ISBN 0-315 56105-X

Performance Analysis of an Input Access Scheme in a High Speed Packet Switch

Mojtaba Youssefi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering at
Concordia University
Montreal, Quebec, Canada

March 1990

ABSTRACT

Performance Analysis of an Input Access Scheme in a High Speed Packet Switch

Mojtaba Youssefi

The throughput and delay performance analysis of an Input Access Scheme in a High-Speed Packet Switch is presented for Broadband ISDN. In this architecture, each input of the packet switch maintains a separate queue for each of the outputs, resulting in a total of n^2 input queues for an $n x n$ switch. Using synchronous operation, at most one packet per input and output will be transferred in any slot. This means that in the devised access scheme the packets are selected among those available at the head of line of each input queue such that the overall throughput of the switch is maximized. An Exact throughput analysis of this access scheme is presented from its analogy to a well known combinatorial problem, and then an approximate analysis is derived which easily leads to numerical results for various switch sizes (≤ 32). These results are then compared to the simulation results of a Neural Network implementation in [16] which proves to be close. For larger switches analytically derived upper and lower bounds show results close to the optimal throughput. The bounds are very tight and approach to one for switch sizes on the order of a hundred for any traffic load.

The mean packet delay is then derived for the access scheme under consideration and its variance is bounded. Further, the application of these results for different classes of periodic traffic shows that for Voice and Teleconferencing the switch delay is almost negligible compared to the packet interarrival time, while for Video a priority scheme is described.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to Dr. M. Mehmet-Ali for his precious help, support and guidance during the entire development and preparation of this thesis.

Also, I gratefully acknowledge the support of Dr. J. Hayes for the opportunity to work on the research project which provided much support for the completion of the master program.

Table of Contents

Chapter 4

Chapter 5

Table of Figures

Table of Tables

CHAPTER 1

INTRODUCTION

The large transmission bandwidth offered by optical fibers is promising to change the telecommunications drastically. Multi-Gigabit rates have been successfully demonstrated [1,2] and therefore make it possible for a wide variety of new and different network architectures, but to satisfy the diverse Broadband requirements [3] of next generation services, new and integrated Network architectures must be considered. Broadband networks may provide quality services such as *High Definition Television* (HDTV), Videophony, and High-speed data, however, realization of these networks will require a unified and integrated transport mechanism to accommodate for all classes of services [4]. Currently there are two main types of Networks in operation, the first is based on Circuit Switching for real time services, such as Voice or Video, and the second is based on Packet Switching for all the Data transfer applications. However sharing of services within separated Networks is difficult since each must be maintained and controlled separately [5]. Further, this kind of hybrid Network is inflexible with respect to the mix of services it can switch [5,6].

Although there has been attempts to use both Circuit Switched and Packet Switched calls in an integrated switch by the process of minipacketization of the Circuit Switched calls, many of the administration and maintenance functions are still duplicated [7]. The trend is then towards Integrated Switching to accommodate for a wide and varying mix of services.

For a Circuit Switched based Integrated transport, the huge transmission bandwidth of optical fibers offers the possibility of providing Broadband services, howcver switch limitations in handling different traffic rates would not make it suitable for broadband Networks.

As an alternative, High-speed Packet Switching offers more flexibility in providing multi-rate services, because a variable bandwidth may be assigned on demand. It also has the advantage of being resource sharing in terms of the interconnection fabric and buffers. These considerations make Packet Switching one of the most promising switching methods to realize Network Integration.

1.1 ATM and Packet Switching

Recently, it was decided by the CCITT that the ultimate Broadband network will be based on the *Asynchronous Transfer Mode* (ATM) of operation [8]. As opposed to the *Synchronous Transfer Mode* (STM) frame references are not required, hence many synchronization problems are avoided. Instead of frames, information generated by any service is divided and labelled into separate blocks which are referred to as Cells or Packets. The flexibility of ATM is based on the bandwidth sharing, since there are no slot assignments and packets are only created when the information arrives at the Network interface.

The key element of the ATM implementation is the Packet. Each packet is comprised of two parts, the information field , carrying user data, and the header, which functions to identify the source and destination addresses of the packet. The size of the packet affects the overall performance of the system in two ways. The transmission efficiency is set by the percent overhead as embodied in the relative size of the header. The packetization delay is the time necessary to stuff the required number of bits into the defined block. The size of the packet is a balance of these two factors. At a greater cost of control, variable length packets may be used to achieve an even greater flexibility to avoid data fragmentation.

1.2 Architecture of a Packet Switch

There are two fundamental components required that make up any packet switch, the switching fabric, and the buffers. The interconnection network of the switching fabric has the function of routing the packets to their appropriate destinations and buffering smoothes fluctuations in traffic. Each of these has an affect on the overall system performance.

1.2.1 Interconnection Networks

Perhaps the simplest structure for routing packets is the single stage interconnection network. This is a crossbar configuration where there exists a different path for every input going to any of the outputs. There are n paths per input, which results in a total of n^2 communication paths (Figure 1.1). The difficulty with this method is that number of switching elements increases with the square of the number of input/output ports. Therefore this configuration is not practical for large switches.

An alternative are Multistage Interconnection Networks which are comprised of many identical and independent switching elements (usually 2x2) organized in a number of stages with interconnections between the stages arranged in such a way that each switching element may be controlled by a single digit of the packet header. The first digit(bit) of the header would set the first stage of the switch, the second digit to control the second and so on.

In this case the number of stages is $\log_2 n$ which results in a total of $\frac{n}{2}\log_2 n$ switching elements. So there is a significant reduction in terms of communication paths when compared to a fully connected network topology as described above.

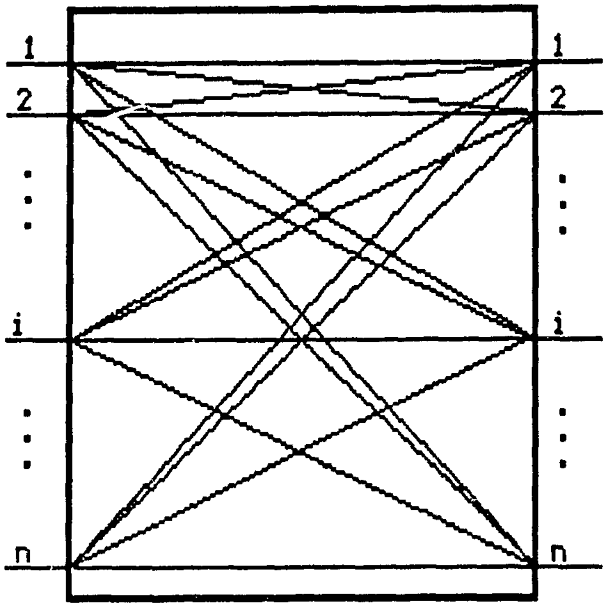


Figure 1.1. Single Stage Interconnection Network.
Δ Number of Communication Paths= n^2
Δ Number of Stages= 1

The performance of the Multistage Interconnection Networks is dependent on the number of communication paths of each switching element, the number of stages in the network, and the interconnection paths between the switches. A number of structures such as the Omega, Flip or the Banyan networks have been analyzed in [9],[10].

Multistage Interconnection Networks may be classified as Internally blocking or non-blocking, we consider each, in turn.

Since Multistage configurations are not fully connected, we may have contention between different packets to gain access to an output within the stages of the switch. For Banyan Networks [11], since there exists a unique path from every input to an output of the switch, packet contention at any stage would result in dropping one of the packets as illustrated in Figure 1.2.

One method to solve this problem is to introduce buffering within the stages of the switch to increase the overall throughput, however it has been shown in [12] that this will cause a variable delay through the switch which is undesirable for some time constrained applications. It is noted that contention is still possible between packets destined for different output ports.

Internally nonblocking Interconnection Networks provide a fixed delay through the switch fabric making them suitable for time constrained demands. As an example in this class we have the Batcher-Banyan Networks which consists of a bitonic sorter [13] to sort the packet destination addresses in an increasing or decreasing order, followed by the Banyan Network to route all the sorted packets. The sorter functions to prevent packets destined different output ports from conflicting inside the fabric.

Banyan Networks have the property that there is a single path from any input to any output and that each switching element has equal number of input/outputs. The Batcher-Banyan Networks are internally nonblocking, however contentions may still occur at the last stage of the switch among the packets trying to gain access to the same output port.

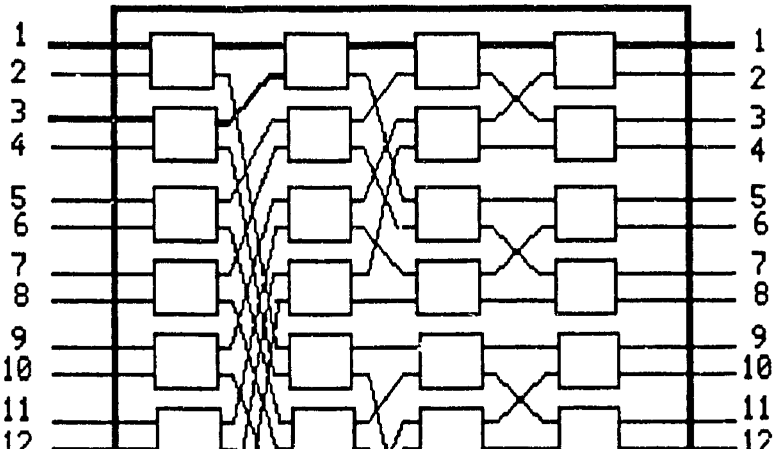


Figure 1.2. A 4 stage Banyan Multistage Interconnection Network.

Δ Number of switching elements= $\frac{n}{2}\log_2 n$.

1.2.2 Queueing Architectures

Two performance constraints imposed on any packet switch are the throughput and the delay. It is required that packets experience a minimum delay through the system while keeping the overall switch throughput at a maximum. These are often conflicting objectives in a Network and require some trade-offs.

For the class of Interconnection Networks of interest, internally nonblocking and displaying the self-routing property, we realize that without any externai buffering, the switch would experience a minimal delay but at the expense of a decreased throughput. Because of the random nature of traffic to the switch, buffering may be provided either at the input or at the output stage of the switch to increase the overall throughput. In this manner packets arriving to a busy input or those in conflict for access to an output port may be kept in the buffer to be transmitted in other time slots. These buffers may be thought as queues where the arriving packets line up to get served, usually in a First Come First Served (FCFS) basis. In the following a performance summary of the approaches to provide queueing (buffering) for a high-speed packet switch is presented.

a) Input Queueing:

Input queueing perhaps the simplest form of all, and with this method a separate buffer is provided at each input port of the switch and the arriving packets for each of the inputs are placed in these buffers and are served on a FCFS basis as illustrated in Figure 1.3.

The first packets in each of the input queues waiting for service are referred to as Head Of Line (HOL) packets, and with Input Queueing the HOL packets may contend for access to the switch outputs. If all HOL packets were destined for different outputs all the time, they may all be routed and transmitted, but this is not always true and there will be contention among the HOL packets destined for the same output(s). This is known as HOL blocking [14] and reduces the throughput to a sub-optimal value. It is shown in [1] that the maximum attainable throughput per input port is 0.5858 for the Input Queueing architecture.

b) Input queueing with a window:

To increase the throughput obtained in the above, a window scheme is proposed in [14] that considers not only the HOL packets in each of the queues, but up to a total of w packets (Figure 1.4). The switch throughput is then improved as the window size increases. So the algorithm for choosing the packets in the previous case m?y have to be iterated up to w times, thereby increasing the processing time.

c) Input Smoothing:

At each input, a frame structure is implemented with a size of b slots. At every slot, b packets per input or a total of nb packets may be transmitted, as shown in Figure 1.5. This scheme achieves a higher throughput however it requires a larger switching fabric, (nb x nb).

d) Output Queueing:

Placing the buffers immediately after the last stage of the switch fabric is known as Output Queueing (Figure 1.6). In any time slot, all packets arriving at the input ports of the switch are routed to the appropriate outputs and enter the queue to be served on a FCFS basis. In this manner, if one or more inputs have their HOL packets addressed to the same output, they may all be routed to that output. In order to implement this, the switch must operate n times faster than the Input/Outputs; nevertheless, the HOL blocking encountered with Input Queueing is avoided in this architecture. Thereafter, it is shown in [14] that

Output Queueing achieves an optimal throughput/delay performance, and a number of architectures have been proposed based on this type of buffering. As an example of Output Queueing the Knockout Switch proposed in [15] utilizes a fully-interconnected topology from each input to each output, and combined with a concentrator circuitry, chooses between packets destined to an output by a tournament algorithm which is then followed by an output buffer(queue). Although this method achieves ideal throughput/delay results, the implementation and packaging of output buffered systems are usually complex as the switch size n becomes large.

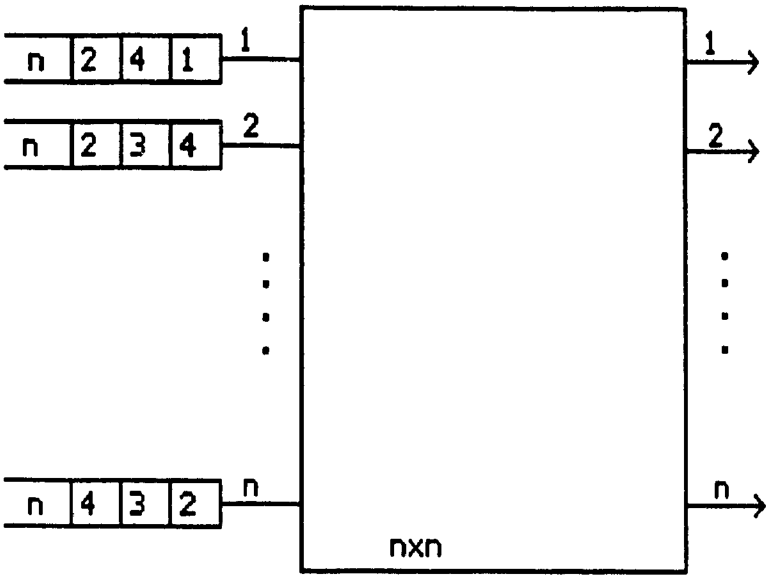


Figure 1.3. Input Queueing Architecture
Δ A single Queue per Input Port

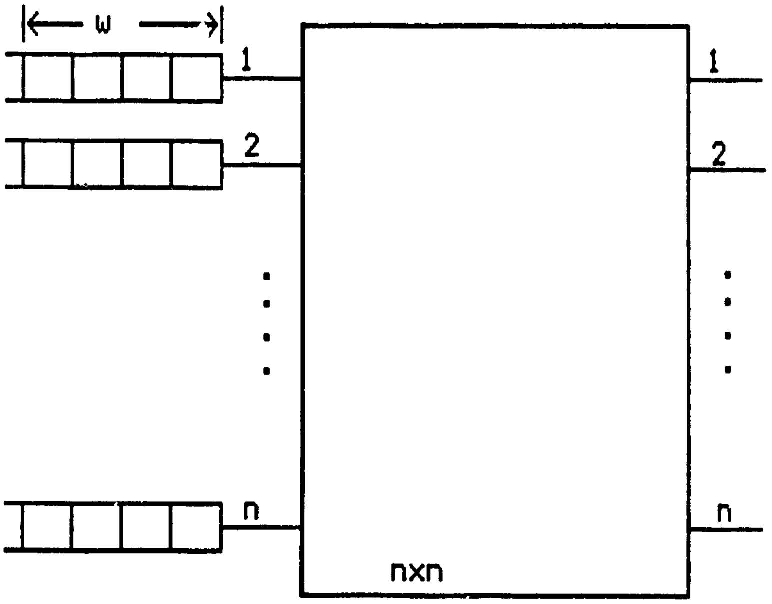


Figure 1.4. Input Queueing with a window
Δ A single Queue per input with a window

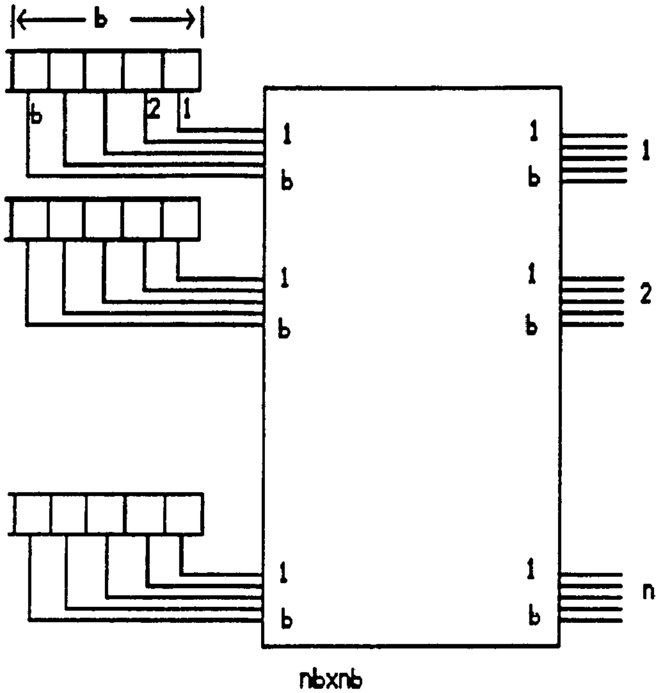


Figure 1.5. Input Smoothing
Δ A single Queue per input with a Frame

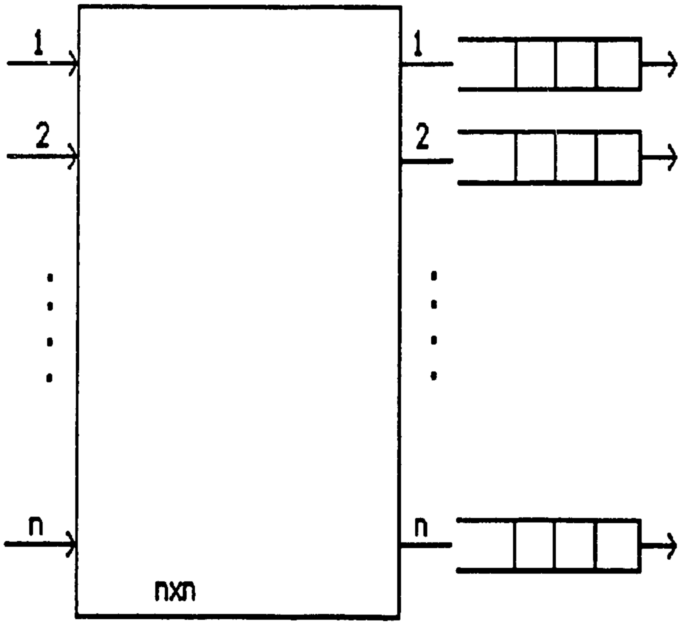


Figure 1.6. Output Queueing Architecture.
Δ Queues placed at each of the Output Ports
Δ Switch Operates n times faster than the Input/Outputs

1.3 The Proposed Input Access Scheme

The major difficulty with the input buffering scheme is the inherent HOL blocking which limits the throughput of the switch to 0.5858. Contending packets for an output port will result in long delays as the switch throughput gets close to the above value.

