broadband-ISDN. We focus on switches designed to be implemented electronically. Photonic switches would not be considered in this work.

The switches can be classified into two categories: internally blocking and nonblocking switches. In an internally blocking switch, the blocking can occur at the internal links in the switch fabric. Let us assume synchronous operation of the switch. An internally blocking switch is the one in which all input packets with distinct output addresses cannot be routed to the respective output ports in a time slot, due to the contention for the internal links. For example, a Banyan network is a blocking switch. In contrast, an internally nonblocking switch is the one, in which all packets with distinct output addresses can be routed to the respective output ports within a time slot, as shown in Figure 2.1. Although the switches are mainly classified based on internal blocking, there are two more types of blocking, as discussed below.

Output Blocking

When more than one packet is destined to the same output port (destination con-

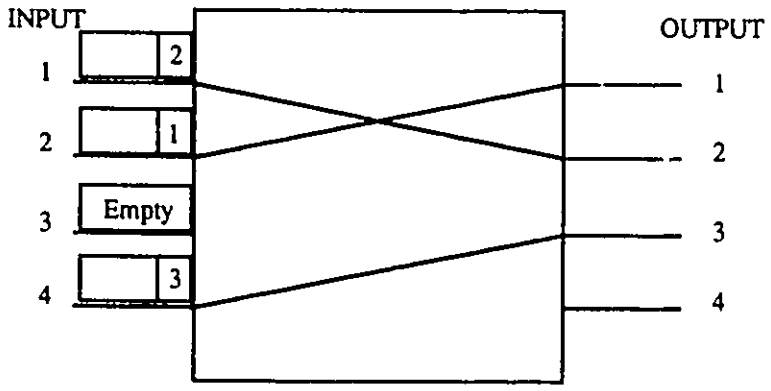


Figure 2.1: Internally Nonblocking Switch

flict), the blocking occurs at the output of the switch if the switch cannot route all the conflicting packets to that output. If only one packet can be routed among the conflicting packets, buffers should be provided at the input ports to minimize the packet-loss. Such a switch is often called an input buffered switch. The packets, which lose in the contention, are buffered and they contend again in the next time slot. When a switch fabric can route all the conflicting packets in a time slot, buffers are to be provided at the output ports to receive multiple packets. This is called an output buffered switch. There is also a trade-off between an input buffered and an output buffered switch. In such a switch, only a limited number of conflicting packets are routed to a given output port. This is called a speed-up switch and the number of packets that can be routed to an output port in a time slot is called the speed-up factor. For the reasons which are obvious from the above discussion, the speed-up switch requires buffers at the output as well as at the input ports. The speed-up switch does suffer from the output blocking, but the blocking diminishes rapidly with the speed-up factor. An $N \times N$ output buffered switch has a speed-up equal to N and it does not suffer from output blocking.

Head of Line (HOL) Blocking

This type of blocking occurs at the input queues of the switch. Let us consider two input queues with head of line packets contending for the same output. One of these HOL packets wins the contention, while the other HOL packet is blocked. The blocked packet may prevent the routing of the next packet in the input queue destined for an

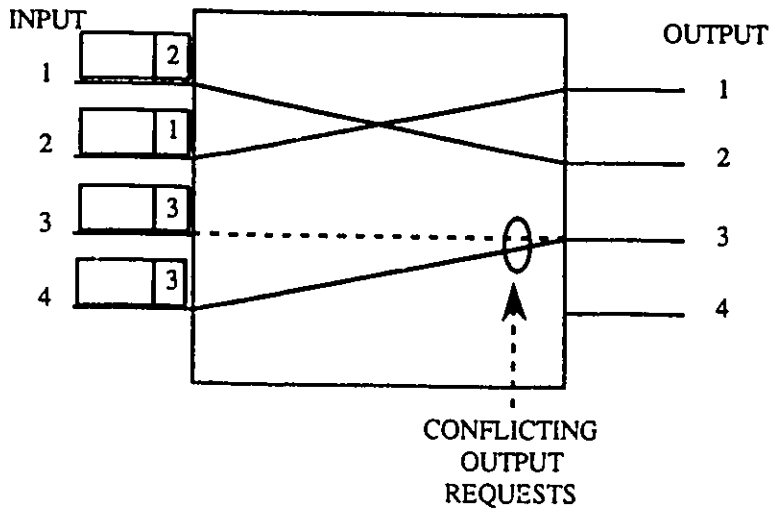


Figure 2.2: Output Blocking in an Internally Nonblocking Switch

output, which is idle in this time-slot. Figure 2.3 shows the HOL blocking in a 4×4 switch. Although no HOL packet is destined to output 4, a packet for output 4 is blocked at input 1, because the HOL packet at input 1 loses in contention against the HOL packet at input 3. It is clear that the output blocking and the HOL blocking adversely affect the throughput of the switch.

2.2 Blocking Switches

Some multistage interconnection networks have been proposed to implement internally blocking fast packet switch fabrics. These networks are generally based on Banyan or topologically equivalent structures. Because the performance of the Banyan networks is inadequate, the buffered Banyan, dilated Banyan, replicated Banyan, Banyan networks with internal speed-ups and Banyan networks with output channel grouping have been considered to enhance the performance. In this section, we will briefly review some of Banyan based networks.

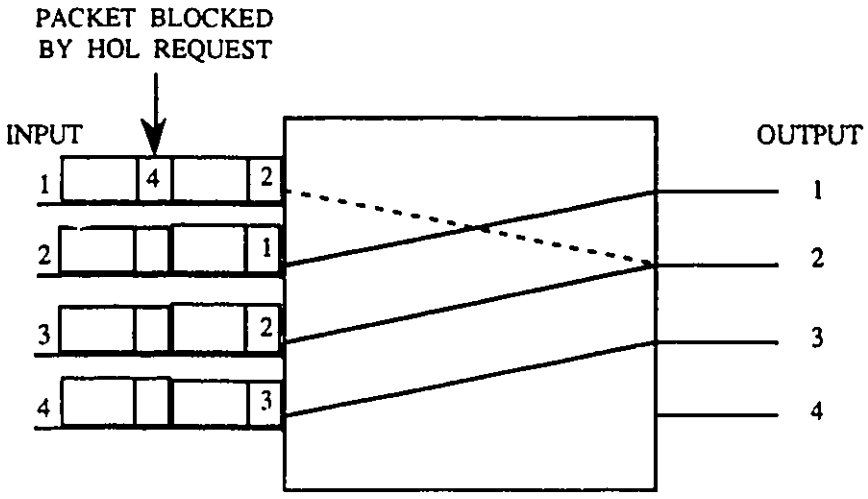


Figure 2.3: Head of Line (HOL) Blocking in an Internally Nonblocking Switch

2.2.1 Banyan Network

Wu and Feng [10] have shown that Banyan networks, baseline networks, omega networks (Shuffle exchange), and flip networks (inverse shuffle exchange) are topologically equivalent. Each of these networks consists of N inputs, N outputs, $log_2 N$ stages of switching elements and interconnection links. The switching elements are 2×2 crossbar switches operating synchronously for fast packet operation. In a time-slot, a switching element can be set either in a direct connection or in a crossed connection state. Figure 2.4 shows a 16×16 Banyan network.

A Banyan interconnection network has the unique path property, if the sequence of nodes connecting an input to an output is unique for all input-output pairs [11]. An important property of the Banyan networks is that they are self-routing. In a n-stage switch (i. e. $N = 2^n$), each packet has an n-bit header. This header contains all the information needed to route a packet through the switching network and such routing is done using a very simple mechanism. The switching element at the first stage routes the packet up or down according to the first bit of the header ('zero' or 'one' indicating up or down routing respectively) and then removes the first bit from the header. The succeeding switching elements will perform the same routing function by removing one

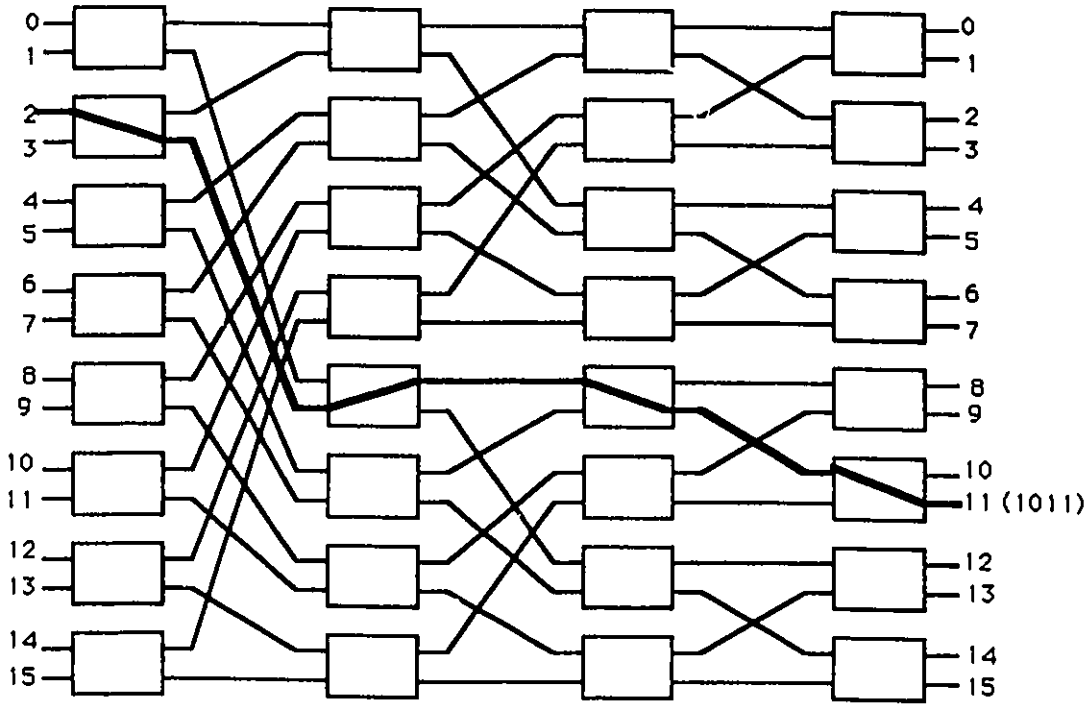


Figure 2.4: Structure of a 16 × 16 Banyan Network and Self-routing

bit from the header and routing the packet to the next stage until the packet reaches its destination output port. It is easy to see that the header is simply the binary address of the output at the last stage and is independent of the input port at the first stage. Figure 2.4 shows the routing of a packet to output port 11.

2.2.2 Buffered Banyan Network

Because of the internal blocking, the throughput of the Banyan network is low and decreases with an increase in the number of stages in the network. To improve its performance, buffers can be provided at the inputs of every switching element. Whenever a conflict arises for an output of a switching element (i. e., for an internal link), one packet is routed and the other one is buffered, which will again contend in the next time-slot.

Jenq [12] presented an analysis of Banyan networks, for the case when a single buffer is provided at each input of every switching element (Figure 2.5). He assumed that, in order for a packet to be able to move forward, either the buffer at the next stage is

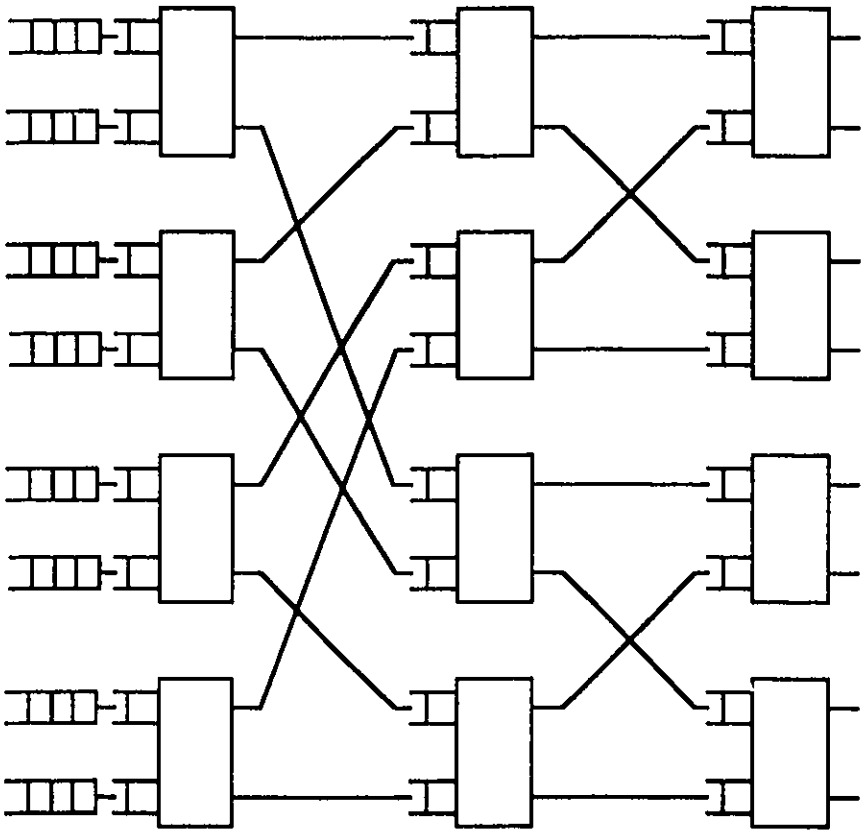


Figure 2.5: Structure of a 8 × 8 Buffered Banyan Network

empty or there is a packet in the buffer and that packet is able to move forward. The network operates synchronously. In the first part of each time slot, control signals are passed across the network from the last stage towards the first stage, so that every packet knows whether it should move forward one stage or stay in the same buffer. Then, in the second part of the time-slot, packets move in accordance with control signals and this ends the time-slot. Since packets can be prevented from moving forward, a relatively large buffer at each input port of the network is provided to minimize the overflow due to the back-pressure.

Providing buffers at the inputs of the switching elements improves the performance substantially. In a 10-stage switch, by providing a single buffer at each input, a maximum throughput of about 0.45 is achieved, in comparison to the 0.30 in an unbuffered switch. Performance of buffered Banyan networks has been studied in [13] when buffers of various

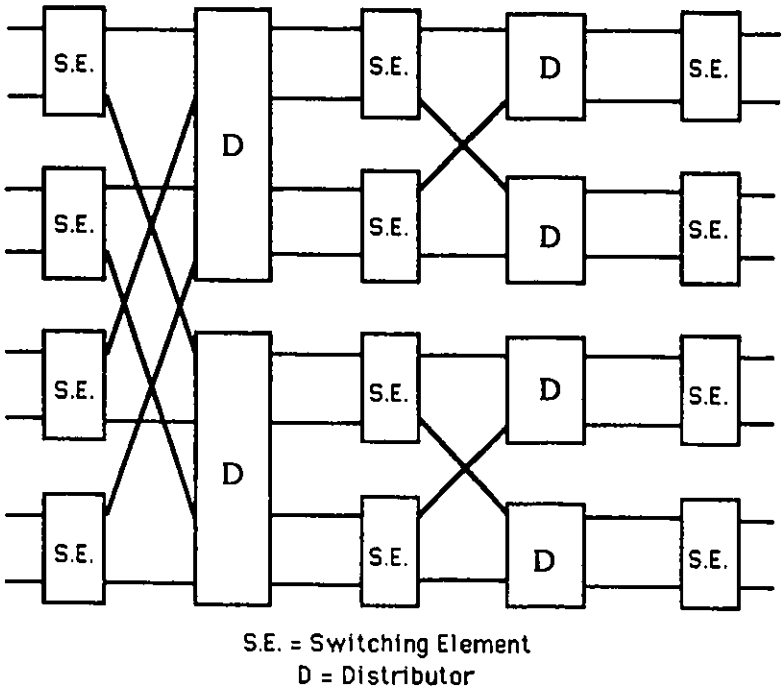


Figure 2.6: Block Diagram of 8 × 8 Switching Network with Internal Speed-ups

sizes are provided. It was demonstrated that as the buffer size is increased up to 5 buffers, the throughput increases considerably, but any further increase in the buffer size improves the throughput only marginally.

An approximate analysis of such a switch under bursty traffic conditions has been performed in [14].

2.2.3 Buffered Banyan Network with Internal Speed-ups

Kim and Leon-Garcia [15] have analyzed multistage switching networks consisting of 2×2 switching elements, distributors and buffers located between stages and the output ports. The switching network requires a speed-up of two. Figure 2.6 shows the block diagram of an 8×8 network. If the distributors are removed, the linking pattern is the same as that of an 8×8 Banyan network. The function of the distributors is to distribute its input packets evenly across its output buffers and to provide a balanced load to the input ports of the switching elements in the next stage.

The 2×2 switching elements operate synchronously. The time slot is divided into two subintervals and even if the two packets at the inputs of an switching element have the same destination, they can be routed to the desired output within two subintervals. The basic element of the distributor is the reverse Banyan network. The distributor distributes the packets in cyclic fashion across its output buffers, which results in multiple routes for the input-output pair. However, it may result in packets arriving out of sequence at their destination. The performance analysis of the switch under uniform traffic pattern shows that the maximum throughput approaches 100% as the size of interstage buffers is increased.

2.3 Internally Nonblocking Switch

Nonblocking Switches have been of great interest for fast packet switching. The nonblocking switches have better performance than the internally blocking switches. Although, nonblocking switches do not suffer from internal blocking, they may have output blocking. As explained earlier, this depends upon the position of the buffers in the switch. A crossbar switch is the simplest type of a nonblocking switch.

2.3.1 Crossbar Switches

Crossbar switches have been considered as a base for packet switching, although, they were originally introduced for circuit switching. An $N \times N$ crossbar switch consists of a square array of N^2 crosspoint switches, one for each input-output pair (Fig. 2.7). In general, a crossbar has m inputs, n outputs and mn crosspoints. By making a contact at the $(i,j)^{th}$ crosspoint, we establish a physical connection between input line i and output line j. It is possible to connect N pairs of input-output lines simultaneously, given that these pairs are disjoint. Thus, the crossbar switch is characterized as internally nonblocking. Furthermore, for routing a packet, no prior knowledge of the destination of the packet is required. For example, for routing a packet from input i to output j, the packet propagates at input line i, until it reaches crosspoint (i,j), where the address

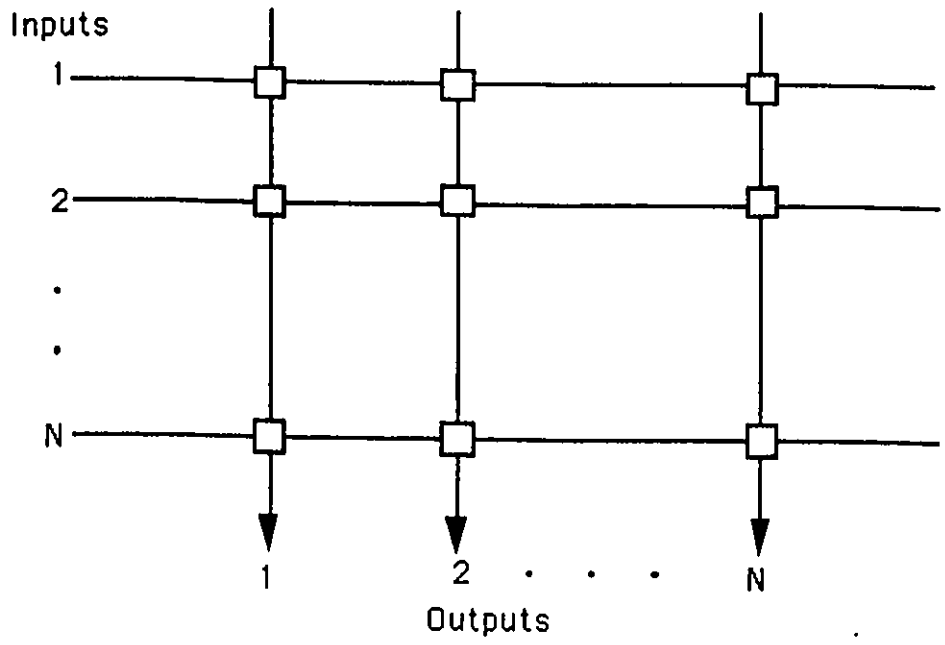


Figure 2.7: Crossbar Switching Fabric

filter on recognizing (decoding) its address closes the crosspoint. In a similar manner, the corresponding setting of the crosspoints is done by each packet individually, using the output address in their headers, which makes the crossbar switch as self-routing. Of course, the packet after closing the crosspoint (i, j) locks all other crosspoints for the output j, otherwise the packets from different inputs would be scrambled. As noted before, an $N \times N$ crossbar requires N^2 crosspoints, i. e., the hardware complexity has the square growth and, therefore the crossbar is not suitable for large size switches.

If in a time slot, more than one packet is destined for the same output, i. e., the output port contention occurs, then only one of the contending packets can be routed to the output and the remaining packets will have to be buffered. In principle, there are three possibilities for the location of buffers in a crossbar switch: (a) at the input of the crossbar matrices, (b) at the crosspoints of the switch, or (c) at the output of the matrices. We will discuss in a little detail about these buffering schemes.

(a) Input Buffering: There is a separate queue at each input to the switch (Fig. 2.8). The switching function is to select a packet out of the packets having the

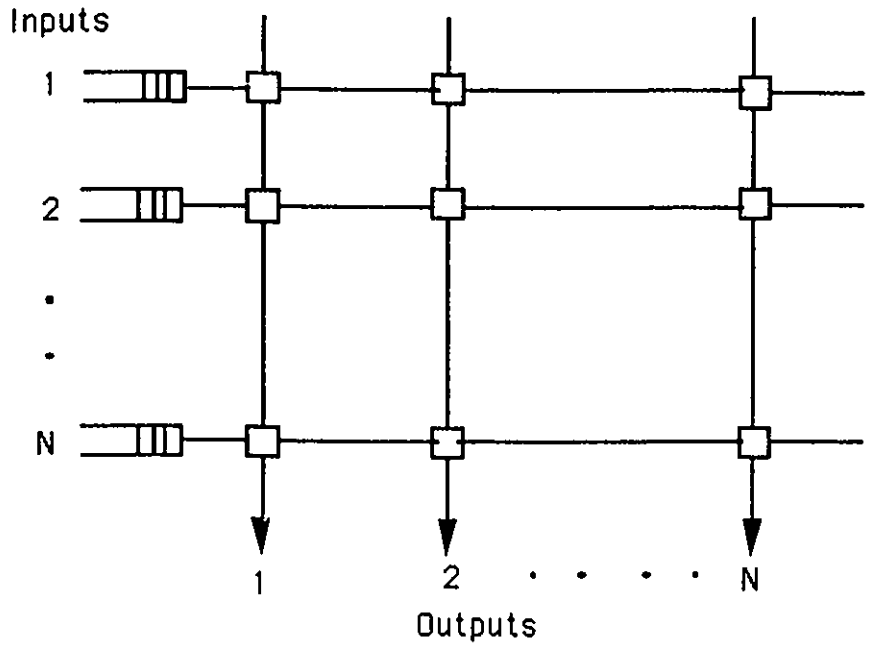


Figure 2.8: Crossbar Switch Fabric with Input Buffering

same destination and waiting at the heads of different input queues.

An implementation of such a switch, which has a distributed control, is shown in (Fig. 2.9) [8]. To resolve the output port contention, a separate arbiter is provided for each output port. This arbiter allows only one of the input ports to be switched through at a time to the corresponding output port by a fair algorithm. This is accomplished by a separate control to each input queue which stops all but one of those queues that compete for the same output port. This proposed switch does not require separate request lines from the input queues to the arbiters because it has the address decoder function distributed in the crosspoints of the matrix.

The performance of an input buffered crossbar switch depends upon the service discipline used in selecting packets from input queues and the degree up to which the output port contention can be resolved. The simplest discipline from control and implementation point of view is the first-come-first-serve (FCFS) in which only the head of line (HOL) packets of each input queue contend in a time slot. The maximum achievable throughput for such a switch, in the liniting case of $N = \infty$, is 0.586 [16]. To increase the throughput

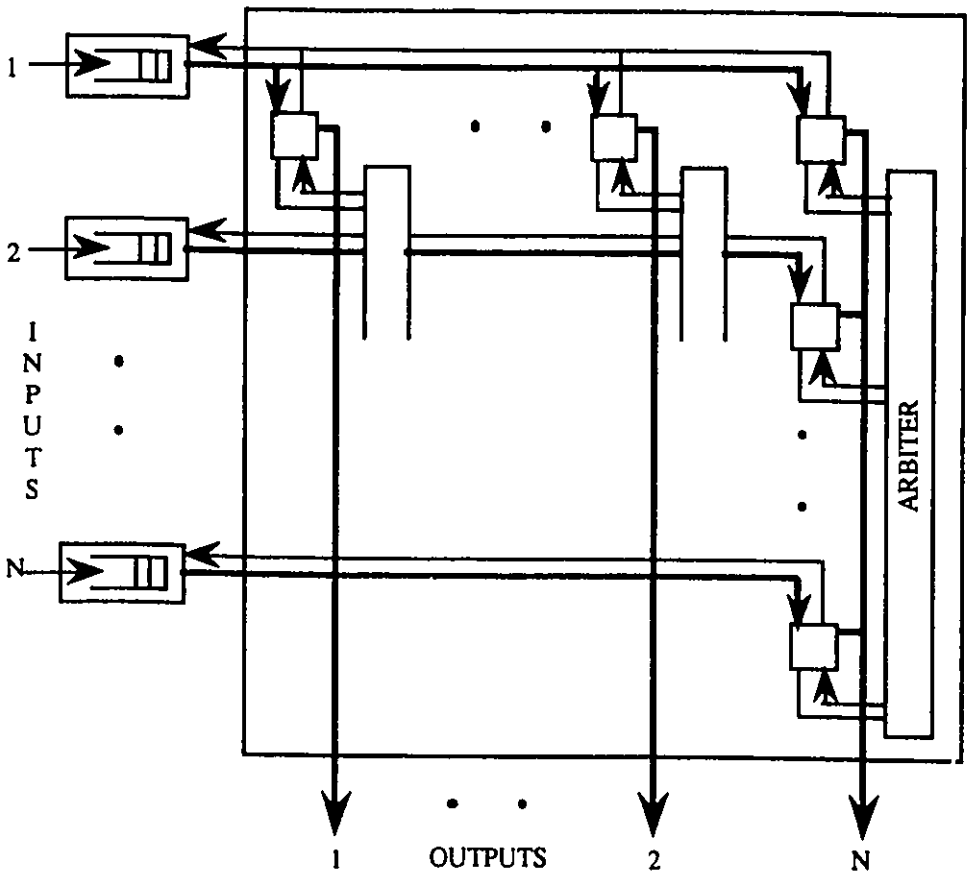


Figure 2.9: Crossbar Switch Arbiter

beyond 0.586, in a time slot, we can allow packets behind the HOL position to contend for the switching array. This is referred to as look-ahead contention resolution or bypass queueing. The performance of input buffered nonblocking switch has been discussed in detail in the next chapter.

(b) Buffering at the Crosspoints: A second class of crossbar switches has been proposed, which provides the queueing function at the crosspoints themselves (Fig. 2.10). There is a FIFO queue preceded by an address filter (AF) at each crosspoint of the switch. Packets from the inputs are sent directly into the matrix. A packet can only pass through the filter, if its address matches the destination address. The queues on a matrix column which belong to one specific output port have to be served in a fair way, e. g., in a round-robin fashion.

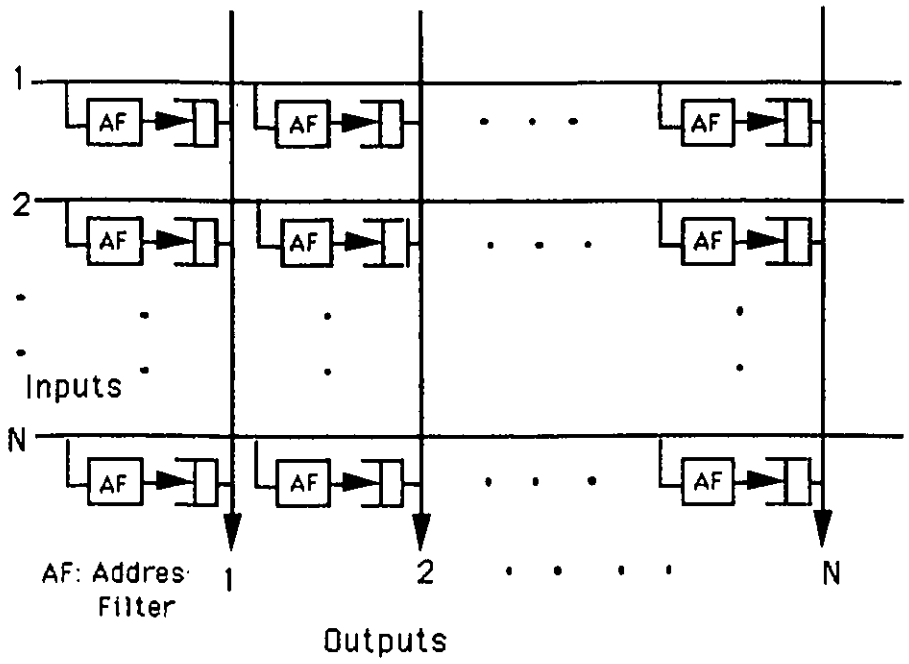


Figure 2.10: Buffering at Crosspoints in a Crossbar Switch

Placing the buffers at the crosspoints is similar to achieving output queueing, with the difference that the queue for each output is distributed over N buffers. In such an arrangement, switch fabric speed can be equal to the speed of input-output port. However, as mentioned in [1], there are two drawbacks to this approach. First, the total memory required for a given packet loss rate is greater than that required for output queueing with complete partitioning, because the output queue is distributed over N buffers and there can be no sharing among these. The second drawback from a hardware layout point of view is that, combining the buffer memory within the switching fabric would severely limit the size of the fabric implementable on a single chip. The "Bus Matrix Switch" (BMX) described in [17] is an example of such an architecture. BMX has been realized using bus width of 8 bits. With ECL technology, the bus clock can be up to 10 ns and, therefore, bus capacity of 800 Mbits/s can be achieved. Each crosspoint has an address filter that selects only packets with corresponding output port address. The packets are then stored in the crosspoint memory (XPM). The XPM can be realized using a FIFO buffer. If priority control is required, a random access memory can be used. The design

of the switch was restricted to a 2×2 matrix, realizable on a single LSI chip due to hardware limitations. A crossbar switch with FIFO buffers at each crosspoint was also discussed in [18] under the name of Butterfly Switch. In a recent paper, Kato et al. [19] described the implementation details of a crossbar switch having a dual port RAM of 16 cells (packets) at each crosspoint.

(a) Output Buffering: An $N \times N$ output buffered crossbar switch requires the switch fabric to run N times faster than the speed of input-output port. To reduce the required speed-up, parallel links can be used to carry multiple bits of a packet. Another way to introduce parallelism is to provide N links per output port. This leads to an expanded $N \times N^2$ switch matrix. Due to these constraints, only a small size switch with output buffering can be realized. Although there is no output blocking in the switch, output port contention occurs because of the statistical nature of the packet arrivals. In a time slot, there may be multiple arrivals (up to N) to an output port. Buffers at the output ports are required to receive multiple packets in a time slot. The buffer at the output ports can be organized in several different ways, as discussed in the next chapter. Mostly, either the buffer is completely shared among all the output ports or there is a separate queue at each output of the switch as shown in Figure 2.11. The output buffered switch has the best throughput-delay performance.

Recently, some other variants of crossbar switches have been suggested. The Phoenix switch, which is based on 2×2 crosspoint buffered switching elements, is being developed at AT&T Bell Labs [20], [21]. For implementing multistage interconnection networks, the switching elements are connected using a Banyan network. The switch fabric runs at speed three times the speed of the input-output port. Buffers are also provided at input and at output ports of the Banyan network.

A crossbar switch which has a combination of FIFO crosspoint buffering and output buffering was proposed in [22]. A speed-up factor of three has been suggested to reduce the amount of buffers at each crosspoint.

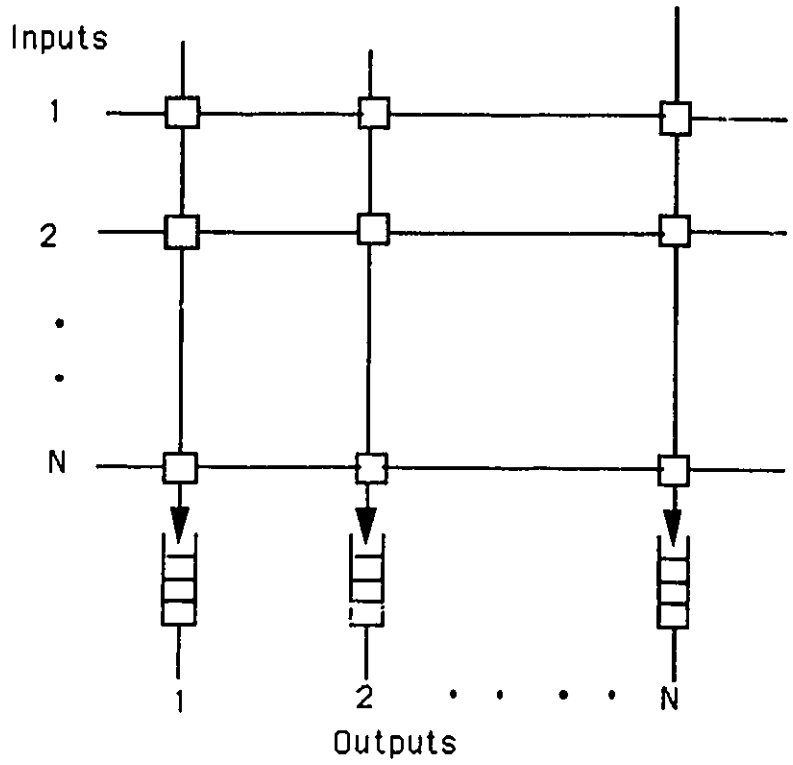


Figure 2.11: Crossbar Switch Fabric with Output Buffering

2.3.2 Batcher-Banyan Networks

Apart from an $N \times N$ crossbar switch, a Batcher-Banyan network can also be used for constructing an internally nonblocking and self-routing packet switch. Banyan networks are internally blocking, i. e., two packets destined for two different outputs may compete for the same internal link and thus collide in the switch fabric. However, if the connection requests at the inputs of a banyan network are compact (i. e., at input ports the packets are placed consecutively without any gaps) and if their destinations are in strictly increasing order, the Batcher-Banyan network becomes nonblocking.

Therefore, a method of building an internally nonblocking switch is to first sort the packets in an ascending order by a Batcher network and then feed the output of the sorting network to a Banyan network (Fig. 2.12). The Batcher-Banyan network is internally nonblocking, provided there is no output conflict among the input packets.

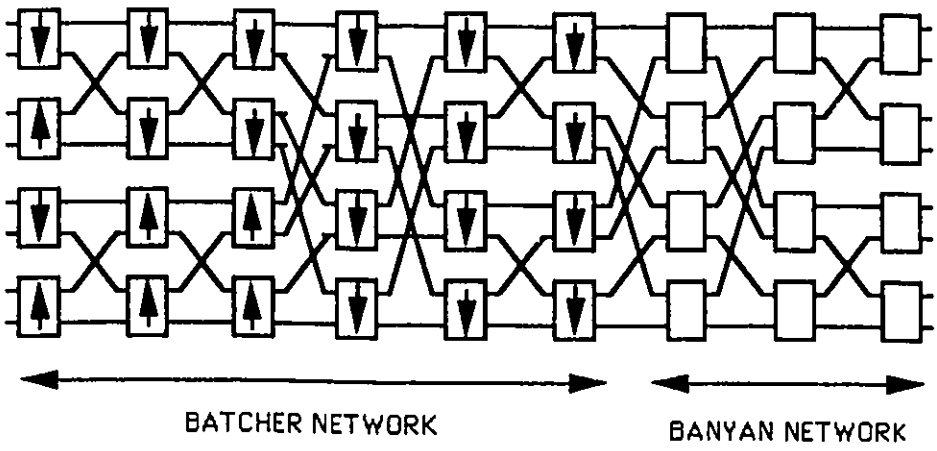


Figure 2.12: The Batcher-Banyan Network

Packets with the same destination address will be placed next to each other after being sorted by the Batcher-Banyan network. Then, if we purge all packets except one per output, the packets at the inputs of the Banyan routing network will not be compact, rendering the Batcher-Banyan network into a blocking switch. A number of methods have been proposed to overcome this problem in the Batcher-Banyan network.

(a) Starlite Switch

The very first switch fabric implemented using sort-Banyan structure is the Starlite switch [23] (Fig. 2.13). To overcome the output port contention problem, the Starlite approach uses a trap network between the sort and the Banyan networks. The trap network detects the packets having the same destinations and the packets with repeated addresses are separated from the ones with distinct addresses. The packets with repeated addresses are recycled back into the sorting network and they again contend in the next slot. A buffering stage may be provided to store recycled packets, in order to reduce packet loss. To maintain the packet sequence integrity, the packets are given priority based on their age.

The Starlite approach has a few drawbacks. First, it requires a sort network of larger size, as half of the input ports are dedicated for reentry. Second, it requires extra hardware to implement the trap and skew network. Hui [24] proposed a switch architecture based on the Batcher-Banyan network using a three-phase algorithm to overcome the output

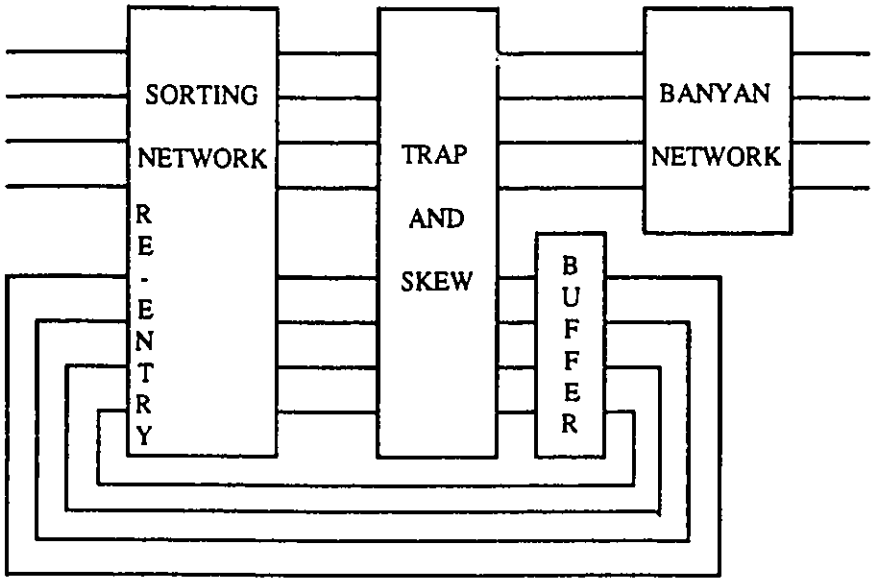


Figure 2.13: Starlite Switch Fabric

port contention problem.

(b) Three-Phase Batcher-Banyan Network

The three-phase algorithm can be used to resolve output port contention in a Batcher-Banyan network [24]. In the first phase (called arbitration phase), each input port i sends a short request packet, which is just a source-destination pair (i, j_i). The requests are sorted by the Batcher network and one request per output port is selected. In the second phase, the request packets (i, j_i) which won the arbitration send an acknowledgment for their request, and then transmit the full packet in the final phase, through the same Batcher-Banyan network, without any conflict for the output ports. The input ports which fail to receive an acknowledgment retain the packet in the input buffer for a retry in the next slot. The buffering at the input ports causes the head of line blocking and, thereby, reduces the throughput.

The first two phases constitute overhead processing and therefore, the switch fabric would have to be speeded up by a fraction which depends upon the switch size. The performance of this switch is the same as that of an input queueing switch.

(c) Sunshine Switch

Sunshine combines a Batcher sorting network with parallel Banyan routing networks [25]. The k parallel Banyan networks provide k independent paths for packets to access the output queues. If during a time slot more than k packets request a particular output port, then the excess packets overflow into a shared recirculating queue to be resubmitted to the switch in the next time slot. This queue consists of T parallel loops and the T dedicated inputs on the Batcher sorting network, each providing access for one packet.

The packet loss can occure by the overflow from the output buffer and the circulating queue. The performance of the Sunshine depends upon k and T. For a relatively large value of T, the performance is very close to the speed-up switch which is discussed in detail in Section 3.5.

2.3.3 ATOM Switch

The ATOM switch [26] is an output queueing switch based on time division multiplexed bus, as shown in Figure 2.14. Information cells on input ports are converted to a parallel form and transferred to the high speed data bus in a time division multiplexed manner. On the output port side, each address filter (AF) detects the physical address which represents port destination address of the cell, and receives the cells destined for the outgoing port. The received cells are stored in buffer memories and read out on a first-in-first-out basis. The data bus and buffer memory speed is N times faster than the speed of the input/output port. This speed-up is reduced by using multiple planes in parallel. To construct large size switches, a multistage interconnection has been proposed by the authors. They have also proposed modifications to the above switch architecture to accommodate multicast functionality.

The performance of the ATOM switch is the same as that of an output buffered switch.

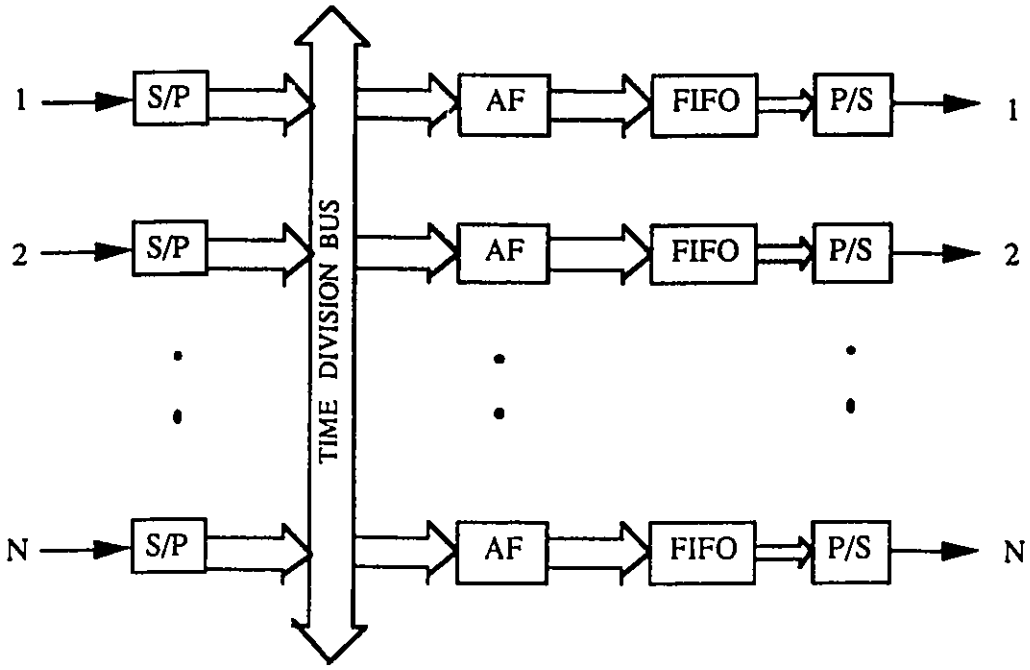


Figure 2.14: ATOM Shared-bus Architecture

2.3.4 Knockout Switch

Another switch fabric proposed in the literature, which supports the output buffering, is the knockout switch [27]. In the knockout switch, each input port has a broadcast bus to which all output ports are tapped (Figure 2.15), through a bus interface. This bus interface has three major components. The first component is the set of N address filers, one for each input line, which selects packets addressed to the corresponding output port and discards the other packets. This operation is performed in parallel and thus each bus interface is capable of receiving N packets simultaneously. The second component is the concentrator, which selects up to a fixed number of packets, say L, out of those accepted by the filters. If more than L packets are destined to the same output port in a given slot, only L are selected from the incoming packets. The selected packets are stored into a shared buffer, which constitutes the third component of the bus interface. The logic behind the N to L concentration is that a small value of L in comparison to N is required to keep the packet loss below a threshold value. For example in a large size switch, at 80%

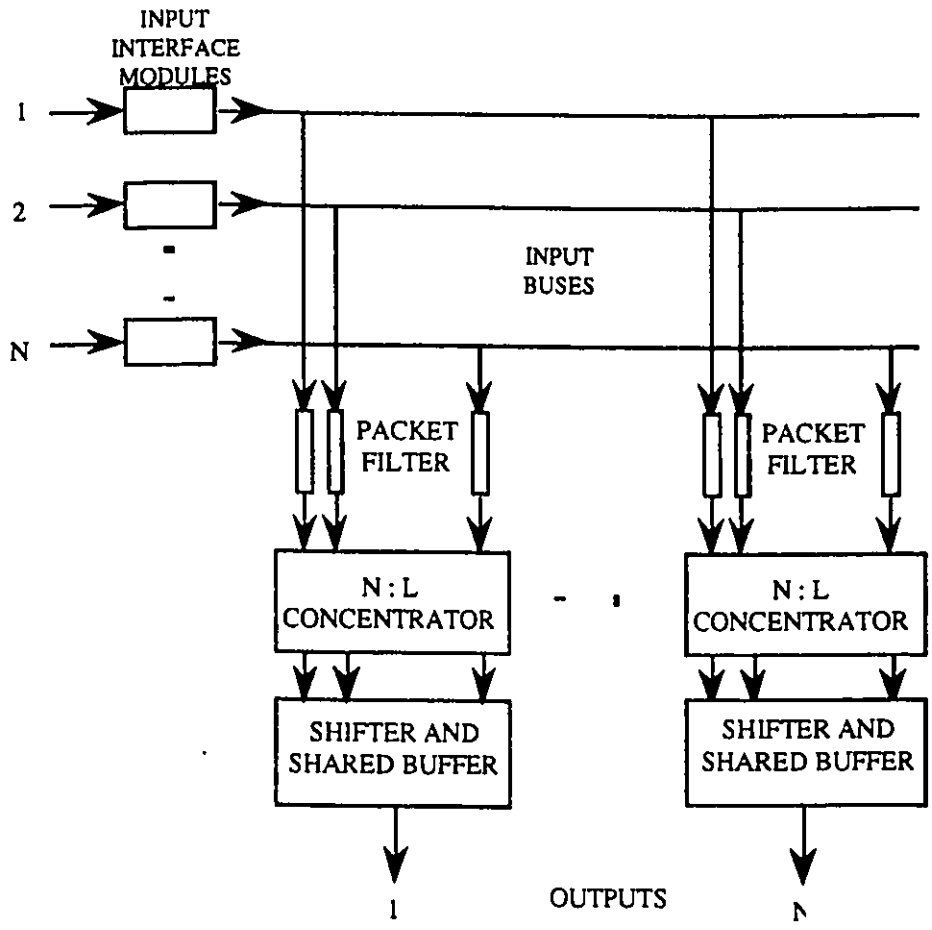


Figure 2.15: Knockout Switch Architecture

input load, a packet loss rate of less than 10^{-6} can be achieved by providing $L = 8$. If the traffic pattern is nonuniform, L has to be higher for the same packet loss rate, depending on the non-uniformity of the traffic [28].

Recently, many switch architectures based on the knockout principle have been proposed to build large size ATM switches in modular fashion. We discuss here some of these architectures.

(a) Growable Packet Switch: The Growable Switch architecture [29] is depicted in Figure 2.16, where a partition is made between a front-end cell distribution network and a column of output switch modules. The distribution network is assumed to be a memoryless network and its function is to route the incoming cells instantaneously to the

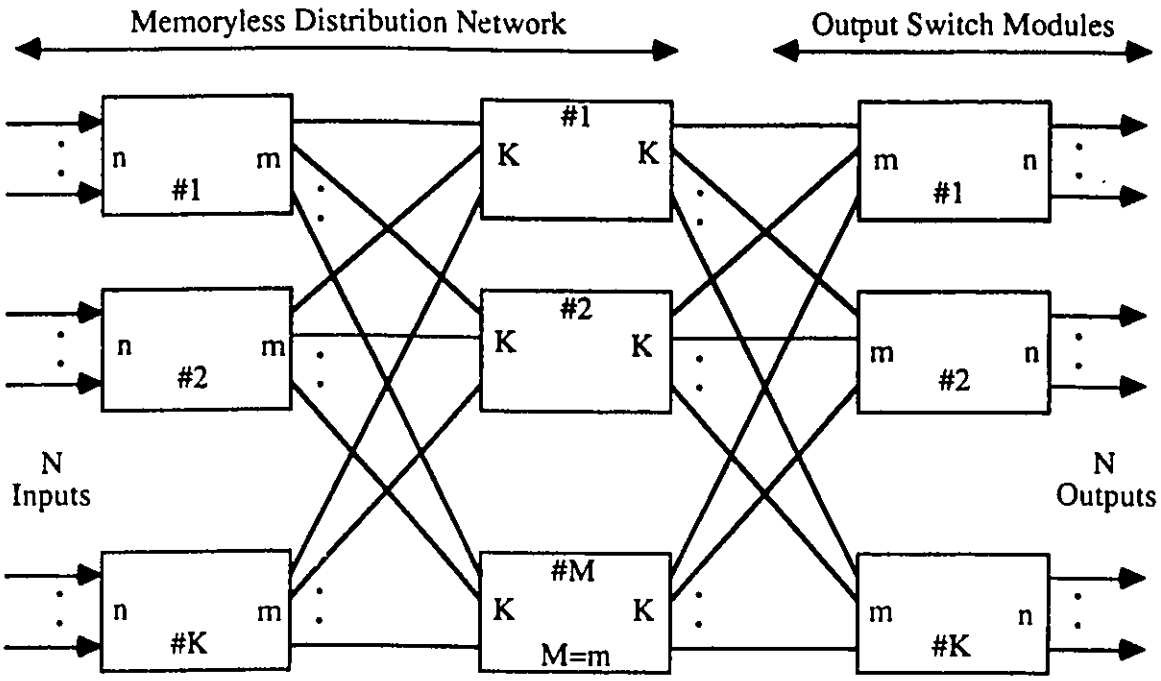


Figure 2.16: Growable Switch Architecture

output switch modules according to the destination addresses of the incoming cells. To achieve the best delay performance, buffers are provided only at the outputs of the output switch modules. The outputs are divided into groups of n lines each. Each output switch is of size $m \times n$ $(m > n)$, where m and n are chosen according to the tolerable cell loss rate, and the considerations involved in the implementation of the switch module. In an $N \times N$ packet switch, up to N cells can arrive simultaneously for a particular output group, but only a maximum of m cells can enter a given output switch module in a time slot. The rest of the cells are knocked-out in the distribution network.

The cell distribution network is of the size $N \times (\frac{m}{n})N$. Implementation of the distribution network and the output switch module have been proposed in [30]. An optical star-coupler based arrangement for the cell distribution network was later proposed in [31].

(b) Switch with Shared Concentration and Output Queueing(SCOQ): An $N \times N$ SCOQ switch [32], [33] is composed of a sorting network and L identical switching modules(Figure 2.17). The sorting network (SN) is an $N \times N$ Batcher bitonic network which sorts the incoming packets according to their destination addresses. The

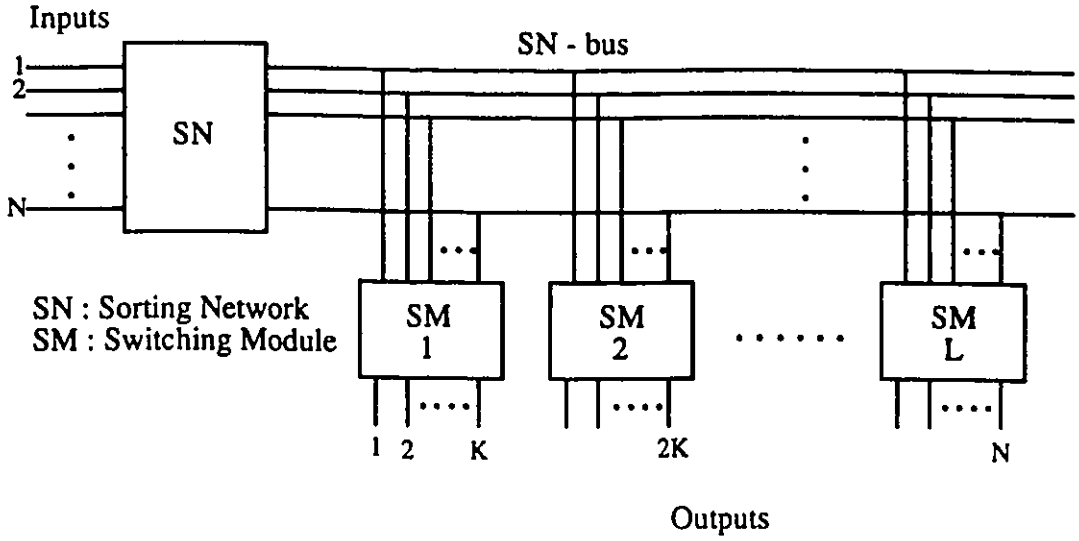


Figure 2.17: Switch with Shared Concentration and Output Queueing(SCOQ)

switching modules are all connected on the SN bus. With an identical structure, each of the switching modules concentrates and routes packets to a group of $K = N/L$ outputs. Each of the L switching modules has N input ports and K output ports and consists of three components: a bank of N packet filters, a routing network of L Banyan $K \times K$ switching elements, and K output buffers.

The concentration is performed inside the switch module in such a way that at most L simultaneously arriving packets will be routed to each output buffer. The SCOQ switch is not suitable for large size switches as the complexity of the switch modules is dependent on the switch size.

Some other Modular structures based on the knockout principle can be found in [34], [35].

2.3.5 Shared-Memory Switches

The switches in this category consist of a common memory shared by all the input and output lines. Packets arriving on all the input lines are fed to a common buffer for storage; internal to the memory, packets are organized into separate output queues, one for each output line. The output ports read out the packets from the memory according to the

control signal given by a centralized control. In this type of architecture, two main design constraints must be satisfied [1]. First, the processing time required to determine where to enqueue the packets and to issue the proper control signals should be sufficiently small to keep up with the flow of incoming packets. The second and most important design constraint pertains to the shared memory. The memory bandwidth should be sufficiently large to accommodate simultaneously all input and output traffic. For these reasons such a switch is limited to a small size. Shared-memory switches utilize buffers very efficiently and are least sensitive to an unbalanced and bursty traffic. Now, we describe some of the shared memory switches.

(a) Prelude Switch: The Prelude switch, developed at the Centre National d'Etudes des Télécommunications (CNET) in France, uses a common memory for all input-output ports [36] (Figure 2.18). The underlying design principle of the Prelude switch is quite similar to a Time Division Multiplex switch, which switches the position of packets through reading and writing of packets from a common memory. The Prelude switch considers a packet size of 16 bytes: 15 bytes of user data and one byte header. The design of the Prelude switch assumes the switch size N to be equal to the number of bytes per packet. First, packets pass through the serial-to-parallel (S/P) conversion stage and then packets on all input lines enter a clock adaptation and phase alignment stage. The packets are aligned in such a way that the packets are time-shifted by one byte from one input line to the next. These diagonally aligned packets are next fed to a rotative space division switch, which delivers the header of each packet on to the first output of the space division switch, and the subsequent 15 bytes of each packet are switched sequentially to the remaining 15 output lines, each byte being offset in time with respect to the previous one, by one byte. This way the headers from all packets are routed to the controller for processing and the packets get stored in the memory in parallel. The controller processes the headers, determining the output line on which to transmit the packets. At the output port, a packet is reconstructed by demultiplexing and converting from parallel to serial.

The throughput and delay performance of the Prelude switch is the same as that of an output buffered switch. The packet loss rate for a given amount of buffer is the

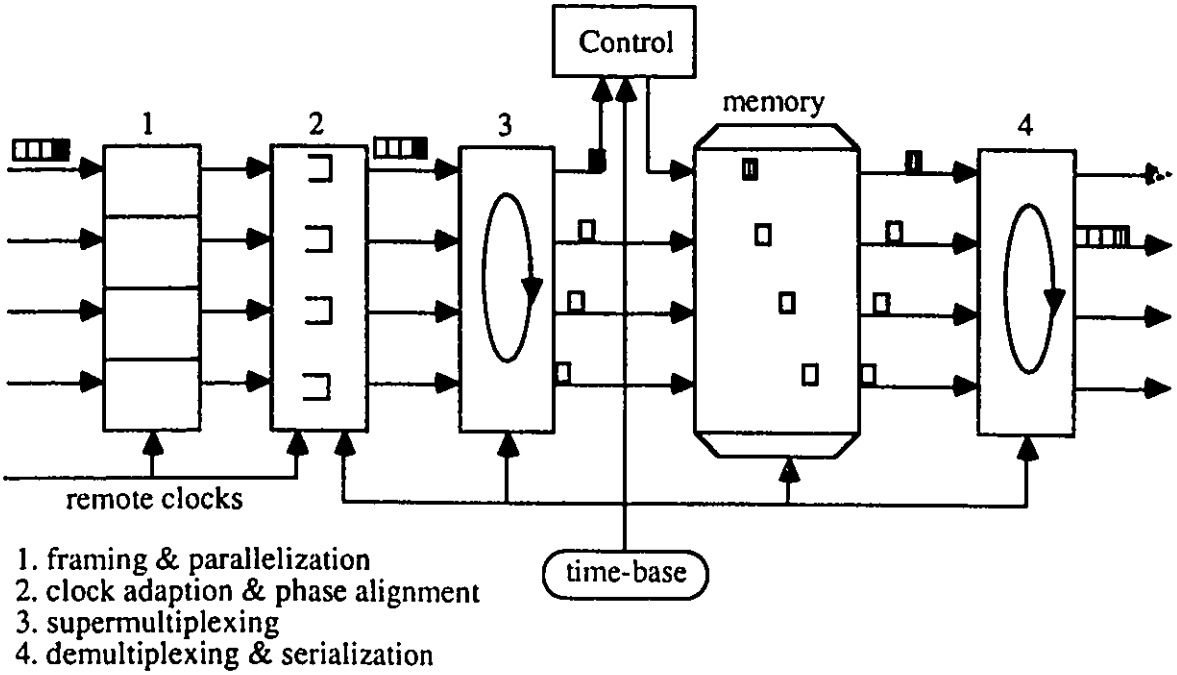


Figure 2.18: Prelude Switch Architecture

same as that of a completely shared output buffered switch, which is discussed in the next chapter.

(b) HITACHI's Shared Memory Switch: Another example of the shared-memory switch is a switch proposed by Kuwahara et al. [37]. In this architecture, the memory is entirely shared by all the output ports. The output queues are organized in the form of linked lists.

The Input/Output LSI chips convert the incoming cells from serial to parallel data format. Then the header converter LSI chips process the virtual circuit number in each packet. The packets are then written one by one into the buffer according to the information provided by the header converter which determines which linked list to put it on. At this time all appropriate pointers are updated. An Idle Address Buffer keeps track of all the empty buffer space. Similarly, at each time slot, one packet from each linked list is identified, and retrieved for transmission. Simultaneously, the pointers and the contents of the idle address buffer are updated. The packets are then converted to a serial data format from a parallel data format and transmitted on to the output lines.

It is possible to accommodate multicast and priority functions in such a switch structure.

2.4 Summary

We have presented an overview of the fast packet switch architectures. Some of these switches have been experimentally developed in the last few years. In general, the performance of the internally blocking switches is inadequate for fast packet switching. To enhance the performance many different types of Banyan networks have been proposed in the literature. Although we presented only two variants of the Banyan networks (buffered Banyan, and Banyan networks with internal speed-ups) here, some other variations also have been of interest to the researchers.

Internally nonblocking switch architectures have been of main interest, especially to the telecommunication industry. Switches based on a crossbar switch fabric have been developed at Fujitsu Laboratories and NTT Laboratories. Modular switch architectures based on the knockout principle also have been of large interest in the recent past. From the performance point of view, the switch structures employing output shared buffer would be the natural choice, but, from the complexity point of view, they may be hard to implement especially for a switch of large size.

Some switch architectures have been proposed to support broadcasting and multicasting connections using fast packet switching principles (e. g. [38], [39]), which we have not covered here.

Chapter 3

Review of Fast Packet Switch Performance

3.1 Introduction

In this chapter, we will review the performance of the fast packet switches. We will focus only on the performance of internally nonblocking switches. As explained earlier, the performance of a nonblocking switch is governed by its ability to cope with output blocking and head-of-line (HOL) blocking. For example, an input buffered nonblocking switch suffers most severely from the output blocking and HOL blocking, and as one would expect, its performance is considered to be inadequate for fast packet switching. On the other hand, an output buffered switch does not suffer from output and HOL blockings, and hence, gives the best throughput-delay performance. In the discussion of the performance, we do not assume any particular switch architecture, as different architectures will yield the same performance, as long as the position of the buffers and the manner in which they cope with output and HOL blockings are the same.

Uniform random traffic has been assumed in many analyses for the study of the performance of different switch architectures. Under uniform random traffic, packets arrive at each input according to independent Bernoulli process and they are uniformly distributed among all output ports. Successive packet arrivals are independently destined

from those of previous arrivals. Recently, non-uniform and bursty traffic has also been considered to study the behaviour of different switch architectures. In such cases, the performance under uniform random traffic serves as the basis for the comparison and it facilitates the measurement of deviation in performance under non-uniform and bursty traffic. Unless mentioned, the following discussion applies to the performance under uniform random traffic.

3.2 Performance of Input Buffered Packet Switch

The performance of an input buffered nonblocking packet switch is mainly restricted by the output blocking. The HOL blocking renders the switch non-work-conserving. To alleviate the effect of blocking, many schemes have been suggested in the literature. In this section, first we discuss the performance of an input buffered switch, and then we discuss some of these schemes.

The simplest discipline from a control and implementation point of view is First-Come-First-Served (FCFS) in which only the HOL packet of each input queue may contend to the switching array. If a HOL packet wins the contention, then it is removed from the HOL position, and the next in line moves to the HOL position. The packets, which lose in the contention, remain at the HOL positions and contend again in the next time slot. It has been shown that the maximum achievable throughput decreases with switch size N and reaches a limit rapidly as $N \rightarrow \infty$ [40]. Table 3.1 and Figure 3.1 give the maximum throughput under the assumptions that the input queues are saturated and they are served in FCFS order. For a 2×2 switch (a building block of Banyan network), the maximum throughput is 0.75. In the limiting case of $N = \infty$, the maximum throughput has been shown to be $2 - \sqrt{2} = 0.5858$ [16] [24].

For the case of $N = \infty$, Hui and Arthurs[24] obtained an upper bound on the packet loss rate (P_{loss}) at input buffers, under FCFS discipline. It is given by,

$$P_{loss} < \frac{\rho(2-\rho)}{2(1-\rho)} \quad \frac{\rho^2}{2(1-\rho)^2} \quad b_i$$

N	Maximum Throughput
2	0.75
3	0.6825
4	0.6553
5	0.6399
6	0.6302
7	0.6234
8	0.6184
⋮	⋮
∞	0.5858

Table 3.1: Maximum Throughput of an Input Buffered Switch with FCFS Discipline

where ρ is the average amount of load per input-output port and b_i is the size of buffer per input port. Figure 3.2 shows the packet loss as a function of load for various values of buffer size in an infinitely large switch [24]. With 20 buffers per input line, the loss rate below 10^{-6} can be achieved up to 50% input load.

San-qi Li [41]-[44] has extensively studied the performance of an input buffered switch under non-uniform and correlated traffic. In [41], he has considered input imbalanced and output imbalanced traffic. If all inputs (outputs) do not have the same load, then it is referred to as input (output) imbalanced traffic. Examination of a large switch ($N \rightarrow \infty$) under the input imbalanced traffic, where 20% of the input ports (first group of input ports) have unity load and the rest of the inputs (second group of input ports) have only 50% load, shows that the throughput per output port is 0.438 (which corresponds to throughput of 0.73 per input port of the first group and that of 0.365 per input port of the second group). As the size of the first group is increased, the maximum throughput per output port increases, and approaches to 0.586 as the first group size approaches to N. Similarly, in the case of output imbalanced traffic significant reduction in throughput is observed.

Now, we discuss some of the techniques discussed in the literature to enhance the performance of an input buffered switch.

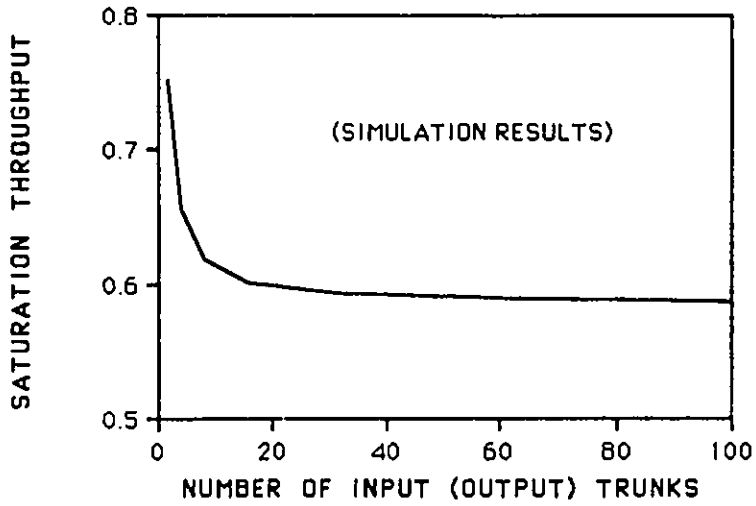


Figure 3.1: Maximum Throughput of an Input Buffered Switch with FCFS Discipline

(a) Nonblocking Switch without Input Buffers: A higher switch throughput can be achieved by dropping packets [16]. In this case, whenever k packets are addressed for a particular output in a time-slot, only one is transmitted over the output trunk and the remaining $k-1$ packets are just dropped from the switch, i. e. the input queues are eliminated. In an $N = \infty$ switch, the probability that a packet wins the output port contention is given by $(1 - e^{-\rho})/\rho$, which represents the throughput of the switch at a given load ρ. When the input load is more than 0.882, throughput of more than 0.586 is achieved. At $\rho = 1.0$, the maximum throughput is about 0.632. But this increase in throughput is achieved at the expense of higher packet loss rate which is about 36.8% at unity input load. Such a high packet loss rate cannot be tolerated in high speed switching.

(b) Look-ahead Contention Resolution: A higher switch throughput can also be achieved by relaxing the strict FCFS discipline at the input buffers [40]. Each input still sends at most one packet into the switch fabric per time slot, but not necessarily the first packet in its queue, and no more than one packet is allowed to pass through the switch fabric to each output. Consider a scenario, where at the beginning of each time slot, up to the first w packets (called 'window size') in each input queue sequentially contend for access to the switch outputs. The packets at the heads of the input queues contend first for access to the switch outputs. Those inputs not selected to transmit the first packets

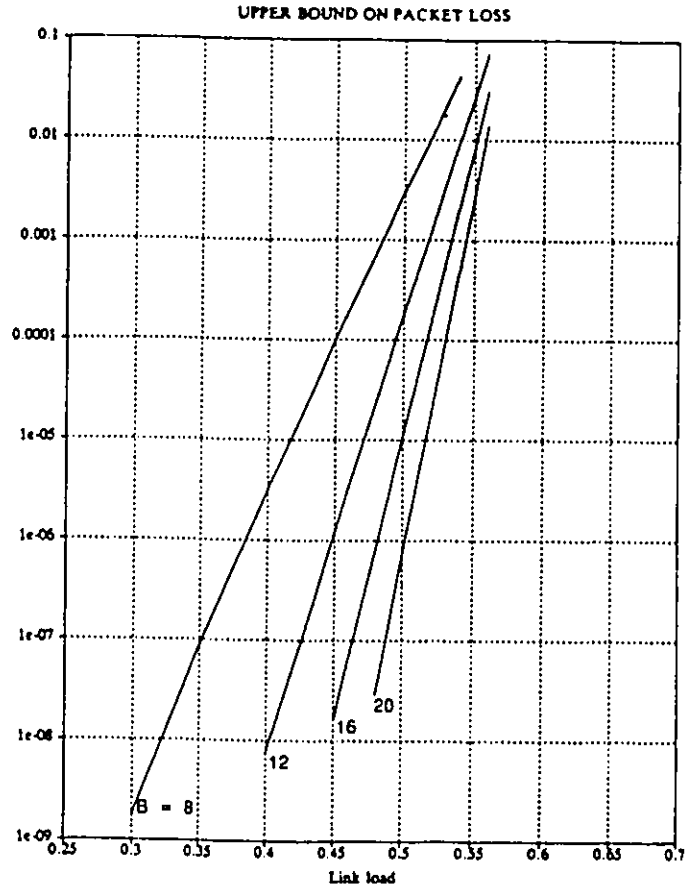


Figure 3.2: Packet Loss Rate in an Input Buffered Switch with FCFS Discipline

in their input queues then contend . . . their second packets for access to any remaining idle outputs, and so on. This scheme is referred to as window selection policy [40], look-ahead contention resolution [45], and input queue by-pass [46]. A window size of $w = 1$ corresponds to the input queueing with FCFS buffers. Simulation results have shown that the throughput increases with increasing values of w, but with diminishing improvements beyond $w = 4$. Table 3.2 shows the maximum throughput for various switch and window sizes [40]. This table shows that the look-ahead contention resolution is most effective when N is small.