In the following, we consider a variation of the input queueing with a kind of window for synchronous time slotted operation. It will be assumed that each input will have a separate queue for each output, so for a switch of size n, there will be n queues at each input, one per output resulting in a total of n^2 input queues as shown in Figure 1.7. The choice of the available packets in any slot, is made in accordance with selecting at most one packet from each input port that corresponds to different outputs. This way, once a packet is selected from any of the inputs, no other packet in the subsequent inputs may be selected that corresponds to the same output.

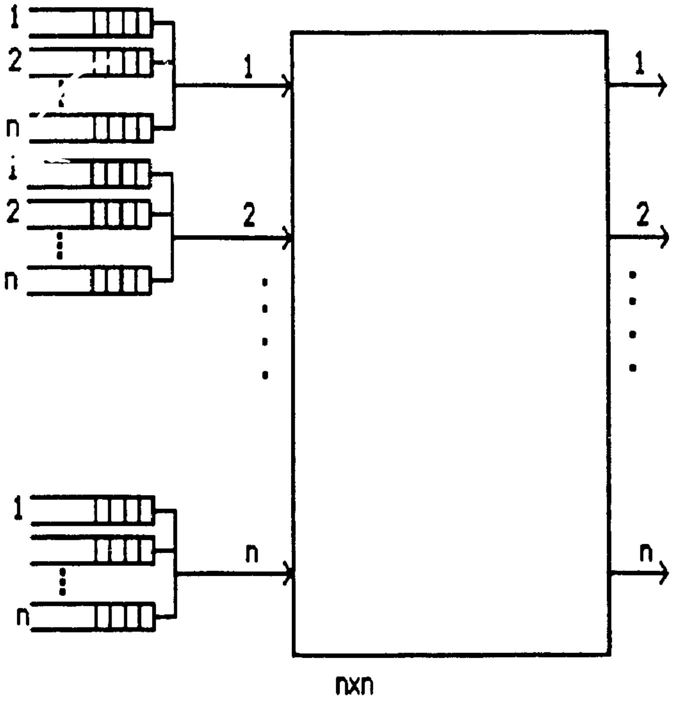


Figure 1.7. The Input Access Scheme Architecture.
Δ n Queues per Input Port

Each of the input queues will operate according to the FCFS discipline, and for synchronous operation, packets are chosen in each time slot from these HOL packets and are routed through the switch fabric and transmitted to the appropriate destinations.

1.3.1 Throughput Performance

We define the system throughput to be the maximum number of packets that may be picked during any slot. One way to determine the throughput is to find the maximum number of packets picked for each of the 2^{n^2} possible configurations corresponding to the n^2 input queues, and find their expected value for a given probability that an input queue is busy. This is the approach taken in chapter 2, where the exact analysis of the throughput is found using a combinatorial approach to solve for each of the 2^{n^2} configurations. The solution for any one configuration, is similar to a classical problem in combinatorics known as the rooks problem which makes use of an expansion method to find the maximum number of elements(in reference to packets) chosen to transmit in any slot. Using this technique, the throughput may be found as a function of the load, but application of this method for all of the 2^{n^2} configurations becomes quite cumbersome, even for switches of size 4.

Since with the above approach, a large number of calculations must be performed to determine the expe:ted throughput, we introduce an approximate solution in obtaining the throughput in chapter 3. There exists a distinct number of ways in placing any given number of packets in the system, and fo⁻ each of these configurations, the maximum number of packets chosen must be determined. The frequency of appearance of each maxima (i=0,1,...,n) is approximated using a combinatorial approach for a given number of packets(m=0,1,...,n^2) in the system. The expected throughput is then found as a function of the load, by taking the ratio of the frequency of each maximum to the total number of all frequencies of occurrences, to find the probability distribution of the number of packets picked for any number of packets in the system.

The approximate expected throughput found for a number of switches proves to be very close to the results of simulations obtained from the Neural Network implementation of the Input Access Scheme described in [16]. These results however, are limited to ($n \leq 32$) because of computational complexities, therefore further approximations were made to obtain a lower and an upper bound on the throughput so that switches of practical sizes may be analyzed. In fact it is shown in section 3.2.1 that for large switches (n>128), the expected throughput per input port is one for all loading conditions.

This result is quite significant for *Broadband ISDN*, since practical switch sizes for such applications are expected to be adequately large to support the number of users. So in effect, the throughput performance of the proposed scheme is ideal.

1.3.2 Delay Performance

In addition to the throughput, the second constraint imposed on any packet switch is its delay performance. If some time-constrained packets are to be transmitted through the switch, it is quite critical that they arrive to their destinations with a minimum delay, which should be less than the time it takes for a subsequent packet of the same service to arrive. This is elaborated upon in the sequel, but for now let us define the system delay. A basic assumption in the delay analysis is that the processing time of the packets is assumed to be negligible with respect to the switching delay, this means that the overall packet delay arises mainly from the switching network. The switching delay consists of the packet waiting time to get into service plus the packet transmission time and may be determined depending on the queueing architecture considered. Here we will discuss some of these architectures as well as the proposed Input Access Scheme.

a) Input Queueing Architecture:

It was shown that the limiting throughput for the input queueing architecture is 0.586; accordingly the delay is infinite at operating loads approaching that value. This decrease in throughput is caused by the HOL blocking which results in larger queue lengths, and ultimately when the throughput approaches the above value, the queue length would be infinite.

b) Output Queueing Architecture:

Output queueing achieves ideal performance results in terms of delay as well as the throughput because all arriving packets are sent to the appropriate output ports as soon as packets become available at the input ports of the switch. As mentioned, in this case the switch operates n times faster than the input/output ports such that even if all packets at the input ports are destined to the same output port, they may be properly routed to that output port.

The packet delay in this case will consist of the associated queueing delay at the output ports and the service time which is constant in this case. An M/D/1 queueing model is used to analyze the switch delay. As shown in chapter 4 the related packet delay is a few slots and the switch saturates only at very high loadings(>0.9).

c) Input Access Scheme:

It is shown in chapter 3 that the normalized throughput per input port is one for all practical switches and this assumption is applied in chapter 4 to determine the overall expected delay. Therefore, it follows that a single packet will be chosen from each of the input ports, in each time slot. Further this selection will be of a random nature among the available HOL packets, because it is equally likely that any one of the available packets will be chosen to transmit in a slot.

The expected delay is then approximated in the analysis in chapter 4 using the M/D/1 queueing model and results are found as a function of the load. It will also be required to determine the variance of this delay in order to evaluate an expected worst case scenario. Keeping in mind that the variance is a function of the order of service, we bound the order of service for this scheme as follows.

As discussed in the above, the selection among the available packets for each of the input ports is random, but each of the n^2 input queues operates according to FCFS. So we would expect the overall delay variance to be higher than the variance of a completely FCFS based system, because the random selection of the packets at the ports might leave some packet in the system for a long time. The overall variance is also less than that of a pure random selection scheme since the operation of each of the n^2 input queues is based on FCFS. Therefore we expect the overall variance to be bounded by the variance of FCFS and the random order of service.

The proposed Input Access Scheme performs as well as output queueing in terms of delay, hence the results are optimal for practical switches.

It was assumed that the overall delay would only consist of the switching delay and the worst case expected packet delay was found as a function of the load. Now it is interesting to examine how different classes of traffic such as Voice, Data, and Video are effected by this expected worst case delay. As it turns out for Voice, the delay is much less than the associated packet interarrival time, even when the switch is quite loaded.

For Data this delay may not be as critical as Voice, nevertheless calculations show that the delay encountered by the Data packets is also much less than the packetization delay.

The large bandwidth required for Video however, makes it more sensitive to delay variations, as shown in the results. Therefore the delay variations must be kept at a minimum to accommodate Video. A suggested method to improve these results would be to give priority to the time constrained packets such as those of Video and Voice at the expense of the Data services that are less critical to delay, however, even with this scheme, as the number of priority packets increases in the switch, the packet delay will become critical.

Another important issue to consider is the buffer size, since packets arriving to a full buffer are dropped. This is also considered in chapter 4 where the distribution function of the buffer length may be found for the Input Access Scheme as a function of the load. Appropriate selection of the buffer size may then be made to ensure that the probability of the buffer overflow is within an acceptable range.

Finally, it is noted that the Input Access Scheme described may be practically implemented by a proposal given in [17], suggesting its implementation using Neural Networks. A number of simulations were performed to this effect for a number of switches ($n \leq 16$).

CHAPTER 2

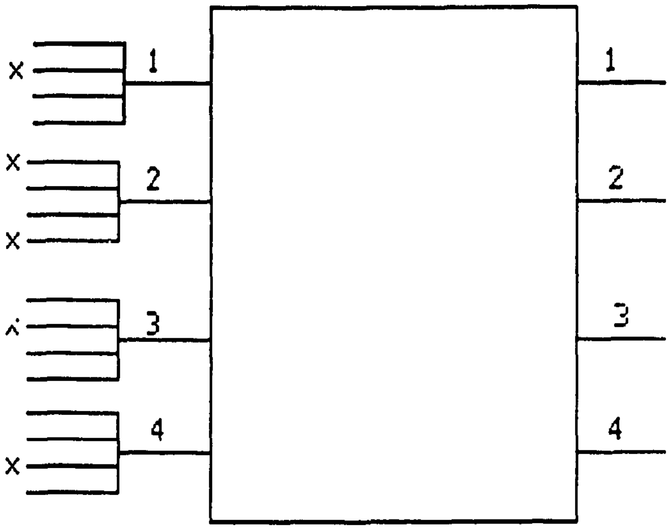
AN EXACT THROUGHPUT ANALYSIS OF THE INPUT ACCESS SCHEME

The throughput performance of the Input Access Scheme is presented in this chapter using an exact mathematical model. This model of the access mechanism is based on a classical combinatorial problem whose solution requires permutations with forbidden positions. We maintain that each Input has a seperate queue for each output, thus in an nxn switch, there are a total of n^2 Input queues resulting in 2^{n^2} possible configurations.

Using this model we may determine the maximum number of packets chosen to be transmitted in a slot for each of the 2^{n^2} configurations and finally find the average number of packets transmitted for a given probability of a busy queue ρ.

2.1 The Input Access characteristics

The input access scheme under consideration requires a maximum number of packets to be chosen during a slot for any configuration. The n^2 input queues may be represented by a square matrix (nxn) as shown in Figure 2.1, such that the rows represent the inputs and the columns would correspond to the outputs. It will be assumed that the probability of an input queue being busy is independent of the other queues, we also assume slot to slot independence. The V_{ij} element of the matrix will indicate the status of the j'th output queue at the i'th input, a nonzero element indicating a busy queue, a zero an idle queue. Further, each of the input queues may have a nonzero element in any slot with probability ρ, and if so, the nonzero element is marked by a cross in the input matrix in its appropriate cell as shown in Figure 2.1. Since each element may be zero or a one, for an nxn switch there would be a total of 2^{n^2} different configurations, and for each of these configurations we define the maximum throughput, T, to be the maximum number of nonzero elements that may be picked during any slot such that only one nonzero element is chosen from every row and column.



n=4	1	2	3	4
1	0	X	0	0
2	X	0	0	X
3	0	X	0	0
4	0	0	X	0

X Indicates a HOL packet is present

Figure 2.1 Matrix representation of the n^2 input queues.

2.2 Exact evaluation of the Throughput

In the following, we are interested in finding the number of ways that a maximum number of nonzero elements, for a given configuration can be picked. This distribution would indicate the maximum throughput of the switch for any of the 2^{n^2} configurations. Since at most only one nonzero element may be chosen from every row and column in any given configuration, by chosing any nonzero element, we must delete(not consider) the remaining nonzero elements corresponding to the same row and column of the chosen element. By chosing a nonzero element in a row and deleting the remaining elements in its corresponding column, we may be removing the only or one of the few packets that a subsequent row(input) may have for the same output(column). Hence the overall choice of packets must be in accordance with picking the maximum number of packets such that one packet is chosen at most in every row and column.

2.2.1 Rooks problem and Permutations with Forbidden positions

Next we describe a classical problem in combinatorics known as the Rook's problem [18] which is identical to ours. A rook is a chess board piece which takes on rows and columns. Rook's problem is placing k nontaking rooks with no two rooks on the same row or column, on a chess board of size n. Further, if some of the board cells are not accessible to placing a rook, they are referred to as forbidden positions. So the problem becomes placing k nontaking rooks on an arbitrary board. We make a distinction at this point that an arbitrary board may have any kind of shape, size, and perhaps be disconnected. This problem is analogous to the throughput analysis explained in the previous section, the nonzero elements correspond to the permitted regions and the zero elements are the forbidden regions(zones). And placing the maximum number of nontaking rooks corresponds to the choosing of the maximum number of packets that may be transmitted from an arbitrary board.

The solution to this problem requires permutations with forbidden positions, since each position(cell) in the board is distinct. This is similar to finding the number of permutations of n distinct elements, which we take as 1 to n, such that element k may not appear in the k^{th} position.

This is known as the 'Rencontres Problem' and may be illustrated by the following example. It is required to find the total number of permutations of 3 colored balls, taken as Red, Green, and Blue into 3 boxes of the same colors with the restriction that a ball can not be placed into a box having the same color, for example the Red ball may not be placed into the Red box and so on. Figure 2.2 shows this problem in the matrix format and the '0' represents the cells forbidden for choice. In this format when a ball is placed into one of the permitted boxes of the first row, the second ball may not be placed in the same column as the selected cell of the first row and this would continue until all the boxes are filled.

	R	G	B
R	0	X	X
G	X	0	X
B	X	X	0

'0' Forbidden cells for choice
'x' Permitted cells for choice
R=red G=green B=blue

Figure 2.2 Board representation of the 'Rencontres problem' for n=3.

However the above example only represents a class of what is required to solve our problem since the distribution of the forbidden positions corresponding to those idle queues may be quite arbitrary and located in any of the n^2 cells, not only confined to the k^{th} position of element k. The analogy to our problem is that the rows represent the individual inputs and the columns are the outputs of the switch. At the beginning of each slot, the forbidden positions(cells of the input matrix without any packets) are identified and the number of ways to pick the maximum number of elements is determined. In general a square of side n has n^2 cells and a nonzero element may or may not appear in each cell, clearly there are 2^{n^2} problems possible (this would include permutations without restrictions). Many of these problems(configurations) however, are not distinct since for enumeration purposes, it is the relative rather than the absolute position of the nonzero elements that is important. For example all input configurations having one permitted cell represent essentially the same problem. But is noted in [18] that the exact number of these distinct problems is not known.