An approximate analysis of maximum throughput as a function of window size has been carried out in [47] for an infinitely large size switch. In [48], an exact analysis of

	Window Size, w							
N	1	2	3	4	5	6	7	8
2	0.75	0.84	0.89	0.92	0.93	0.94	0.95	0.96
4	0.66	0.76	0.81	0.85	0.87	0.89	0.91	0.92
8	0.62	0.72	0.78	0.82	0.85	0.87	0.88	0.89
16	0.60	0.71	0.77	0.81	0.84	0.86	0.87	0.88
32	0.59	0.70	0.76	0.80	0.83	0.85	0.87	0.88
64	0.59	0.70	0.76	0.80	0.83	0.85	0.86	0.88
128	0.59	0.70	0.76	0.80	0.83	0.85	0.86	0.88

Table 3.2: Maximum Throughput of an Input Buffered Switch with Window Selection Policy

delay and packet loss rate has been presented for the by-pass queueing discipline in a 2×2 switching element. Cell by-pass queueing has also been discussed in a buffered Banyan Network in [49] and it has been shown through simulation that the cell loss rate for a given amount of buffer space is one hundredth of that for FCFS discipline at 85% offered load. Sarkies [50] has considered a switch architecture consisting of parallel planes of internally blocking switch networks and buffers are provided at input and output ports. He has shown that the by-pass queueing in such an architecture improves the throughput of the switch and thereby reduces the number of switch planes required.

(c) Other Methods for Improving the Performance: Apart from by-pass queueing, other methods suggested for improving the performance of an input buffered switch are input port expansion, output port expansion, and switch speed-up. A comparative study of these methods has been performed in [45]. A survey of different types of speed-up switches has been presented in [51]. Here we discuss these methods in brief.

Input Port Expansion: In this scheme, each input port is expanded into s ports before packets enter an asymmetric $Ns \times N$ switch. In a time slot, up to s packets from each input queue can be presented to the switch fabric for contention. With $s = w$, one would

expect the input port expansion switch to have a higher throughput than the look-ahead scheme, because it is possible for more than one packet to be cleared from each input queue in the same time slot.

Output Port Expansion: If there are more output ports than input ports, the offered load per output port (and therefore contention among inputs for available outputs) is reduced. Basically, the output port expansion scheme is the same as the output channel-grouping. With r output ports provided for each output address, up to r packets can access any output address simultaneously. However, no more than one packet can be routed from any given input port in a time slot. Advantages of output channel grouping in Batcher-Banyan networks have been considered in [52].

Switch Speed-up: In a switch that operates at v times the input-output line speed, each external time slot is divided into v mini-slots within the switch, and the effective offered load is reduced by a factor of v. This allows up to v packets to be routed from each input queue and, similarly, each output can receive up to v packets in a time slot. In this case, a FCFS buffer is needed at each output port to temporarily store packets when multiple packets arrive at an output. It is also possible to have a switch where either the input ports have higher switching capacity or only the output ports have higher switching capacity. We can denote these switching capacities respectively by L_{input} and L_{output}. (The Switching capacity can be defined as the number of packets that can be routed in a time slot from (to) an input (output) port.) The switch with $L_{input} = 1$ and $L_{output} = L$ $(1 < L < N)$ is often called a speed-up switch [53] and it has been extensively studied in the literature. For this reason, we discuss the performance of the speed-up switch in a separate section later in this chapter.

3.3 Performance of Output Buffered Packet Switch

An output buffered switch has the best throughput-delay performance [40]. A crossbar switch with FIFO queues at each crosspoint will also have the same throughput-delay performance. Output buffers in a switch can be organized into five different ways, similar

to the schemes discussed in [54] for the storage in a store and forward computer network node. The total buffer size required, so as not to exceed a given maximum packet loss rate, depends upon the way the memory is shared. The five sharing schemes are: (a) complete partitioning, (b) complete sharing, (c) sharing with maximum queue lengths, (d) sharing with minimum allocation, and (e) sharing with a maximum queue and minimum allocation. Among these, complete partitioning and complete sharing have been studied in detail in the literature, and we summarize these results in this section.

3.3.1 Complete Partitioning

In complete partitioning, the memory is divided into N separate sections, each one allocated to a particular output port. Assuming uniform random traffic, the arrival process to each output queue has the binomial distribution. As $N \rightarrow \infty$, the arrival process becomes Poisson and each output queue behaves as an $M/D/1$ queue [16]. Mean waiting time in an output queue of an $N \times N$ switch is given by,

$$\overline{W} = \frac{N-1}{N} \ \overline{W}_{M/D/1}$$

where $\overline{W}_{M/D/1}$ represents the mean waiting time in an $M/D/1$ queue. Figure 3.3 shows the mean waiting time as a function of load per port (p) for several values of N [16].

Assuming a b packet FIFO buffer at each output, a packet destined to output i is lost, if all b buffers at the output are occupied. Figure 3.4 shows the packet loss probability as a function of the number of packet buffers allocated to each output queue [40]. The load per input port is 0.8. For a given b, we note that the loss rate increases with N because the variance of the arrival process to each queue increases with N [1]. We also note that the results obtained for $N = \infty$ represents a good approximation for finite size switch of $N \geq 32$. Figure 3.5 shows the packet loss rate in an infinitely large switch as a function of b, for different values of input load [40].

The effect of bursty traffic on the buffer requirement of the switch has been studied in [55], [56]. In [55], an infinite buffer per output port has been assumed and then using paradigm matrix-analytic methodology to deal with infinite Markov chain, the distribu-

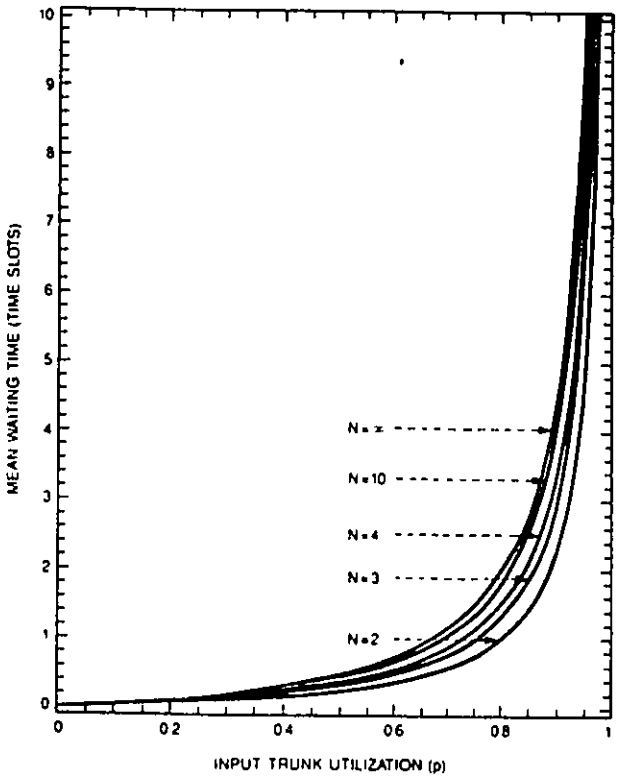


Figure 3.3: Mean Waiting Time in an Output Buffered Switch

tion of the infinite queueing model has been computed. An upper bound to the packet loss rate of the finite buffer model is achieved by summing up the tail distribution. It has been shown that the uniform random traffic assumption underestimates the buffer requirement of the switch. For a given traffic load, the required buffer size increases linearly as the average burst length of the traffic increases. An approximate analysis of the finite buffer case has been carried out in [56].

3.3.2 Complete Sharing

In place of providing a separate buffer for each output, the buffers can be pooled into one completely shared buffer. With complete sharing, we save on the total amount of buffering needed to achieve a desired packet loss rate. Figure 3.6 illustrates the packet loss rate for various values of N at 80% input load [40]. We observe that for a given

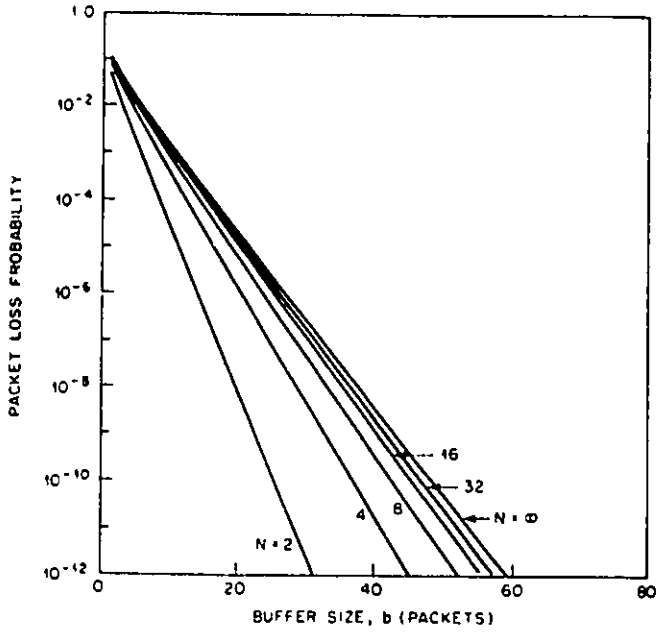


Figure 3.4: Packet Loss Rate in an Output Buffered Switch with Complete Partitioning as a Function of Buffer Size (b)

amount of buffers per output port (b), the packet loss rate decreases with increasing N. The reason is that the sharing among the output ports increases with N, resulting in more efficient use of the buffers.

In [37], the buffer reduction ratio, which is the ratio between the buffer size of the shared buffer switch and that of the complete partitioned buffer switch for the same packet loss rate, was computed. It was shown that a shared buffer switch of size 32×32 requires only 14% buffer of the complete partitioned buffer switch for packet loss rate of 10^{-9} at 80% uniform input load.

In [40], the arrival process to a given output was assumed to be independent to the arrival processes to the other outputs. But in a finite size switch, the numbers of total packets that arrive at each time-slot destined for the individual output ports are not independent, and in fact they are negatively correlated [57]. This negative correlation causes the sum of the queues for the output ports to be stochastically smaller than what this sum would be, were the queues to be independent. Eckberg and Hou [57] have

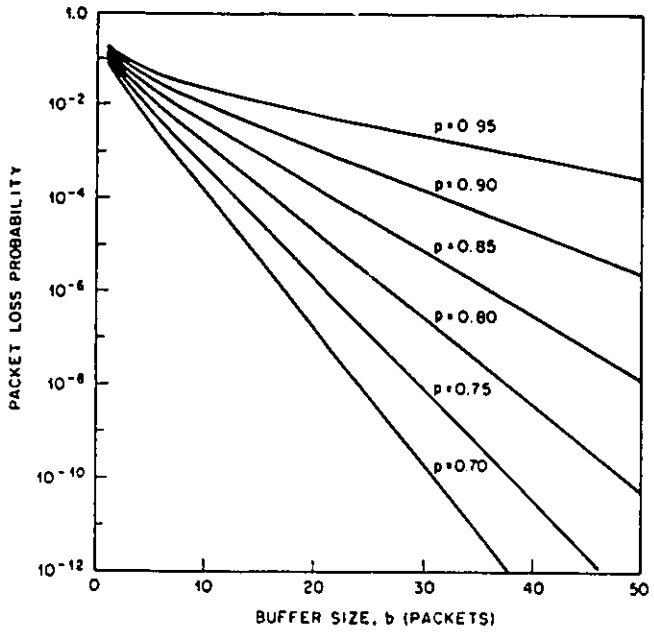


Figure 3.5: Packet Loss Rate in an Output Buffered Switch ($N = \infty$) with Complete Partitioning as a Function of Buffer Size b and Offered Load p

analyzed the effect of this negative correlation on the buffer size in a shared buffer switch, and have demonstrated that in a finite size switch, to achieve an objective level of packet loss rate due to buffer overflow, buffers would be considerably over-estimated, if the negative correlation between the queues were to be ignored. However, as N increases, the arrival process to each output becomes an independent Poisson Process and the results of [40] agree with those of Eckberg and Hou [57].

A comparative study of complete partitioned and complete shared buffered switches under uniform random traffic, imbalanced traffic, and bursty traffic has been presented in [58]. In this study, a hardware emulation method has been used in order to obtain the cell loss rate of the order of 10^{-9}. An 8×8 switch was used in the experiment. The result indicates that under imbalanced traffic, the advantage of buffer sharing is reduced and the cell loss rate at all output ports increases when the load on one particular output port is increased.

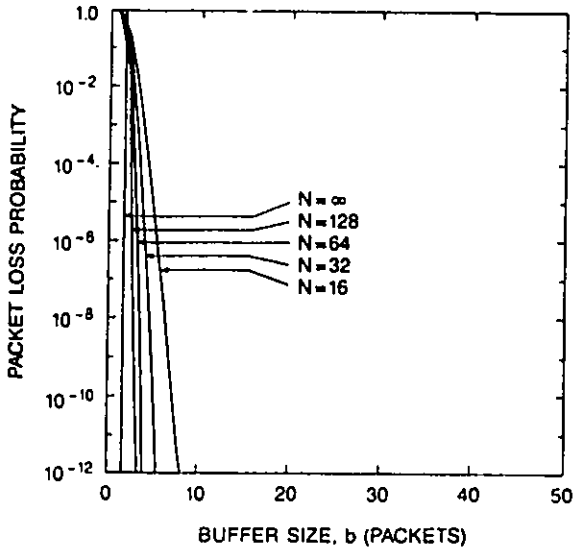


Figure 3.6: Packet Loss Rate in a Completely Shared Output Buffered Switch

3.3.3 Other Sharing Schemes

In practice, in the case of non-uniform traffic, it is likely that some output queues become momentarily heavily loaded and monopolize the use of the shared buffers, to the detriment of the other output queues. Such transient effects have not been considered in the results presented above. The appropriate memory-sharing policy must lie somewhere between the two extreme cases presented above, such as, for example, sharing with maximum queue and minimum allocation constraints for each output [1]. Recently, such a policy has been analyzed in brief in [59].

In [58] sharing with maximum queue length (SMXQ) scheme has also been studied under uniform random traffic and imbalanced traffic. It has been shown that this scheme is effective in preventing the increase in cell loss rate in the case of imbalanced traffic.

In [60], an analysis of output buffer sharing scheme has been presented, where one output port is assumed to be continuously overloaded. The buffer pool is completely shared, but in addition to that a new buffer management policy called "drop on demand" has been applied. Under this policy, if an arriving packet for port i ($i = 1, \ldots, N$) finds the buffer full and port j has more packets in the buffer than any other port, the following

action is taken: if $i = j$, the arriving packet is dropped; if $i \neq j$, the arriving packet joins the buffer and one of the packets of port j is purged. It has been shown that the drop on demand policy yields a greater switch throughput and lower packet loss probability.

3.4 Performance of Knockout Switch

A knockout switch is an output buffered switch with a variation that the arriving packets are passed through an N to L concentrator before storing them in the output buffer. If there are k packets arriving in a time slot for a given output, all packets will pass through the concentrator, when $k \leq L$. If $k > L$, then only L packets will pass through the concentrator and $k - L$ packets will be dropped within the concentrator. The value of L is so chosen that the packet loss rate is below a tolerable limit. Under uniform random traffic, the packet loss rate in given by [27],

$$Pr[packet\ \ loss] = \frac{1}{\rho} \sum_{k=L+1}^{N} (k - L) \begin{pmatrix} N \\ k \end{pmatrix} \left(\frac{\rho}{N}\right)^k \left(1 - \frac{\rho}{N}\right)^{N-k}$$

where ρ is the load per input port.

Figure 3.7 shows the packet loss rate as a function of L at 90% input load, for $N = 16, 32, 64$, and infinity [27]. It is to be noted that a concentrator with only eight outputs achieves a probability of lost packet less than 10^{-6} for arbitrarily large N. Each additional output added to the concentrator beyond eight results in an order of magnitude decrease in the lost packet probability.

3.5 Performance of Speed-up Switch

To improve the performance of an input buffered switch, the switching capacity of output ports can be increased to L $(1 < L < N)$, i. e., up to L packets can be routed to an output port in a time slot. However, in a time slot, no more than one packet can be routed from any given input port. Such a switch requires buffers at the input as well as at the output ports. This is called speed-up switch. It is clear that in such a switch packet loss can occur

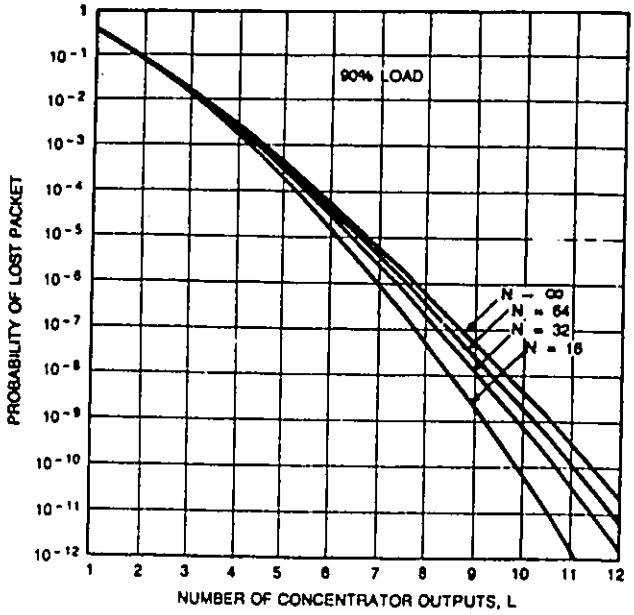


Figure 3.7: Packet Loss Rate in a Knockout Switch

by the overflow of input as well as output buffers. Oie et al. [53] have analyzed the effect of speed-up $1 \leq L \leq N$ in an $N \times N$ switch. They obtained the maximum throughput in a large speed-up switch ($N = \infty$) for the case when infinite buffers are provided at each input and output port. The maximum throughput is given for several values of L in Table 3.3.

Chen and Stern have analyzed a speed-up switch without a back-pressure mechanism in [61]- [63], where packets have exponentially distributed service-time. A speed-up switch with output channel grouping has been presented in [64], assuming packets of fixed length. In both the above models, the packet loss can occur at output as well as at input buffers.

3.6 Summary

In this chapter, we reviewed the performance of some nonblocking fast packet switches. As maximum throughput of an input buffered switch is limited to 58.6%, many schemes have suggested to enhance the performance. Although, look-ahead contention resolution

L	Maximum Throughput
1	0.5858
2	0.8845
3	0.9755
4	0.9956
5	0.9993
6	0.9999
$\vdots$	$\vdots$
∞	1.0000

Table 3.3: Maximum Throughput in a Speed-up Switch with Input and Output Buffers

scheme with large window size improves the performance significantly, but it increases the complexity of the switch to a great extent. A speed-up switch with a speed-up factor of three ($L = 3$) appears to be a good solution, but in such a switch packet loss can occur at input as well as at output buffers.

An output buffered switch provides the best throughput-delay performance. From the buffer utilization point of view, a completely shared output buffer is very attractive. To prevent monopolizing the use of the shared buffers by an overloaded output port, some control over the sharing of these buffers is required. Sharing scheme with maximum queue length and minimum allocation is a possible solution. Complete sharing with 'drop on demand' scheme also looks promising but it needs to be investigated in detail, and its implementation complexity should be examined.

Chapter 4

Nonblocking Packet Switch with Speed and Output Buffer Constraints

4.1 Introduction

In this chapter, we consider an $N \times N$ synchronous nonblocking switch for high-speed packet switching networks transporting fixed length packets. In practice, such a switch may have buffers at input as well as at output ports (Fig. 4.1). The buffers at the input and output ports are provided when the switch fabric operates faster than an input/output trunk but slower than the total speed of the input/output trunks, i. e. when at most L packets can be routed from inputs to any given output port in a time slot, although the switch can transfer only one packet from each input to outputs in any time slot. This we call speed constraint (i. e., $1 < L < N$). Even if there is no speed constraint (i. e., $L = N$), buffers at input and output ports may be provided if the switch has to be implemented in a single VLSI chip. In such an implementation, the output buffers are part of the VLSI chip and therefore only a limited amount of buffers (e.g., 10 to 20 buffers) can be provided per output port. The input buffers can be considered as a part of the input line interface, hence their amount could be relatively larger than that of the output buffers. This we call

output buffer constraint and the number of buffers per output port is denoted as b_o. We assume that whenever an output queue is full, the packets from the head of input queues are not transferred to that output queue and head of line blocking occurs. This implies that packet loss does not occur at the output queues.

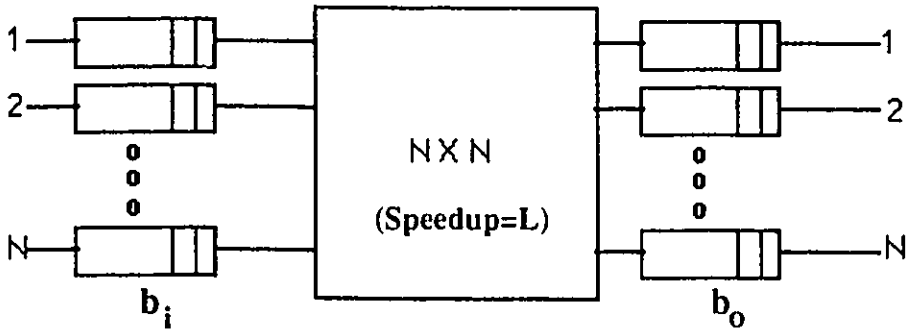


Figure 4.1: ATM Switch with Input and Output Buffers

In [51], the performance of the nonblocking packet switch (NPS) with input and output queueing was surveyed. Oie et al. [53] considered the speed constraint in a switch with infinite input and output buffers. The knockout switch [27] may be viewed as having speed constraint but no input buffers. In a knockout switch, L is the number of outputs of each concentrator. It has been shown that a value of $L = 7$ is required to achieve packet loss probability of 10^{-6} at 70% load, if an infinite number of buffers is provided at each output. Our proposed speed-up switch with $L = 3$ and with moderate amount of input and output buffers can achieve packet loss probability of less than 10^{-6} at even higher loads. The results show that for sizable amount of output buffers, the switch with $L = 3$ would perform almost as well as the switch with $L = N$. For moderate loads, the switch with $L = 2$ can also be implemented.

In the analysis of the switch, we assume that there are infinite number of input and output ports. The performance parameters obtained from the analysis of such a large switch are important because they serve as pessimistic bounds for a finite size switch, which is difficult to analyze. For example, maximum throughput obtained in the limiting case of $N = \infty$ is the lower bound on the maximum throughput of a finite size switch,

for a given number of buffers per output port (b_o) and speed-factor (L). Similarly, the average waiting time obtained for such a large switch gives the upper bound on the average waiting time in a finite size switch.

Our objective is to analyze the effect of speed constraint and output buffer constraint on performance parameters in a high-speed nonblocking packet switch (NPS). This analysis has been divided into two parts. In the first part, we consider only output buffer constraint, i. e., there is no speed constraint ($L = N$). Note that the analyses for $L \geq b_o + 1$ and $L = N$ would be identical. If $b_o + 1$ or more packets arrive at the head of input queues to be transmitted to a given output port (assuming it is idle at this point of time), $L \geq b_o + 1$ ensures that one of these packets would be routed directly to the output port and the other b_o packets will be queued at the output buffer. This implies that head-of-line (HOL) blocking at input queues would occur only when the output queue is full. This case is presented in Section 4.2. In the second part, which is presented in Section 4.3, we analyze the maximum throughput and mean delay of the switch having speed as well as output buffer constraints. The input buffers are sized according to the stringent requirements of packet loss rate.

In Section 4.4, we describe the implementation details of a speed-up switch with a back-pressure mechanism. We also discuss the issue of fairness in serving the input buffers of the switch.

The results of Section 4.2.1 and 4.2.3 were also independently discovered by I. Iliadis and W. E. Denzel in [65], [66]. They, however, did not study the following aspects that we cover in Section 4.2.2 and 4.2.4:

1. Service-time distribution of HOL packets, and

2. Sizing of input buffers according to desired packet loss rate.

Similarly, the analysis of Section 4.3.1 and 4.3.2 were also independently published by Bruzzi and Pattavina [67], but they did not sized the input buffers in a speed and output buffer constrained switch, which we do in Section 4.3.3. Our results of Section 4.2.1 to 4.3.1 appeared in [68], and the results of Section 4.3.2 and 4.3.3 were published in [69].

4.2 NPS under Output Buffer Constraint

We assume that the number of buffers at each input port, b_i is infinite, but the number of buffers at each output port, b_o is finite. A random uniform traffic model is assumed, which is as follows. In a time slot, packets arrive at each input port according to a Bernoulli process with probability ρ_o. Each packet has an equal probability $(1/N)$ of being addressed to any given output. The packet arrival at any input port is independent of arrivals at other inputs and successive packet arrivals are independently destined for their respective output ports.

4.2.1 Maximum Throughput Analysis

In this section, we obtain the maximum throughput for a switch with input and output queueing where the number of output buffers b_o is finite, $b_o < L \leq N$ and in the limit N goes to infinity.

In a nonblocking switch with input and output buffers, during a time slot packets from heads of the input queues move to output queues, depending upon the buffer space available at output queues at the beginning of that slot. For any output port, whenever more packets arrive at the head of input queues than the available buffers in that output queue, HOL blocking occurs. Let us consider the packets destined for output port i. We define H_m^i as the number of HOL blocked packets for output i at the end of the m^{th} time slot. Let us also define A_m^i as the number of packets moving to the head of input queues at the beginning of the m^{th} slot and destined for output i. A_m^i includes the fresh arrivals in input queues at the beginning of the slot m, which were empty at the end of the previous slot. Q_m^i indicates the queue length at output port i at the end of the m^{th} slot. These three random variables are related as follows:

$$Q_m^i = max(0,\ Q_{m-1}^i + min(b_o + 1 - Q_{m-1}^i,\ H_{m-1}^i + A_m^i) - 1) \tag{4.1}$$

But with the assumption that $b_o < L$, we have $H_m^i = 0$ when $Q_m^i < b_o$, and we also have $Q_m^i = b_o$ when $H_m^i > 0$.

Under these conditions, it is interesting to note that the output queue (Q^i_m) and HOL blocked packets (H^i_m) can be considered as a single queue (Fig. 4.2), where the arrival process is given by the distribution of A^i_m. The occupancy of this virtual queue is

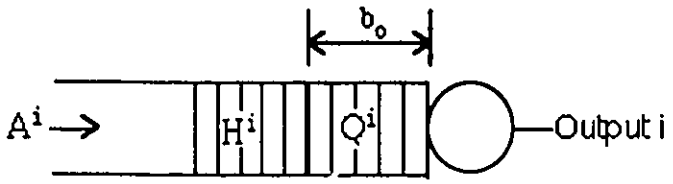


Figure 4.2: The Virtual Queue- M/D/1 Model

defined as:

$$S^i_m = Q^i_m + H^i_m \tag{4.2}$$

As all the queues are identical under the assumed uniform random traffic, therefore in steady-state, the above equation becomes $S^i = Q^i + H^i$ or simply $S = Q + H$. Similarly, in steady-state A^i_m would be written as A^i or simply A. Notice that

$$S^i_m = max(0,\ S^i_{m-1} + A^i_m - 1) \tag{4.3}$$

It has been proved in [16] that the arrival process at the head of inputs becomes sequence of independent Poisson variables as N approaches infinity. Therefore the distribution of the virtual queue length would be identical to that of a discrete time $M/D/1$ queue. Although, it is possible to have a closed form solution for the steady-state queue size of the virtual queue, but as mentioned in [16], it leads to inaccurate results for the tail of the distribution. Therefore, a Markov chain (refer to Fig. 4 of [16]) is used to obtain the following steady-state queue size probabilities [16],

$$s_0 \equiv Pr(S = 0) = \frac{(1 - \rho_o)}{a_0} \tag{4.4}$$

$$s_1 \equiv Pr(S = 1) = \frac{(1 - a_0 - a_1)}{a_0} s_0 \tag{4.5}$$

$$\vdots$$

$$s_n \equiv Pr(S = n) = \frac{(1 - a_1)}{a_0} s_{n-1} - \sum_{j=2}^{n} \frac{a_j}{a_0} s_{n-j} \qquad n \geq 2 \tag{4.6}$$

where ρ_o is the average number of packets moving from input ports to any given output port, in a time slot. In other words, ρ_o is the normalized throughput of each output port. And a_j's are defined as,

$$a_j \equiv Pr(A = j) = \frac{\rho_o^j \, e^{-\rho_o}}{j!} \qquad j = 0, 1, \ldots \tag{4.7}$$

Now, we are interested in the distribution of random variables Q and H, which are obtained as follows:

$$q_0 \equiv Pr(Q = 0) = s_0 \qquad b_o \geq 1 \tag{4.8}$$

$$\vdots$$

$$q_{b_o-1} \equiv Pr(Q = b_o - 1) = s_{b_o-1} \qquad b_o \geq 1 \tag{4.9}$$

$$q_{b_o} \equiv Pr(Q = b_o) = \sum_{j=b_o}^{\infty} s_j \qquad b_o \geq 0 \tag{4.10}$$

The distribution of H is,

$$h_0 \equiv Pr(H = 0) = \sum_{j=0}^{b_o} s_j \tag{4.11}$$

$$h_1 \equiv Pr(H = 1) = s_{b_o+1} \tag{4.12}$$

$$\vdots$$

$$h_n \equiv Pr(H = n) = s_{b_o+n} \qquad n \geq 1 \tag{4.13}$$

To obtain the maximum throughput, we assume that the input queues are saturated so that the packets are always waiting at every input queue. When a packet is removed from any input queue, a new packet immediately moves to the head of the input queue. Under above saturation condition, the following equation is satisfied [16],

$$\sum_{i=1}^{N} A^i = N - \sum_{i=1}^{N} H^i \tag{4.14}$$

and by taking expectation of both sides of the above equation and dividing by N, results in

$$\rho_o = 1 - \overline{H^i} \tag{4.15}$$

Note that $\overline{H^i}$ itself depends upon the ρ_o. Therefore the maximum throughput ρ_o is iteratively optimized using equations (4.4)-(4.7), (4.11)-(4.13) and (4.15). For different number of output buffers (b_o), the maximum throughput is shown in Figure 4.3. Figure 4.3 also shows the maximum throughput obtained by simulation of a 16 × 16 packet switch. As mentioned earlier, the throughput is higher for a finite size switch than the infinitely large switch. In a finite size switch, the HOL blocked packets destined for the individual output ports are not independent, but are correlated. This correlation results in a smaller value of $\overline{H^i}$ and thus higher throughput.

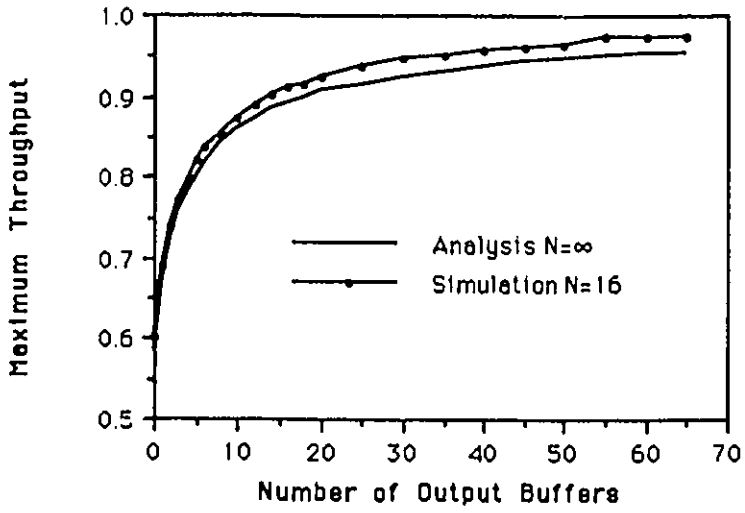


Figure 4.3: Effect of Output Buffer Size on Maximum Throughput

Certainly, $b_o = 0$ implies that the switch has queues only at input ports and gives $\rho_o = 0.5857$, exactly as obtained in [16]. By providing 5 buffers per output port, theoretically a throughput of more than 0.8 can be achieved, assuming infinite number of buffer space at each input. As intuitively clear, providing more buffers at output ports gradually diminishes the HOL blocking and thus increases the throughput of the switch.

4.2.2 Service-Time Distribution of HOL Packets

Karol et al. [16] have obtained the service-time distribution for an infinitely large switch having only input buffers and assuming that the head-of-line (HOL) packets are served in random order. In such a case, the HOL packets behave like a discrete-time $M/D/1$ queue and the delay distribution of this queue gives the service-time distribution of HOL packets. In this section, we compute the service-time distribution of HOL packets in a switch with finite buffers at each output port.

In a time slot, one or more packets move to the HOL position for a given output, which constitute a batch. Let us concentrate on a packet ('tagged packet') from a batch, moving to the head of an input queue at the beginning of the m^{th} slot and assume that it is destined for output i. Let the random variable B_m represent the batch-size of the tagged packet and e_n be the probability that the tagged packet belongs to the batch size $B_m = n$. Then from [70],

$$e_n \equiv Pr(B_m = n) = n\, a_n/\overline{A} \qquad n = 1, 2, \ldots \tag{4.16}$$

where a_n are given by equation (4.7). Now we would observe the number of slots required 'to serve' this packet. 'To serve' means to route the tagged packet to the output queue i. At the end of the m^{th} slot, one or more packets would be served from the head of input lines and routed to the output queue i, depending upon the number of empty buffers at the output. We assume that the HOL blocked packets are given higher priority over the HOL arrivals, i. e., FIFO among HOL packets having the same destination. In a time slot, if more than one packet arrive at the head of input lines for a given output port, they receive lower priority than blocked HOL packets for that output port, but randomly prioritized among themselves.

Let T represent the number of slots required to serve the tagged packet. The service-time would depend upon the occupancy of the virtual queue at the end of the previous slot (S^i_{m-1}) and the batch-size B_m to which the tagged packet belongs. The tagged packet may be served at the end of the m^{th} slot, only if $S^i_{m-1} \leq b_o$. In such a situation, it would definitely be served at the end of the m^{th} slot if $B_m \leq b_o - S^i_{m-1} + 1$ or it would be served

with probability $(b_o - S^i_{m-1} + 1)/B_m$ in case of $B_m > b_o - S^i_{m-1} + 1$. From this argument, the steady-state probability that the tagged packet would be served in the slot it arrives is:

$$\begin{aligned} Pr(T=1) &= \sum_{k=0}^{b_o} s_k \left[\sum_{n=1}^{b_o-k+1} e_n + \sum_{n=b_o-k+2}^{\infty} \frac{(b_o-k+1)}{n} e_n \right] \\ &= \sum_{k=0}^{b_o} s_k \left[\sum_{n=0}^{b_o-k} \frac{\rho_o^n \, e^{-\rho_o}}{n!} + \sum_{n=b_o-k+1}^{\infty} \frac{(b_o-k+1)}{(n+1)} \frac{\rho_o^n \, e^{-\rho_o}}{n!} \right] \end{aligned} \tag{4.17}$$

The service-time would be more than one slot with the following probabilities:

$$Pr(T=j) = \sum_{k=0}^{b_o+j-1} s_k \left[\sum_{n=b_o+j-k-1}^{\infty} \frac{1}{(n+1)} \frac{\rho_o^n \, e^{-\rho_o}}{n!} \right] \qquad j > 1 \tag{4.18}$$

Now, we will find the expression for the average total delay through the switch.

4.2.3 Transmission-Delay through the Switch

The transmission delay of a packet consists of waiting time in the input queue until it reaches the head-of-line, waiting time at the head of input due to the HOL blocking and waiting time at the output queue.

We have assumed that, packets arrive at the head of input queues at the beginning of the time slot and if they win contention, they are transmitted to the output queues at the end of the slot and they suffer no delay in the switch fabric. In a real switch, the contention resolution and transmission of the packets through the switch fabric would begin immediately after the arrivals and it will take one slot for routing the packets to the output queues. It implies that after reaching the head of input queue, it takes a minimum of one time slot for a packet to be transmitted to the output queue. As explained in Section 4.2.1, the sum of HOL blocked packets for a given output and packets in the output queue at any time slot is stochastically identical to the number of packets in an $M/D/1$ queue, and therefore the sum of waiting time at the head of input queue and the output queue is equal to the waiting time in the $M/D/1$ queue plus one slot.

Assuming that the input load is less than the maximum throughput of the switch for a given number of output buffers (b_o), every input queue is a stable discrete-time

$Geom/G/1$ queue, where the probability mass function of the random variable T gives the general service-time distribution. We use the expression obtained in [71] for the average waiting time in discrete-time $Geom/G/1$. Therefore the expected delay through the switch is

$$\overline{D} = \frac{\rho_o \overline{T(T-1)}}{2\,(1-\rho_o\,\overline{T})} + \frac{\rho_o}{2\,(1-\rho_o)} + 1 \tag{4.19}$$

where the first component is the average waiting time of a packet in the $Geom/G/1$ queue till it reaches the HOL position. The second component in the above equation is the average waiting time in the $M/D/1$ queue [72] plus one extra slot delay that is suffered in the switch fabric. Figure 4.4 shows the average delay suffered (in terms of number of slots) through the switch when different numbers of buffers are provided at the output ports. The curve for $b_o = 0$ corresponds to the case of NPS with input queueing and is similar to the curve obtained in [16]. However there are two differences. First, instead of random selection among HOL packets, the service-order we considered is FIFO, which results in better delay performance. The second difference is that the delay of one time slot required to route the packet through the switch fabric has been included in our analyses. On the other hand, $b_o = \infty$ corresponds to the case where there is no HOL blocking and the arriving packets join the output queues after suffering a delay of one time slot, which is required to route the packets. The curve for $b_o = \infty$ is similar to the curve of output queueing in [16] except that the routing delay of one time slot has been added in Figure 4.4.

4.2.4 Packet Loss Rate Analysis

In this section, we obtain the distribution of the number of packets waiting in an input queue including the HOL packet. As all the input queues are identical, we consider a particular (i. e., tagged) input queue. In the beginning we assume that there are infinite buffers at each input port and find the steady-state distribution p_j, i. e., the probability that there are j packets in the tagged input queue immediately after the arrival instant. In the case of finite number of buffers (b_i) per input port, the packet loss probability

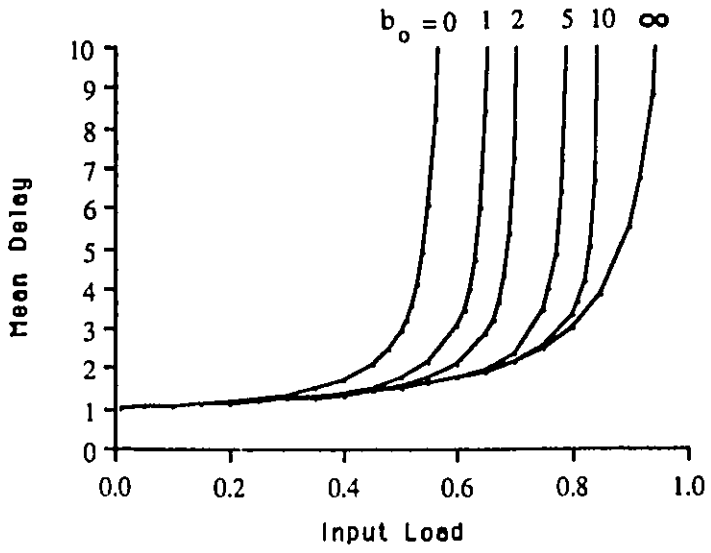


Figure 4.4: Effect of Output Buffer Size on Average Delay

(P_{loss}) is upper bounded by summing up the tail of the distribution, i. e.,

$$P_{loss} < \sum_{j=b_o+1}^{\infty} p_j$$

To find the distribution p_j, we first need to compute the probability c_m, that m packets arrive during the service-time of a HOL packet. We can determine c_m from

$$\begin{aligned} c_m &= \sum_{k=1}^{\infty} Pr(m \text{ arrivals in } k \text{ slots}).Pr(\text{service-time of } HOL \text{ packet is } k \text{ slots}) \\ &= \sum_{k=max(1,m)}^{\infty} \binom{k}{m} \rho_o^m (1-\rho_o)^{k-m} \, Pr(T=k) \qquad m \geq 0 \end{aligned}$$

p_j are related to c_m by the matrix equation [71],

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} c_0 & c_0 & 0 & 0 & \dots & \dots \\ c_1 & c_1 & c_0 & 0 & \dots & \dots \\ c_2 & c_2 & c_1 & c_0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \end{bmatrix} \tag{4.20}$$

This requires an iterative solution for probabilities p_j, but we know that

$$p_0 = 1 - (\overline{H} + \rho_o) \tag{4.21}$$

Now we can directly solve for p_j, $j \geq 1$

$$p_1 = \frac{p_0\ (1 - c_0)}{c_0} \tag{4.22}$$

$$p_2 = \frac{p_1\ (1 - c_1) - p_0\ c_1}{c_0} \tag{4.23}$$

$$\vdots$$

$$p_j = \frac{p_{j-1}\ (1 - c_1) - p_0\ c_{j-1} - \sum_{n=2}^{j-1} p_{j-n}\ c_n}{c_0} \qquad j > 2 \tag{4.24}$$

Figures 4.5-4.7 show the number of input buffers required per port to achieve different packet loss rates at load of 0.7 to 0.85 as a function of the amount of output buffers. At 80% input load, to achieve a packet loss rate of 10^{-8}, the results show that, about 48 buffers are needed at each input port if there are 8 buffers at each output port, while only 32 buffers are required per input port when 12 buffers are provided at each output port, i. e. a total of 44 buffers per port. If we further increase the amount of output buffers, the reduction in input buffers becomes almost linear and so the total number of buffers required per port remains about the same. To clarify this point, about 38 buffers per port are needed to meet the same requirements in case of a switch having only output queueing.

4.3 NPS under Speed and Output Buffer Constraints

In this section we analyze the nonblocking packet switch having speed constraint and output buffer constraint ($L \leq b_o$), assuming the same traffic model as described in Section 4.2.

4.3.1 Maximum Throughput Analysis

Under these constraints, it is clear that the sum of HOL blocked queue and the output queue can no longer be modelled as an $M/D/1$ queue. For the tagged output i, a two dimensional Markov Chain of H^i and Q^i random variables is required to find the joint

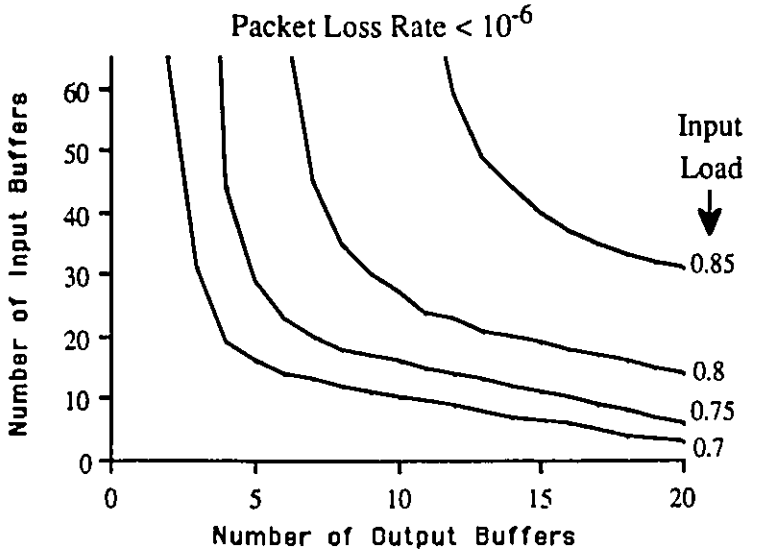


Figure 4.5: Output Buffer Size vs. Input Buffers Required for $P_{loss} < 10^{-6}$

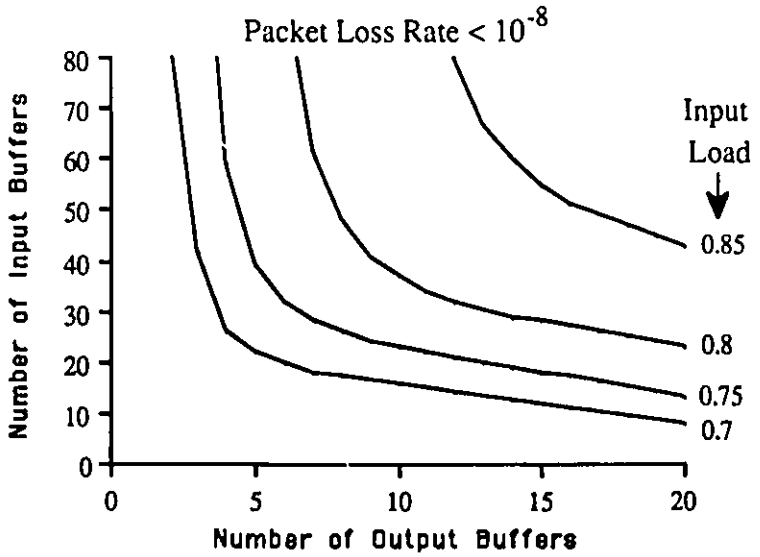


Figure 4.6: Output Buffer Size vs. Input Buffers Required for $P_{loss} < 10^{-8}$

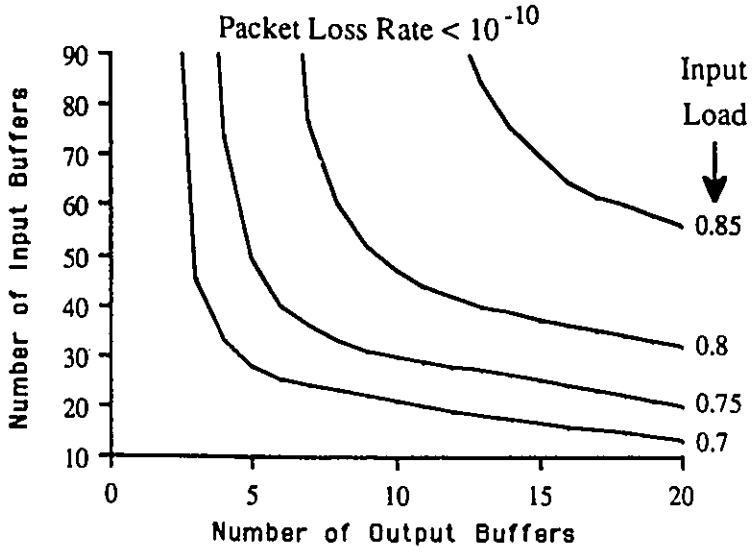


Figure 4.7: Output Buffer Size vs. Input Buffers Required for $P_{loss} < 10^{-10}$

distribution of H^i and Q^i. Figure 4.8 shows the Markov Chain for the case of $L = 3$ and $b_o = 5$. Although all transitions have not been shown in the figure, it explains the concept of output queueing and the HOL blocking under the above two constraints. We define $r_{n,m}$ as the joint distribution of H^i and Q^i as follows

$$r_{n,m} \equiv Pr\ (Q^i = n,\ H^i = m) \qquad 0 \le n \le b_o,\ m \ge 0$$

The following equations give the steady-state probabilities

$$r_{n,0} = s_n \qquad 0 \le n \le L-1 \tag{4.25}$$

$$r_{n,m} = 0 \qquad 0 \le n \le L-2,\ m \ge 1 \tag{4.26}$$

$$r_{n,m} = \sum_{k=0}^{L+m} a_k\, r_{n-L+1,L+m-k} \qquad L-1 \le n < b_o,\ m \ge 1 \tag{4.27}$$

$$r_{n,0} = \frac{r_{n-1,0}}{a_0} - \sum_{j=1}^{L}\sum_{k=0}^{j} \frac{a_k}{a_0}\, r_{n-j,j-k} \qquad L \le n \le b_o \tag{4.28}$$

$$r_{b_o,m} = \sum_{j=1}^{L}\sum_{k=0}^{j+m} a_k\, r_{b_o-j+1,j+m-k} \qquad m \ge 1 \tag{4.29}$$

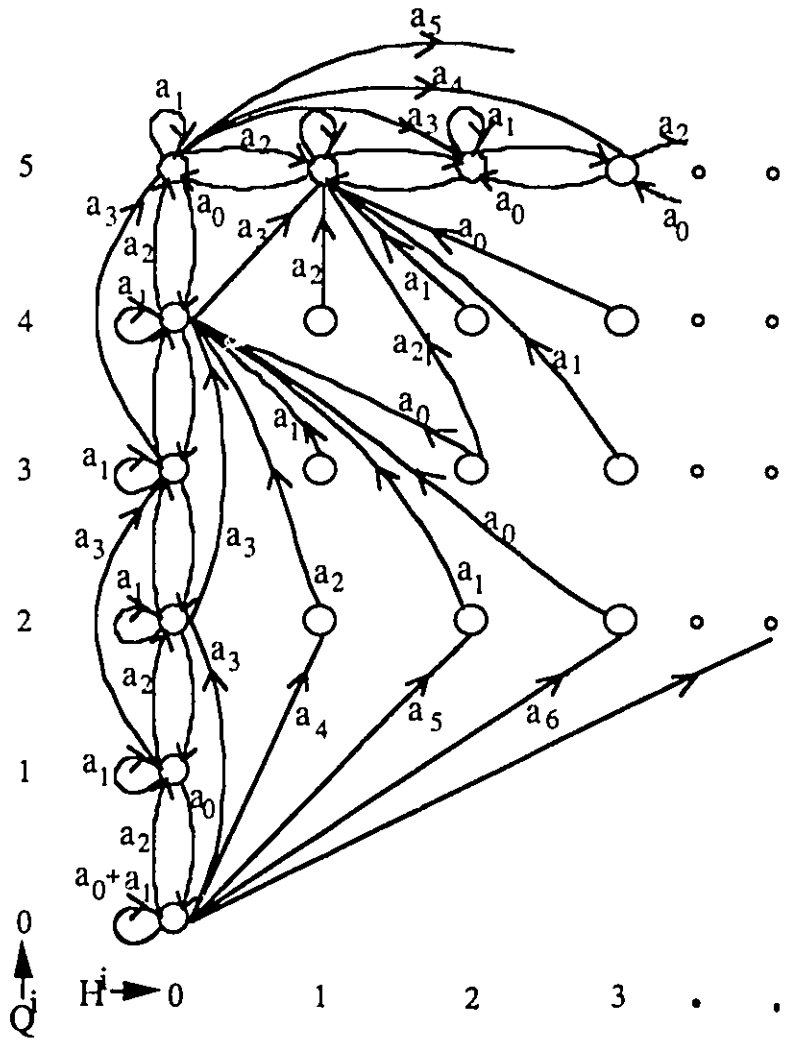


Figure 4.8: Markov Chain State Transition Diagram for (H^i, Q^i), when $L = 3$, $b_o = 5$

The maximum throughput is obtained by solving the above equations. Figure 4.9 and Table 4.1 show the maximum throughput as a function of the number of output buffers and speed-factor L. The maximum throughput for the case of $b_o = \infty$ (the last row of Table 4.1) was analyzed in [53]. The maximum throughput for the case of $L > b_o$, which was analyzed in Section 4.2.1, is also shown in the table for the comparison purpose. For $L = 2$, the throughput gets saturated when 50 buffers are provided per output port and any further increase in the number of output buffers does not improve the throughput. Note that when $L = 3$, the maximum throughput approaches the maximum throughput of the case of $L = N$ for sizable amount of buffers per output port. Figure 4.10 shows the maximum throughput of a switch with speed-factor of 4. The simulation results of a 16×16 and 32×32 are also shown for the comparison. The maximum throughput of a 32×32 switch is less than 1% higher than the maximum throughput obtained by analysis of a very large switch. It proves the accuracy of the analysis.

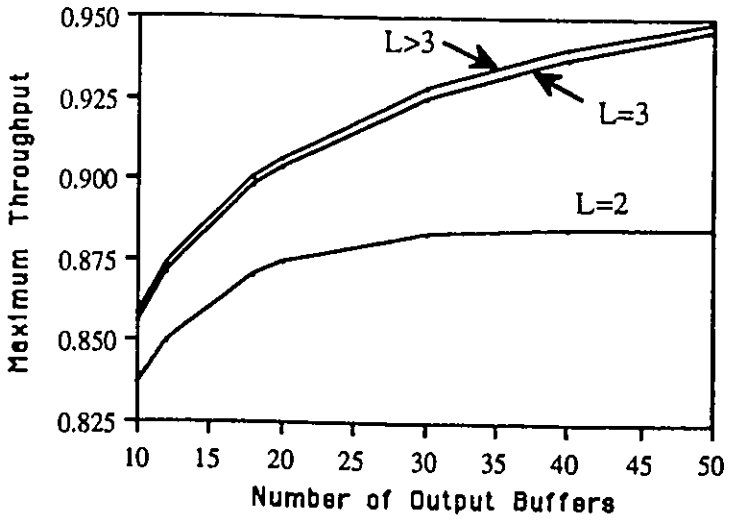


Figure 4.9: Effect of Output Buffer Size and Speed-factor on Maximum Throughput

	Maximum Throughput			
b_o	$L = 2$	$L = 3$	$L = 4$	$L > b_o$
0	0.5857	0.5857	0.5857	0.5857
1	0.6713	0.6713	0.6713	0.6713
2	0.7176	0.7228	0.7228	0.7228
3	0.7484	0.7567	0.7576	0.7576
4	0.7709	0.7818	0.7830	0.7831
5	0.7883	0.8011	0.8026	0.8028
8	0.8233	0.8402	0.8423	0.8425
12	0.8496	0.8708	0.8731	0.8735
18	0.8702	0.8974	0.8999	0.9003
20	0.8741	0.9036	0.9062	0.9066
30	0.8830	0.9247	0.9276	0.9280
40	0.8843	0.9373	0.9402	0.9407
50	0.8845	0.9457	0.9487	0.9492
⋮	⋮	⋮	⋮	⋮
∞	0.8845	0.9755	0.9956	1.0000

Table 4.1: Maximum Throughput for Different Output Buffer Sizes and Speed-factor

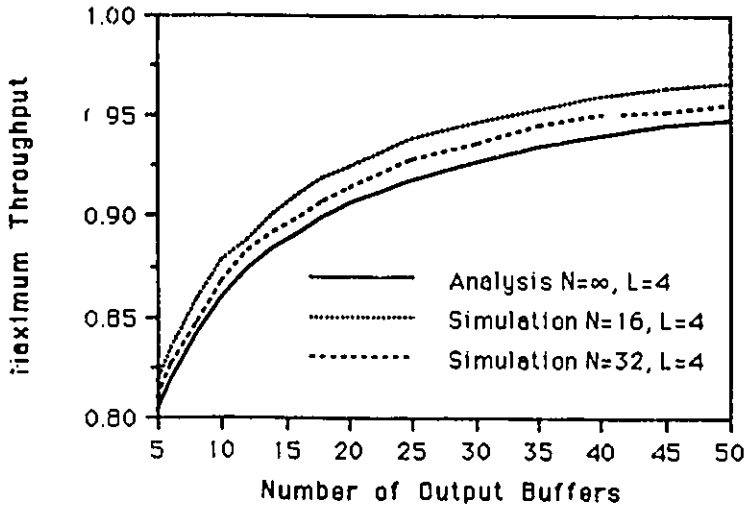


Figure 4.10: Maximum Throughput of the Switch with Speed-factor of 4

4.3.2 Transmission-Delay Analysis

In this section, first we compute the service time distribution of a tagged HOL packet, and then we find the mean delay of a packet through the switch. Our approach here is similar to the procedures used in Section 4.2.2 – 4.2.3. The service time distribution of the HOL packets for output buffer and speed constrained switch was also obtained independently in [67], but we present it here in a method which is simple and computationally more efficient.

Let us concentrate on a packet ('tagged packet') moving to the head of an input queue at the beginning of an arbitrary slot and assume that it is destined for output i. During a slot, one or more packets would be served from head of input lines and routed to the output queue i, depending upon the switch speed and the number of empty buffers at the output. We assume that the HOL blocked packets are given higher priority over the HOL arrivals, i. e. FIFO among HOL packets having the same destination.

Let T represent the number of slots required to serve the tagged packet. The service time would depend upon the switch speed, the number of buffers empty at the output

queue, the HOL blocking and the batch-size to which the tagged packet belongs. The steady-state probability that the tagged packet would be served in the slot it arrives is:

$$Pr(T=1) = \sum_{k=0}^{b_o-L+1} \sum_{l=0}^{L-1} r_{k,l} \left[\sum_{n=1}^{L-l} e_n + \sum_{n=L-l+1}^{\infty} \frac{L-l}{n} e_n \right] + \sum_{k=b_o-L+2}^{b_o} \sum_{l=0}^{b_o-k} r_{k,l} \left[\sum_{n=1}^{b_o-k-l+1} e_n + \sum_{n=b_o-k-l+2}^{\infty} \frac{(b_o-k-l+1)}{n} e_n \right] \quad (4.30)$$

where e_n are given by equation (4.16). The first expression on the right hand side of (4.30) belongs to the case when L or more buffers are empty in the output queue and therefore up to L packets can be served out of the HOL blocked and HOL arrivals. As HOL blocked packets are given priority over HOL arrivals, the number of HOL blocked packets (l) must not be more than $L-1$. In such a situation, if the batch-size of HOL arrivals is $L-l$ or smaller, they are served in the same slot, otherwise $L-l$ packets are served out of n. The second expression in (4.30) corresponds to the case when less than L buffers are empty in the output queue and hence, the output buffer is the constraint rather than the speed.

The service-time would be more than one slot with the following probabilities:

$$Pr(T=j) = \sum_{k=0}^{b_o} \sum_{l=0}^{min(b_o-k+j,\, jL)-1} r_{k,l} \left[\sum_{n=n_{low}}^{\infty} \frac{n_s}{n} e_n \right] \qquad j > 1 \quad (4.31)$$

where n_{low} is the lower bound of the batch-size n and n_s is the number of packets to be served in the j^{th} slot (out of n HOL arrivals). Depending upon the empty buffers in the output queue, up to $min(b_o-k+j, jL)$ packets can be served in j slots. At least one HOL arrival has to be served in the j^{th} slot, therefore HOL blocked packets must not be more than $min(b_o-k+j, jL)-1$. This explains the upper limit of l in (4.31). Coming to n_{low}, it is clear that $min(b_o-k+j-1, (j-1)L)$ packets are served in $j-1$ slots and at least one packet is served in the j^{th} slot. From this argument,

$$n_{low} = max|min(b_o-k+j-1, (j-1)L) + 1 - l, 1| \quad (4.32)$$

Now we find the expression for n_s in (4.31). It will be the minimum of the following three values:

1. $L-$ Number of HOL blocked packets (if any) to be served in the jth slot

   $= L - max(l - (j-1)L, 0)$

2. Number of buffers available in the output queue i in the j^{th} slot

   = Number of buffers initially empty + Number of packets transmitted from the output port i in j slots − (Number of HOL blocked + HOL arrivals served during $(j-1)$ slots)

   $= (b_o - k) + j - (l + n_{low} - 1)$

3. Out of n, number of HOL arrivals waiting for service in the jth slot

   $= n - (n_{low} - 1)$

Therefore,

$$n_s = min[L - max(l - (j-1)L, 0),\ b_o - k + j - l - n_{low} + 1,\ n - n_{low} + 1] \quad (4.33)$$

This completes the computation of the service-time distribution of HOL packets.

The expected delay through the switch is

$$\overline{D} = \frac{\rho_o\ \overline{T(T-1)}}{2\ (1 - \rho_o\ \overline{T})} + \overline{T} + \overline{Q^i} \quad (4.34)$$

where the first two components are of the average delay in the $Geom/G/1$ queue [71] and the third component is the average delay in the output queue. The mean delay of a packet in the HOL position and in the output queue $(\overline{T} + \overline{Q^i})$ is equal to the mean waiting time in an $M/D/1$ queue plus one slot delay that is suffered in the switch fabric.

Figure 4.11 shows the average delay suffered (in slots) through the switch for $L = 2$ and 4 when $b_o = 4, 10$ and 50. The delay performance of the switch operating at $L = 2$ is not much different from that at $L = 4$ when $b_o = 4$, but the performance is quite distinct when $b_o \geq 10$ and when the input load is larger than 0.8. The average delay for the case of $L = 3$ is very close to that of $L = 4$, for $b_o = 4, 10, 50$ and so it has been shown only for $b_o = 50$. The delay performance of $L = 4$ is identical to the case of $L = N$.

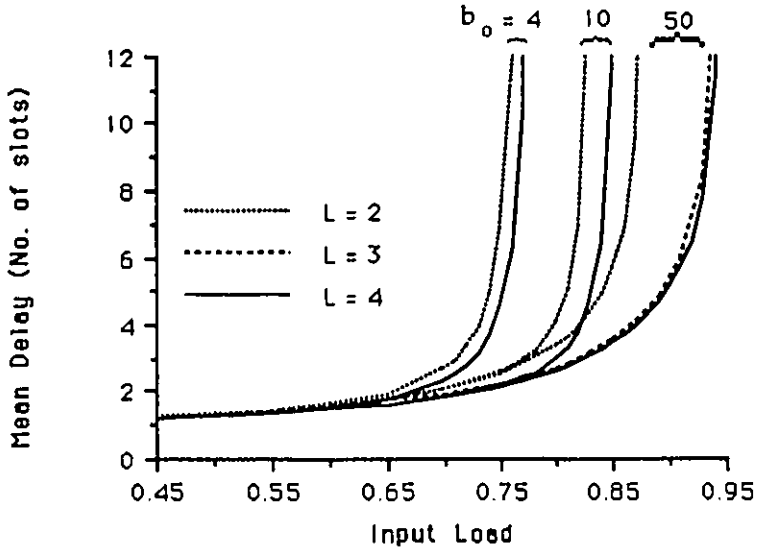


Figure 4.11: Effect of Output Buffer Size and Speed-factor on Average Delay

4.3.3 Packet Loss Rate at Input Queues

Now, we obtain the distribution of the number of packets waiting in an input queue including the HOL packet. Assuming that there are infinite buffers at each input port, we find the steady-state distribution p_j, i. e., the probability that there are j packets in the tagged input queue immediately after the arrival instant. To find the distribution p_j and the upper bound on packet loss rate in the case of finite buffers, we use the technique similar to the one used in Section 4.2.4.

Figure 4.12 shows the upper bound on packet loss rate as a function of the amount of input and output buffers for a switch with $L = 2$ at an input load of 0.7. To obtain the packet loss rate of less than 10^{-10}, we can provide 12 buffers at each output port and 20 buffers at each input port, or alternatively, we can provide 20 buffers at each output port and 14 buffers at each input port. Since we want to limit the number of output buffers and the total number of buffers to a minimum for a given performance, the first option is more suitable. Figure 4.13 shows the packet loss rate for the switch with $L = 2$ at an input load of 0.8. To achieve the packet loss rate of less than 10^{-10}, $b_o = 28$ and $b_i = 30$

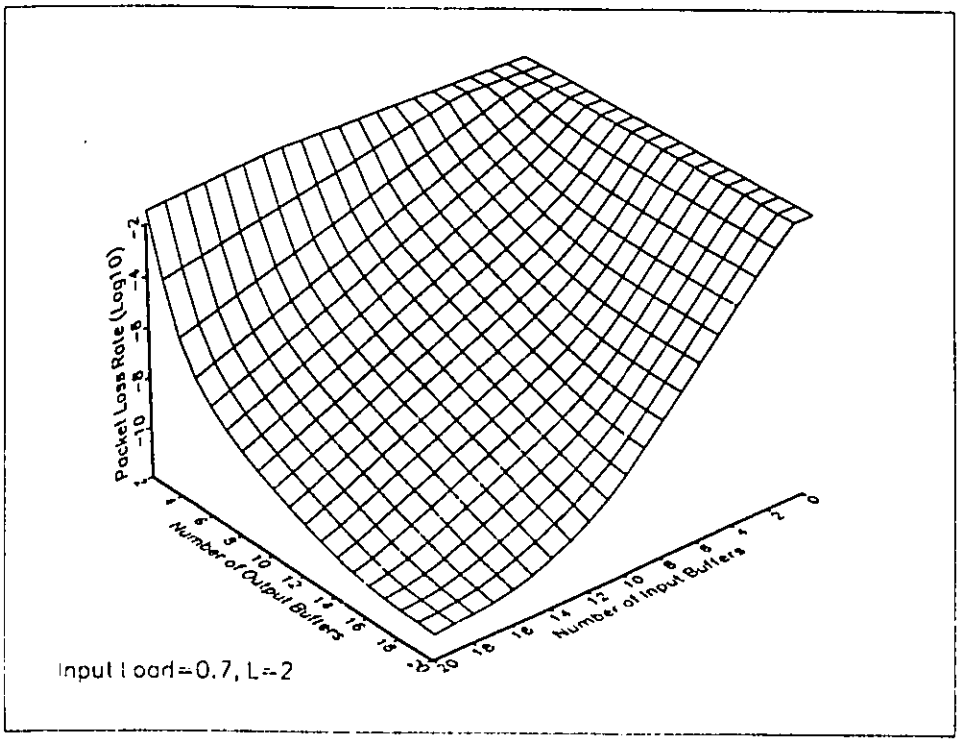


Figure 4.12: Packet Loss Rate in NPS with $L = 2$ at an Input Load = 0.7

appears to be the right choice.

Figure 4.14 gives the upper bound on packet loss rate for various sizes of input and output buffers in a switch operating at $L = 4$ and at an input load of 0.7. It is to be noted from Figure 4.12 and Figure 4.14 that at a load of 0.7, the switch with $L = 2$ or $L = 4$ perform very close to each other. For a packet loss rate of less than 10^{-10}, the switch with $L = 4$ requires $b_o = 12$ and $b_i = 18$ in comparison to $b_o = 12$, $b_i = 20$ for a switch operating at $L = 2$. Figure 4.15 shows the packet loss rate for the switch with $L = 4$ at an input load of 0.8. Providing 25 buffers per output port and 20 buffers per input port results in packet loss probability of less than 10^{-8}. To achieve the loss rate of less than 10^{-10}, we require about $b_o = 36$ and $b_i = 20$.

Figure 4.16 shows the packet loss rate at input loads varying from 0.7 to 0.84 in a switch operating at $L = 2$ and the total number of buffers at input and output ports is 32 (i. e., $b_i + b_o = 32$). For a given allocation of buffers between input and output ports, the load vs. packet loss rate has a straight line, which indicates that as the load increases the packet loss rate increases logarithmically. For different buffer allocation between input

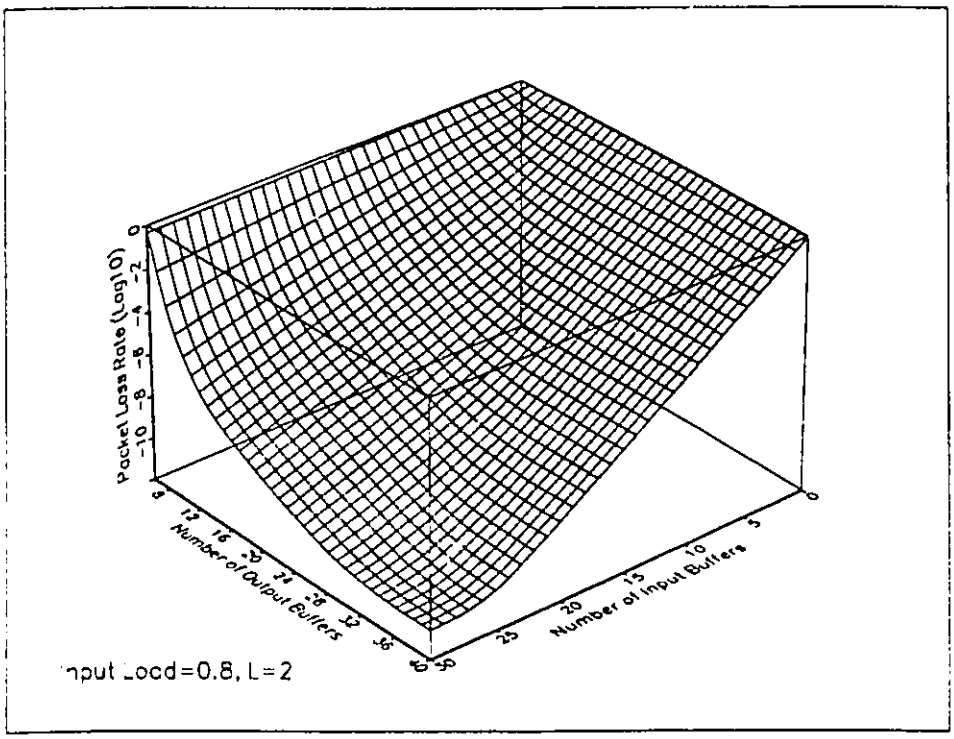


Figure 4.13: Packet Loss Rate in NPS with $L = 2$ at an Input Load = 0.8

and output ports, the slope of the lines is different. Another observation from Figure 4.16 is that at a given load, the packet loss rate is a convex function of the size of input and output buffers, which is obvious at higher loads, especially at input load of 0.84. This suggests that for a known average input load of the switch and for a given total number of buffers per port, there exists a optimum allocation of buffers between input and output ports which minimizes the packet loss probability. For example, at a load of 0.7 the optimum allocation is $b_o = 12$, $b_i = 20$ and at the load of 0.84, the optimum allocation is $b_o = 17$, $b_i = 15$.

4.4 Implementation of the Speed-up Switch

From a practical point of view, a speed-up switch with back-pressure mechanism is a more appropriate solution for high-speed networks. Such a switch provides adequate performance for the high speed networks. In this section, our objective is to suggest a simple implementation of the switch having speed as well as output buffer constraints.