2.2.2 Rook Polynomials

In the following we determine the number of ways that k nontaking rooks may be placed on an arbitrary board. The approach in this case according to Riordan [18], is to consider the permitted positions of the board and to determine the number of ways of placements for a subset. In this subset, the number of ways of the placement of k nontaking rooks may be easily found and expanding it one step at a time by adding a new permitted cell, a new result may be found. At each step we determine the polynomial corresponding to the distribution of the number of ways, and finally at the last step, when all the permitted cells are covered, the final distribution is obtained.

Let us assume that for k nontaking rooks on a board, that

r_k Number of ways of putting k nontaking rooks on a board.

And the generating function of r_k is defined to be

$$R(X) = \sum_{k=0}^{\infty} r_k X^k \tag{2.1}$$

$R(X)$ is called the rook polynomial and depicts the distribution of the number of elements into the permitted regions. The coefficients of the rook polynomial, r_k show the number of ways that k rooks may be placed for a given configuration. And in such a case, the first nonzero coefficient in r_k $(k = n, n-1, \ldots, 1)$ represents the maximum number of rooks(packets) that may be picked.

The major difficulty in finding the rook polynomial is determining the rook coefficients, r_k. So next we consider a systematic approach in finding r_k by denoting $R(X,C)$ as the rook polynomial of a chessboard C which may be of arbitrary size.

1) If the board C has i disjunct parts, which we take as $C_1, C_2, \ldots, C_i$, such that no cell of C_1 may be in the same row or column of the cells of $C_2, C_3, \ldots, C_i$ and so on, then the rook polynomial of the board C is defined by

$$R(X,C) = R(X,C_1)R(X,C_2)..R(X,C_i) \qquad (2.2)$$

This is illustrated by the example shown below,

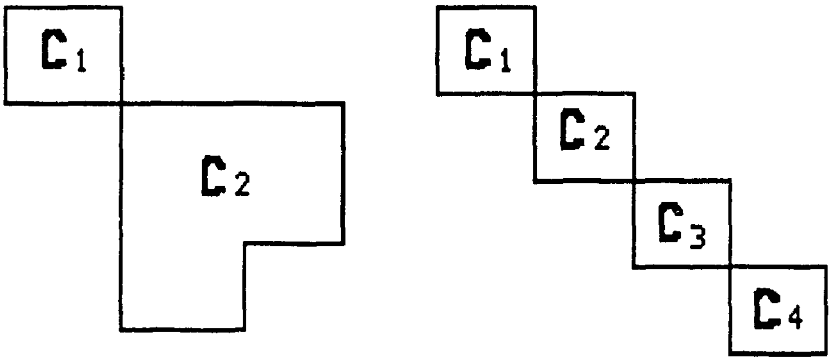


Figure 2.3 Representation of boards with disjunct parts

2) If the board can not be seperated into disjunct parts, then the *Expansion method* is used to find the rook polynomial. In general, the arrangements of k rooks on a board C may be thought of as the number of arrangements of k-1 rooks on another board C_i, not containing the row or the column of the k^{th} rook, plus the number of ways of placing k

rooks on another board C_e which is still a subset of the board C only that it does not contain the new cell.

This is an expansion process in which we start from a minimal array of cells, for which the number of arrangements is obvious or is easily found, adding a cell at a time and finding the number of arrangements for that board, to finally expanding to the desired board. Mathematically this may be expressed as

$$r_k(C) = r_{k-1}(C_i) + r_k(C_e) \tag{2.3}$$

and its corresponding generating function as

$$R(X,C) = XR(X,C_i) + R(X,C_e) \tag{2.4}$$

Next we attempt to solve an example to show the validity of the above. In Figure 2.4 we consider the case where the expansion method is used to find the rook polynomial. We start by considering the top left corner of the board (3 cells), Figure 2.4a, and matching it to a known polynomial, $R_1(X)$, which is

$$R_1(X) = 1 + 3X + X^2 \tag{2.5}$$

Next we expand the above board by adding an extra cell to get the board, shown in Figure 2.4b whose rook polynomial is

$$R_2(X) = R_1(X) + XL_2(X) \tag{2.6}$$

Where $L_2(X)$ is the polynomial obtained by removing all cells relating to the rows and columns of the new cell. In this case the corresponding polynomial for $L_2(X)$ is 1+2X ,hence

$$R_2(X) = 1 + 3X + X^2 + X(1 + 2X) \tag{2.7}$$

$$= 3X^2 + 4X + 1 \tag{2.8}$$

So continuing to add a new cell we get a new board, Figure 2.4c and find $L_3(X)$,

$$R_3(X) = R_2(X) + XL_3(X) \tag{2.9}$$

Deleting row 3 and column 3, we match the new board to find $L_3(X)$

$$L_3(X)=1+3X+X^2 \tag{2.10}$$

Using (2.10) in (2.9) we get

$$R_3(X)=1+5X+6X^2+X^3 \tag{2.11}$$

Adding the last cell to the board we obtain the final rook polynomial for the completed board (Figure 2.4d).

$$R_4(X)=R_3(X)+XL_4(X) \tag{2.12}$$

Deleting row 3 and column 1 we get $L_4(X)$

$$L_4(X)=1+3X+X^2 \tag{2.13}$$

Again we replace (2.13) in (2.12) to get $R_4(X)$

$$R_4(X)=1+6X+9X^2+2X^3 \tag{2.14}$$

So having completed all the permitted cells of the board, we regard $R_4(X)$ as the final rook polynomial corresponding to the shown configuration. The coefficients r_k, show the number of ways of picking k nonzero elements, and for this example there are two ways to pick a maximum of 3 rooks(packets). Thus the coefficient of the highest power of X is the desired result.

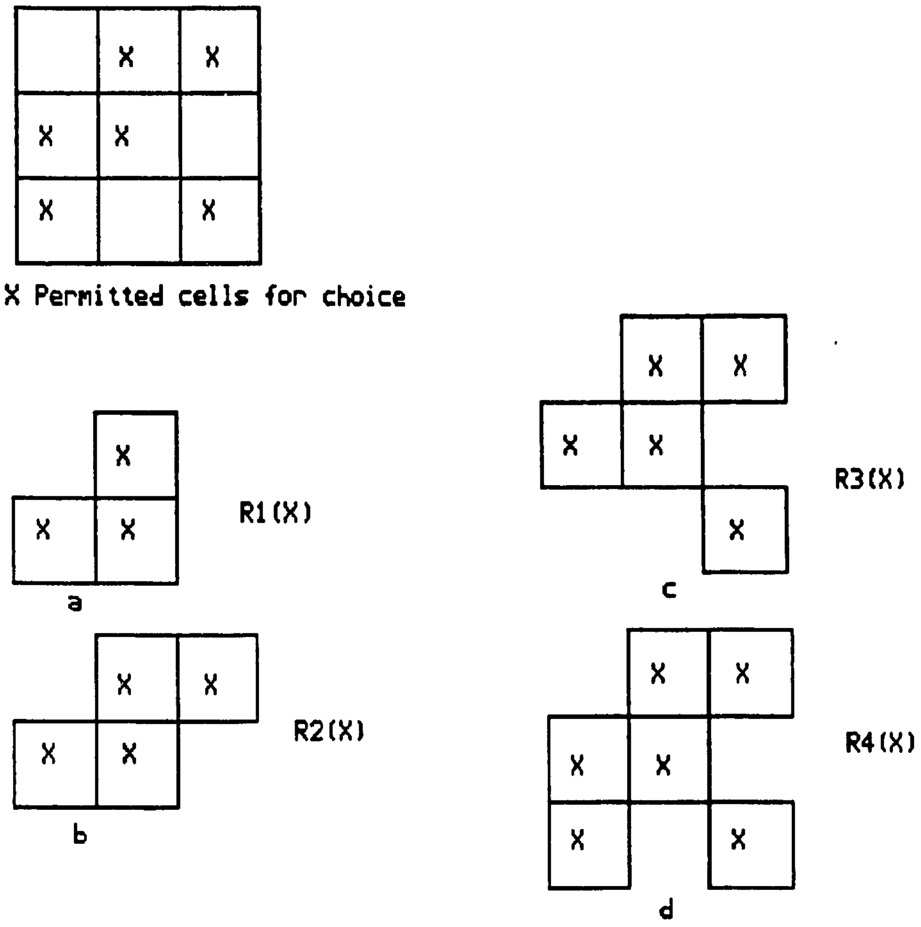


Figure 2.4a,b,c,d Example illustrating the expansion method to find the rook polynomial for a given configuration.

2.2.3 Board Complements

In some of the board configurations, there are fewer permitted cells than forbidden cells and calculations of $R(X)$ is much simpler when the number of cells considered is minimal. In cases where there are more forbidden cells than the permitted ones, we define the complement of the board, considering the forbidden cells as the permitted ones and proceediı,g with the calculations of the rook polynomial. The rook polynomial of the complement of the board may then be used to find the rook polynomial of the actual permitted cells.

Assume that $Q(X)$ is the rook polynomial of the complement of a board and $R(X)$ is related to $Q(X)$ by the transformation derived in [18],

$$Q(X) = \sum_{k=0}^{n} r_k(-X)^k S_{n-k}(X) \tag{2.15}$$

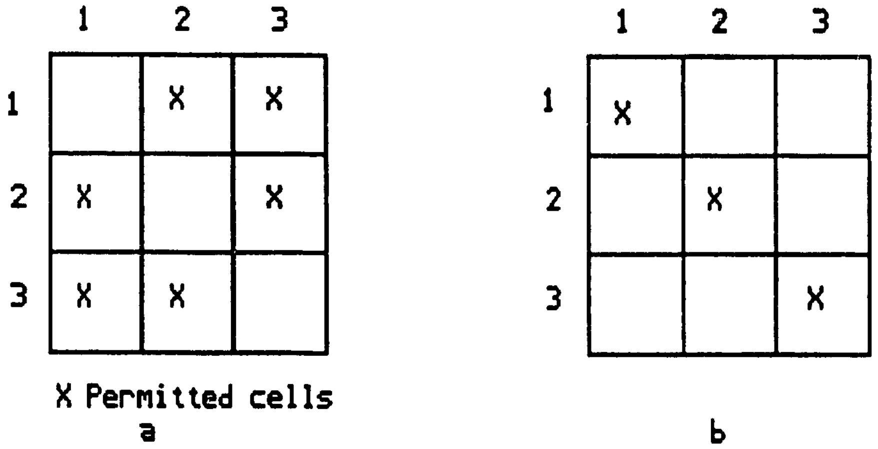
where $S_j(X)$ is the rook polynomial of a square board of size j having j^2 permitted cells. For example $S_2(X)$ is the rook polynomial of square of 4 cells, which turns out to be $1+4X+2X^2$.

However as the size of the square board j increases, calculations of $S_j(X)$ become long and tedious, so to simplify the matter, a recurrence relationship was derived in [18] which relates S_{n+1} to S_n and S_{n-1}, and is given by

$$S_{n+1}(X) = \{1+(2n+1)X\}S_n(X) - n^2X^2S_{n-1}(X) \tag{2.16}$$

Given the above relationship and (2.15), the polynomial of the complement of the board, $Q(X)$ may be found using $R(X)$.

To illustrate this point, let us consider the 'Rencontres problem' introduced earlier in Figure 2.2, where the placement of element k was forbidden for the k^{th} position.



	1	2	3
1		X	X
2	X		X
3	X	X	

X Permitted cells

a

	1	2	3
1	X		
2		X	
3			X

b

Figure 2.5a,b Example where the complement of a board is used to determine the rook polynomial.

As may be seen in Figure 2.5, the rook polynomial of the forbidden positions is simpler to find than that of the permitted ones, hence we find the rook polynomial of the complement of the board and call it $R(X)$ and then use the above transformation to obtain $Q(X)$, the polynomial of the permitted region of the configuration.

Since the forbidden positions are not in the same row or column, we use the relationship defined in (2.2) to solve for $R(X)$. For a single cell the rook polynomial is $1+X$, therefore $R(X)$ for a board with 3 disjunct cells is

$$R(X,C) = (1+X)(1+X)(1+X) \tag{2.17}$$

$$= (1+X)^3 \tag{2.18}$$

$$= 1+3X+3X^2+X^3 \tag{2.19}$$

Since $R(X)$ is the polynomial of the forbidden positions, we find $Q(X)$ by equation (2.15)

$$Q(X) = \sum_{k=0}^{n} r_k(-k)^k S_{n-k}(X) \tag{2.20}$$

$$= S_3(X) - 3XS_2(X) + 3X^2S_1(X) - X^3S_0(X) \tag{2.21}$$

Similarly $S_n(X)$ may be found using (2.16) and

$$S_0(X) = 1 \tag{2.22}$$

$$S_1(X) = 1 + X \tag{2.23}$$

$$S_2(X) = 1 + 4X + 2X^2 \tag{2.24}$$

$$S_3(X) = 1 + 9X + 18X^2 + 6X^3 \tag{2.25}$$

Now we substitute the above to get $Q(X)$

$$Q(X) = 1 + 6X + 9X^2 + 2X^3 \tag{2.26}$$

So now the coefficients of $Q(X)$, namely q_k, show the distribution of the number of nonzero elements that may be picked for the given configuration.

As it can be seen by the above examples, calculation of the rook polynomials could be a lengthy process even for small board sizes (4x4). Given an arbitrary board(input matrix configuration), we must expand cells from a known configuration(whose rook polynomial is known) to finally complete all the cells and get the final rook polynomial.

These calculations would have been simplified if the board equivalences were known, because some boards that are not geometrically compatible (with different forbidden regions) have the same rook polynomial; also sometimes it may be easier to find the polynomial of the complement of the board, in which case the true rook polynomial may be found by the transformation shown in (2.15).

Therefore for an nxn switch, It is theoretically possible to determine the maximum number of packets chosen for each of the 2^{n^2} configurations using the above, and then to determine the average number of packets that may be transmitted for a given ρ, however computational complexities involved would definitely limit these results to n in the order of 4 and that is why no numerical results are presented based on this solution.

CHAPTER 3

AN APPROXIMATE THROUGHPUT ANALYSIS OF THE INPUT ACCESS SCHEME

In this chapter the throughput is determined for a switch with symmetric random inputs using an approximate mathematical model of the Input Access Scheme in selecting the maximum number of packets. An Upper and a Lower bound to the throughput is also devised in section 3.2 to determine the bounds for large switches (n>32).

The throughput results of the Input Queueing Architecture described in Chapter 1, are analysed and compared in section 3.3 with those of the Input Access Scheme. These results display the superior throughput performance of the switch using the Input Access Scheme.

3.1 Throughput Analysis

In order to find the optimum, it is required is to find the throughput based on picking the maximum number of allowable packets for any of the switch configurations. For random input matrices we shall assume that the random variable denoting whether an input queue is busy or not V_{ij} , as defined in the previous chapter, is independent and identically distributed for symmetric traffic with Bernoulli distribution as,

$$V_{ij} = \begin{cases} 1 \text{ with probability } \rho \\ 0 \quad " \qquad\quad " \ (1-\rho) \end{cases}$$

Moreover, as in the rook problem, we introduce the allowed and forbidden positions of the input matrix to correspond to the nonzero and zero elements, respectively; and placement of maximum number of rooks to the choice of maximum number of packets in the switching problem. In this approach, we shall condition on the number of nonzero elements in the input matrix (number of busy input queues) and will then distribute these nonzero elements in distinct ways which result in the same maximum number of packets chosen.

Now we attempt to determine first the conditional and then unconditional distribution of the number of chosen packets. The probability distribution of the number of chosen packets would then lead us to obtain the throughput for any arbitrary switch size n.

Let us define,

n Switch size

m The conditional value of the number of nonzero elements, $m \leq n^2$.

$C_n(i,m)$ The number of ways that a maximum number of i nonzero elements may be chosen for a given m,n where $0 \leq i \leq m$.

$P_n(i \mid m)$ as the conditional probability that i nonzero elements will be chosen.

Further define

$$C_n(m) = \sum_{i=0}^{n} C_n(i,m) \tag{3.2}$$

Since $C_n(i,m)$ denotes mutually exclusive events, therefore,

$$P_n(i \mid m) = \frac{C_n(i,m)}{C_n(m)} \tag{3.3}$$

then the unconditional probability of choosing i nonzero elements is

$$P_n(i) = \sum_{m=0}^{n^2} P_n(i \mid m) \binom{n^2}{m} \rho^m (1-\rho)^{(n^2-m)} \tag{3.4}$$

The approach taken here is to determine $C_n(i,m)$ implicitly without enumerating each distinct case so as to minimize the calculations.