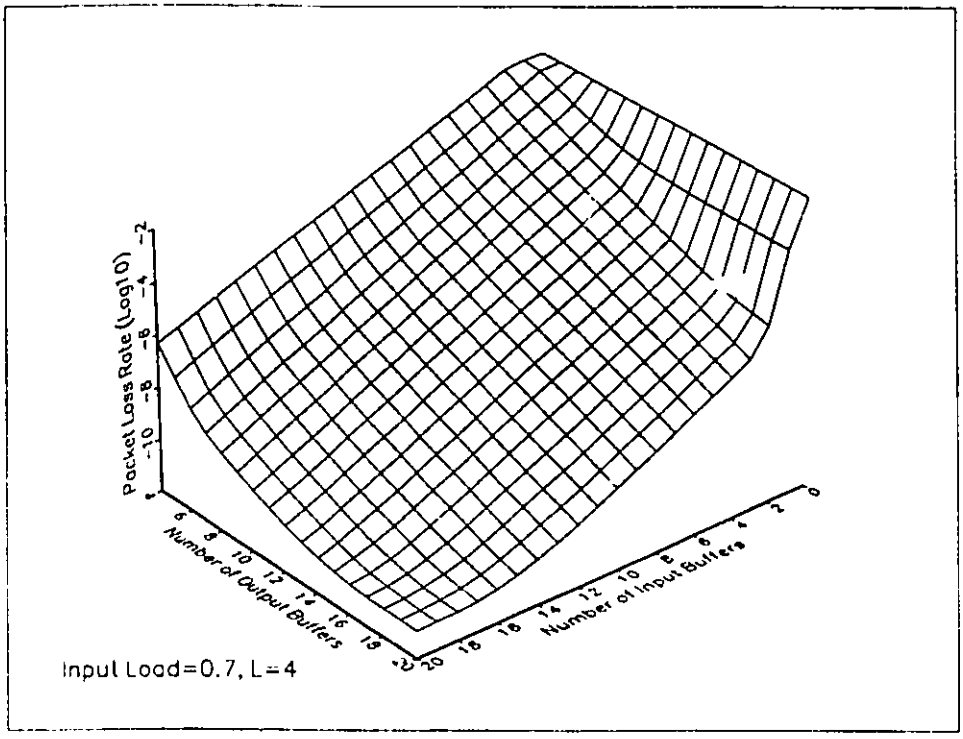


Figure 4.14: Packet Loss Rate in NPS with $L = 4$ at an Input Load = 0.7

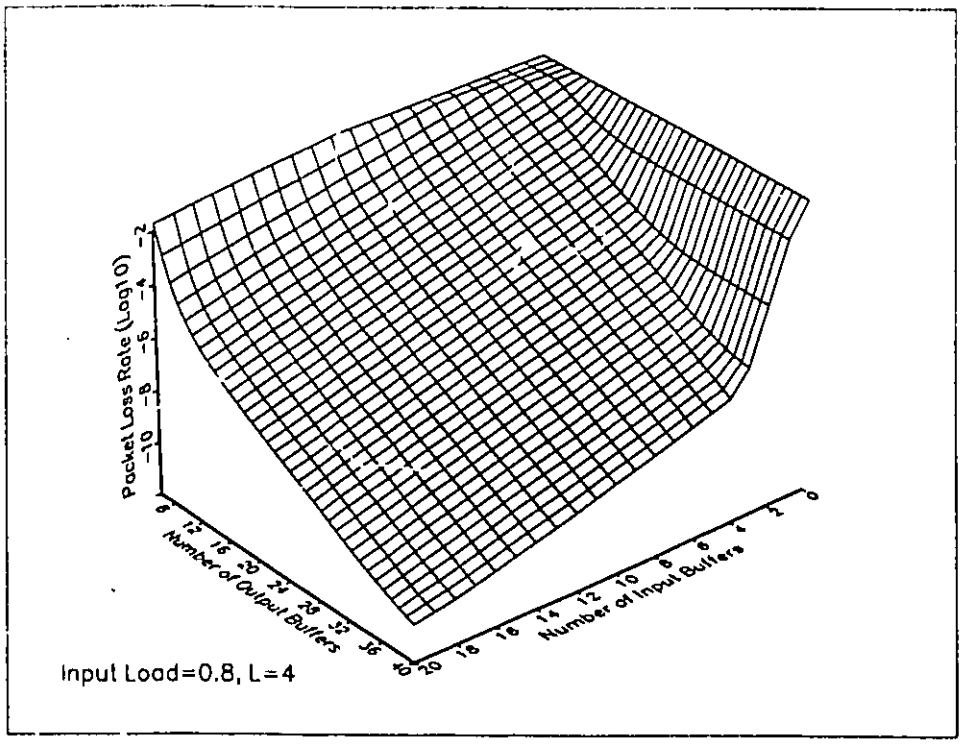


Figure 4.15: Packet Loss Rate in NPS with $L = 4$ at Input Load = 0.8

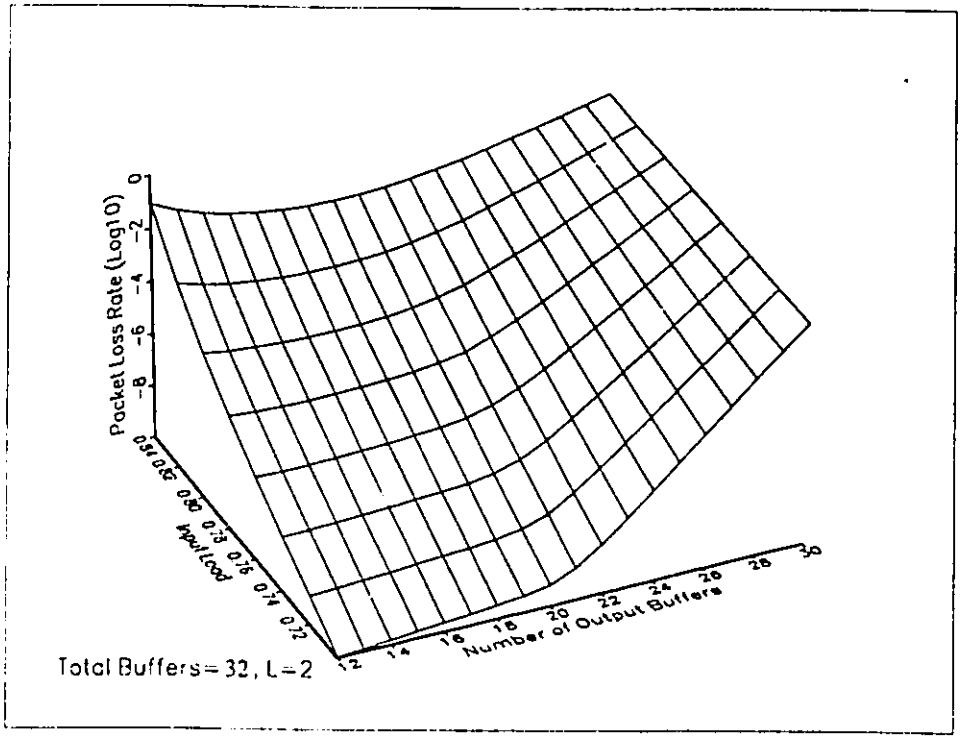


Figure 4.16: Packet Loss Rate as a Function of Input Load in NPS with $L = 2, b_o + b_i = 32$

4.4.1 Crossbar Switch Fabric for the Speed-up Switch

A speed-up switch can be implemented by an $N \times N$ crossbar fabric operating at L times faster than the speed of input-output ports. To reduce the required speed-up, parallel links can be used to carry multiple bits of a packet. For example, by using a bus of 8 bits width, a switch supporting input-output links of 150 Mbps and speed-up of $L = 4$, would require the switch fabric to operate at 75 Mbps. In fact, the switch fabric would be required to operate at a slightly higher rate, in order to allow the overhead processing at the beginning of each time slot to resolve the contention for the output ports. Another way to reduce the speed-up is to provide L links per output port (Figure 4.17). Although the performance of speed-up and parallelism are the same, the implementation of speed-up may be difficult, especially when each input-output link is required to run at 600 Mbps. For this reason, we suggest an implementation of speed-up switch with L parallel links per output port. Each link itself may be carrying 8 bits in parallel. To evaluate the switching overhead, we assume serial transmission on the links in the switch fabric. We can represent the output links in the switch fabric by l_k^o, where o is the output port

number (i. e., $o = 1, 2, \ldots N$) and k identifies a particular link for that output port (i. e., $k = 1, 2, \ldots L$).

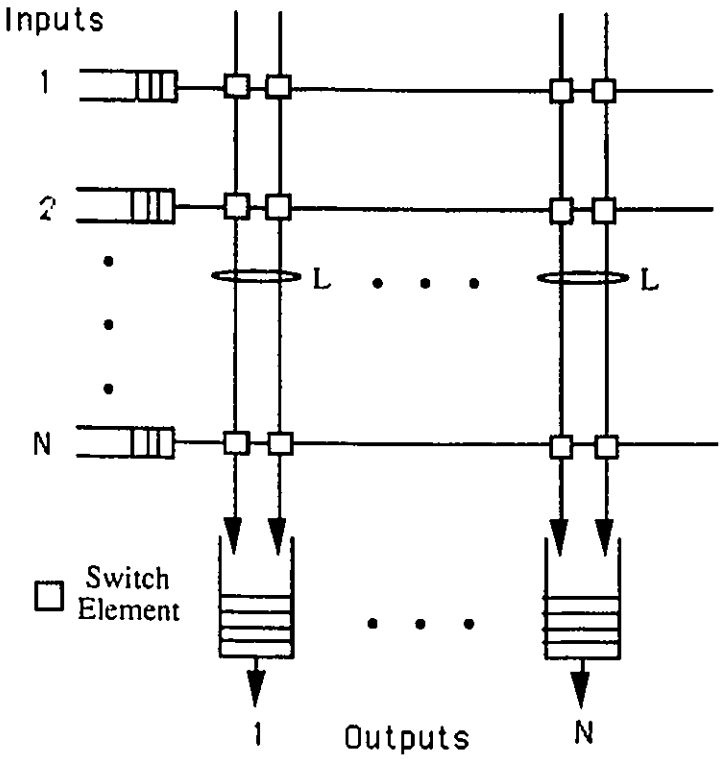


Figure 4.17: Implementation of the Speed-up Switch using a Crossbar Architecture

Each time slot is divided into two sub-slots, where the first sub-slot is used to resolve the contention and the second sub-slot is used to transfer the user packets from the input buffers to the output buffers. The first sub-slot is further divided into N mini-slots, one mini-slot for each input port. The mini-slots are denoted as $M_1, M_2, \ldots M_N$. Each mini-slot is of n bits, where $n = \log_2 N$.

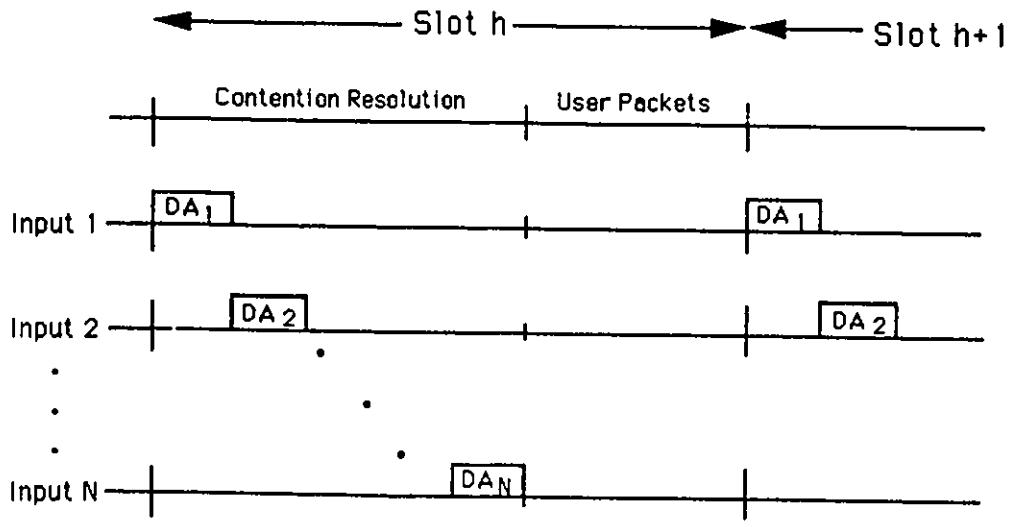
For a moment, let us ignore the issue of fairness in serving the input queues, and assume that inputs are served in cyclic order, starting always with input port 1. In this case, the HOL packets of input buffers that are at the beginning of the switch cycle have better chance to be routed to their destined output buffers, while the HOL packets that are served at the end of the cycle are less likely to find a link to their destination output ports and are less likely to find a free-space in the output buffers.

Let us indicate the destination addresses of HOL packets of input buffers $1, \ldots N$

as $DA_1, \ldots DA_N$. Each mini-slot is used by one input port to reserve a link to the destination of the HOL packet. In the first mini-slot, the destination address of the HOL packet of input port 1 (DA_1) is transmitted. In subsequent mini-slots, other inputs, one by one, transmit the destination address of their HOL packets (for timing diagram, see Figure 4.18). For the time being, let us also ignore the back-pressure from the output buffers. During each mini-slot, at the most one link (out of L links) per output port is 'active', i. e., only one link (per output port) is looking for the corresponding output port address. At the beginning of each time slot, links $l_1^o, o = 1, \ldots N$ are active. When a link (let us say l_1^o) identifies the address (say in mini-slot M_i), l_1^o is reserved for input i to switch its data packet for output o, in the second sub-slot of the time slot. In the next mini-slot, l_2^o is active for the output port o. If more than L packets are destined for a given output port (say j), then all L links are reserved and $j - L$ packets are blocked at the HOL positions. For a given output port, a link remains active in consecutive mini-slots till it is reserved by an input port or till the mini-slot period (i. e., the first sub-slot) ends An input port which is successful in reserving a link, gets an 'ack' immediately from the switching element. A cross-point corresponding to a reserved link l_k^o for input i can be identified by (i, o, k)-tuple.

At the beginning of the second sub-slot of a slot, the selected crosspoints are set to bar-state and all other crosspoints are set to cross-state (see Figure 4.19). All the input ports which are successful in reserving a link, transmit their HOL packets simultaneously in the second sub-slot of the slot. The major advantage of this scheme is that the contention resolution is done completely in a distributed fashion at the cross-points of the switch fabric, and an arbiter is not required for each output port, as proposed in [73]. The switch also has the self-routing property.

In the above N mini-slots, each of n bits represent the switching overhead. Assuming user packets of 53 bytes, a 16×16 switch will have an overhead of 15% and a 32×32 switch will require an overhead processing of about 38%. An implementation of the speed-up switch using Batcher-Banyan network was proposed in [74]. A 'probe-ack' contention resolution mechanism was adopted to select the conflict-free paths for the packets to be



DA_i : Destination Address of HOL Packet at Input-i

Figure 4.18: Timing Diagram for the Contention Resolution and Packet Switching in the Crossbar Architecture

transferred through the interconnection network. In such an implementation, considerable additional hardware is needed. The speed-up required is also quite high because several stages of the network have to be traversed by the probe, ack, and data packets in any given time slot.

Now, we discuss how the back-pressure mechanism can be implemented in a simple manner in the crossbar speed-up switch. In the beginning of a slot, if there are less than L free buffers at an output port, then some of the output links are made accordingly inactive for that output port in that slot. As one packet is transmitted from a non-empty output buffer in every slot, therefore, in each slot, at least one buffer is available at every output port. This is the basis for making the first link of each output port $(l_1^o, o = 1, \ldots N)$ active, at the beginning of every time slot. In the case when there are only k $(1 \leq k < L)$ free buffers at a given output port, then the first k links can be made active during that slot. This ensures that the output buffers never overflow. To make the higher $L - k$ links inactive for an output port, a signal from the buffer control unit is generated during the mini-slots to indicate to the switching elements that these links

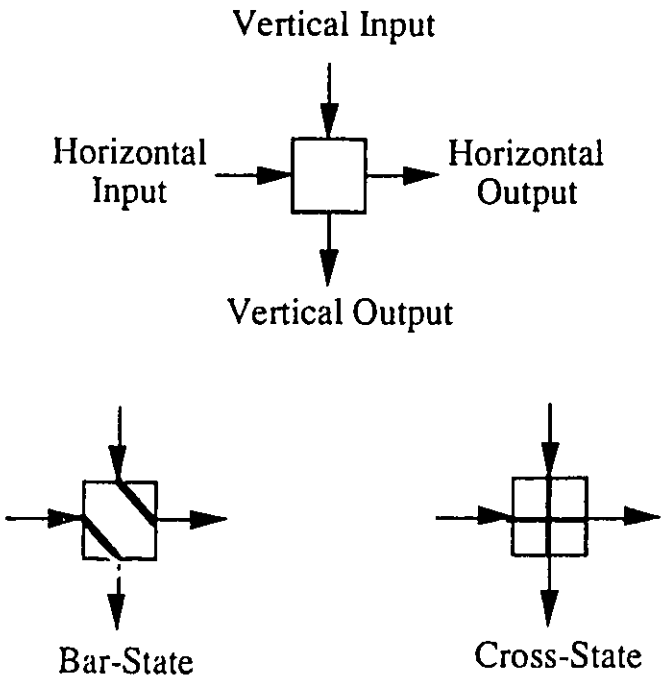


Figure 4.19: A Switching Element and its States

are not available (Figure 4.20). In a multistage interconnection network, composed of speed-up switch modules, the back-pressure may have to be carried from one stage to the previous stage, such that the packet loss occurs only at the input buffers of the first-stage. In that case, we may not be able to transmit one packet from a non-empty output buffer of intermediate switch modules in every slot, and in a time slot all the L links of an output port can be inactive.

4.4.2 Fairness in the Speed-up Switch

Now, we come back to the unfairness problem. As mentioned earlier, the cyclic scanning mechanism results in uneven distribution of the switching capacity among the input ports. The unfairness problem in a pure input buffered Batcher-Banyan switch has been considered in [75], and for an output buffer constraint switch it has been addressed in [76], [77].

To overcome the unfairness problem, it is obvious that several selection mechanism are possible. A simple modification to the cyclic scanning is to increment in each cycle the

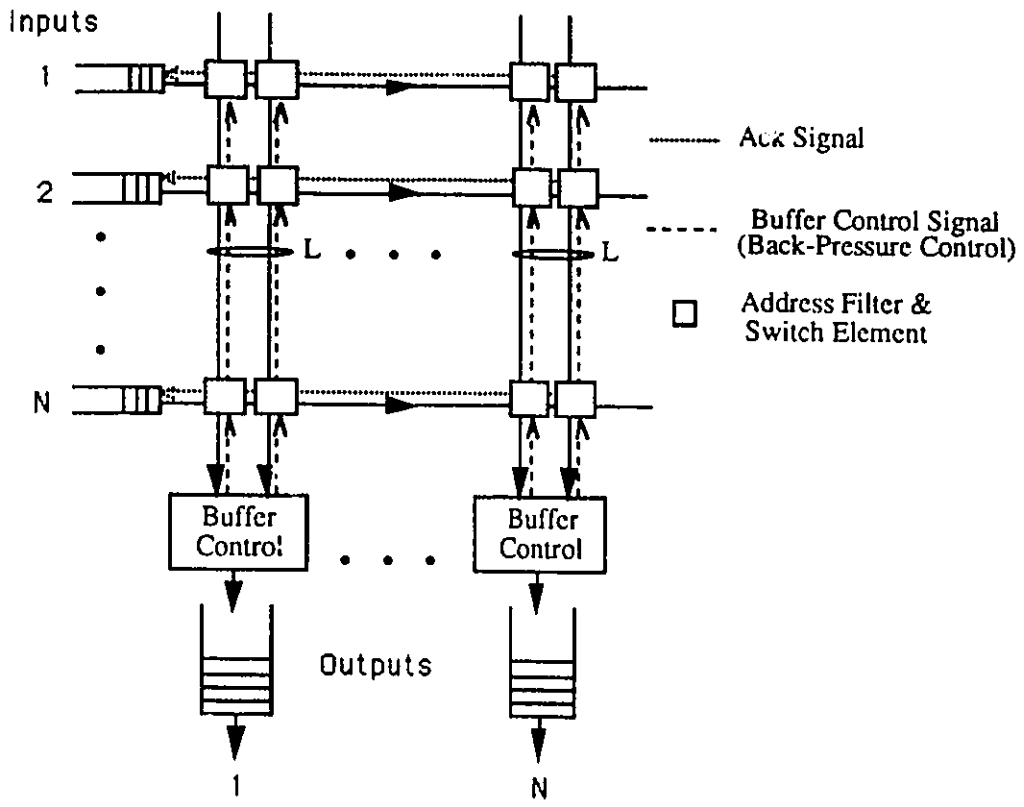


Figure 4.20: Implementation of the Back-Pressure Mechanism in a Speed-up Switch

input port number by 1 at which the selection starts. Once we start selection from input port N, in the next cycle, we again start scanning from input port 1. In addition to the modified cyclic scanning, two more service mechanisms have been considered in [76], [77]. In one of these service mechanism, packets which appear at the HOL position in a given time slot (say h) will be selected before all the other packets at later slots ($h+1, h+2, \ldots$), which are destined for the same output port. Among those cells which arrive in the same time slot and are contending for the same output queue, cyclic selection is performed. The other service mechanism is basically a combination of the above service mechanism and the modified cyclic scanning. It has been shown that this service mechanism offers the best performance in terms of fairness and cell loss rate, but more difficult to implement.

4.5 Summary

In this chapter, a synchronous nonblocking switch has been considered. Such a switch may have speed constraint, output buffer constraint or both. Under output buffer constraint, the performance parameters studied are the maximum throughput, mean delay through the switch and the packet loss probability. Although an infinitely large switch was considered, simulation demonstrates that these results are equally applicable to a finite size switch of 16×16 or larger.

The switch was also analyzed under the speed and output buffer constraints and results show that the switch with speed-factor of 3 gives performance very close to the input and output buffering switch with no speed constraint. This result is of a great practical importance as a switch with higher speed-factor may be difficult to implement. The buffer allocation in such a switch was also considered. To achieve a given packet loss rate, the switch with $L = 2$ requires almost the same amount of input and output buffers as with $L = 4$ up to about 70% input load, but as the load increases beyond 70%, the switch with $L = 4$ would require more output buffers and less number of input buffers when compared to the switch operating at $L = 2$. The performance of the switch with $L = 3$ is very close to the switch with $L = 4$ and therefore not considered in these discussions.

We also studied a simple implementation of the switch with speed and output buffer constraints. Using a back-pressure mechanism from the output queues, the packet loss at the output queues is avoided.

Chapter 5

Limited Intermediate Buffer Switch Modules and their Interconnection Networks

5.1 Introduction

Our second approach to achieve a high-performance packet switch is based on the crossbar switch fabric. A crossbar switch with only input buffers suffers from output blocking and the head-of-line (HOL) blocking. To reduce the HOL blocking, buffers can be provided within the switch fabric.

A bus matrix switch was described in [17], providing 16 Kbytes of FIFO buffer at each cross-point (a variable packet length of 3 to 512 bytes was considered). The design of the switch was restricted to a 2×2 matrix, realizable on a single LSI chip due to the hardware limitations. A crossbar switch with FIFO buffers at each cross-point was also discussed in [18] under the name of a Butterfly Switch. In a recent paper, Kato et al. [19] described the implementation details of a crossbar switch having a dual port RAM of 16 cells (packets) at each cross-point.

Here, we propose to have a single buffer at each crosspoint of the switching array. This reduces the HOL blocking and improves the performance considerably. We call it

Limited Intermediate Buffer (LIB) switch. The most important feature is that the switch fabric operates at the speed of the input-output port. We consider several policies for selecting a packet to be forwarded to an output port from the intermediate buffers and show that the performance of the switch module is influenced by the selection policy. A novel scheduling scheme based on HOL blocking is proposed, which improves the performance significantly. For a uniform random traffic, a 16×16 LIB switch can achieve a throughput equal to 87.5%. We also examine the switch performance under two delay dependent priority classes and show that the achievable throughput can be increased to 91%.

To build large size switching systems, a multistage interconnection network is used, which meets the demands of large scale ATM switch design, such as (1) modularity, (2) relaxed synchronization, (3) guaranteed high performance (i. e., high throughput, low variability of delay) without requiring internal speed-up, and (4) maintaining the packet sequence integrity. The simulation results of three-stage interconnection networks demonstrate the efficacy of the LIB switch architecture and the proposed scheduling scheme.

5.2 Limited Intermediate Buffer Switch

In this section, we describe the LIB switch architecture and discuss some scheduling policies for selecting a packet to be transmitted from the intermediate buffers belonging to a specific output port.

5.2.1 Crossbar Switch with Limited Intermediate Buffer

Let us focus on a crossbar switch fabric which has a FIFO queue preceded by an address filter [8], [1] at each cross point (Figure 5.1). A packet can only pass through the filter whose address matches the packet's destination address. This does not require a centralized controller and the switch fabric speed can be equal to the speed of the input/output port. However, as mentioned in [1], there are two drawbacks to this approach. First, the

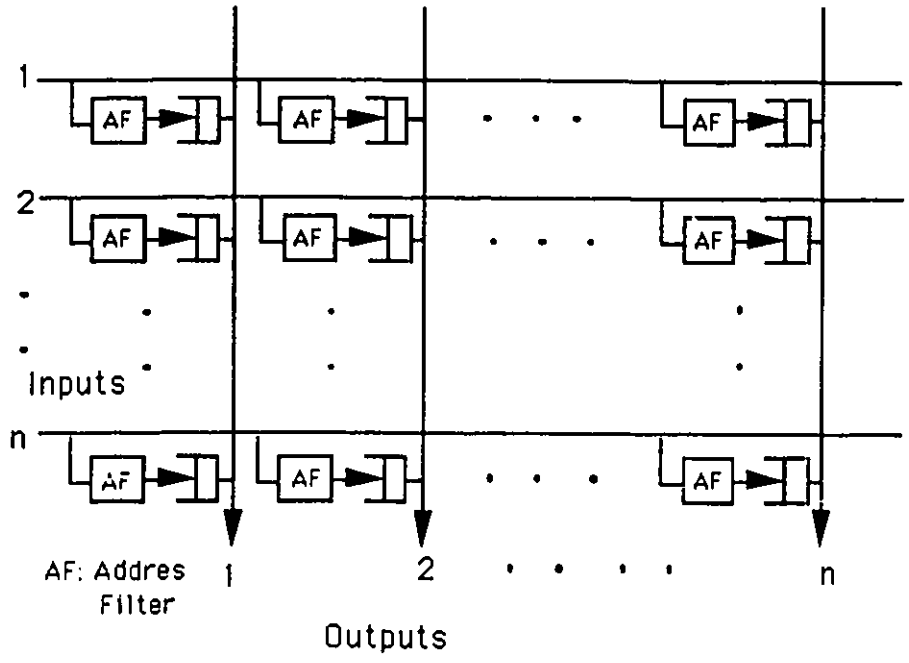


Figure 5.1: A Crossbar Switch with FIFO Buffers at Crosspoints

total memory required is much greater, because there are n^2 queues (in a switch module of $n \times n$) and there is no sharing among them. The second drawback is that combining such a large buffer memory with the switching fabric would drastically limit the size of the switch implementable on a single chip.

To overcome these problems, we suggest that there should only be a limited buffer at each crosspoint, but input queues be provided to minimize the packet loss rate to the desired limit (Figure 5.2). By providing a dual port buffer of 53 bytes capable of storing an ATM cell at each cross-point, a throughput of 87.5% can be achieved in a switch of size 16×16. We call this Limited Intermediate Buffer (LIB) switch. The input buffers are outside the switching fabric and thus do not pose any implementation problem. With current VLSI technology, such a 16×16 or even larger size switch module can be fabricated on a single chip. To understand the functioning, we consider an equivalent model of the packet switch of size $n \times m$ as shown in Figure 5.3, which is similar to the

abstract model for space division type switches presented in [1]. In this model, the

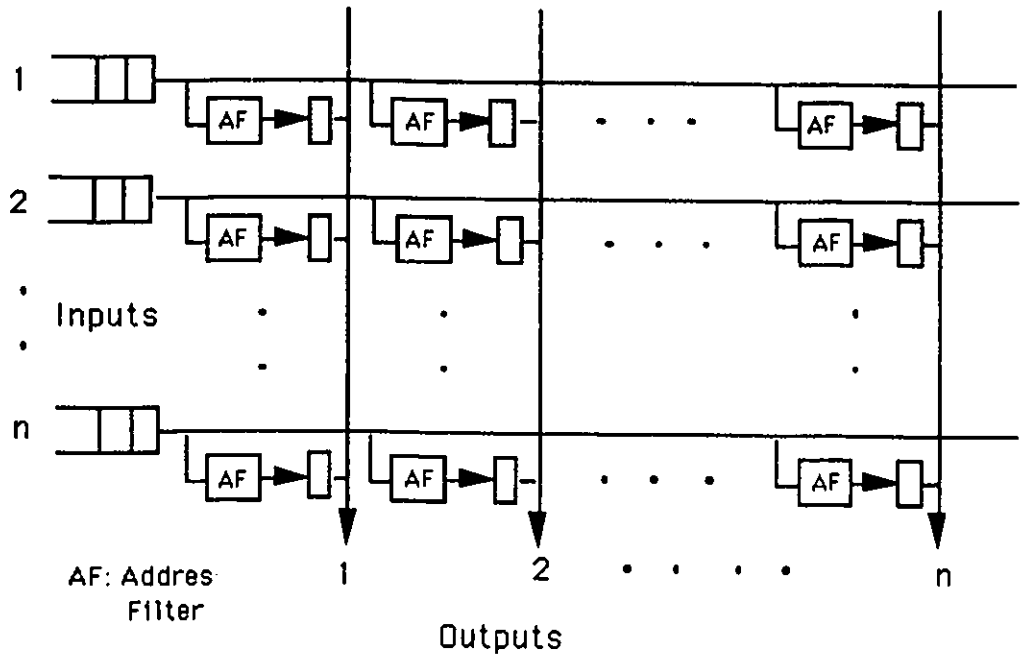


Figure 5.2: A Crossbar Switch with Single Buffer at Crosspoints and Input Queueing

intermediate buffers correspond to the buffers provided at the cross-points of the crossbar switch. For each input line i, there is a router (demultiplexer) which routes its packets to m separate intermediate buffers, numbered $(i, 1)$ through (i, m) according to the output address of the packets. At each output line j, there is a concentrator (multiplexer) which connects n intermediate buffers $(i, j), i = 1, 2, \ldots n$ to output line j, one from each router. The switch operates synchronously, i. e. all the routers and concentrators of the module are synchronized to a single clock. In each time slot, the concentrator selects one packet, if any, from its intermediate buffers, for transmission on to the output line. Similarly, the router forwards a packet in each time slot from the head of its input queue to the intermediate buffer, if it is available. If the corresponding intermediate buffer is not available, the packet remains at the head of the input queue. In other words, the back-pressure is applied and the packet waits at the head of its input queue. It is easily understood that by providing nm intermediate buffers, the HOL blocking is reduced considerably and this

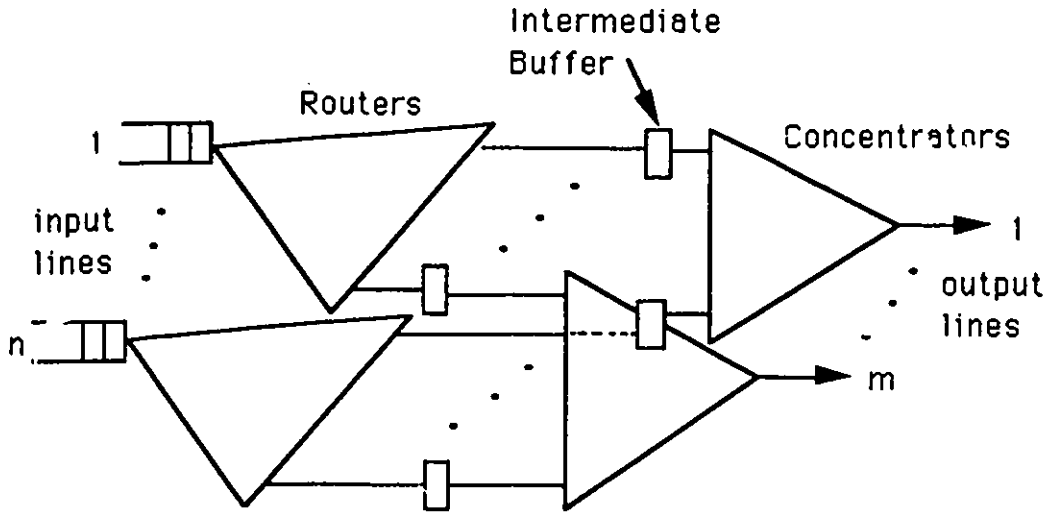


Figure 5.3: A Model for the Space-Division Packet Switch having Limited Intermediate Buffers

improves the performance of the switch. The implementation of such a switch is simple, as each router only requires information of the packet, which is at the head of the input queue, and the concentrator selects one packet for transmission on to the output port from n intermediate buffers. The self-routing property of the switch is maintained.

5.2.2 Scheduling policies in a LIB Switch

It is obvious that several scheduling policies are possible for selecting a packet to be transmitted from the n intermediate buffers belonging to the specific output port. As we will see, the performance of the switch module is sensitive to the selection policy adopted. We consider the following three policies:

- *Random Selection:* the concentrator randomly selects a packet from the set of contending packets for the given output.
- *FIFO Selection:* the concentrator selects packets for transmission on a FIFO basis with reference to the packet arrival at its intermediate buffers. However, in any time

slot, if more than one packets arrive at its intermediate buffers, they are randomly prioritized among themselves.

- *Selection Based on Head-of-Line Blocking:* in this selection policy, the concentrator selects a packet from that intermediate buffer which is causing head-of-line blocking at its input queue. If there is no such packet, it selects a packet on a FIFO basis. We refer to this policy as HOL priority selection.

For the clear understanding of the HOL priority selection scheme, let us consider that a packet (i. e., 'tagged' packet) at the input line i is destined to output port j. If the intermediate buffer (i, j) is not available, the tagged packet is blocked at the head of the input queue i. At this moment, the router i sets a bit corresponding to the buffer (i, j), to indicate that the packet of the buffer (i, j) is causing HOL blocking and, therefore, should be given higher priority. In this time slot, if only one such request is received by the concentrator j, the packet is selected for transmission at the end of the slot and in the next slot the tagged packet moves to the intermediate buffer. If, there is more than one such packet, any packet is selected from among those causing HOL blocking.

We found that a big increase in the achievable throughput is possible in the HOL priority selection case in comparison to the random or FIFO selection. Surprisingly, the HOL priority selection also reduces the variability of the delay through the switch, which is an important consideration in Broadband ISDN. However, the implementation of the priority scheme requires some sort of hand-shaking protocol between the routers and concentrators. The performance of the switch module under the above three selection policies for different traffic patterns is discussed in the next section. It is to be noted that the ordering of the packets is maintained in all these selection policies.

5.3 The Performance of the LIB Switch

In this section, we present the simulation results for the Limited Intermediate Buffer switch under three selection policies, namely, random, FIFO and selection based on head-

of-line blocking (i. e., HOL priority selection). The performance parameters studied are the maximum throughput, the mean delay and the variance of the delay. We assume that there are infinite buffers at each input port.

5.3.1 Traffic Models

To study the performance, we consider different traffic models as follows:

- *Uniform Traffic:* In a time slot packets arrive at each input port according to a Bernoulli process. Each packet has equal probability of being addressed to any given output. The packet arrival at any input port is independent of arrivals at other inputs and successive packet arrivals are independently destined for the outputs.

- *Unbalanced Traffic:* This traffic model is as follows. The input (output) ports are divided into two equal sets called I_1 and I_2 (O_1 and O_2). A packet arriving at any input of the set I_1 (I_2) is assigned any output from the set O_1 with equal probability p_{11} (p_{21}) or it is assigned any output from the set O_2 with equal probability p_{12} (p_{22}). Obviously, $p_{11} + p_{12} = p_{21} + p_{22} = 1.0$. For simplification, we take $p_{11} = p_{22}$ (Figure 5.4). The difference in p_{11} and p_{12} represents the amount of imbalance in the traffic, i. e., the more the difference in p_{11} and p_{12}, the more the imbalance in the traffic. It is assumed that the packet arrival at each input is an independent Bernoulli process and each output carries the same amount of load as in the case of uniform traffic.

- *Bursty Traffic:* The input traffic can be characterized by a two state Markov chain which alternates between burst and idle periods. The burst length, as well as the idle period, are geometrically distributed. The minimum burst length and the idle period are of one packet-length each. We consider two types of bursty traffic: the one with a mean length of 10 packets and the other with a mean burst length of 15 packets. However, in each case, it was assumed that an arriving packet has equal probability of being addressed to any given output port. The basis of this

assumption is that the peak cell rate of each connection is only a small fraction of the total capacity of an input-output port, and therefore, the assignment of output addresses within a burst may be quite random.

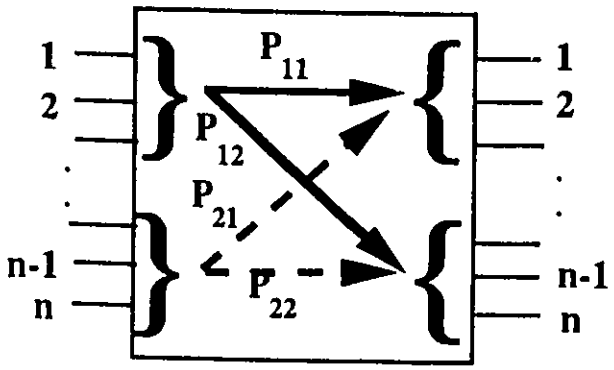


Figure 5.4: Unbalanced Traffic Model

In all above traffic models, each input-output port is equally loaded on the average. In practice, the average traffic carried by each input-output port should be less than a fixed percentage of the maximum (saturation) throughput, e. g., 90%. Let us call this the threshold value. In reality, the load on each input/output port can be limited to the threshold value by monitoring the sum of all virtual calls on each input-output port. A call is added on to an input-output port pair on the basis that the sum of loads of virtual calls on the input as well as the output port does not exceed the fixed threshold value. If it exceeds that, the call is rejected or routed through some other alternate path. Therefore, each input-output port carries a load equals to or less than the threshold value. This justifies the assumption that each input-output port carries the same amount of load, in the above traffic models.

5.3.2 The Performance of a Symmetric LIB Switch

First, we consider the performance of an $n \times n$ LIB switch under the uniform traffic. To examine the maximum (saturation) throughput of the switch module, we assume that the

input queues are saturated so that packets are always waiting in every input queue. In an input buffered nonblocking packet switch, the maximum throughput decreases with n and approaches 58.6% as n goes to infinity [16]. In the LIB switch, under FIFO and random selection, we note an interesting phenomenon. The maximum throughput first decreases and then increases with n, as shown in Figure 5.5. The reason for this is as follows: there are two factors which are influencing the performance of the switch. First, the HOL blocking increases with n, which has a negative effect on the throughput. The second factor is that with n the amount of the intermediate buffers increases (n^2 packets), which has a positive effect on the throughput. The net result of the above two factors determines the saturation throughput. As illustrated in Figure 5.5, the saturation throughput is only slightly better in the case of the FIFO compared to that of the random selection. The HOL priority selection leads to a big increase in the achievable throughput. For example, a 16×16 switch has 87.5% throughput under the HOL priority selection in comparison to that of about 81% under FIFO/random selection and of 60% in pure input-buffered nonblocking packet switch (NPS), which is also shown in Figure 5.5.

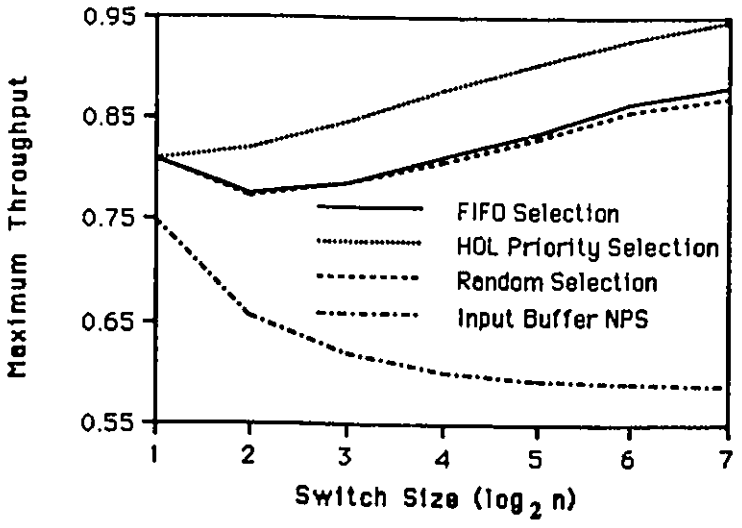


Figure 5.5: Maximum Throughput Achievable in a LIB Switch Module for Uniform Traffic

In fact, the performance of the LIB switch architecture is a compromise between the input-buffered NPS and the look-ahead (windowed) resolution scheme [40] (also referred to as a by-pass queue). The performance of an $n \times n$ LIB switch can be compared with that of an $n \times n$ input-buffered NPS employing look-ahead contention resolution of window-size (w) equals to n (since, in a LIB switch there are n intermediate buffers attached to each input). However, the LIB switch is not exactly the same as the look-ahead contention resolution NPS. The working of the LIB switch implies that the contention resolution is limited to a depth (i. e., window size), where we hit a second packet destined to an output port. To illustrate this point, let us consider a LIB switch under FIFO selection scheme. From an input queue, one packet in each slot can move to the intermediate buffers until the HOL packet is blocked. This happens when the HOL packet has same destination as that of one of the earlier packets of this input queue, which are still in the intermediate buffers. This implies that the LIB switch is somewhat similar to providing a look-ahead contention resolution of variable window size, which ranges from 1 to n, depending upon at what position we find a packet with a repeated address in the input queue. This also implies that the window size in a slot is dependent on the window of the previous slot. In the case of the HOL priority selection scheme, after finding a packet with a repeated address, we try to make the window size (in the next time-slot) independent of that in current slot, by giving higher priority to the packet which is causing HOL blocking. This automatically leads to an enlarged window in the subsequent slots and thus higher throughput. There is one more difference, in the working of the LIB switch and that of the look-ahead contention resolution NPS. In a time slot, it is possible that in a LIB switch more than one packet from the intermediate buffers of an input are transmitted to the output ports, whereas in a look-ahead contention resolution NPS more than one packet from an input queue is never transmitted. Figure 5.6 shows the maximum throughput of the LIB switch and that of the look-ahead contention resolution NPS with window size equals to 4 and 8. The performance of a 16 $\times$ 16 LIB switch under FIFO selection is nearly the same as a 16 $\times$ 16 NPS with $w = 4$, and under the HOL priority selection it is close to the performance of NPS with $w = 8$.

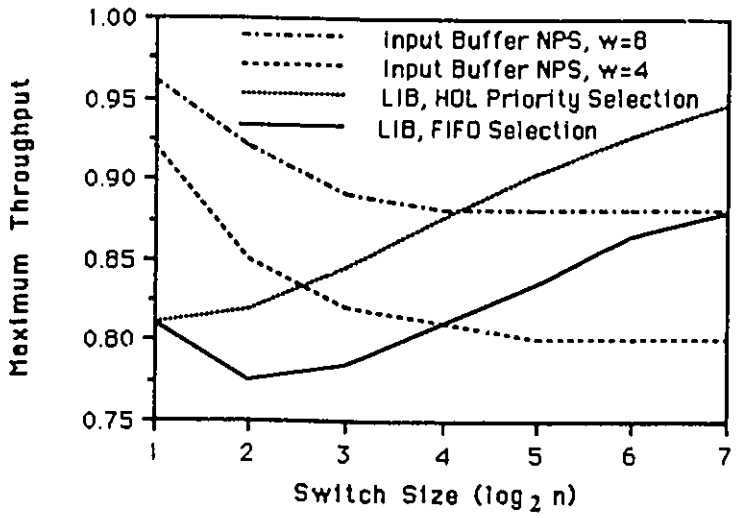


Figure 5.6: A Comparative Study of Maximum Throughput in a Look-ahead Contention Resolution NPS and in a LIB Switch (Uniform Traffic)

From the overhead processing point of view, as mentioned earlier, the look-ahead contention resolution requires that the centralized controller should scan all the input queues up to a depth of w. The queues should also be organized in such a way as to permit the extraction of the packets selected for transmission.

Now, we examine the delay performance of the LIB switch. The delay of a packet consists of waiting time in the input queue until it reaches the head-of-line (HOL), waiting time at the head-of-line due to the HOL blocking and waiting time at the intermediate buffer. Figure 5.7 shows the mean delay (number of slots) through the switch of size 16×16 under the above three selection policies. It is to be noted that when the input load is 80% of the saturation throughput, the mean delay is two time-slots and in case of 90% load it is three slots, which is tolerable for most applications. The variance of the delay through the switch is illustrated in Figure 5.8, which shows that the HOL priority selection results in the lowest variance of the delay at a given load and therefore, it should be preferred over the other two selection policies.

Figures 5.9 and 5.10 show the maximum throughput for unbalanced traffic under FIFO and priority selection scheme, respectively. The throughput for the uniform traffic

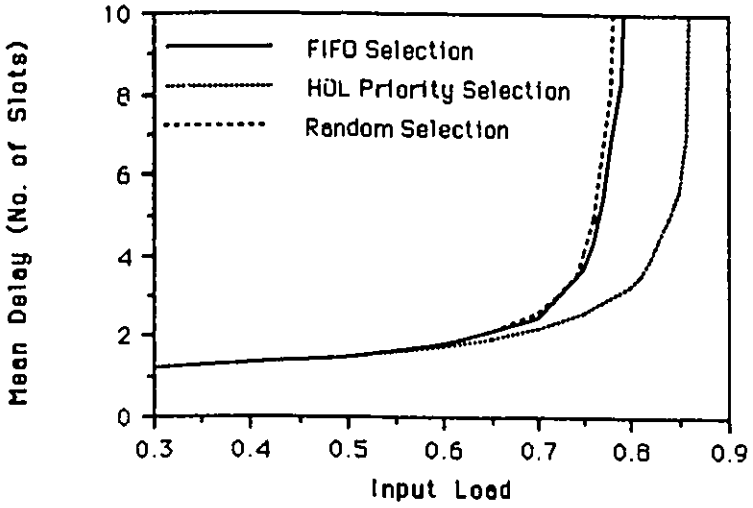


Figure 5.7: Mean Delay in a 16 × 16 LIB Switch with Uniform Traffic

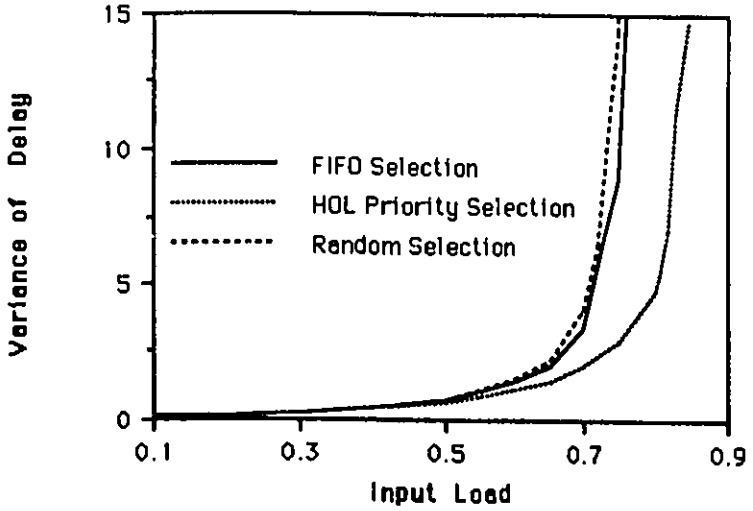


Figure 5.8: Variance of Delay in a 16 × 16 LIB Switch with Uniform Traffic

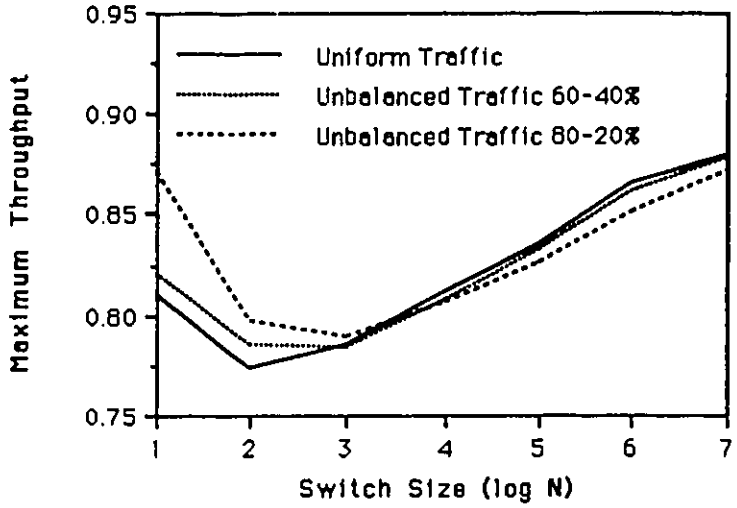


Figure 5.9: Maximum Throughput Achievable in a LIB Switch with Uniform and Unbalanced Traffic under FIFO Selection

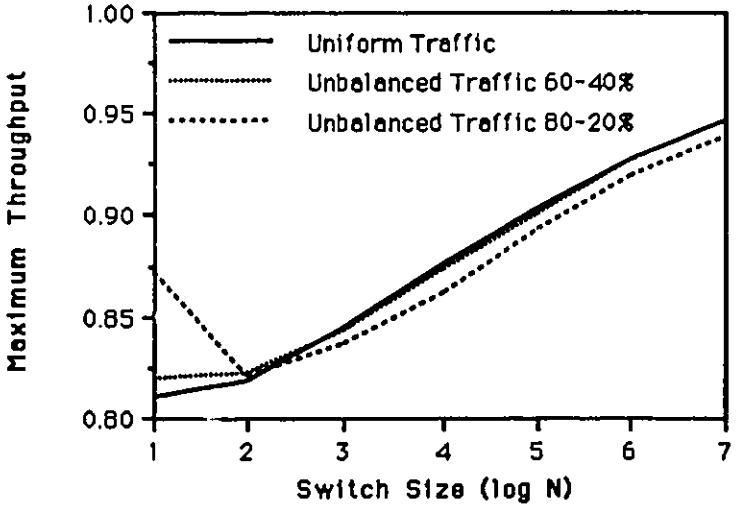


Figure 5.10: Maximum Throughput Achievable in a LIB Switch with Uniform and Unbalanced Traffic under HOL Priority Selection

is also shown in these figures. Two different sets of output port assignment probabilities are considered. First, $p_{11} = p_{21} = 0.6$ and $p_{12} = p_{22} = 0.4$, which is indicated as unbalanced traffic of $60 - 40\%$ on figure 5.9 and 5.10. In the second set, $p_{11} = p_{21} = 0.8$ and $p_{12} = p_{22} = 0.2$, and it is indicated as unbalanced traffic of $80-20\%$. The ratio of $80-20\%$ represents that the traffic is more unbalanced than the first case of $60 - 40\%$. For FIFO selection (Figure 5.9), the switch of size 8×8 or larger has almost the same performance for the uniform and the $60 - 40\%$ unbalanced traffic. For a 16×16 switch the performance does not degrade appreciably under unbalanced traffic of $80 - 20\%$. In this case, the maximum throughput is about 87% in comparison to 87.5% for the uniform traffic. However, as would be expected, the performance is better under unbalanced traffic than that under the uniform traffic for a smaller switch size. The examination of the switch under the priority selection scheme for the unbalanced traffic (Figure 5.10) suggests that the degradation in the performance is similar to the degradation in the case of FIFO selection (Figure 5.9). However, for the 16×16 switch, the maximum throughput for $80 - 20\%$ unbalanced traffic drops to 86% from 87.5% for the uniform traffic. The delay of a 16×16 switch was observed to be little more but not much different in the case of unbalanced traffic than in the case of uniform traffic.

Figure 5.11 shows the mean delay under the HOL priority selection for the bursty traffic and also for the uniform traffic for comparison purpose. The mean delay is almost the same for the uniform and the bursty traffic when the input load is less than 40%. But as the load increases beyond this, the performance for the bursty traffic deviates from that of uniform traffic and the deviation is proportional to the burst length. At 80% input load this deviation is about one slot for the traffic with mean burst length equal to 10 packets and is two slots for a load with mean burst length of 15 packets, with respect to the uniform traffic. It is to be noted that the uniform traffic at 80% input load is equal to the bursty traffic of mean burst length 5 packets.

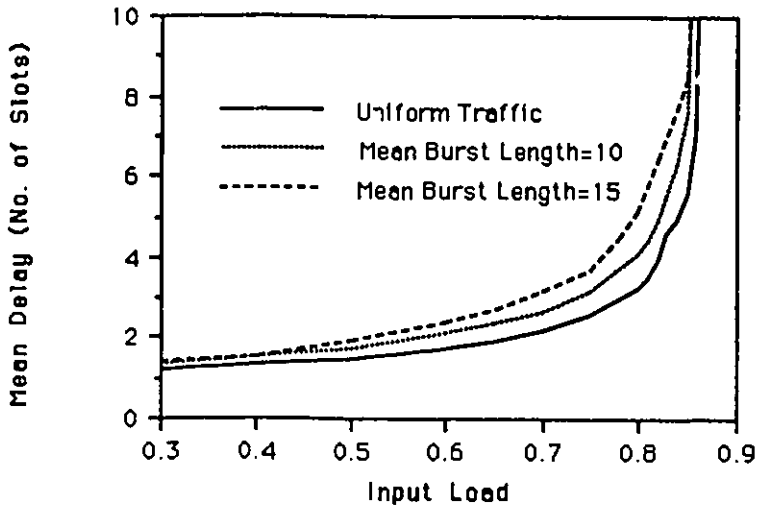


Figure 5.11: Mean Delay in a 16 × 16 LIB Switch with Uniform and Bursty Traffic (HOL Priority Selection)

5.3.3 Performance of a 16 × 16 LIB Switch under Two Delay-Priority Classes

At the ATM Adaptation Layer (AAL), four classes of service have been defined by CCITT on the basis of the timing relation required between source and destination, bit rate (constant or variable), and connection oriented/ connectionless traffic [78]. We believe that at the ATM layer, these four classes of service may be implemented using two classes of delay dependent priorities, the higher level priority for isochronous traffic and the lower level priority for non-isochronous traffic. Such a priority can be implemented in the LIB switch, in a simple manner as follows. A high priority packet arriving at an input moves ahead of all the low priority packets waiting for the service in the input queue, and if the HOL position is currently occupied by a low priority packet, it is preempted by the high priority one. There are many possible ways for a concentrator to select a packet from its intermediate buffers. To maintain the simple structure, we can just have the FIFO selection or the HOL priority selection at the intermediate buffers level as discussed

in the Section 5.2.2. Or, a more complex concentrator at each output port can provide isochronous/ non-isochronous traffic priority in addition to the HOL priority. In this section, we study the performance of a 16 × 16 LIB Switch under the above two possible scenarios.

Assume that for the uniform traffic, λ_H and λ_L be the rate of packet arrivals for high and the low priority traffic, respectively. It has been shown that in an input-buffered nonblocking packet switch, the maximum throughput under two priority classes exceeds that of a single priority class traffic [79]. The reason is that in a two priority switch, the preemption of low priority packets from the HOL positions by the high priority arrivals modifies the distribution of the HOL packets, and thus, the destinations of the contending packets in the successive slots change randomly. In other words, the arbitration of the HOL contending packets increases from slot to slot, which contributes to the higher throughput. In the case of the LIB switch, the increase in the maximum throughput is more noticable.

Under the two priority classes, the maximum throughput is illustrated in Figure 5.12, as a function of high priority load λ_H. In this case also, we assumed that the input queues are saturated. Providing priority on the basis of two priority classes at the intermediate buffers does not alter these curves. It can be seen that under the HOL priority selection, the maximum throughput increases to a maximum of about 0.911 (in comparison to 0.875 under single priority traffic) at $\lambda_H \simeq 0.62$. Under FIFO selection it reaches to about 0.848 at $\lambda_H \simeq 0.54$. This figure demonstrates that a gain of about 3.5% is possible in the maximum throughput in the case of two priority classes.

Figure 5.13 shows the mean delay (number of slots) of the high priority packets (for $\lambda_H = 0.2$ to 0.5) under HOL priority selection as a function of low priority load (λ_L). Although low priority traffic affects the delay performance of the high priority, it is not severe for total loads ($\lambda_L + \lambda_H$) below 70%. For example, for $\lambda_H = 0.3$ the delay increases from 1.1 slots (at zero low priority load) to 1.8 slots at $\lambda_L = 0.4$.

We also studied the performance under bursty two priority traffic, and observed that the performance of the high priority traffic declined almost negligibly, although the

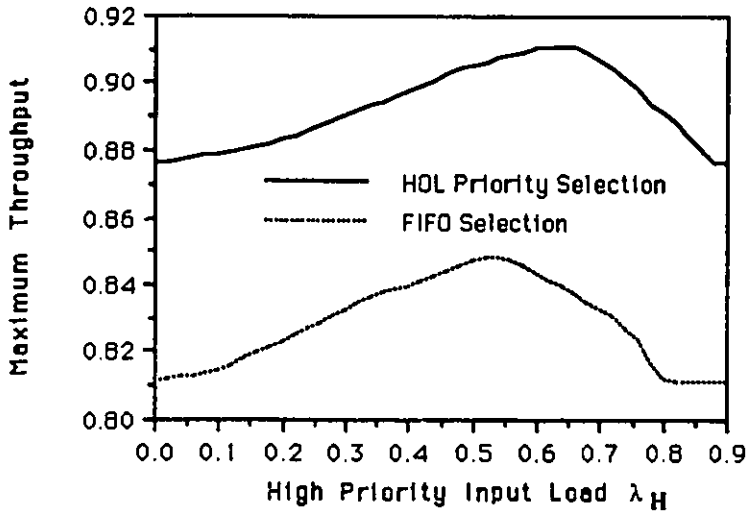


Figure 5.12: Maximum Throughput Achievable under Two Priority Classes (16 × 16 Switch, Uniform Traffic)

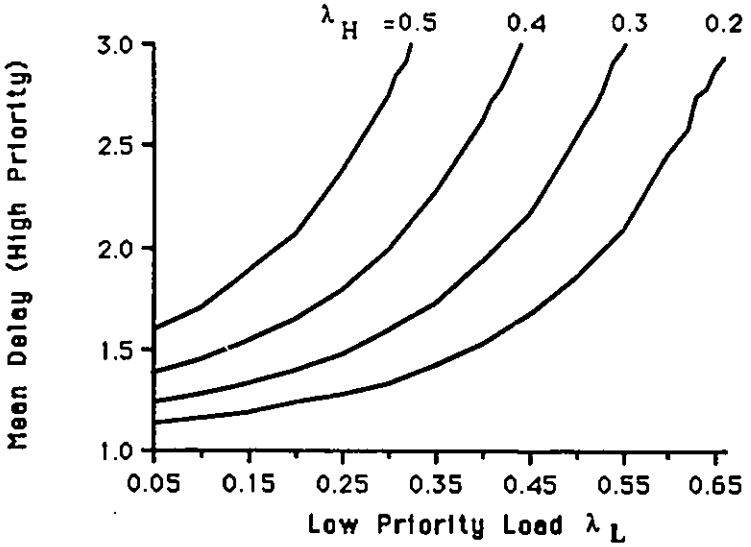


Figure 5.13: Mean Delay for High Priority Class Traffic under HOL Priority Selection (16 × 16 Switch, Uniform Traffic)

delay for low priority increased significantly. Figure 5.14 illustrates the mean delay for low priority under bursty and also under uniform traffic. The high priority load was assumed to be 0.3 for these curves. We observe that as the low priority load increases, the

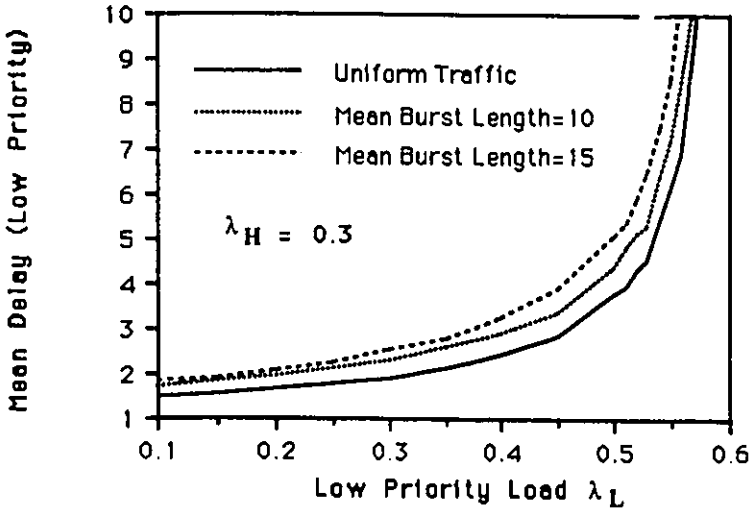


Figure 5.14: Mean Delay for Low Priority Class under Uniform and Bursty Traffic (16×16 Switch, HOL Priority Selection)

performance for the bursty traffic deteriorates from that of uniform and the deviation is proportional to the burst length. At $\lambda_L = 0.5$ ($\lambda_L + \lambda_H = 0.8$), the deviation is about 1.3 slots for the traffic with a mean burst length of 15 packets, with respect to the uniform traffic.

Figure 5.15 provides the comparative study of mean delay for high priority traffic under the two types of selection schemes at the intermediate buffers level. When the selection was based on HOL priority and the two priority classes, the packets causing HOL blocking were given higher priority with respect to the packets of isochronous traffic. In this case also, the presence of low priority packets influences the performance of the high priority packets, but only slightly. This is caused by the low priority packets (in the intermediate buffers) blocking high priority packets at the head of input queues, although it might be just for one slot duration.

The average delay suffered by the low priority traffic, in the presence of high priority traffic of rate $\lambda_H = 0.2$ to 0.5 is illustrated in Figure 5.16, when the selection is based on HOL priority and the two delay-priority classes.

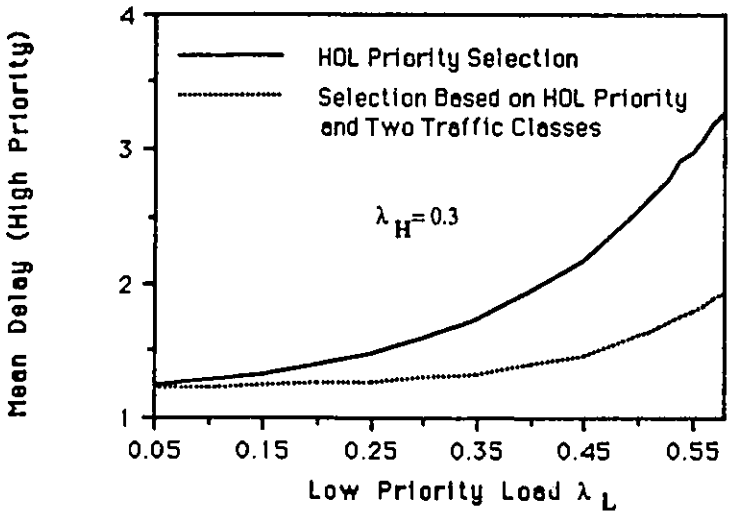


Figure 5.15: Comparative Study of Mean Delay for High Priority Traffic under HOL Priority, and under the selection based on HOL Priority and Two Traffic Classes (16×16 Switch, Uniform Traffic)

5.3.4 The Performance of Asymmetric LIB Switch

In the next section, we have considered three-stage networks for interconnecting the LIB modules to build large size switching systems. In addition to the 16×16 modules, the networks consist of switching elements of asymmetric size and the performance of the interconnection networks depends upon it. Therefore, in this subsection, we would briefly study the performance of the asymmetric LIB switch modules.

Figure 5.17 shows the maximum throughput of a $16 \times m$ switch, where m varies from 16 to 32. This figure exemplifies that the performance of the switch improves drastically by increasing m from 16 to 20, 22, 24 and 26 with diminishing improvements thereafter. Another striking feature is that as m increases, the throughput under FIFO selection

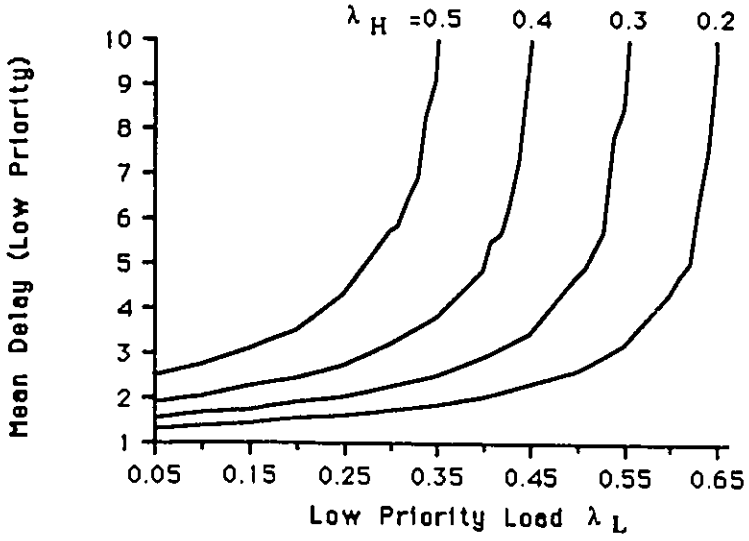


Figure 5.16: Mean Delay for Low Priority Traffic, When selection is based on HOL Priority and Two Taffic Classes (16 × 16 Switch, Uniform Traffic)

approaches to that of the HOL priority selection, and at $m = 32$ the performance is almost the same under the two selection schemes. The reason is that as m increases, the head-of-line blocking decreases and thus the benefit of the HOL priority selection also diminishes.

Under HOL priority selection, the LIB module of size 16 × 26 gives throughput of about 97.5% and module of size 16 × 32 yields throughput of about 98.6%. This suggests that the implementation of the 16 × 26 module may be preferred over the 16 × 32, as the improvement is not significant enough to justify the additional complexity.

5.4 Multistage Interconnection Networks

Several large scale switch architectures have been proposed for high-speed networks [15], [30], [34], [35], [47], [80], [81]. It is difficult to realize a Batcher-banyan switch of large scale because of the stringent synchronization requirements for the self-routing elements at each stage of the switch [80]. From the synchronization point of view, a relatively larger switch

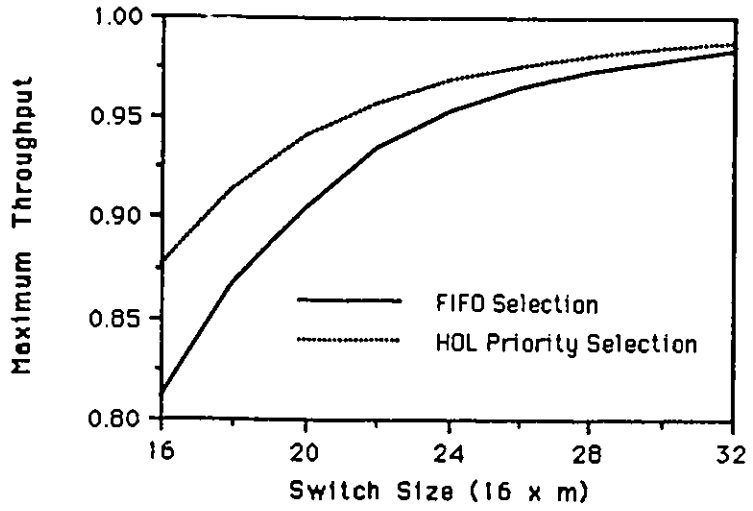


Figure 5.17: Maximum Throughput Achievable in the Asymmetric LIB Switch Module (Uniform Traffic)

module, which can be implemented on a single VLSI chip, appears to be more attractive than a switch module of size 2×2. The crossbar switches with limited intermediate buffer are our basic building blocks for the Multistage Interconnection Networks (MIN). A three-stage network, as drawn in Figure 5.18, can be used to achieve this objective. The switch structure looks exactly like the three-stage Clos Network [82], except that the switch modules are the LIB modules. A three-stage Clos network with N input and output ports can be denoted by $C_{N,n,m}$, where $n \times m$ is the size of the first stage modules. A 5-stage network can be used to build larger size switching systems. In this section, we would consider the case, where $n = 16$, $N = n^2$ and m varies from 16 to 32. When $m = 16$, the network resembles the Benes [83] network. The three-stage network provides m multiple paths between any pair of input and output ports. A path from this set is chosen at the call set-up time and, during the connection of the call, the packets belonging to this call are constrained to follow the same path. This ensures that the sequencing of the packets is maintained. Such an approach for Benes networks has been adopted in [84].

A three-stage interconnection network has been proposed in [29], [30], which consists

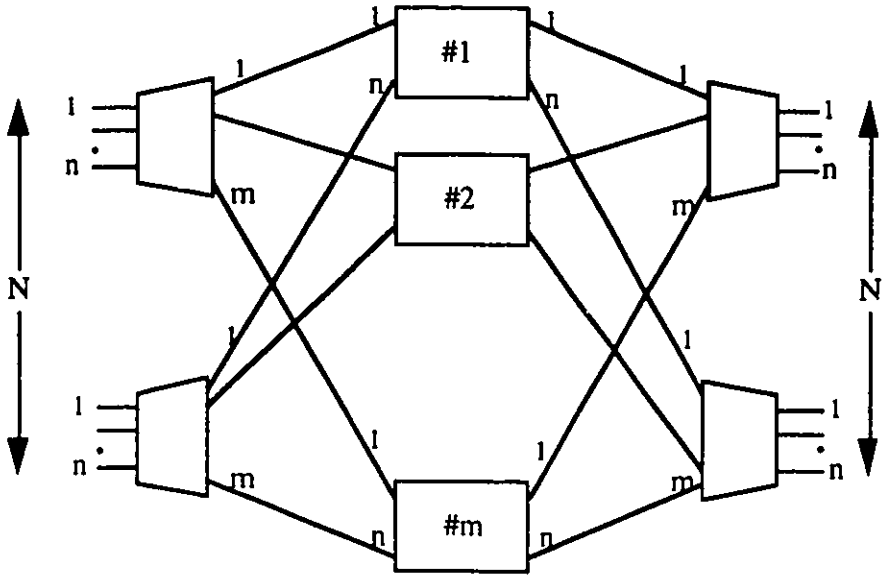


Figure 5.18: A Three-Stage Interconnection Network

of two-stages of memoryless front-end distribution network followed by a third stage of small size switches with output buffers. The implementation of such a network requires the switch fabric to operate at the speed much higher than the input-output port speed, and the speed-up needed grows linearly with the number of stages in the distribution network. Because of the speed constraint and the output port contention within the network, we feel that the buffers inside the MIN are almost impossible to avoid, especially in a high-performance large-scale switching system.