We shall now present two statements that will help to determine $C_n(i,m)$, *Statement I:*

The number of chosen nonzero elements for a given distribution of nonzero elements is equal to i if and only if the maximum number of nonzero elements that may be picked with no more than one of them on any row or column is equal to i.

Statement II:

Let us define x and y as the number of rows and columns with at least a single nonzero element. Then the maximum number of nonzero elements that may be chosen, i, is bounded by $i \leq \min(x, y)$. This is true because i is bounded by the allocation of the minimum of the inputs or outputs. This is shown in the example (Figure 3.1).

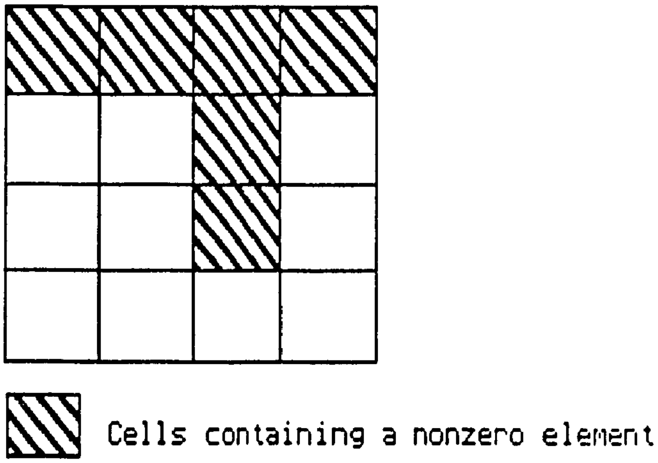


Figure 3.1 An example showing the upper bound of the chosen elements, (x=3,y=4)

From now on, the chosen nonzero elements will be referred to as the *essential elements*

Let us assume a given number of nonzero elements m, then a choice of i essential elements partitions the input matrix into four distinct regions. They are defined as the common elements of,

Region I: Chosen rows and columns.

Region II: Chosen rows and non chosen columns.

Region III: Chosen columns and non chosen rows.

Region IV: Non chosen rows and columns

If the number of essential elements is to be i, then they must be located in Region I which contains i^2 cells, in conformity with statement I such that each essential element would be located on a different row and column. Note that this is similar to the distribution of i nontaking rooks on a board of size i^2.

In Region I the total number of ways to distribute i essential elements is $i!$. Also the rows(columns) of these essential elements may be chosen independently out of the possible n, resulting in, $\binom{n}{i}^2$ ways to choose these rows and columns.

Having distributed the i essential elements, it remains to distribute the remaining $m-i$ nonzero elements into unoccupied cells without violating statements I or II. Of the regions introduced, region IV may not contain any of these remaining elements because placement of any nonzero element in that region would change the maximum number of essential elements hence contradicting statement I.

Region IV contains a total of $(n-i)^2$ cells in which none of the nonzero elements will be distributed in. The unoccupied cells of region I (a total of i^2-i) and all the cells of region II or III may have the remaining nonzero elements. This means that the total number of cells to distribute the remaining $(m-i)$ elements would be $n^2 - i - (n-i)^2$.

Careful inspection of the above shows that the distribution of the $m-i$ elements may not include both Regions II and III simultaneously. But the remaining cells of the region I however, do not conflict with statements I or II. Some specific configurations as seen in the example (Figure 3.2) would violate statement I when regions II and III are used simultaneously. Therefore, we shall restrict our choices to the rows(columns) of the i essential elements as to conform with both statements.

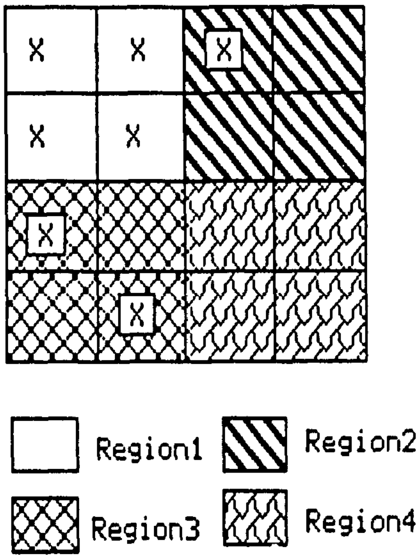


Figure 3.2 Example illustrating violation of statement I if regions II and III are to be used simultaneously (n=4,m=7,i=2).

So the available cells to distribute the remaining $m-i$ elements is $(ni-i)$ where ni corresponds to the total number of cells in the row(column) of the essential elements and i is the number of cells already occupied by the said essential elements. The total number of ways of distributing $m-i$ nonzero elements into $(ni-i)$ cells is defined by the Binomial coefficient $\begin{pmatrix} ni-i \\ m-i \end{pmatrix}$.

Finally for a given m and i, the above distribution of essential and non essential elements result in a distinct distribution of the nonzero elements into the cells. The number of ways that a maximum of i nonzero elements may be chosen is given by,

$$C_n(i,m) = i!\begin{pmatrix} n \\ i \end{pmatrix}^2 \begin{pmatrix} ni-i \\ m-i \end{pmatrix} \qquad m \geq n \quad (3.5)$$

Having determined the distribution of the essential and non essential elements for a given m and i, then $P_n(i \mid m)$ is given by,

$$P_n(i \mid m) = \frac{C_n(i,m)}{C_n(m)} = \frac{i!\begin{pmatrix} n \\ i \end{pmatrix}^2 \begin{pmatrix} ni-i \\ m-i \end{pmatrix}}{\sum_{k=0}^{n} k!\begin{pmatrix} n \\ k \end{pmatrix}^2 \begin{pmatrix} nk-k \\ m-k \end{pmatrix}} \qquad (3.6)$$

Now the unconditional distribution of the number of nonzero elements chosen is given by (3.4),

$$P_n(i) = \sum_{m=0}^{n^2} P_n(i \mid m)\begin{pmatrix} n^2 \\ m \end{pmatrix} \rho^m (1-\rho)^{(n^2-m)}$$

Replacing the expression found for $P_n(i \mid m)$, we get a simple closed form solution for $P_n(i)$ equal to,

$$P_n(i) = \sum_{m=0}^{n^2} \left\{ \frac{i!\begin{pmatrix} n \\ i \end{pmatrix}^2 \begin{pmatrix} ni-i \\ m-i \end{pmatrix}}{\sum_{k=0}^{n} k!\begin{pmatrix} n \\ k \end{pmatrix}^2 \begin{pmatrix} nk-k \\ m-k \end{pmatrix}} \right\} \begin{pmatrix} n^2 \\ m \end{pmatrix} \rho^m (1-\rho)^{(n^2-m)} \qquad (3.7)$$

Next we define the expected throughput $E\{T\}$ per port to be ,

$$\frac{E\{\Gamma\}}{n}=\frac{\sum_{i=0}^{n} iP_n(i)}{n} \tag{3.8}$$

Note that a port refers to the n queues at an input collectively.

3.1.1 Approximations

For a switch of size n having n^2 cells, there are a total of 2^{n^2} possible configurations, and according to the above definitions this must be equal to

$$\sum_{m=0}^{n^2}\sum_{i=0}^{n} C_n(i,m)=2^{n^2} \tag{3.9}$$

This relation otherwise states that each of these configurations must map to only one value of a maximum i (i=0,1,...,n). Theoretically, we must determine the maximum number of nonzero elements that may be picked for a given configuration, regardless of the number of combinations for which a maximum occurs. This means that the analysis presented in the previous section is not exact but an approximation, because according to the definition of $C_n(i,m)$, the maximum number of nonzero elements are picked in a distinct manner such that for any configuration, the same number of chosen packets i, may be counted many distinct times which is not required.

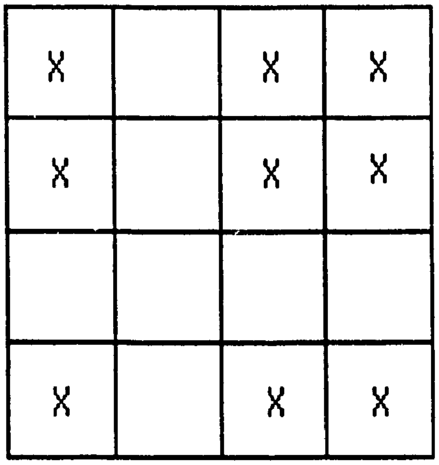
Perhaps this may be clarified by the example illustrated in Figure 3.3, where the 9 nonzero elements form a specific configuration which is a subset of the 2^{16} possible configurations. For this example, we may clearly pick a maximum of 3 nonzero elements and enumeration of all other combinations in which the maximum 3 occurs is not required. For a configuration such as Figure 3.2, a maximum of 3 chosen packets is counted 9 times, whereas in reality we are interested in choosing only one way to map a specific configuration.

The overcounting of the maximums in the approximate analysis, results in a total number of configurations that exceeds 2^{n^2}, but as we defined $P_n(i,m)$ in (3.3) the probabilities would be normalized to the total maximum number $C_n(m)$.

3.1.2 Results

Using this approximate analysis, a number of runs were made for different values of n where the throughput per input port was found as is shown in Figure 3.4. Moreover, it was mentioned that the physical implementation of the above controller uses Neural Networks. Simulation results of a Neural Network implementation presented in [16] for various switches, would provide verification of the results of the performance analysis. This is illustrated by comparing the results of the two methods in Figure 3.5 where different cases of n, are shown. The above figure shows that this analysis is a close approximation to the simulation results of the Neural Networks for all values of n.

Clearly, for each given input matrix, there is a maximum number of nonzero elements that may be chosen. However, usually there are several ways of choosing this maximum number of elements. It turns out that this analysis for each given matrix determines the total number of ways of choosing the maximum number of elements. Fortunately, the distribution of the chosen packets is more like an impulse function located at n, results are very close.



X		X	X
X		X	X
X		X	X

Figure 3.3 Configuration of 9 nonzero elements and maximum i set at 3.

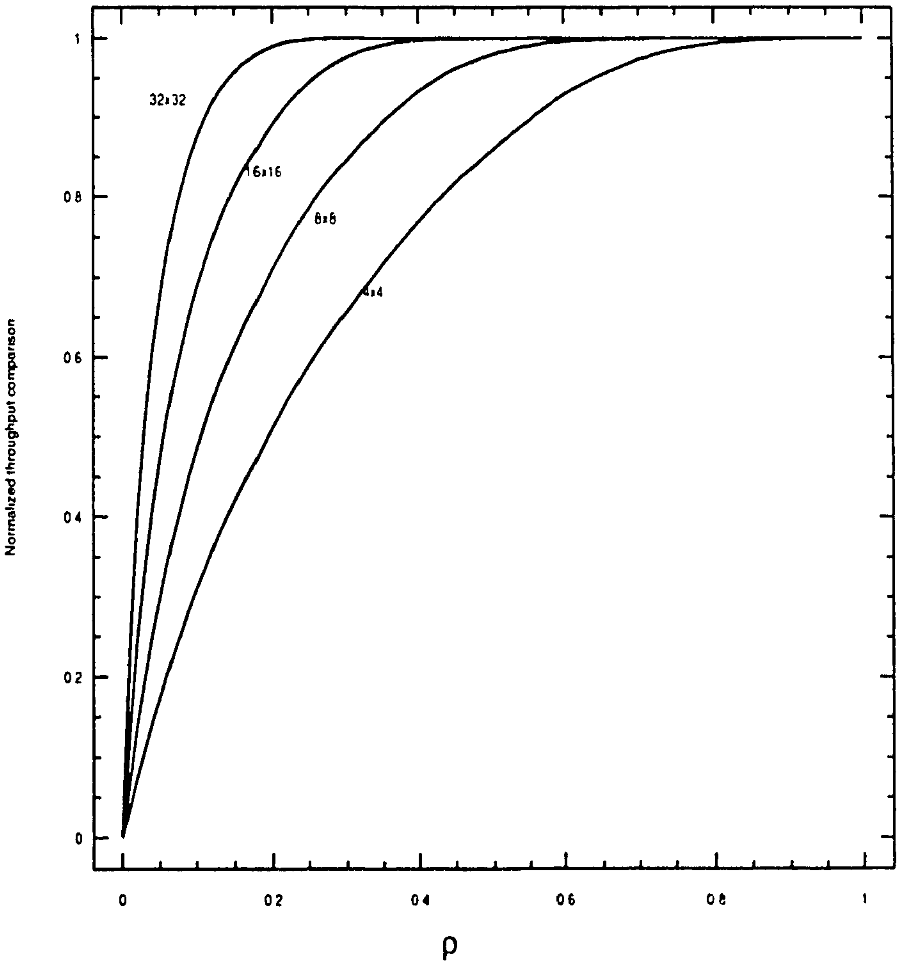
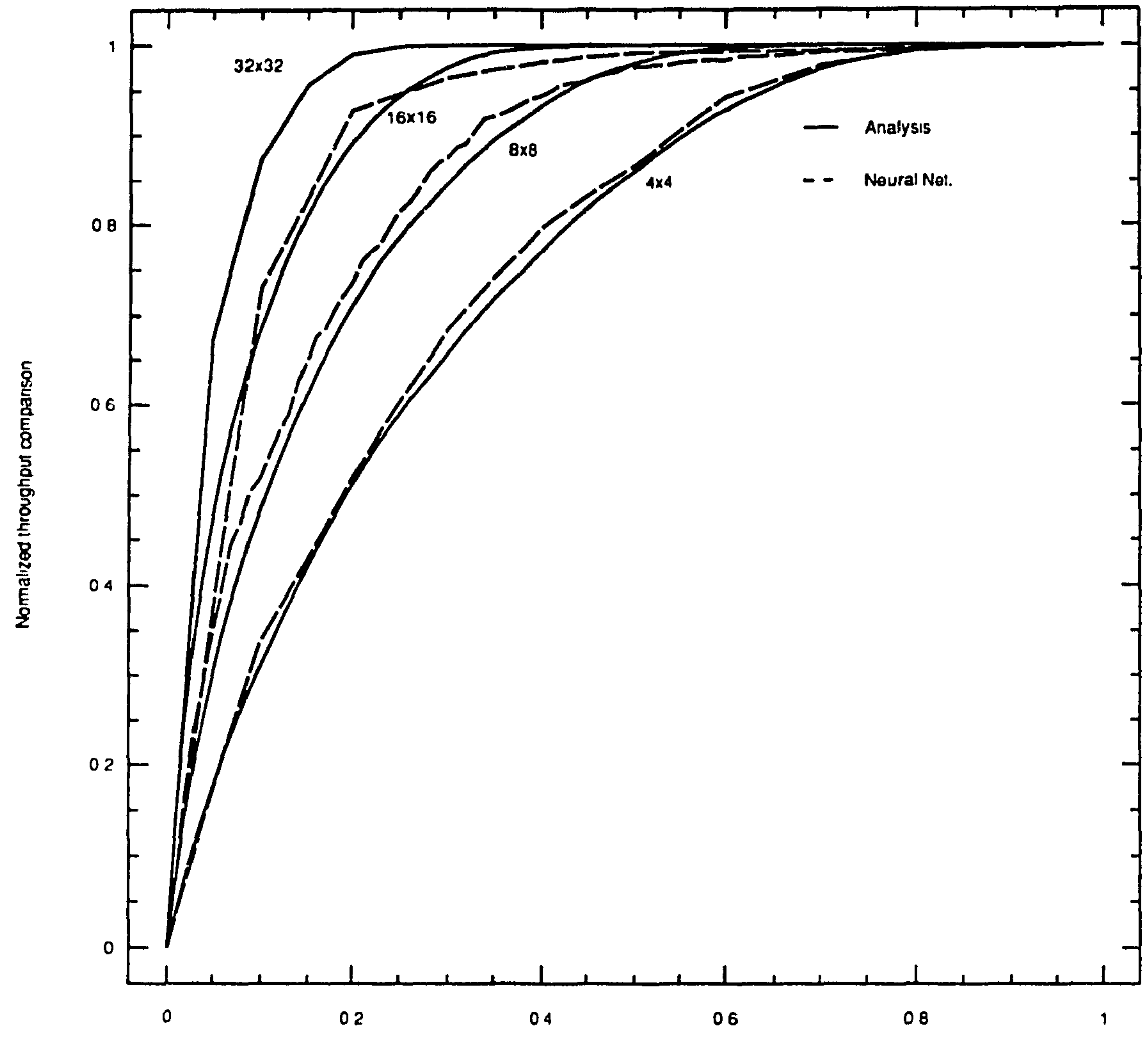


Figure 3.4 Throughput results of the analysis for various switch sizes.