5.4.1 The Routing Mechanism

Even distribution of the traffic within the interconnection network is required to avoid the link congestion. This can be done on a static basis, i. e., at the call set-up time. When a control packet is received by the switching system requesting to set-up a call, the central controller selects a path suitable for this call based on some criteria. If a suitable path is not found, the call is rejected. In case the path is selected, the routing information is supplied by the controller to the corresponding input line interface. During the connection

of the call, when a packet arrives at the input line , this routing information is appended by the interface to the packet header. Based on this information, the packet is self-routed in the interconnection network. Each input line interface maintains a table containing entries, one entry per call passing through it, and it is responsible for appending routing information to the packets. The controller makes the routing decision only once for a call. This centralized routing on a static basis has two advantages: first, the load is more properly distributed and second the processing overhead per packet reduces in the network as it becomes self-routing.

A path is determined out of the possible multiple paths between a pair of input-output ports such that the quality-of-service (i. e., buffer overflow rate, delay) for the existing calls and the new call being set-up is maintained on the route. The criterion for selecting a path has to be on the following basis:

- Existing load on the alternative links.
- Anticipated characteristics of the traffic of the call being set-up (e. g., mean rate, peak rate and average burst length).
- The call belongs to the isochronous or non-isochronous application.

5.4.2 The Performance of the Interconnection Networks

There are two possible ways for interconnecting the modules. In the first possibility, the output of a stage is directly connected to the input of the next stage, except in the last stage where an output acts as a perfect sink. In such an arrangement, the queues are provided only at the inputs of the first stage. Whenever any input port (the router) is unable to accept a packet due to blocking at its intermediate buffer, the output of the previous stage is informed of this blocking situation and the packet is retained there. In this manner, the back-pressure may be carried all the way to the input queues at the first-stage. The length of the input queues is tailored according to the packet loss rate acceptable for a given traffic. The second possible way for interconnecting the

modules is to provide a small amount of buffer between the stages. In some situations, this interstage buffer helps reducing the back-pressure and thus improves the performance of the interconnection network.

First, we consider the scenario where interstage buffers are not provided. However, one packet buffer is assumed in each router of the switch modules, to store the packet temporarily. The uniform traffic model was assumed in the simulation. Figure 5.19 shows the maximum throughput achievable in the 256×256 networks under FIFO selection and under the HOL priority selection schemes, when m varies from 16 to 32. Under HOL priority selection, for small values of m ($\simeq$ 16 to 22) the throughput is quite less than that of a single $16 \times m$ module, but as m increases, the traffic is more spread out and the throughput of the network approaches that of a single module (for comparison, see Figure 5.17). For small values of m, the throughput is limited by the back-pressure from second and third stages, but, it is intuitively clear that as m increases the back-pressure diminishes, leading to better performance. The three-stage network with $m = 26$ can achieve a throughput of 95%. In the case of FIFO selection, the back-pressure is more critical, because a packet at a router may be blocked for several consecutive slots.

We also studied the delay performance of the three-stage network. Figure 5.20 shows the mean delay under HOL priority selection for the case of $m = 26$ and 32. Since, the performance of $m = 26$ is very close to the case of $m = 32$, this again suggests that $m = 26$ provides a better practical solution. The mean delay is less than eight slots at 90% input load.

To ease the back-pressure in the case of FIFO selection or for small values of m in the case of HOL priority selection, the interstage buffers can be useful. Figure 5.21 illustrates the maximum throughput achievable (in the case of FIFO selection and $m = 32$) as a function of the number of interstage buffers, which varies from 1 to 10. The results suggest that a buffer of about 7 packets is good enough for reducing back-pressure considerably from one stage to the other stage.

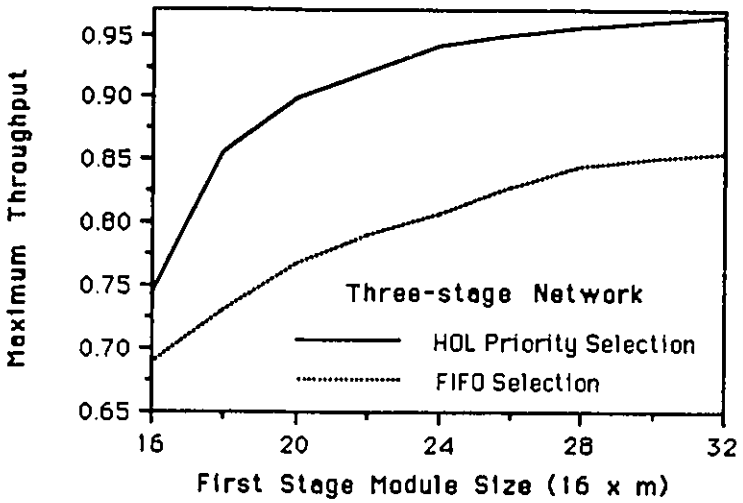


Figure 5.19: Maximum Throughput Achievable in a Three-Stage Network as a Function of m ($N = 256$, $n = 16$)

5.5 Summary

We have presented a high-performance crossbar switch having a single buffer at each cross-point, and also the interconnection networks comprising of such switch modules. The simulation results demonstrate that a switch of excellent performance is possible by introducing buffers of one packet in the switch fabric at places where the output port contention takes place. The most important feature is that the switch fabric operates at the speed of the input-output port, which is quite desirable in high-speed switching systems. Three selection schemes (namely, random, FIFO and the selection based on HOL blocking) were considered for selecting a packet to be transmitted to a given output. Among these, the selection based on HOL blocking results in the best performance, i. e., higher saturation throughput, low mean delay and low variability of delay. The examination of a 16×16 switch module under two delay dependent priority traffic reveals that the maximum throughput can be increased by about 3.5%, in comparison to that of a single priority traffic. The performance of the switch was also examined under two priority bursty traffic.

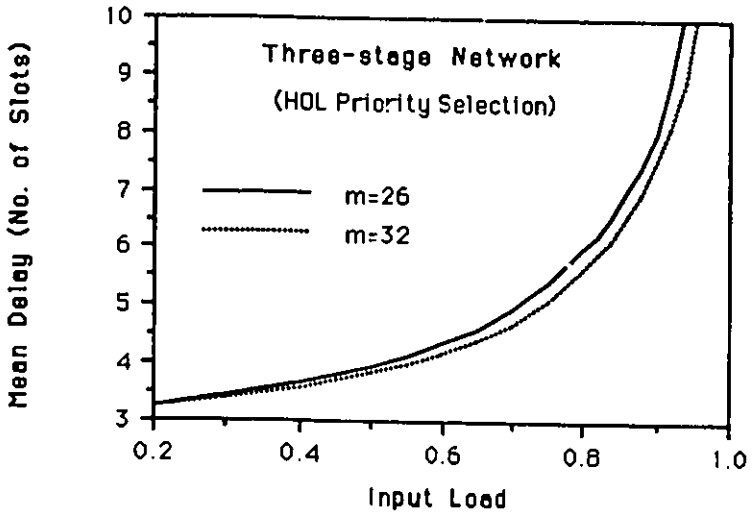


Figure 5.20: Mean Delay in a Three-Stage Network for $m = 26$ and $m = 32$ under HOL Priority Selection ($N = 256, n = 16$)

The results show that the performance of the limited intermediate buffer switch does not degrade appreciably under unbalanced and bursty traffic in comparison to that of uniform traffic.

The real benefit of the LIB switch architecture and the HOL priority selection scheme is demonstrated by the performance of the 3-stage interconnection networks. From the trade-off between complexity and performance, first-stage modules of size 16×26 provide better solution than the larger size switch modules. Although the interconnection network of size 256×256 was considered, networks of larger sizes can also be built.

We have published the results of this chapter in [85], [86], [87].

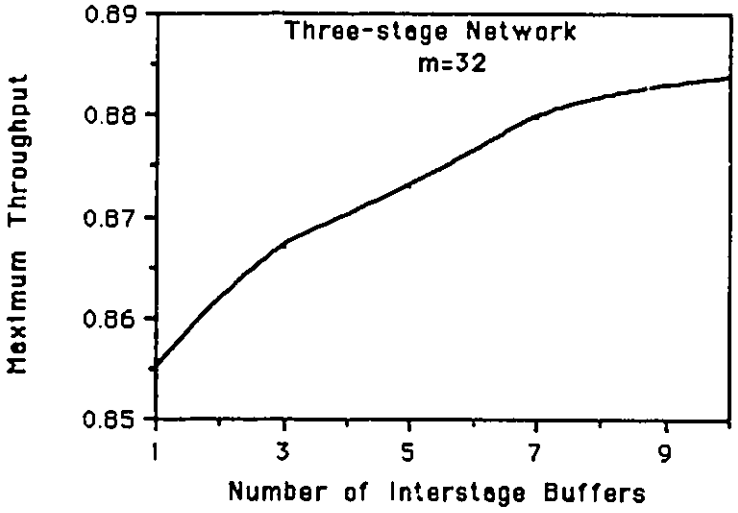


Figure 5.21: Maximum Throughput Achievable in a Three-Stage Network as a Function of Number of Interstage Buffers ($N = 256, n = 16, m = 32$), FIFO Selection

Chapter 6

Priority Analysis of Input Buffered Switches

6.1 Introduction

The Asynchronous Transfer Mode (ATM) will support a wide variety of communication services of different characteristics and requirements. In such a network, high efficiency of the resources is obtained by statistical multiplexing of packets (called cells in ATM) from different services. In order to provide services of differing requirements, the network should be capable of controlling the access to the network resources, according to the quality of service (QOS) demanded by the B-ISDN services.

The QOS of an ATM connection is mainly described by two parameters: end-to-end transfer delay of a cell and the cell loss probability. Therefore, we need to distinguish cells based on their delay and loss priorities. A service, for example packet video, requires low end-to-end delay, as well as low packet loss rate. Ordinary telephony can accept a medium delay [24] and a cell loss probability up to 10^{-3} without showing any appreciable degradation [88]. Electronic mail packets, on the other hand, need not be delivered quickly, but this service requires a low packet loss rate [89]. These examples show that different services require different combinations of delay and packet loss rate performance. By de-linking the handling of these two priorities, the protocols of the ATM layer can be

kept simple. In the next section, we discuss further the delay and loss sensitive priorities in ATM networks.

In this chapter, we study delay priorities in switch structures with input buffers. First, we consider an $N \times N$ nonblocking packet switch (Figure 6.1) with two classes of traffic. We analyze their delay performance under non-preemptive priority schemes. The major part of the delay of a packet, is the waiting time in the input queue to reach the head-of-line (HOL) position, rather than the contention time to access the output port. Therefore, identifying packets in the switch fabric on the basis of their class is not important and this simplifies the protocols in the switch fabric. This part is presented in Section 6.3.

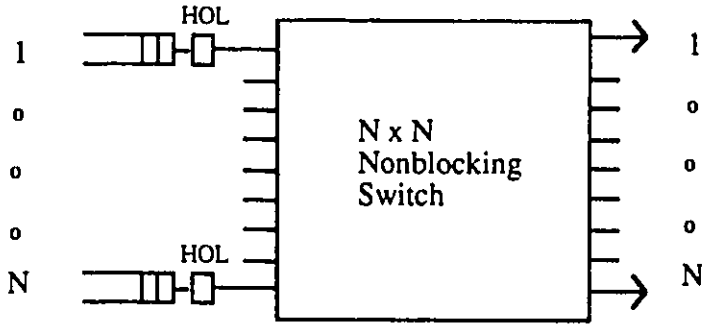


Figure 6.1: An $N \times N$ Nonblocking Packet Switch with Input Queues.

A packet switch with only input buffers has inadequate performance, although such a switch is easily realized since it does not require the switch fabric to work at higher speed than the input/output port speed. To improve the throughput of the switch, we present a dual plane switch architecture. As shown in Figure 6.2, two switch planes are connected in parallel to form a load sharing arrangement. Each plane is an $N \times N$ nonblocking switch with input queues. A dual plane switch structure with common input queues was discussed in [90]. The traffic of different delay priorities is shared by the two planes and here we extend the number of traffic classes to three. The highest priority traffic is carried by the second switching plane and the traffic of the other two classes is served by the first plane. The input port controller routes the incoming packet to either plane, depending

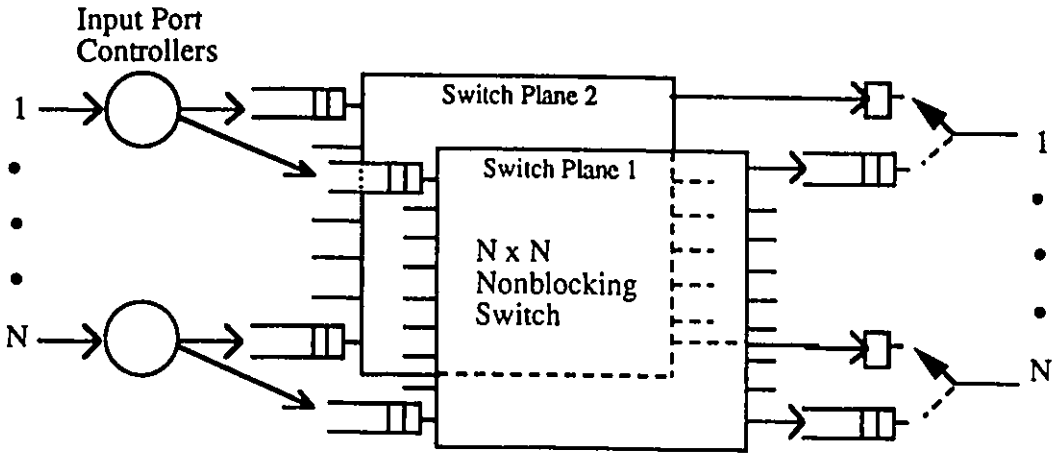


Figure 6.2: Dual Plane Switch Architecture

upon the delay priority of the packet. A buffer is provided at each output of the first plane to queue the packets, whenever the output port controller is busy transmitting packets from the second plane. We compute the mean delay in the output buffer using a conservation law. This is presented in Section 6.4.

6.2 Priorities in ATM Networks

Different levels of delay and cell loss priorities are being standardized for supporting traffic of various performance requirements. Mainly two implementation schemes for loss priority management have been proposed in the literature: threshold [91], [92] and overwriting schemes [88].

In the first scheme, a threshold R is assigned and packets of low priority are accepted in the input buffer if the current queue length is smaller than R. As a result, the input buffer of size b_i (per input port) is partitioned into two parts: the first one is of size R, where all the packets can be stored, and the second one of size $b_i - R$, where only the high

priority packets can be stored. In the second scheme, a dynamic sharing of the buffer b_i among the different loss sensitive packets is obtained. Any packet can be stored if buffer space is available, but a low priority cell can be pushed out by a high priority one. In this scheme, an extra care has to be taken to maintain the packet sequence integrity.

Similarly, there are different delay priority schemes possible in the switch. The delay priority level of a connection can be decided at the time of connection request, and it does not change for the duration of the connection. In the network, this priority is known from the Virtual Connection Identifier (VCI), and, therefore, the cell header need not indicate the delay priority.

The cell loss due to uncorrectable bit errors is supposed to be very small in the ATM networks. The end-to-end cell loss resulting from buffer overflow must also be kept low. The cell loss probability requirement varies for different services. The cells within a service may also have different loss sensitivity, e. g., a video service adopting a layered coding. For this purpose, CCITT reserved one bit in the cell header, for explicit cell loss priority indication. This bit may be used by the user or by the service provider to indicate lower priority cells. At the time of network congestion, the network will know the cells to be discarded without violating the QOS parameters.

Different broadband services would require different combinations of delay and packet loss rate performance. The delay priority can be indicated to the network at the time of call/ connection establishment, but the loss priority is to be decided at the call set up time as well as depending upon the type of service, it should be decided for individual cells. Incorporating these together may make protocols very complex. By de-linking the handling of these two priorities, the protocols of the ATM layer can be kept simple. This is a major consideration in a broadband network, due to the high transmission speeds. Iyengar and Zarki [89] have suggested that the packet delivery can be more dynamic and robust by separating the two types of priorities.

6.3 Performance of an $N \times N$ switch with two priority classes

In this section, we study an $N \times N$ synchronous nonblocking packet switch with input queues (Figure 6.1). We consider that there are two priority class packets, arriving at each input according to the independent Bernoulli processes. Let us assume that λ_H (λ_L) is the arrival rate of high (low) priority packet. We also assume that packets are uniformly distributed among all the outputs and successive packet arrivals are independently destined from those of previous arrivals.

At each time slot, the following events occur in sequence: arrival of packets at the input ports, contention among the packets at the head-of-line (HOL) positions having the same destination, and finally the departure of the packets, who win the contention. The service time of a HOL packet consists of the number of time slots required to win the contention. The service time depends on the priority class of the packet, the number of packets contending for the output and the type of priority used. We consider the following three priority schemes, which differ in their implementation complexity and performance.

- *Preemptive Priority:* a high priority packet arriving at an input moves ahead of all the low priority packets waiting for the service in the input queue, and if the HOL position is currently occupied by a low priority packet, it is preempted by the high priority one. Similarly, high priority packets are served first in comparison to the low priority packets contending for the same output.

- *Non-preemptive Priority Scheme-A:* a high priority packet arrival moves ahead of all the low priority packets in the input queue, but it does not preempt a low priority packet from the HOL position. Similar to the preemptive case, a high priority packet is favored among the packets contending for the same output.

- *Non-preemptive Priority Scheme-B:* it is the simplest to implement among the three priorities discussed here. This scheme is similar to Scheme-A except that, in

the switch fabric, the packets are not distinguished on the basis of their priority, i. e., the packets of both classes have identical delay distribution at the HOL position.

For all the above priority schemes, the packets within each priority class are served on a First-Come-First-Serve (FCFS) basis at all the input queues.

In preemptive and non-preemptive Scheme-A, the priority handling has to be implemented at two different places in the switching system. First, the priority of an incoming packet is taken care of at the input line interface before buffering it in the input queue. The second place is within the switch fabric, where the priority has to be considered, to resolve the contention in favor of a high priority packet. We assume that for each output, one of the high priority contending packets is randomly selected and then transmitted through the switch. If no high priority packet is contending for an output, then one among the low priority contending packets is chosen randomly, for the transmission.

On the other hand, in Scheme-B, the priority handling is required only at the input line interface. As one would expect, in this case the performance of high priority packets deteriorates to that of Scheme-A.

In the preemptive discipline, the high priority packets queue up as in the single priority system and the presence of low priority packets is transparent to them. Therefore, the analysis of single priority system [24], [16] is applicable to the high priority packets. Chen and Guérin [79] have studied the performance of low priority packets in the preemptive case. In [79], first an average service time of a low priority HOL packet was obtained, and then this measure was used to derive other performance parameters of interest. It is clear that such an analysis provides approximate results. In this section, we analyze the switch performance under non-preemptive priority schemes. We consider a large switch of $N \times N$ and determine the delay suffered through the switch by a low and by a high priority packet for the limiting case of $N = \infty$. The results are compared with those of simulation of a 64×64 switch.

6.3.1 Non-preemptive Priority Scheme-A

Here we analyze the delay performance of high and low priority packets in an input queueing nonblocking switch, employing non-preemptive priority Scheme-A.

With the assumption of $N = \infty$, the packets in the HOL positions can be considered as N independent virtual queues, where each virtual queue consists of HOL packets having the same destination. We fix our attention to one such virtual queue for a tagged output. The arrival process to this queue can be approximated as being composed of two Poisson processes with rate λ_H for the high priority and λ_L for the low priority packets. The total offered load ($\lambda_H + \lambda_L$) is assumed to be less than the saturation throughput of 0.586 [16]. Each virtual queue has one server, i. e. the output port. In each time slot, one of the high priority packets is randomly selected for the service from this queue. If there is no high priority packet for the tagged output, then one among the low priority packets is randomly chosen. From this discussion, it is clear that the virtual queue behaves as an $M/D/1$ queue, where packets are served in random order, and a low priority packet is served only when, there is no high priority packet in the queue. A two dimensional Markov Chain is required to find the steady-state joint distribution of low and high priority packets in the queue. This part of the analysis is presented in Section 6.5.

The delay suffered by a high priority packet in the virtual queue is not affected by the presence of low priority packets, and therefore the delay distribution (let us represent it by T_h) can be obtained from the analysis in [16]. To find the delay distribution of a low priority packet at a HOL position, we proceed as follows. We tag a low priority packet when it enters the virtual queue. Let us define $P_{m,l,h}$ as the conditional probability, that in a given time slot, immediately after the arrival instant, there are l low priority packets and h high priority packets in the queue, and the remaining delay is m time slots until the tagged packet completes the service. Since, a low priority packet cannot be served in the presence of any high priority packet, we have

$$P_{m,l,h} = 0 \qquad m \leq h$$

It is easy to see that the following equations, which are similar to the single priority case

of [16], hold in our case, when $h = 0$,

$$\begin{aligned} P_{1,1,0} &= 1 \\ P_{m,1,0} &= 0 \qquad m \neq 1 \\ P_{1,l,0} &= \frac{1}{l} \qquad l \geq 1 \end{aligned}$$

To obtain a general expression for $P_{m,l,h}$, where $m > 1$, $l \geq 1$, and $h \geq 0$, we recognize two different cases depending upon the value of h. The first case corresponds to $h = 0$. In this case, a low priority packet (other than the tagged one; for the reason of $m > 1$) is served in the current slot. In this slot, at the most $m - 2$ high priority arrivals can be considered, as the remaining delay for the tagged packet is $m - 1$ slots. Then, by simple recursion on m, we have :

$$P_{m,l,0} = \frac{l-1}{l} \sum_{i=0}^{\infty} \sum_{j=0}^{m-2} P_{m-1,l-1+i,j} \; \frac{\lambda_L^i \, e^{-\lambda_L}}{i!} \; \frac{\lambda_H^j \, e^{-\lambda_H}}{j!} \qquad m > 1, \; l > 1 \qquad (6.1)$$

In the second case, which corresponds to $h > 0$, a high priority packet is served in the current slot and the recursive equation becomes,

$$P_{m,l,h} = \sum_{i=0}^{\infty} \sum_{j=0}^{m-1-h} P_{m-1,l+i,h-1+j} \; \frac{\lambda_L^i \, e^{-\lambda_L}}{i!} \; \frac{\lambda_H^j \, e^{-\lambda_H}}{j!} \qquad m > 1, \; l \geq 1, \; 0 < h < m \qquad (6.2)$$

Averaging over l and h, the tagged packet delay (denote it by T_l) has the following probabilities:

$$Pr(T_l = m) = \sum_{l=1}^{\infty} \sum_{h=0}^{m-1} P_{m,l,h} \; Pr[l \textit{ low priority and } h \textit{ high priority packets in the virtual queue, immediately after the tagged packet (of low priority) arrives}]$$

where $P_{m,l,h}$ is given by equation (6.1) or (6.2), depending upon the value of h. The probability that the tagged packet belongs to a batch of size n, is given by $n \, l_n / \lambda_L$ [70], where l_n is the probability that in a time slot there are n low priority arrivals to the virtual queue. As observed before, in the limiting case of $N = \infty$, the arrival process to

the virtual queue becomes a Poisson process with a rate λ_L for the low priority and with a rate λ_H for the high priority. Therefore,

$$Pr(T_l = m) = \sum_{l=1}^{\infty} \sum_{h=0}^{m-1} P_{m,l,h} \left(\sum_{i=0}^{l-1} s_i \frac{\lambda_L^{l-i-1} e^{-\lambda_L}}{(l-i-1)!} \right) \left(\sum_{j=0}^{h} r_j \frac{\lambda_H^{h-j} e^{-\lambda_H}}{(h-j)!} \right) \quad (6.3)$$

where s_i (r_j) are the steady-state queue size probabilities of low (high) priority packets, just before the arrival instant in any given slot. These marginal distributions of the low and high priority packets can be obtained from the joint distribution $q_{i,j}$, i. e., the probability that there are i low priority and j high priority packets in the $M/D/1$ queue, immediately after the service completion. The joint distribution is computed in Section 6.5.

The transmission delay of a packet consists of the waiting time in the input queue until it reaches head-of-line and the delay at HOL position due to the output port contention. The high priority packets wait in the input queue as in a $Geom/G/1$ system of a single class, seeing themselves only, except for the added residual service time due to the packets of both classes, that might have been in HOL position at the arrival instant of a given high priority packet. Since the equivalent of a discrete time $Geom/G/1$ queue is an $M/G/1$ system in continuous time, we quote the result from [72] of the mean waiting time in an $M/G/1$,

$$\overline{W}_{M/G/1} = \frac{\lambda \overline{T^2}}{2\,(1 - \lambda \overline{T})} \quad (6.4)$$

where $\overline{T}$ and $\overline{T^2}$ are the first and second moments of the service time in a single priority queue, respectively. In the above equation, λ is the mean rate of Poisson arrival process. From [71], the mean waiting time in a $Geom/G/1$ queue is,

$$\overline{W}_{Geom/G/1} = \frac{\lambda\,(\overline{T^2} - \overline{T} \cdot \Delta t)}{2\,(1 - \lambda\,\overline{T})} \quad (6.5)$$

where Δt is the slot duration. In other words, the time between any two consecutive arrivals in an input queue is an integral multiple of Δt. In the limiting case ($\Delta t \rightarrow 0$) the binomial input distribution approaches a Poisson distribution with the same mean rate, equation (6.5) reduces to equation (6.4). Now, we can modify equations (2-85) and (2-86)

of [93] to give the mean delay of high priority and low priority packets in a $Geom/G/1$ queue. The mean delay of a high priority packet $\overline{D_h}$ (number of slots) is,

$$\overline{D_h} = \frac{\lambda_H(\overline{T_h^2} - \overline{T_h}) + \lambda_L(\overline{T_l^2} - \overline{T_l})}{2\,(1 - \lambda_H\,\overline{T_h})} + \overline{T_h} \tag{6.6}$$

where $\overline{T_h^2}$ and $\overline{T_l^2}$ are the second moments of delay in the $M/D/1$ virtual queue, for the high and the low priority packets, respectively. Similarly, the mean delay of a low priority packet $\overline{D_l}$ is

$$\overline{D_l} = \frac{\lambda_H(\overline{T_h^2} - \overline{T_h}) + \lambda_L(\overline{T_l^2} - \overline{T_l})}{2\,(1 - \lambda_H\,\overline{T_h})\,(1 - \lambda_H\,\overline{T_h} - \lambda_L\,\overline{T_l})} + \overline{T_l} \tag{6.7}$$

Figure 6.3 shows the average delay suffered (in terms of number of slots) by the low priority traffic in the presence of high priority traffic of rate $\lambda_H = 0.05$ to 0.25. Note that, this delay measure includes one slot time required to route the packet through the switch fabric. The symbols indicate the simulation results of a 64×64 switch. Mean delay of the high priority packets is plotted in Figure 6.4 as a function of low priority load. Although low priority traffic affects the delay performance of high priority, it is not severe at moderate loads. For example, the delay is between 1.5 and 1.9 time slots when the total traffic remains at 0.5 and the high priority load varies from 0.05 to 0.25.

6.3.2 Non-preemptive Priority Scheme-B

In this priority scheme, once a packet reaches at the HOL position, we do not distinguish between the two classes of packets, and therefore, packets of both classes experience the same delay at head-of-line. This simplifies the protocols within the switch fabric because any packet can be selected among the contending packets, whenever contention occurs for a given output. It is obvious that the high priority packets now experience more delay in comparison to Scheme-A. In this subsection, we examine the amount of degradation in the performance of high priority packets.

The service time distribution of a HOL packet can be calculated using analysis of a single priority $M/D/1$ queue. Let $\overline{T}$ and $\overline{T^2}$ represent the first and second moments of service time, respectively. Equations (6.6) and (6.7) can be modified to yield mean

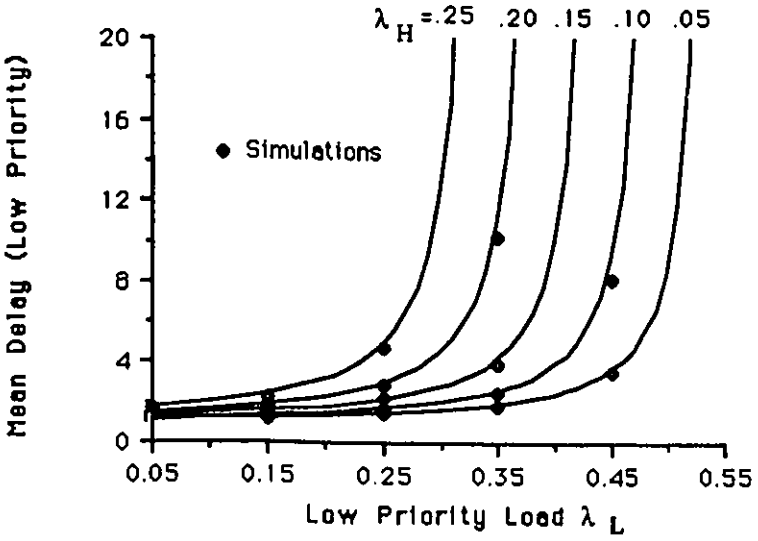


Figure 6.3: Mean Delay for Low Priority Traffic in Scheme-A.

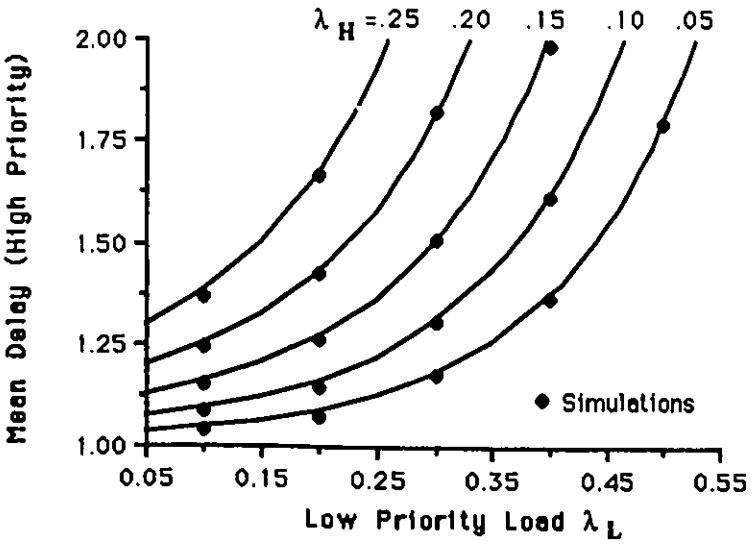


Figure 6.4: Mean Delay for High Priority Traffic in Scheme-A.

delay of high and low priority packets in Scheme-B, by substituting $\overline{T_h} = \overline{T_l} = \overline{T}$ and $\overline{T_h^2} = \overline{T_l^2} = \overline{T^2}$.

Figure 6.5 shows the mean delay of high priority traffic as a function of λ_L and λ_H. The performance of high priority traffic does not degrade appreciably in Scheme-B in comparison to Scheme-A. For example, the delay increase is less than 0.3 slot when the total load (i. e., $\lambda_H + \lambda_L$) is 0.5, and the increase is about 0.1 slot when the total load remains at 0.4. In Scheme-B, the performance of low priority packets improves by a small amount, which is of no consequence, and so we do not discuss it in detail. In non-preemptive priorities, a low priority packet does not suffer as much delay as in the case of preemptive priority.

Figure 6.6 provides the comparative study of the mean delay of high priority packets under three priority schemes. In preemptive priority, the performance of high priority

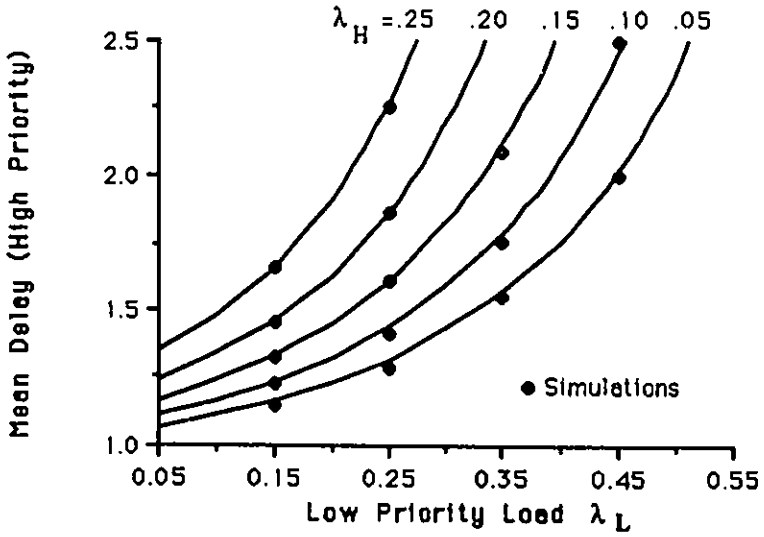


Figure 6.5: Mean Delay for High Priority Traffic in Scheme-B.

packets is not influenced by low priority traffic at all. Figure 6.6 suggests that at higher loads, the major part of the delay is the waiting time in the input queue to access the HOL position, and it is not the service time, i. e., the contention time at the head-of-line.

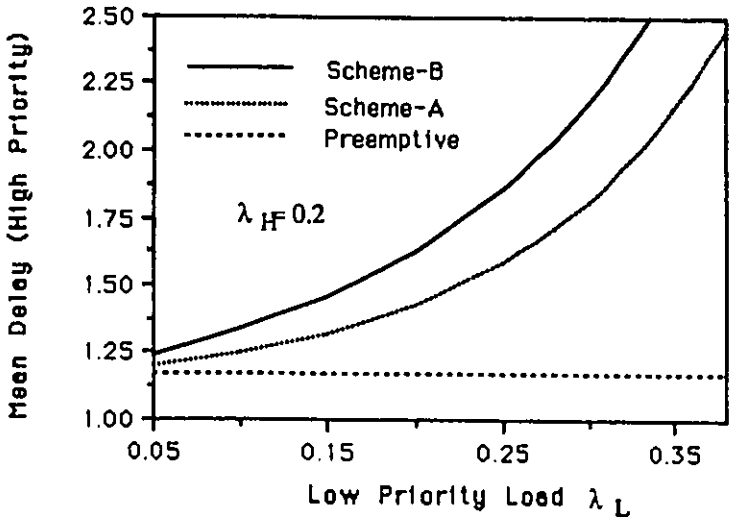


Figure 6.6: Comparison of Mean Delay for High Priority Traffic at $\lambda_H = 0.2$.