Probability that an input queue is busy, ρ

Figure 3.5 Comparison of the throughput results of Neural Networks and the performance analysis for various switch sizes.

3.2 Evaluating Throughput Bounds

Numerical throughput results of the previous analysis were only shown for switches of r-ximum size 32x32. Results for larger switches were not found because of the computational complexity and the overflow of floating point numbers.

It is often required to analyze larger switches (>32) and their throughput performance, so we shall devise a new approach that would yield a lower bound result and its extension to obtain an upper bound.

It is equally interesting to observe the limiting behavior of the throughput as the switch size gets larger and approaches infinity $n \rightarrow \infty$, but unfortunately the previous analysis would not easily provide such results. However using a lower bound analysis we are able to find the limiting throughput and show that the throughput per input port approaches one as $n \rightarrow \infty$ for $\rho > 0$. This might have been suspected by the results of the former analysis for n=4,8,16,32 since as the switch size gets larger, there will be enough busy queues at each input port to allow a choice of a packet without having a conflict at the outputs.

3.2.1 Lower Bound Analysis

In this section, we determine a lower bound of the throughput for the switch under consideration. As we had previously defined in (3.1), the random variable denoting whether an input queue is busy or not ,V_{ij}, is independent and indentically distributed with Bernoulli distribution,

$$V_{ij} = \{ \begin{matrix} 1 \text{ with probability } \rho \\ 0 \quad " \qquad\quad " \ (1-\rho) \end{matrix}$$

In the input access scheme described earlier, we try to choose a nonzero element from each row with no more than one of them from the same column such that a chosen element in a given row eliminates the remaining nonzero elements in the same column for consideration in the following rows still not considered. So starting with the first row, we pick a nonzero element and then eliminate its corresponding column for the choice of the following rows. In the first column there are a total of n elements available. Having chosen one of them, in the second row there would be $(n-1)$elements available and $\{n-(i-1)\}$ elements for the i^{th} row.

Let u_i denote the value of the element chosen from row i and

$$u_i = \{\begin{matrix} 1 & \text{if chosen element was nonzero} \\ 0 & \text{otherwise} \end{matrix} \tag{3.10}$$

u_i is a Bernoulli random variable with a probability distribution given by

$$Pr\{u_i = 1\} \quad = Pr\{\text{at least one nonzero element among } \{n-(i-1)\} \text{ elements}\}$$

$$= 1-(1-\rho)^{n-(i-1)} \tag{3.11}$$

$$Pr\{u_i = 0\} \quad = Pr\{\text{No nonzero element among } \{n-(i-1)\} \text{ elements}\}$$

$$= (1-\rho)^{n-(i-1)} \tag{3.12}$$

Let us also define y as the random variable denoting the total number of nonzero elements chosen to be

$$y = u_1 + u_2 + \ldots + u_i + \ldots + u_n \tag{3.13}$$

In this case, the expected number of packets chosen during a slot is given by

$$E\{y\} = \sum_{i=1}^{n} E\{u_i\} \tag{3.14}$$

$$= \sum_{i=1}^{n} \{1-(1-\rho)^{n-(i-1)}\} \tag{3.15}$$

Separating the terms of the summation we obtain,

$$E\{y\} = n - \sum_{i=1}^{n} (1-\rho)^{n-(i-1)} \tag{3.16}$$

The above may be expressed in a closed form

$$E\{y\} = n - \frac{(1-\rho)-(1-\rho)^{n+1}}{\rho} \tag{3.17}$$

This expected value is the total number of packets chosen in a slot, and the average numbe of packets chosen per input port is given by

$$T = \frac{E\{y\}}{n} = 1 - \frac{(1-\rho)-(1-\rho)^{n+1}}{n\rho} \tag{3.18}$$

In each row, an element is chosen from the set of available elements in a random manner. On the other hand, the Input Access Scheme further looks ahead to choose an element corresponding to a column with the least number of elements. This obviously results in a lower bound as illustrated in Figure 3.6 where a few columns are eliminated and the only nonzero elements of a subsequent row are contained in those columns.

Next we find the limiting behavior of the switch using the lower bound analysis

$$T_{n\to\infty} = \lim_{n\to\infty}\left[\frac{E\{y\}}{n}\right] \tag{3.19}$$

Substituting equation (3.17) for $E\{y\}$ we get

$$T_{n\to\infty} = \lim_{n\to\infty}\left[1 - \frac{(1-\rho)-(1-\rho)^{n+1}}{n\rho}\right] \tag{3.20}$$

$$= 1 - \lim_{n\to\infty}\left[\frac{(1-\rho)-(1-\rho)^{n+1}}{n\rho}\right] \tag{3.21}$$

$$= 1 - \frac{(1-\rho)-(1-\rho)^{\infty}}{\infty} \tag{3.22}$$

$$= \begin{cases} 1 & \rho > 0 \\ 0 & \rho = 0 \end{cases} \tag{3.23}$$

This result, in effect shows that the average throughput per input port is one under any loading conditions for large switch sizes ($n > 64$).

3.2.2 Upper Bound Analysis

As it was explained in the lower bound analysis, the column of a chosen nonzero element in any row is eliminated and the nonzero element of that column may not be chosen for subsequent rows still not considered. This is not optimum, since some inputs with nonzero elements may remain idle(not be chosen) because of the deleted columns. Ideally we require that a nonzero element is chosen from each row(if possible), if there is at least one or more elements in each row. This way all the n elements in each row are available for making a choice, hence this is the basis for our upper bound results. This is an upper bound because we are dropping the condition that no more than one nonzero element is chosen from each column.

Defining u_i as before, we obtain the probability distribution given by

$$Pr\{u_i = 1\} = Pr\{\text{at least one nonzero element out of } n\}$$

$$= 1-(1-\rho)^n \tag{3.24}$$

$$Pr\{u_i = 0\} = (1-\rho)^n \tag{3.25}$$

In a similar manner to (3.13), we define the random variable y as the sum of u_i

$$y = u_1 + u_2 + \dots + u_i + \dots + u_n \tag{3.26}$$

and its expected value as

$$E\{y\} = \sum_{i=1}^{n} E\{u_i\} \tag{3.27}$$

$$= \sum_{i=1}^{n} 1-(1-\rho)^n \tag{3.28}$$

$$= n[1-(1-\rho)^n] \tag{3.29}$$

In this case, the average throughput per input port T, will be

$$T = \frac{E\{y\}}{n} = 1-(1-\rho)^n \tag{3.30}$$

The limiting behavior of the switch ($n \rightarrow \infty$) using the upper bound results is

$$\lim_{n \rightarrow \infty} \frac{E\{y\}}{n} = \lim_{n \rightarrow \infty} [1-(1-\rho)^n] \tag{3.31}$$

$$= \begin{cases} 1 & \rho > 0 \\ 0 & \rho = 0 \end{cases} \tag{3.32}$$

Thus the limiting throughput per input port is 1 as $n \rightarrow \infty$ for the upper bound as well as the lower bound analysis results.

3.2.3 Lower and Upper Bound Results

The relative simplicity in the analysis of the upper and lower bounds lead to numerical results for larger switch sizes (n=64,128,256,512). This is of particular interest since practical Broadband switches are required to support a large number of inputs.

To show the validity of the bounds, we shall compare the throughput results to those simulation results of the Neural Networks, and the analysis in section 3.1. These results are illustrated in Figures 3.7-3.10, where the normalized throughput is shown for, the Neural Network simulation results or the analysis in section 3.1, the lower bound, and the upper bound for switch sizes n=4,8,16,32 respectively.

These figures show that the upper bound is much closer to the results of the analysis than the lower bound, in fact these results get even closer for the last case(n =32). The discrepancy between the upper bound and the analysis is due to an assumption in the derivation of the former that if there is at least one nonzero element in each row, we are able to choose one packet from each row. This is not true in some cases as seen in the example of Figure 3.11, where at most two packets may be chosen and not 4; But as ρ gets larger the upper bound and the analysis converge to the same value since there are enough nonzero elements in each row such that one may be chosen per input port without conflict at the outputs.

For larger switches (>32), lack of simulation results or those of the approximate analysis would limit the observations to the upper and lower bounds, so figures 3.12-3.15 show the theoretical bounds of the throughput for n=64,128,256, and 512 respectively. Examination of these results and particularly the case of n=512 affirms that the bounds get tighter as the switch size gets larger. These results also confirm the limiting behavior of the switch as found for both bounds. It was shown that the throughput of the switch is 1 for $\rho > 0$ and 0 for $\rho = 0$ as $n \rightarrow \infty$, and a similar trend is observed by the above figures to assert these values. As mentioned in the above, the practical switch size is expected to be large, and these results show that the throughput per input port of these switches will be one for any loading condition. Therefore the simple lower and upper bound results have quite a significance for Broadband applications.

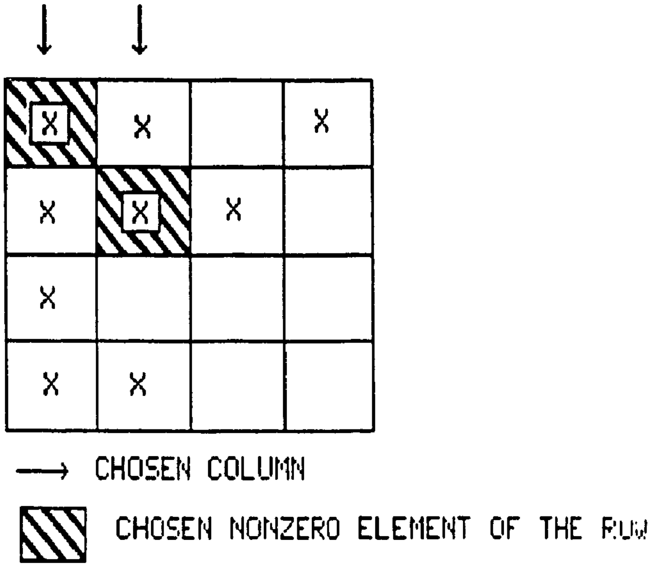


Figure 3.6 Example showing a specific case where the lower bound may fail to pick the maximum number of elements (m=9,n=4).

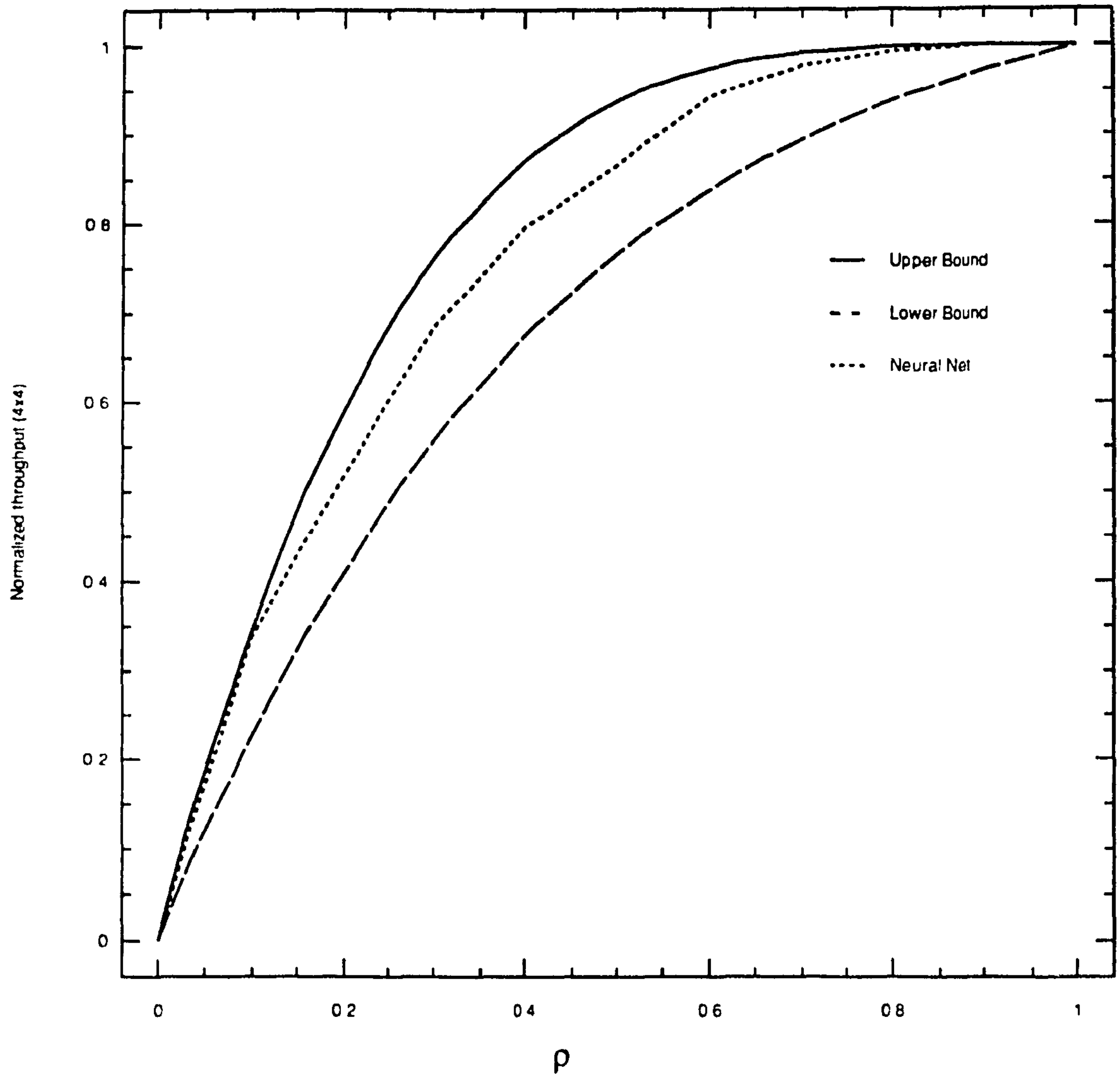


Figure 3.7 Theoretical bounds and Neural Network simulation results of the throughput as function of the probability that an input queue for an output is busy,ρ, (n=4).

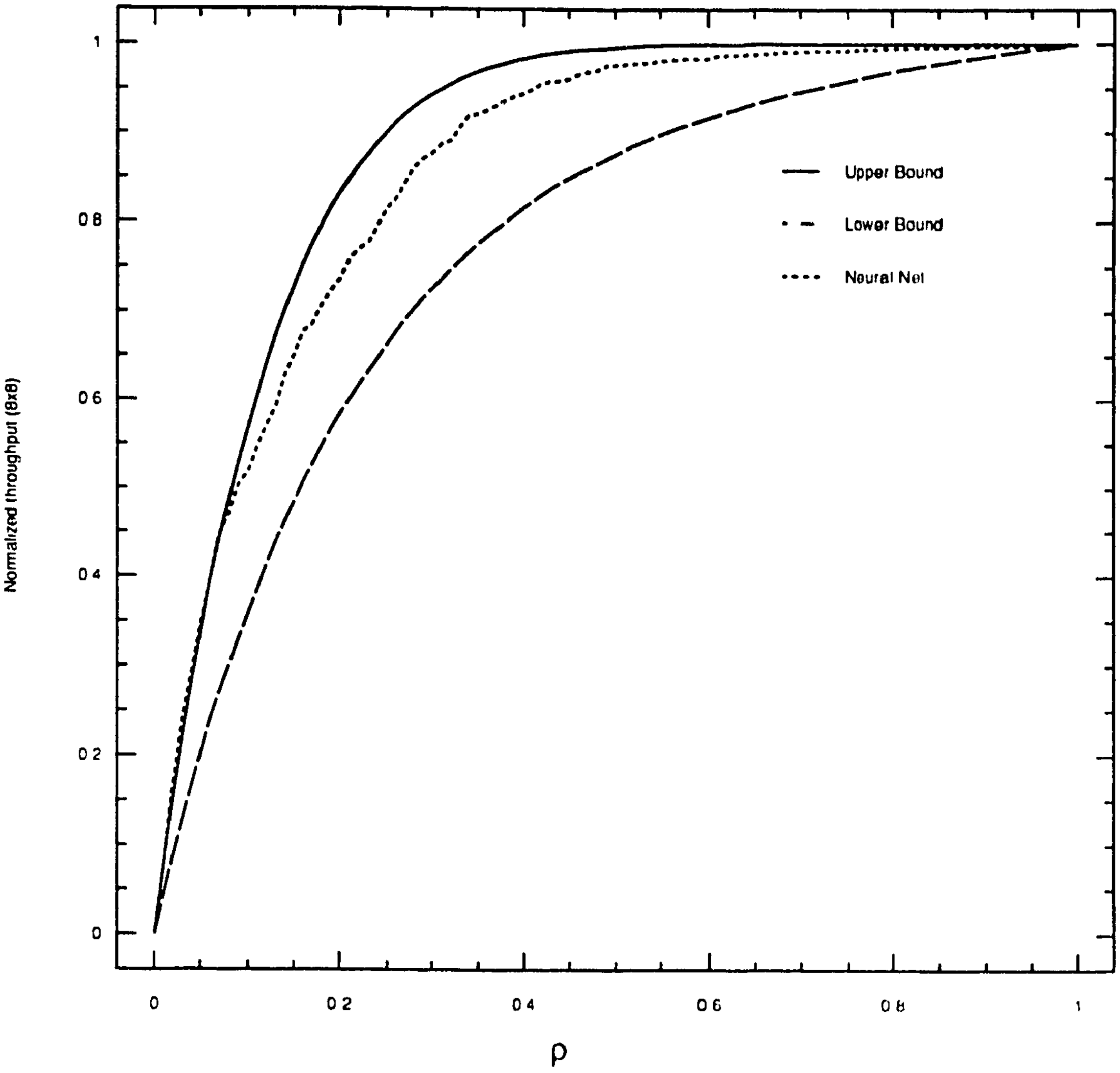


Figure 3.8 Theoretical bounds and Neural Network simulation results of the throughput as function of the probability that an input queue for an output is busy,ρ, (n=8).

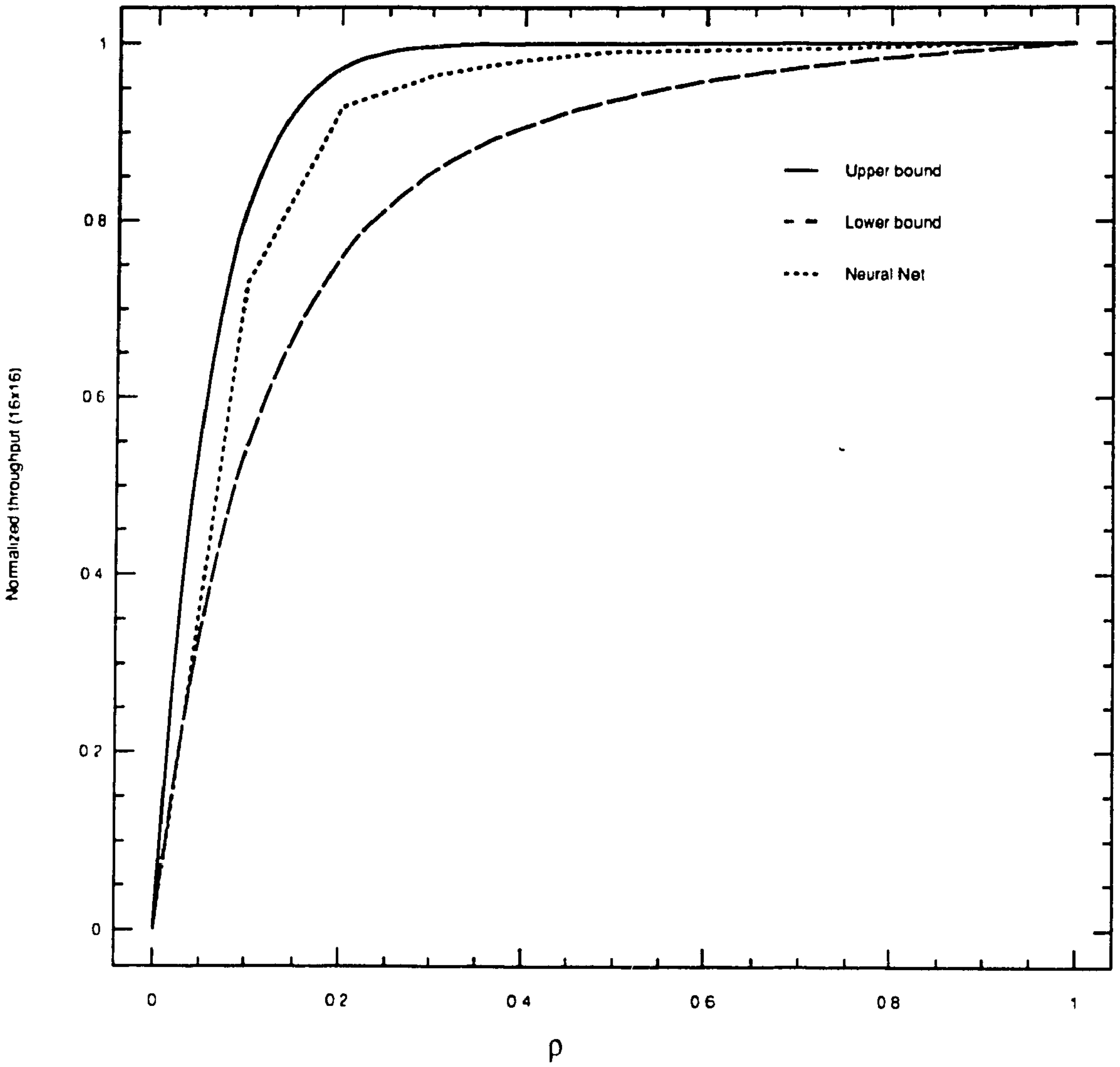


Figure 3.9 Theoretical bounds and Neural Network simulation results of the throughput as function of the probability that an input queue for an output is busy,ρ, (n=16).

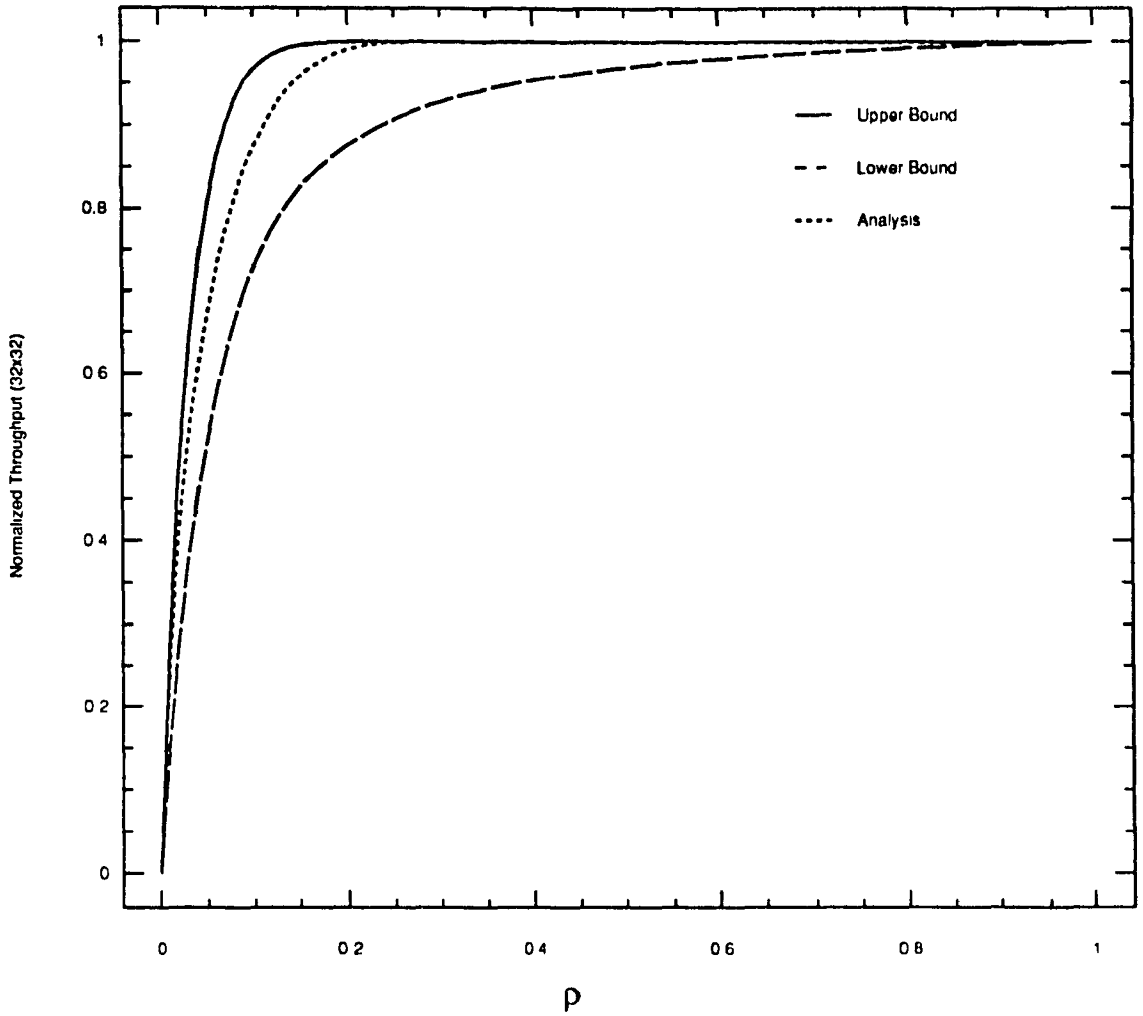


Figure 3.10 Theoretical bounds and the approximate analysis results of the throughput as function of the probability that an input queue for an output is busy,ρ, (n=32).

↓		↓	
X		X	
X		X	
X		X	
X		X	

Figure 3.11 Example of a case where the upper bound fails to pick appropriate number of nonzero elements (n=4,m=8).

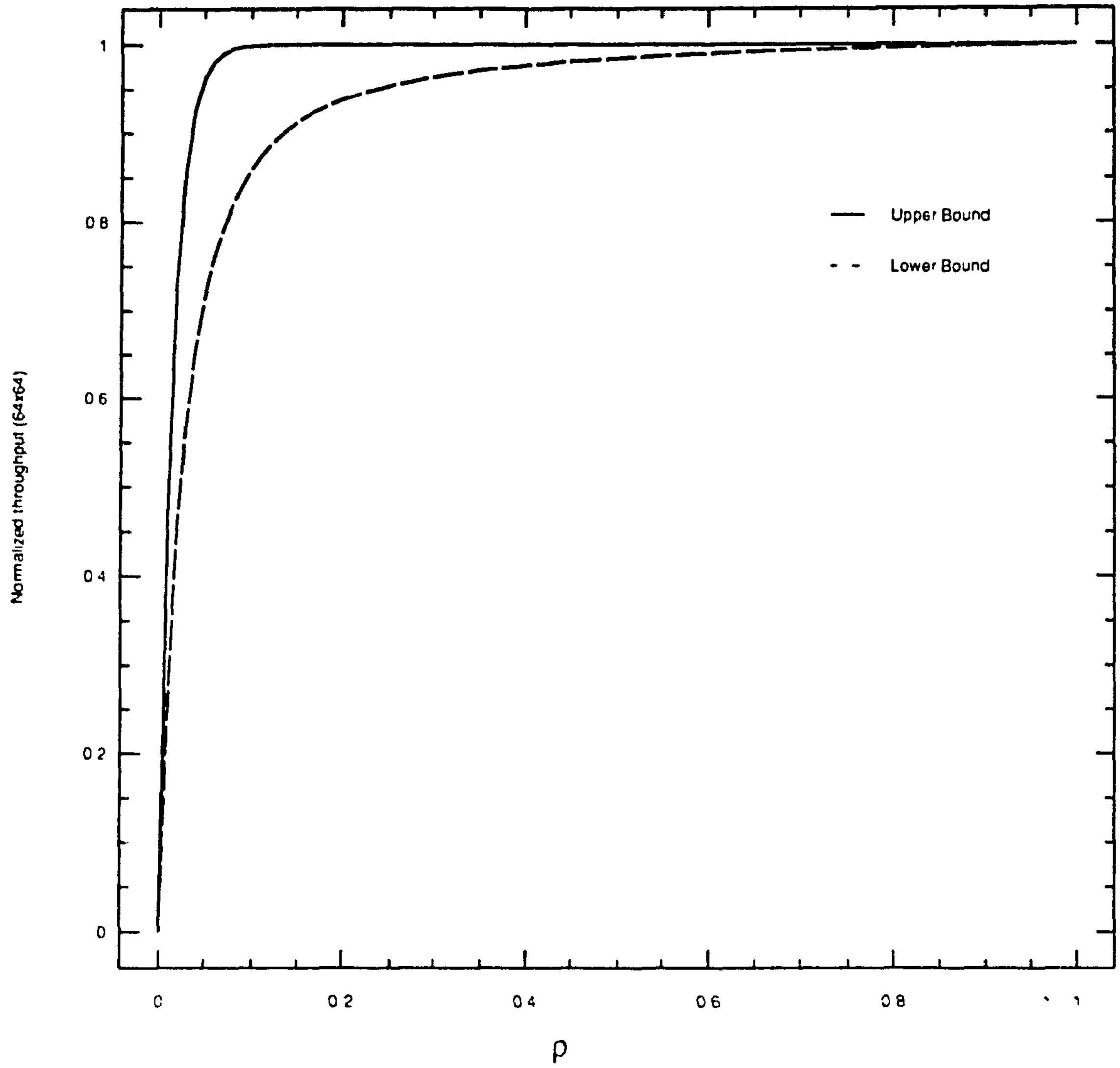


Figure 3.12 Theoretical bounds of the throughput as a function of the probability an input queue for an output is busy, ρ, (n=64).

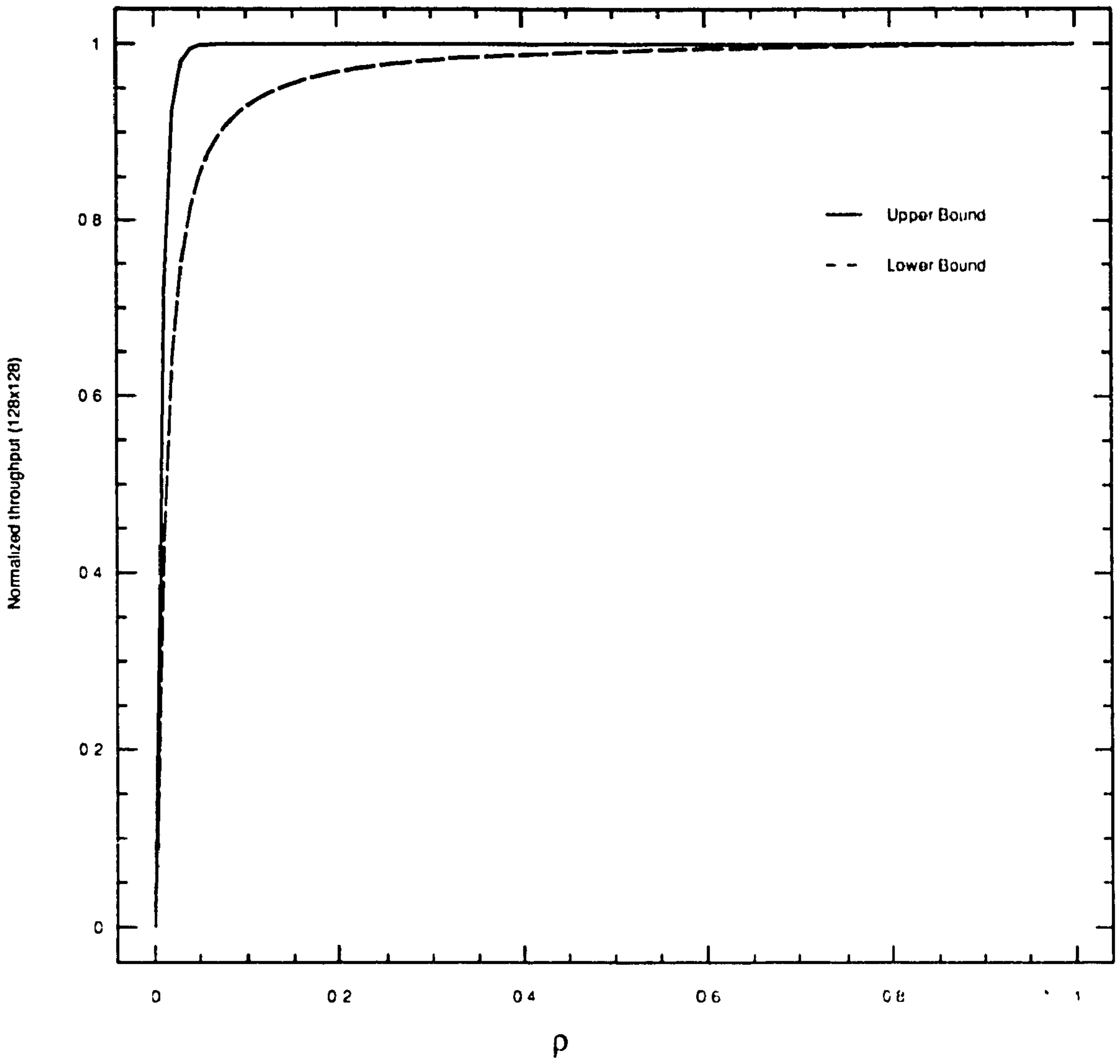


Figure 3.13 Theoretical bound of the throughput as a function of the probability an input queue for an output is busy, ρ, (n=128).