Therefore, any queueing discipline, allowing quick access to high priority packets to the HOL position, would help in reducing the delay. However, within the switch fabric, the packets of both classes may be treated on equal basis.

6.4 Performance of dual plane switch

This section studies the dual plane switch shown in Figure 6.2. Here we extend the number of priority classes to three. Let us call these priority classes as Class P, Class H, and Class L. The traffic of Class H (of rate λ_H) and Class L (of rate λ_L), which are considered in the previous section, is carried by the first switching plane. The second switching plane carries only the highest priority class traffic, i. e., Class P. In order to provide low delay to this class of traffic, we should restrict the load to about less than 40% (i. e., $\lambda_P < 0.4$). Similar to the case of previous section, we assume that packets of this class arrive at each input according to an independent Bernoulli distribution and they are uniformly distributed among all the outputs.

At each time slot, up to two packets can appear at a given output, one from each switching plane. The packet from the second switching plane is preferred over the packet from the first switching plane and therefore, buffers are not required at the outputs of the second plane. However, a single buffer has been provided to store the packet temporarily, before being transmitted on to the output port. The FIFO buffers at the outputs of the first switching plane queue the packets whenever the output port is busy transmitting packets from the second plane. In other words, in a time slot, a packet from an output buffer of the first plane is transmitted, only if no packet is available for that output from the second plane.

It is clear from the above discussion that the presence of the first plane is transparent to the packets of the second plane, and therefore, their performance measures are already available [24], [16]. A packet of the first switching plane experiences delay at the following three places:

1. waiting time within the input queue until it reaches the head-of-line,
2. waiting time at HOL position due to head-of-line contention, and
3. waiting time at the output buffer due to output port contention.

The first and second delay components were calculated in the previous section, for two classes of traffic, where Class H had higher priority over Class L. In order to obtain the third delay component, we make the following assumptions:

- There is an infinite amount of buffer at each output port of first plane and therefore, the output buffer never overflows.
- Packets of Class H and Class L wait in the output buffer for transmission on to the output port on First-Come-First-Serve (FCFS) basis. For non-preemptive priority Scheme-A, it implies that after crossing the switch fabric, packets of both classes are treated on equal basis. In case of Scheme-B, packets of Class H get higher priority over Class L, only in the input queue, and thereafter, they are not distinguished from each other.

Let W_o represent the waiting time in the output buffer, which is the same for class L and class H packets. To compute, the average value of W_o, we proceed as follows.

For a moment, let as consider the packets at HOL positions of the first switching plane. The mean service time for high priority packets, for the reason of simplicity, was represented as $\overline{T_h}$, which is a function of load λ_H. Now, let us represent it by $\overline{T_h}(\lambda_H)$. Similarly, the mean service time of low priority packets is a function of λ_H and λ_L, and can be written as $\overline{T_l}(\lambda_H, \lambda_L)$. Note that $\overline{T_h}(\lambda_H)$ and $\overline{T_l}(\lambda_H, \lambda_L)$ include one slot delay required to route the packet through the switch fabric. Therefore, for a high priority packet the mean waiting time in HOL position, before it wins the contention is given by,

$$\overline{W_h}(\lambda_H) = \overline{T_h}(\lambda_H) - 1$$

and similarly, for a low priority packet,

$$\overline{W_l}(\lambda_H, \lambda_L) = \overline{T_l}(\lambda_H, \lambda_L) - 1$$

According to Kleinrock's Conservation Law for work-conserving $M/G/1$ systems with any non-preemptive queueing discipline, the weighted sum of wait times is always conserved [94]. Since the virtual queue behaves as an $M/D/1$ queue, therefore, the following would hold,

$$\lambda_H \ \overline{W_h}(\lambda_H) + \lambda_L \ \overline{W_l}(\lambda_H, \lambda_L) = (\lambda_H + \lambda_L) \ \overline{W}_{M/D/1}(\lambda_H + \lambda_L) \tag{6.8}$$

where $\overline{W}_{M/D/1}(x)$ is the mean waiting time for a single priority $M/D/1$ queue at load x, and is given by [16],

$$\overline{W}_{M/D/1}(x) = \frac{x}{2(1-x)}$$

In equation (6.8), note that $\overline{W_h}(\lambda_H) = \overline{W}_{M/D/1}(\lambda_H)$.

Now, we come back to the dual plane switch structure. For a tagged output, there are three classes of packets, which form three independent Poisson processes at HOL positions: with rate λ_P in the second switching plane, and with rate λ_L and λ_H in the first plane. All packets have one slot service time, i. e., the transmission time at the output port. From the tagged output port, a packet is always transmitted, whenever there is a

packet available in any of the two virtual queues or in the buffer at that output. With these observations, it is easy to realize that for the tagged output, two virtual queues, one at each plane's HOL positions, and buffer at the output of first plane, can be considered as a single $M/D/1$ queue, where Class P packets have priority over packets of the other two classes. In this equivalent model, we have to ignore the one slot delay required to route a packet through the switch fabric, which being a constant can be ignored without affecting the accuracy of the analysis. We also do not distinguish between the packets of Class H and Class L. Therefore, the mean waiting time of high priority (Class P) packets, $\overline{W_h}(\lambda_P)$, and the mean waiting time of low priority (Class H and Class L) packets, $\overline{W_l}(\lambda_P, \lambda_H + \lambda_L)$, should satisfy the following conservation equation, which is similar to Equation (6.8),

$$\lambda_P \ \overline{W_h}(\lambda_P) + (\lambda_H + \lambda_L) \ \overline{W_l}(\lambda_P, \lambda_L + \lambda_H) \ = \ (\lambda_P + \lambda_H + \lambda_L) \cdot \overline{W}_{M/D/1}\ (\lambda_P + \lambda_H + \lambda_L) \qquad (6.9)$$

Note that for the tagged output, $\overline{W_l}(\lambda_P, \lambda_L + \lambda_H)$ represents the sum of mean waiting time in the HOL virtual queue (of the first plane) and the mean waiting time in the output buffer. Since the virtual queue itself behaves as an $M/D/1$ queue, the mean waiting time in the output buffer, $\overline{W_o}$ is,

$$\overline{W_o} = \overline{W_l}(\lambda_P, \lambda_L + \lambda_H) - \overline{W}_{M/D/1}(\lambda_H + \lambda_L) \qquad (6.10)$$

where $\overline{W_l}(\lambda_P, \lambda_L + \lambda_H)$ can be computed using the analysis of Section 6.3 or from Equation (6.9).

Figure 6.7 shows the mean delay suffered in the output buffer by the Class H and Class L packets, when Class P load, $\lambda_P = 0.20$ to 0.35. At low Class P loads, the delay in the output buffer is very small and it increases slowly with the increase in load of first switching plane $(\lambda_H + \lambda_L)$. However, when $\lambda_P \geq 0.3$, the delay in the output buffer increases sharply with $\lambda_H + \lambda_L$. The reason for this sharp increase is that the total load $(\lambda_P + \lambda_H + \lambda_L)$ is approaching the output port saturation level. The simulation results of a 64×64 switch are also shown on the Figure 6.7, which agree with the analysis.

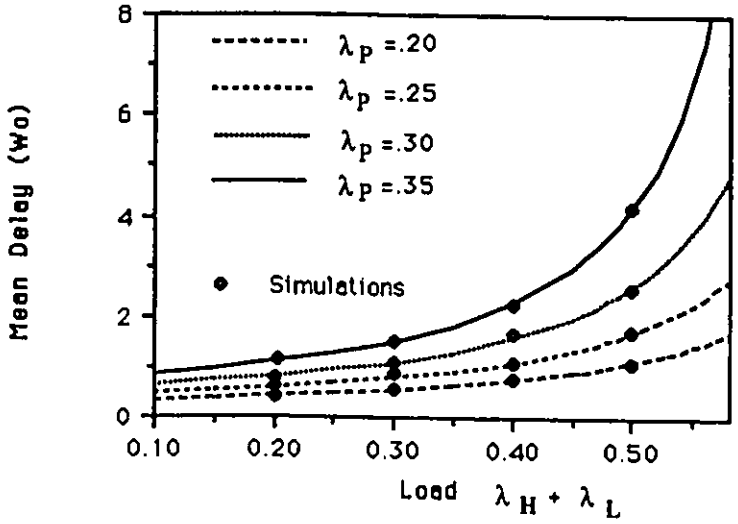


Figure 6.7: Mean Delay in Output Buffer ($\overline{W_o}$).

6.5 Joint Distribution of two Priority Packets in an M/D/1 Queue

In this section, we obtain the steady-state joint distribution of the low and high priority packets in an $M/D/1$ queue, where a low priority packet is served only when there is no high priority packet in the queue. Let $q_{i,j}$ denote the state probability that just after service completion, there are i low priority and j high priority packets in the queue. The following equations describe the system behaviour,

$$q_{i,j} = \sum_{k=0}^{i} \sum_{n=0}^{j+1} q_{k,n}\, l_{i-k}\, h_{j-n+1} \qquad i \geq 0,\ j > 0 \tag{6.11}$$

and for the boundary states $(i,0)$, $i \geq 0$

$$q_{i,0} = \sum_{k=0}^{i+1} q_{k,0}\, l_{i-k+1}\, h_0 + \sum_{k=0}^{i} q_{k,1}\, l_{i-k}\, h_0 \qquad i \geq 0 \tag{6.12}$$

where

$$l_i = \frac{\lambda_L^i\, e^{-\lambda_L}}{i!}$$

$$h_i = \frac{\lambda_H^i \, e^{-\lambda_H}}{i!}$$

In addition to equations (6.11) and (6.12), we need the conservation relationship,

$$\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} q_{i,j} = 1 \tag{6.13}$$

The steady-state joint distribution $q_{i,j}$ can be obtained by solving the above (infinite) set of linear equations, iteratively. In practice, only finite number of states can be considered. We truncate the number of states at a point, where the probability becomes negligibly small. Neuts' method of solving stochastic matrices of $M/G/1$ type [96] may also be used to obtain the steady-state distribution.

6.6 Summary

An ATM packet switch should be capable of handling delay and loss sensitive traffic. First, we argued in favor de-linking handling of these two types of priorities in a switching system, and then we studied the delay priorities in two different ATM switch architectures. In the first architecture, which is a nonblocking switch with input queues, we analyzed the performance of two classes of traffic under two different non-preemptive priority schemes. We found that a queueing discipline, which allows a quick access to the high priority packets would reduce the delay. However, within the switch fabric, the packets of high and low priority packets can be treated on equal basis.

A packet switch with input buffers has inadequate throughput. To enhance the performance and to provide low delay to high priority traffic, we examined a dual plane switch architecture, where each plane is a nonblocking switch with input buffers. The two planes are connected in parallel to form a load sharing arrangement. We extended the analysis of a single plane nonblocking switch to this architecture. In both cases, the performance was compared with the simulation results of a 64×64 switch.

These results were published in [95].

Chapter 7

Conclusions

7.1 Summary

In this thesis, we presented an analysis of three novel switch architectures for Broadband-ISDN. We first reviewed various fast packet switch architectures that have been proposed in the literature. The switches are classified into two categories: internally blocking and nonblocking switches. Since, the nonblocking switches provide better performance in comparison to that of the blocking switches, they are of greater interest to us. The performance of a nonblocking switch depends upon how it manages the packets that are contending for the same output port. In fact, it is directly related to the arrangement of buffers in the switch fabric. There are three places where buffers can be provided in a switch: at the input ports, in the switch fabric, and at the output ports. An input buffered nonblocking switch can only achieve a maximum throughput of 58.6%, although it has the minimal complexity from an implementation point of view. Buffers can also be provided within the switch fabric, for example, a crossbar switch with buffering at crosspoints. Such a switch has the same throughput-delay performance as that of an output buffered switch, but for a given packet loss rate, it requires much larger memory than an output buffered switch. In an output buffered switch, the buffers can be organized in many different ways. An output buffered switch with a separate buffer at each output port performs well under non-uniform traffic but the buffer requirement increases with

the average burst length of the traffic. The implementation of the switch requires high speed-up or high parallelism in the switch fabric. The output buffered switch with a completely shared buffer makes most efficient use of the buffers, but requires a buffer of high memory bandwidth, which is a constraint in the design of a large switch.

In this thesis, our focus is on the high-performance nonblocking switch architectures which have manageable implementation complexity for a considerably large switch. In Chapter 4, we presented an analysis of a switch with input and output buffers, where packet loss can occur only at the input ports. Whenever an output queue is full, a back-pressure is applied and the packets are not transferred from the head of the input queues. In such a switch, we studied the maximum throughput, mean delay through the switch and the packet loss rate. These performance parameters were also studied in the switch when a speed constraint is applied, i. e., when only a limited number of packets (equals to the speed-up factor) are routed from inputs to any given output port in a time slot, although the switch can transfer only one packet from each input to outputs. We found that such a switch with a speed-up of 3 and 20 buffers at each output port can achieve a throughput of more than 90% [68]. It is noted that the performance of this switch is very close to that of an input and output buffered switch with no speed constraint [69]. These results are of a great practical value as a switch with higher speed-ups (e.g., an output buffered switch) may be difficult to realize. We also presented a simple implementation of such a switch.

In Chapter 5, we proposed a switch architecture based on a crossbar switch fabric, where a buffer sufficient to store an ATM cell is provided at each crosspoint of the switch. In addition to this single buffer at each crosspoint, buffers are also provided at the input ports to reduce the packet loss rate. It is called the Limited Intermediate Buffer (LIB) switch [85]. The most important feature of the LIB switch is that the switch fabric operates at the speed of the input-output port, which is quite desirable in high-speed switching systems. We examined the LIB switch performance under different traffic models and under various scheduling policies. We introduced a new scheduling policy called HOL Priority Selection, where the priority is given to a packet at the intermediate buffer,

which is causing the head-of-line blocking. The simulation results demonstrate that HOL priority selection results in the best performance, i. e., high saturation throughput, low mean delay and low variability of the delay. The study of a 16 × 16 LIB switch module under two delay priority traffic shows that the maximum throughput can be increased by about 3.5% [86]. The performance of the LIB switch degrades slightly under unbalanced and bursty traffic in comparison to that of an uniform traffic. We also considered a three-stage interconnection network to build large size switching systems comprising the LIB switch modules. The interconnection network consists of symmetric as well as asymmetric size switch modules. The study suggests that modules of size 16 × 26 are a better choice than the larger size switch modules, for a trade-off between complexity and performance [86], [87]. Advantages of the HOL priority selection scheme are also supported by the study of the interconnection network.

In an ATM network, we need to identify the cells based on their delay and loss priorities. To keep the protocols simple at the ATM layer, we suggest that the handling of these priorities should be de-linked. In Chapter 6, we studied the performance of delay-sensitive traffic in two different switch architectures. First, we considered a nonblocking switch with input queues and analyzed the performance of two classes of traffic under two different non-preemptive priority schemes. The two priority schemes differ in the sense that, in the second scheme the packets are not identified based on their priority within the switch fabric. The important finding is that this does not result in an appreciable degradation in the performance of the high priority traffic, and therefore should be preferred to reduce the complexity of the switch fabric [95]. The second switch architecture to study the delay sensitive traffic performance is a dual plane switch architecture, where each plane is a nonblocking switch with input buffers. Here the number of traffic classes is extended to three. The highest priority traffic is carried by the second switching plane and the traffic of the other two classes is served by the first switching plane. A buffer is provided at each output port of the first plane to queue the packets of lower priority. We analyzed the mean delay in the output buffer and compared the results with the simulation of a 64 × 64 switch.

7.2 Suggestions for Further Research

The three switch architectures presented in this thesis require some further investigation. The issues which require further consideration are discussed below.

In Chapter 4, the nonblocking switch with speed and output buffer constraints was analyzed under uniform random traffic. Although the exact characteristic of the integrated services traffic is not known, the effect of unbalanced and bursty traffic on the performance of the switch should be studied. Furthermore, we did not investigate a suitable interconnection network for building a large size switching system consisting of the above switch modules.

The proposed Limited Intermediate Buffer (LIB) switch architecture was analyzed under several traffic models but its performance needs to be further examined under some other traffic conditions, e. g., hot-spot traffic. We also did not study the packet loss rate at the input ports of such a switch, which is very essential for sizing the input buffers. A VLSI implementation design of the LIB switch architecture can be another interesting area for future research work.

The dual plane switch architecture is an excellent switch architecture considering its capability to provide high-performance to the delay-sensitive traffic. The use of the non-preemptive priority scheme (Scheme-B) which does not distinguish the packets within the switch fabric according to their priority has been justified on the basis of our study under the uniform random traffic. It requires further investigation under other traffic models. In the dual plane switch architecture, an interesting problem is to obtain the distribution of the output buffer occupancy, in order to size the number of buffers for a given packet loss rate.

In all cases, reliability and fault tolerance need to be investigated before these switches can form a basis for future telecommunications networks.

References

[1] F. A. Tobagi, "Fast Packet Switch Architectures For Broadband Integrated Services Digital Networks," *Proceedings of the IEEE*, Vol. 78, No. 1, pp. 133-167, Jan. 1990.

[2] J. P. Coudreuse, P. Adam and P. Gonet, "Asynchronous Time-Division Switching: The way to flexible Broadband Communication Networks," *Proc. of the International Zurich Seminar on Digital Commun. '86*, Mar. 1986.

[3] CCITT Draft Recommendation I.361: B-ISDN ATM Layer Specification. *Study Group XVIII*, Geneva, January 1990.

[4] S. E. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)," *IEEE Communications Magazine*, Vol. 27, pp. 17-24, Sep. 1989.

[5] B. Schaffer, "Synchronous and Asynchronous Transfer Modes in the Future Broadband ISDN," *Proc. IEEE ICC'88*, Toronto, Canada, pp. 1552-1558, 1988.

[6] M. Decina, "Open Issues Regarding the Universal Application of ATM for Multiplexing and Switching in the B-ISDN," *Proc. IEEE ICC'91*, pp. 1258-1264, 1991.

[7] CCITT Recommendation I.121: Broadband Aspects of ISDN. *Blue Book*, Geneva, 1989.

[8] H. Ahmadi and W. E. Denzel, "A Survey of Modern High-Performance Switching Techniques," *IEEE J. Select. Areas Commun.*, Vol. 7, No. 7, pp. 1091-1103, September 1989.

[9] M. A. Henrion, K. J. Schrodi, D. Boettle, M. De Somer and M. Dieudonn, "Switching Network Architecture for ATM Based Broadband Communications," *Proc. ISS'90*, A7.1, 1990.

[10] C. Wu and T. Feng, "On a Class of Multistage Interconnection Networks," *IEEE Trans. on Computers*, Vol. C-29, pp. 694-704, No. 8, Aug. 1980.

[11] J.Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks.* Kluwer Academic Publishers, 1987.

[12] Y. Jenq, "Performance Analysis of a Packet Switch Based on Single-Buffered Banyan Network," *IEEE J. Select. Areas Commun.*, Vol. SAC-1, No. 6, pp. 1014-1021, Dec. 1983.

[13] H. S. Kim and A. Leon-Garcia, "Performance of Buffered Banyan Networks under Nonuniform Traffic Patterns," *Proc. IEEE INFOCOM'88*, pp. 344-353, 1988.

[14] T. D. Morris and H. G. Perros, "Performance Modelling of a Multi-Buffered Banyan Switch under Bursty Traffic," *Proc. IEEE INFOCOM'92*, pp. 436-445, 1992.

[15] H. S. Kim and A. Leon-Garcia, "A Self Routing Multistage Switching Network for Broadband ISDN," *IEEE J. Select. Areas Commun.*, Vol. 8, No. 3, pp. 459-466, April 1990.

[16] M. J. Karol, M.G. Hluchyj and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans. Commun.*, Vol. COM-35, pp.1347-1356, Dec. 1987.

[17] S. Nojima, E. Tsutsui, H. Fukuda and M. Hashimoto, "Integrated Services Packet Network Using Bus Matrix Switch," *IEEE J. Select. Areas Commun.*, Vol. SAC-5, No. 8, pp. 1284-1292, Oct. 1987.

[18] E. P. Rathgeb, T. H. Theimer and M. N. Huber, "Buffering Concepts for ATM Switching Networks," *Proc. IEEE GLOBECOM'88*, 39.3, 1988.

[19] Y. Kato, H. Hayami, J. Kamoi, M. Takeno, K. Murakami and S. Hattori, "A VLSIC for the ATM Switching System," *Proc. ISS'90*, Vol. 3, P5, 1990.

[20] V. P. Kumar, J. G. Kneuer, D. Pal and B. Brunner, "PHOENIX: A Building Block for Fault Tolerant Broadband Packet Switches," *Proc. IEEE GLOBECOM'91*, 8B.1, 1991.

[21] P. Goli and V. Kumar, "Performance of a Crosspoint Buffered ATM Switch Fabric," *Proc. IEEE INFOCOM'92*, 3D.1, 1992.

[22] U. Killat, W. Kowalk and J. Noll, "A Versatile ATM Switch Concept," *Proc. ISS'90*, Vol. 4, A6.4, 1990.

[23] A. Huang and S. Knauer, "STARLITE: A Wideband Digital Switch," *Proc. IEEE GLOBECOM'84*, 5.3, 1984.

[24] J. Y. Hui and E. Arthurs, "A Broadband Packet Switch for Integrated Transport," *IEEE J. Select. Areas Commun.*, Vol. SAC-5, No. 8, pp. 1264-1273, Oct. 1987.

[25] J. N. Giacopelli, W. D. Sincoskie and M. Littlewood, "SUNSHINE: A High Performance Self-Routing Broadband Packet Switch Architecture," *Proc. ISS'90*, Vol. 3, P21, 1990.

[26] H. Suzuki, H. Nagano, T. Suzuki, T. Takeuchi and S. Iwasaki, "Output-buffer Switch Architecture for Asynchronous Transfer Mode," *Proc. IEEE ICC'89*, 4.1, 1989.

[27] Y. S. Yeh, M. G. Hluchyj and A. S. Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching," *IEEE J. Select. Areas Commun.*, Vol. SAC-5, No. 8, pp. 1274-1283, October 1987.

[28] H. Yoon, M. T. Liu and K. Y. Lee, "The Knockout Switch under Nonuniform Traffic," *Proc. IEEE GLOBECOM'88*, Hollywood, FL., 1628-1634, 1988.

[29] K. Y. Eng, M. J. Karol and Y. S. Yeh, "A Growable Packet (ATM) Switch Architecture: Design Principles and Applications," *Proc. IEEE GLOBECOM'89*, 32.2, 1989.

[30] K. Y. Eng and M. J. Karol, "A Growable Switch Architecture: A Self-Routing Implementation for Large ATM applications," *Proc. IEEE ICC'91*, 32.3, 1991.

[31] K. Y. Eng and M. J. Karol, "Gigabit-per-Second ATM Packet Switching with the Growable Switch Architecture," *Proc. IEEE GLOBECOM'91*, 31A.4, 1991.

[32] D. X. Chen and J. W. Mark, "SCOQ: A Fast Packet Switch with Shared Concentration and Output Queueing," *Proc. IEEE INFOCOM'91*, 3A.1, 1991.

[33] D. X. Chen and J. W. Mark, "A Buffer Management Scheme for the SCOQ Switch Under Nonuniform Traffic Loading," *Proc. IEEE INFOCOM'92*, 1D.4, 1992.

[34] H. J. Chao, "A Distributed Modular Tera-bit/sec ATM Switch," *Proc. IEEE GLOBECOM'90*, 805.3, 1990.

[35] Y. M. Kim and K. Y. Lee, "KSMINs: Knockout Switch Based Multistage Interconnection Networks for High-speed Packet Switching," *Proc. IEEE GLOBECOM'90*, 305.6, 1990.

[36] M. Devault, J.-Y. Cochennec and M. Servel, "The Prelude ATD Experiment: Assessments and Future Prospects," *IEEE J. Select. Areas Commun.*, Vol. 6, No. 9, pp. 1528-1537, Dec. 1988.

[37] H. Kuwahara, N. Endo, M. Ogino and T. Kozaki, "A Shared Buffer Memory Switch for an ATM Exchange," *Proc. IEEE ICC'89*, Toronto, Canada, pp. 118-122, 1989.

[38] J. S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE Trans. Commun.*, Vol. 36, No.6, pp.734-743, June 1988.

[39] T. T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE J. Select. Areas Commun.*, Vol. 6, No. 9, pp. 1455-1467, Dec. 1988.

[40] M. G. Hluchyj and M. J. Karol, "Queueing in High-Performance Packet Switching," *IEEE J. Select. Areas Commun.*, Vol. 6, No. 9, pp. 1587-1597, Dec. 1988.

[41] S.-Q. Li, "A Study of Traffic Imbalances in a Fast Packet Switch," *Proc. IEEE INFOCOM'89*, pp. 538-547, 1989.

[42] S.-Q. Li, "Performance of a Nonblocking Space-Division Packet Switch with Correlated Input Traffic," *Proc. IEEE GLOBECOM'89*, 49.1, 1989.

[43] M. J. Lee and S.-Q. Li, "Performance of a Nonblocking Space Division Packet Switch in a Time Variant Non-uniform Traffic Environment," *Proc. IEEE ICC'90*, 308.5, 1990.

[44] S.-Q. Li, "Non-uniform Traffic Analysis on a Nonblocking Space-Division Packet Switch," *IEEE Trans. Commun.*, Vol. 38, No.7, pp.1085-1096, July 1990.

[45] S. C. Liew, "Performance of Input Buffered and Output Buffered ATM Switches under Bursty Traffic: Simulation Study," *Proc. IEEE GLOBECOM'90*, 905.2, 1990.

[46] R. G. Bubenik and J. S. Turner, "Performance of a Broadcast Packet Switch," *Proc. IEEE ICC'87*, 31.6, 1987.

[47] T. T. Lee, "A Modular Architecture for Very Large Packet Switches," *IEEE Trans. Commun.*, Vol. 38, No. 7, pp. 1097-1106, Jul. 1990.

[48] K. Shiomoto, M. Murata, Y. Oie and H. Miyahara, "Performance Evaluation of Cell Bypass Queueing Discipline for Buffered Banyan Type ATM Switches," *Proc. IEEE INFOCOM'90*, pp. 677-685, 1990.

[49] Y. Shobatake and T. Kodama, "A Cell Switching Algorithm for the Buffered Banyan Network," *Proc. IEEE ICC'90*, 316.4, 1990.

[50] K. W. Sarkies, "The Bypass Queue in Fast Packet Switching," *IEEE Trans. Commun.*, Vol. 39, No.5, pp. 766-774, May 1991.

[51] Y. Oie, T. Suda, M. Murata, and H. Miyahara, "Survey of the Performance of Nonblocking Switches with FIFO Input Buffers," *Proc. IEEE ICC'90*, 316.1, 1990.

[52] A. Pattavina, "A Multiservice High-Performance Packet Switch for Broadband Networks," *IEEE Trans. Commun.*, Vol. 38, No.9, pp.1607-1615, Sep. 1990.

[53] Y. Oie, M. Murata, K. Kubota and H. Miyahara, "Effect of Speedup in Nonblocking Packet Switch," *Proc. ICC'89*, 13.4, 1989.

[54] F. Kamoun and L. Kleinrock, "Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions," *IEEE Trans. Commun.*, Vol. COM-28, No.7, pp.992-1003, July 1980.

[55] T.-C. Hou and D. M. Lucantoni, "Buffer Sizing for Synchronous Self-Routing Broadband Packet Switches with Bursty Traffic," *International J. of Digital and Analog Commun. Systems*, Vol. 2, pp. 253-260, 1989.

[56] D. X. Chen and J. W. Mark, "Performance Analysis of Output Buffered Fast Packet Switches with Bursty Traffic Loading," *Proc. IEEE GLOBECOM'91*, 14.3, 1991.

[57] A. E. Eckberg and T.-C. Hou, "Effects of Output Buffer Sharing on Buffer Requirements in an ATDM Packet Switch," *Proc. IEEE INFOCOM'88*, pp. 459-466, 1988.

[58] N. Endo, T. Ohuchi, T. Kozaki, H. Kuwahara and M. Mori, "Traffic Characteristic Evaluation of a Shared Buffer ATM Switch," *Proc. IEEE GLOBECOM'90*, 905.1, 1990.

[59] X. Chen and J. F. Hayes, "A Shared Buffer Memory Switch with Maximum Queue and Minimum Allocation," *Proc. Canadian Conference on Elec. and Comp. Eng.*, Quebec, Canada, 7.1, 1991.

[60] S. X. Wie, E. J. Coyle and M.-T. T. Hsiao, "An Optimal Buffer Management Policy for High-Performance Packet Switching," *Proc. IEEE GLOBECOM'91*, 27.2, 1991.

[61] J. S.-C. Chen and T. E. Stern, "Optimal Buffer Allocation for Packet Switches with Input and Output Queueing," *Proc. IEEE GLOBECOM'90*, 905.5, 1990.

[62] J. S.-C. Chen and T. E. Stern, "Throughput Analysis, Optimal Buffer Allocation, and Traffic Imbalance Study of a Generic Nonblocking Packet Switch," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, pp. 439-449, Apr. 1991.

[63] J. S.-C. Chen and T. E. Stern, "Throughput Reduction due tc Non-uniform Traffic in a Packet Switch with Input and Output Queueing," *Proc. IEEE ICC'91*, 15.2, 1991.

[64] A. Y. M. Lin and J. A. Silvester, "The Effect of switch Speed and Buffer Limitations on the Performance of a Multichannel ATM Switch with Output Queueing," *Proc. IEEE ITS'90*, 21.2, 1990.

[65] I. Iliadis and W. E. Denzel, "Performance of Packet Switches with Input and Output Queueing," *Proc. IEEE ICC'90*, 316.3, 1990.

[66] I. Iliadis, "Head of the Line Arbitration of Packet Switches with Combined Input and Output Queueing," *International J. of Digital and Analog Commun. Systems*, Vol. 4, pp. 181-190, 1991.

[67] G. Bruzzi and A. Pattavina, "Performance Evaluation of an Input-Queued ATM Switch with Internal Speed-up and finite Output Queues," *Proc. IEEE GLOBECOM'90*, 801.5, 1990.

[68] A. K. Gupta and N. D. Georganas, "Analysis of a Packet Switch with Input and Output Buffers and Speed Constraints," *Proc. INFOCOM'91*, pp. 694-700, Miami, Fl., April 1991.

[69] A. K. Gupta and N. D. Georganas, "Buffer Allocation in an ATM Switch with Output Buffer and Speed Constraints," *Proc. Canadian Conf. on Elec. and Comp. Eng.'91*, 42.1, Québec, Canada, Sept. 1991.

[70] P. J. Burke, "Delays in Single-Server Queues with Batch Input," *Operations Research*, Vol. 23, pp. 830-833, July-Aug. 1975.

[71] T. Meisling, "Discrete-Time Queueing Theory, " *Operations Research*, Vol. 6, pp. 96-105, Jan.-Feb. 1958.

[72] L. Kleinrock, *Queueing Systems*. Vol. 1, New York: John Wiley, 1975.

[73] U. Killat, "Asynchrone Zeitvielfachbermittlung fr Breitbandnetze," *Nachrichtentech. Z.*, Vol. 40, no. 8, pp. 572-577, 1987.

[74] A. Pattavina, "A Broadband Packet Switch with Input and Output Queueing," *Proc. ISS'90*, A9.3, 1990.

[75] A. Pattavina, "Fairness in a Broadband Packet Switch," *Proc. IEEE ICC'89*, 13.3, 1989.

[76] H. F. Badran and H. T. Mouftah, "Fairness for Broadband Integrated Switch Architectures under Backpressure Mechanism," *Proc. IEEE ICC'91*, 32.6, 1991.

[77] H. F. Badran and H. T. Mouftah, "Head of Line Arbitration in ATM Switches with Input-Output Buffering and Backpressure Control," *Proc. IEEE GLOBECOM'91*, 11.4, 1991.

[78] CCITT Draft Recommendation I.362: ATM Adaptation Layer functional description for B-ISDN, *Study Group XVIII*, Geneva, January 1990.

[79] J. S.-C. Chen and R. Guérin, "Performance Study of an Input Queueing Packet Switch with Two Priority Classes," *IEEE Trans. Commun.*, Vol. 39, No. 1, pp. 117-126, January 1991.

[80] S. C. Liew and K. W. Lu, "A 3-Stage Interconnection Structure for Very Large Packet Switches," *Proc. IEEE ICC'90*, 316.7, 1990.

[81] J. N. Giacopelli, T. T. Lee and W. E. Stephens, "Scalability Study of Self-Routing Packet Switch Fabrics for Very Large Scale Broadband ISDN Central Offices," *Proc. IEEE GLOBECOM'90*, 805.5, 1990.

[82] C. Clos, "A Study of Non-blocking Switching Networks," *Bell Syst. Tech. J.*, Vol. 32, 3/53, pp. 406-424.

[83] T. Feng,"A survey of interconnection Networks," *IEEE Computer*, pp. 12-27, December 1981.

[84] M. De Prycker and M. De Somer, "Performance of a Service Independent Switching Network with Distributed Control," *IEEE J. Select. Areas Commun.*, Vol. SAC-5, No. 8, pp. 1293-1301, Oct. 1987.

[85] A. K. Gupta, L. O. Barbosa and N. D. Georganas, "16 x 16 Limited Intermediate Buffer Switch Module for ATM Networks," *Proc. IEEE GLOBECOM'91*, 27.5, Phoenix, Az., Dec. 1991.

[86] A. K. Gupta, L. O. Barbosa and N. D. Georganas, "Limited Intermediate Buffer Switch Modules and their Interconnection Networks for B-ISDN," *Proc. IEEE ICC'92*, 354.7, Chicago, June 1992.

[87] A. K. Gupta, L. O. Barbosa and N. D. Georganas, "Switching Modules for ATM Switching Systems and their Interconnection Networks," *International Journal on Computer Networks and ISDN Systems*, (to appear).

[88] G. Gallassi, G. Rigolio and L. Fratta, "Bandwidth Assignments in Prioritized ATM Networks," *Proc. IEEE GLOBECOM'90*, 505.2, 1990.

[89] A. Iyengar and M. E. Zarki, "Switching Prioritized Packets," *Proc. IEEE GLOBECOM'89*, 32.5, 1989.

[90] P. Newman, "A Fast Packet Switch for the Integrated Services Backbone Network," *IEEE J. Select. Areas Commun.*, Vol. 6, No. 9, pp. 1468-1479, Dec. 1988.

[91] J.-Y. Le Boudec, "An Efficient Solution Method for Markov Models of ATM Links with Loss Priorities," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 3, pp. 408-417, Apr. 1991.

[92] K. Rothermel, "Priority Mechanism in ATM Networks," *Proc. IEEE GLOBECOM'90*, 505.1, 1990.

[93] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison-Wesley Publishing Company, 1987.

[94] L. Kleinrock, *Queueing Systems*. Vol. 2, New York: John Wiley, 1976.

[95] A. K. Gupta and N. D. Georganas, "Priority Performance of ATM Packet Switches," *Proc. INFOCOM'92*, pp. 727-733, Florence, Italy, May 1992.

[96] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. New York: Marcel Dekker, 1989.