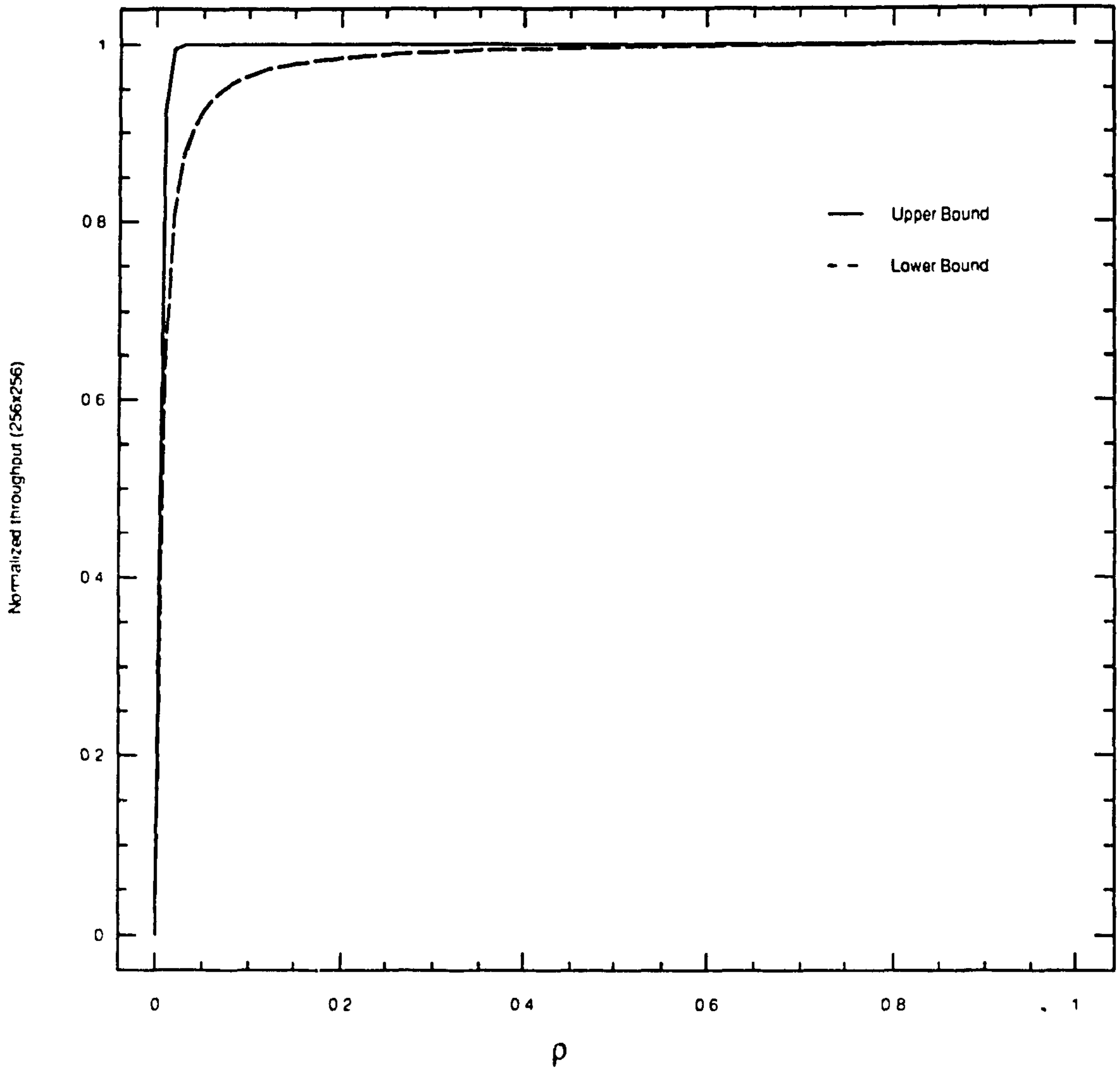


Figure 3.14 Theoretical bound of the throughput as a function of the probability an input queue for an output is busy, ρ, (n=256).

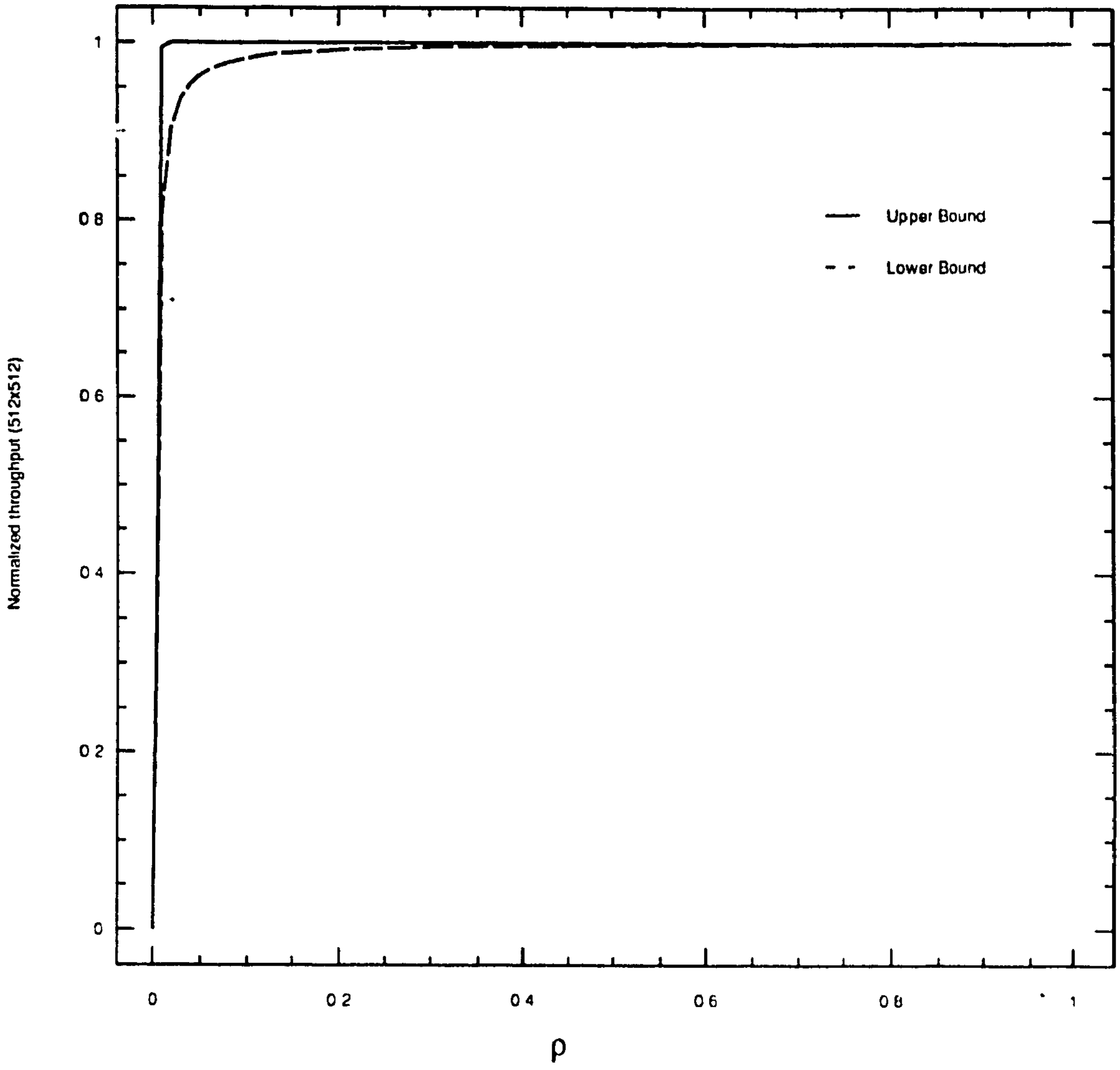


Figure 3.15 Theoretical bound of the throughput as a function of the probability an input queue for an output is busy, ρ, (n=512).

3.3 Throughput Analysis of the Input Queueing Architecture

As a measure of performance comparison, we analyze the input queueing architecture where a single queue is maintained at each of the Input ports of the switch. All traffic destined for any of the outputs arrive at each input port and are then transmitted through the switch if the other HOL packets at the other inputs do not conflict with the output port destination. When a conflict occurs, only one of the HOL packets is transmitted and the rest of the conflicting packets must wait until the next slot, to be considered again.

The inherent drawback of this queueing architecture is HOL blocking, because when a conflicting HOL packet is waiting to gain access to an output, other packets behind it in the queue may be blocked from reaching possible idle outputs on the switch which results in a lower maximum throughput.

3.3.1 Switch Performance

In this section we determine the performance of an input queueing architecture and compare its throughput results to those obtained earlier in our analysis. According to [1], we assume the following

N_j	Number of HOL requests for the same output j
Λ	Probability of packet arrival at each port in a slot
M	Number of input ports not blocked during present slot
T	Throughput per port
A_j	Number of fresh HOL arrivals for outputs j

At steady state, the probability of having a busy queue at any one of the input ports is equal to $\Lambda\overline{m}$, where $\overline{m}$ is the average packet service time. According to the throughput results of the previous section, the throughput per input port is one for large switches (>128) which makes $\overline{m}$=1. Now at each of the n queues per input port, the steady state probability of having a busy queue, ρ, will be

$$\rho = \lambda\overline{m_0}$$

where $\overline{m_o}$ is the average packet service time, and λ is the Poisson arrival rate in each of these queues. At steady state, $\overline{m_o} = n$, which results in $\rho = n\lambda$. Since Λ is the combined arrival rate to each of the input ports, it will be $\Lambda = n\lambda$.

This means that the probability of having a busy queue at the input port or at each of the n queues per input port, is equal to ρ.

Now the expected number of HOL requests for an output j, $E\{N_j\}$ is

$$E\{N_j\} = E\{A_j\} + \frac{E\{A_j(A_j - 1)\}}{2(1 - E\{A_j\})} \tag{3.33}$$

At steady state the above will reduce to (derived in [1])

$$E\{N_j\} = \Lambda + \frac{\Lambda^2}{2(1 - \Lambda)} \tag{3.34}$$

On the average, the number of blocked HOL packets N_b will degrade the throughput by $\frac{E\{N_b\}}{n}$

which is equivalent to

$$\frac{E\{N_b\}}{n} = 1 - \frac{E\{M\}}{n} = 1 - \frac{\Lambda}{\rho} \tag{3.35}$$

So [1] further defines the throughput per Input port, T, to be

$$T = E\{N_j\} - \frac{E\{N_b\}}{n} \tag{3.36}$$

By substitution of terms of (3.35) and (3.34) into (3.36), T becomes

$$T = \Lambda + \frac{\Lambda^2}{(1-\Lambda)} - \left(1 - \frac{\Lambda}{\rho}\right) \tag{3.37}$$

also at steady state we have $T = \Lambda$ which results in

$$\Lambda = \Lambda + \frac{\Lambda^2}{(1-\Lambda)} - \left(1 - \frac{\Lambda}{\rho}\right) \tag{3.38}$$

or

$$(2-\rho)\Lambda^2 - 2(1+\rho)\Lambda + 2\rho = 0 \tag{3.39}$$

So the above equation measures the degree of saturation of the queues with respect to the arrival rate Λ. It is interesting to note that the limiting throughput when the queues are saturated at $\sigma = 1$ is

$$\Lambda^2 + 4\Lambda + 2 = 0 \tag{3.40}$$

Solving for Λ we get $\Lambda = 2 - \sqrt{2} = 0.5858$.

This shows that the blocked packets at the HOL will limit the throughput of this queueing architecture to 0.5858.

Next we attempt to compare these results to the performance of the approximate analysis found earlier in section 3.1. We recall that at steady state, the throughput per port is $T = \Lambda$ assuming an infinite queues, in which case we can find T as a function of ρ. Now we may compare the throughput results of the expression in (3.39) for input queueing to those formerly found in the original analysis. For n=32 and 64, the throughput results are shown in Figure 3.16. It is apparent that the results of the proposed architecture is much superior to input queueing since HOL blocking is avoided by the optimization in chosing the appropriate packets(out of n^2) such that a maximum number of packets is transmitted per slot.

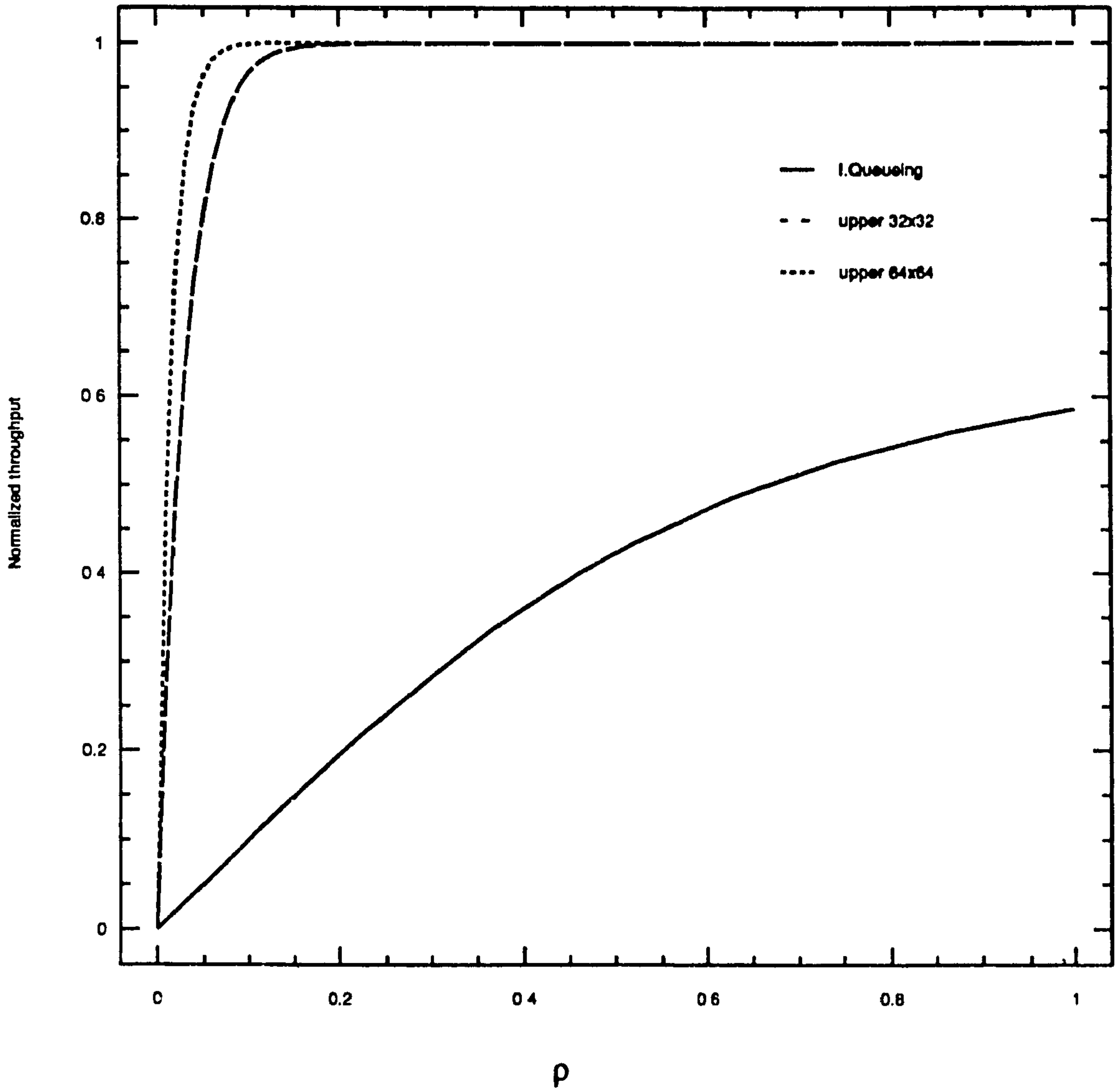


Figure 3.16 Comparison of the results of the input queueing and the analysis for n=32,64.

CHAPTER 4

DELAY AND BUFFER ANALYSIS OF THE INPUT ACCESS SCHEME

A constraint imposed on any packet switch is its delay performance under different loading conditions. All incoming traffic to the switch must be delivered to the appropriate outputs with a minimum delay. The switch delay consists of both the waiting time and the message transmission time.

In this chapter, we shall first present the delay analysis of the Input Access Scheme, and then compare it to that of Output Queueing for its superior performance characteristics.

For the case of Input Queueing, HOL blocking reduces the maximum throughput to 0.5858. This is because of the growing number of blocked packets at the input queue, thereby causing the queue size to become infinitely long for loads approaching the above value.

Output queueing on the other hand allows all the buffering to be performed at the output ports of the switch and HOL blocking is eliminated which results in an optimum delay/throughput results. So it is our intention in this chapter to compare and analyze the delay performance of the input access scheme described in the previous chapter to that of output queueing because of its superior performance compared to input queueing.

It is also equally important to apply these delay results for different broadband traffic services such as voice and video and observe how these services are effected by the delay, as will be shown in the last section of this chapter.

4.1 Delay Analysis of the Input Access Scheme

The study of throughput in the previous chapter has shown that picking the maximum number of packets in the input access scheme results in a normalized throughput of 1 when there are enough packets to choose from. As the switch size n gets larger (n>64), it was shown in chapter 3 that this throughput is one for $\rho > 0$ and 0 for $\rho = 0$. This implies that one packet is chosen in every slot per input port from the possible n queues. Further, we may represent the queueing model of this scheme as the merging of n separate queues into a single one and placing a server at each of the input ports of the switch as shown in Figure 4.1. Thus each of the input ports may be considered as having its own dedicated server.

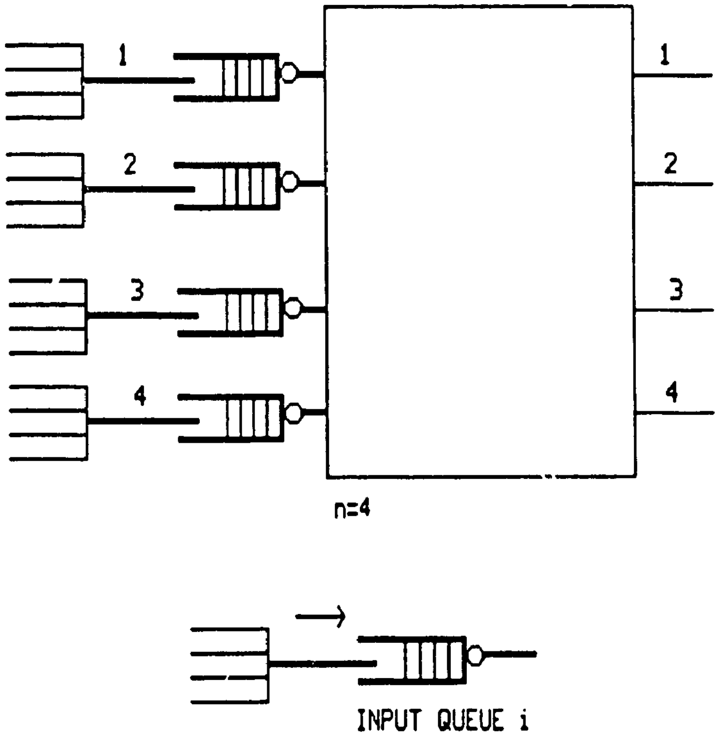


Figure 4.1 Queueing model of the input access scheme.

Also, the proposed Input Access Scheme does not serve all of the packets from the merging queues on a strictly First Come First Served (FCFS) basis, but this does not matter since the queue size and the mean delay are independent of the service discipline. Thus the mean delay can be determined using the M/D/1 queueing model.

In the following we will make the assumption that the processing time of the packet selection using Neural Networks is negligible compared to the overall delay experienced by a given packet.

The arrival to each of the n queues leading to an input port is assumed to be Poisson at λ_i packets/sec, and therefore its sum will also be Poisson with,

$$\Lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_i + \ldots + \lambda_n \tag{4.1}$$

We also assume that the arrival rates to these queues are identical for symmetric traffic such that $\lambda = \lambda_i$, then the total arrival to the merged queue is

$$\Lambda = n\lambda \tag{4.2}$$

Because of the constant packet transmission time, the M/D/1 queueing model may be used to find the expected number of customers in the system assuming infinite buffer size and $n \rightarrow \infty$.

Defining the total switch delay as the waiting time in the queue (w), and the constant packet transmission time (m), we get

$$d = w + m \tag{4.3}$$

Then the expected delay is defined by

$$\overline{d} = \overline{w} + \overline{m} \tag{4.4}$$

For a constant service time, the Laplace transform of the message service time is

$$M(s) = e^{-ms} \tag{4.5}$$

It can be shown that for the above all the moments are given by

$$\overline{m'} = m^i \tag{4.6}$$

The expected waiting time $\overline{w}$, was found for an M/D/1 queue in [19] and is given to be

$$\overline{w} = \frac{\rho\overline{m}}{2(1-\rho)} \tag{4.7}$$

where ρ is the probability that the queue at the input port is busy. The mean delay in number of slots will then be

$$\frac{\overline{d}}{\overline{m}} = \frac{\rho}{2(1-\rho)} + 1 \text{ (slots)} \tag{4.7}$$

$$\frac{\overline{d}}{\overline{m}} = \frac{(2-\rho)}{2(1-\rho)} \text{ (slots)} \tag{4.8}$$

At steady state we also have

$$\rho = n\lambda m \quad \text{or from (4.2)} \tag{4.8}$$

$$\rho = \Lambda m \tag{4.9}$$

Since practical switch sizes will be quite large, we can safely assume that the throughput per input port will be one. This way each input may be thought of as having its own dedicated server, serving its n input queues in a random manner. Of course all the packets in each of these n queues are still served on a FCFS basis.

The variance of the delay is dependent on the service discipline. For the Input Access Scheme as explained above, the order of service is not a strict FCFS, hence the variance of the delay will be higher than the variance of FCFS but lower than that of random order of service. Therefore, the mean delay variations may be bounded by the results of FCFS and random order of service.

So, next we proceed to find the delay variations of the above by denoting σ_d^2, σ_w^2 and σ_m^2 as the variance of the delay, waiting time and of the packet transmission time respectively. The variance of delay may be related to the variance of the waiting time and that of the service time by

$$\sigma_d^2 = \sigma_w^2 + \sigma_m^2 \tag{4.10}$$

Clearly for a constant message service time, the variance would be

$$\sigma_m^2 = \overline{m^2} - \overline{m}^2 = 0 \tag{4.11}$$

this means that

$$\sigma_d^2 = \sigma_w^2 \tag{4.12}$$

Also the variance of the waiting time is defined by

$$\sigma_w^2 = \overline{w^2} - \overline{w}^2 \tag{4.13}$$

For both cases of random order of service and FCFS, $\overline{w}$ is the same and the second moments as defined in [20] are given as follows

$$\overline{w^2}_{(FCFS)} = \frac{\rho}{3(1-\rho)} + \frac{\rho^2}{2(1-\rho)^2} \tag{4.14}$$

$$\overline{w^2}_{(Random)} = \frac{2}{3}\left(\frac{1}{1-\rho}\right)\left(\frac{\rho}{2-\rho}\right) + \frac{1}{(1-\rho)^2}\left(\frac{\rho^2}{2-\rho}\right) \tag{4.15}$$

Also the expected waiting time, $\overline{w}$ was shown in (4.7) to be

$$\overline{w} = \frac{\rho}{2(1-\rho)} \tag{4.16}$$

In the first case, the variance of the delay will be

$$\sigma_{d(FCFS)}^2 = \frac{\rho}{3(1-\rho)} + \frac{\rho^2}{2(1-\rho)^2} - \frac{\rho^2}{4(1-\rho)^2} \tag{4.17}$$

$$= \frac{\rho(4-\rho)}{12(1-\rho)^2} \tag{4.18}$$

Similarly for the random order of service, the variance of the delay will be,

$$\sigma^2_{d(Random)} = \frac{2}{3}\left(\frac{1}{1-\rho}\right)\left(\frac{\rho}{2-\rho}\right)+\frac{1}{(1-\rho)^2}\left(\frac{\rho^2}{2-\rho}\right)-\frac{\rho^2}{4(1-\rho)^2} \tag{4.19}$$

$$=\frac{\rho(2+\rho)}{3(1-\rho)^2(2-\rho)}-\frac{\rho^2}{4(1-\rho)^2} \tag{4.20}$$

$$=\frac{\rho(3\rho^2-2\rho+8)}{12(1-\rho)^2(2-\rho)} \tag{4.21}$$

4.1.1 Delay Results

Having found the mean and the variance of the delay for both orders of service, we present the results as a function of the probability that an input port is busy, namely ρ, in Figures 4.2 and 4.3 where the mean packet delay and the standard deviation are shown for both disciplines.

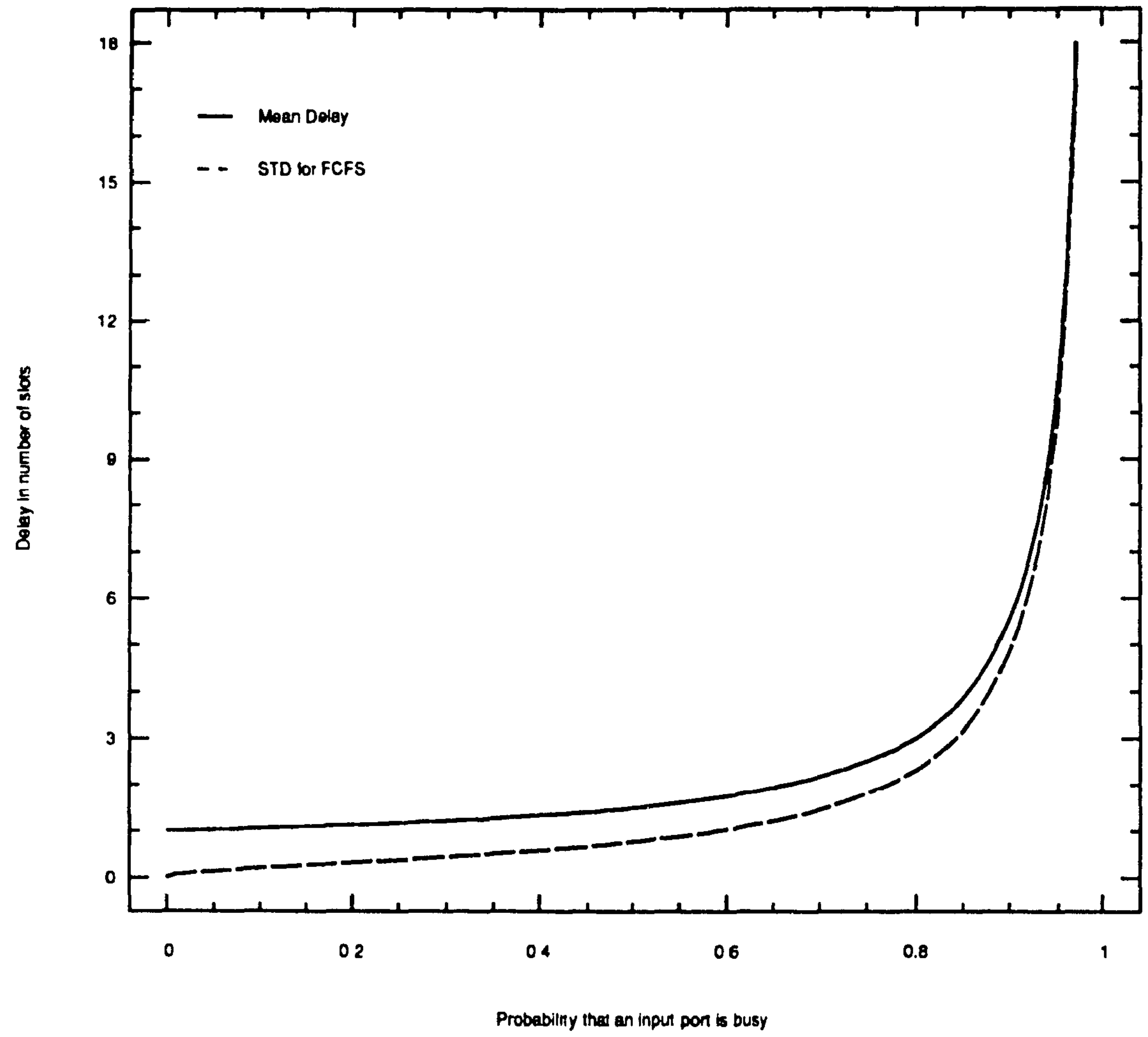


Figure 4.2 Mean delay and standard deviation for the FCFS discipline.

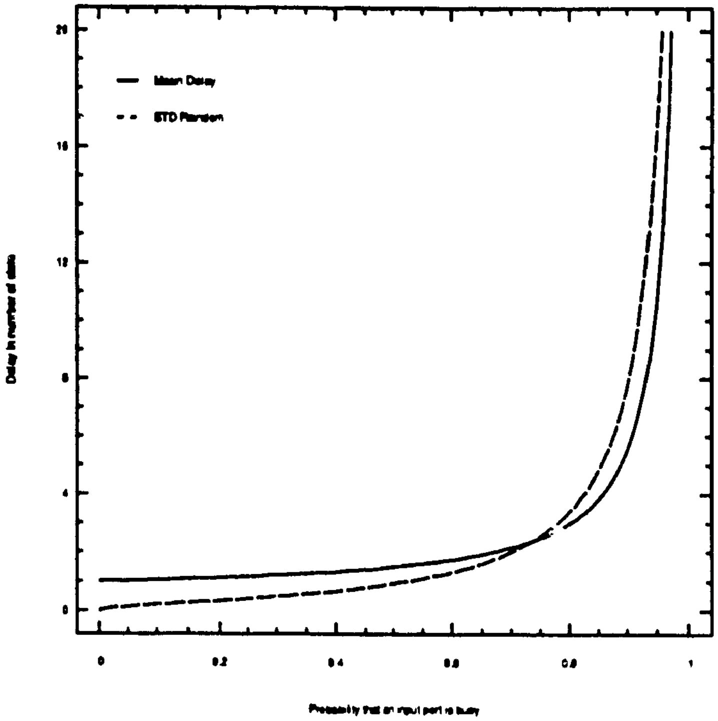


Figure 4.3 Mean delay standard deviation for the random order of service.

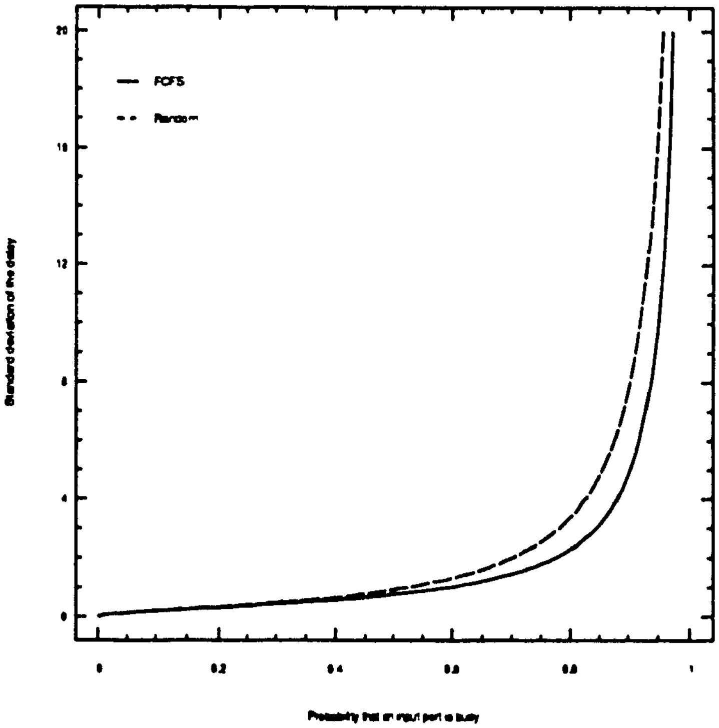


Figure 4.4 Theoretical bounds of the standard deviation of the delay.

Figure 4.4 shows the standard deviation of the two servicing disciplines, and confirms(as expected) that the delay variations are greater for the random servicing. These variations in the delay may be critical for time constrained traffic in the switch, so some measures must be taken as a consequence to control the variable delay, as will be shown in the traffic analysis of the switch.

We present a worst case expected delay as the sum of the mean delay plus its standard deviation for the random order of service as illustrated in Figure 4.5. For the Input Access Scheme, we expect the worst case delay to fall within the shown bounds.

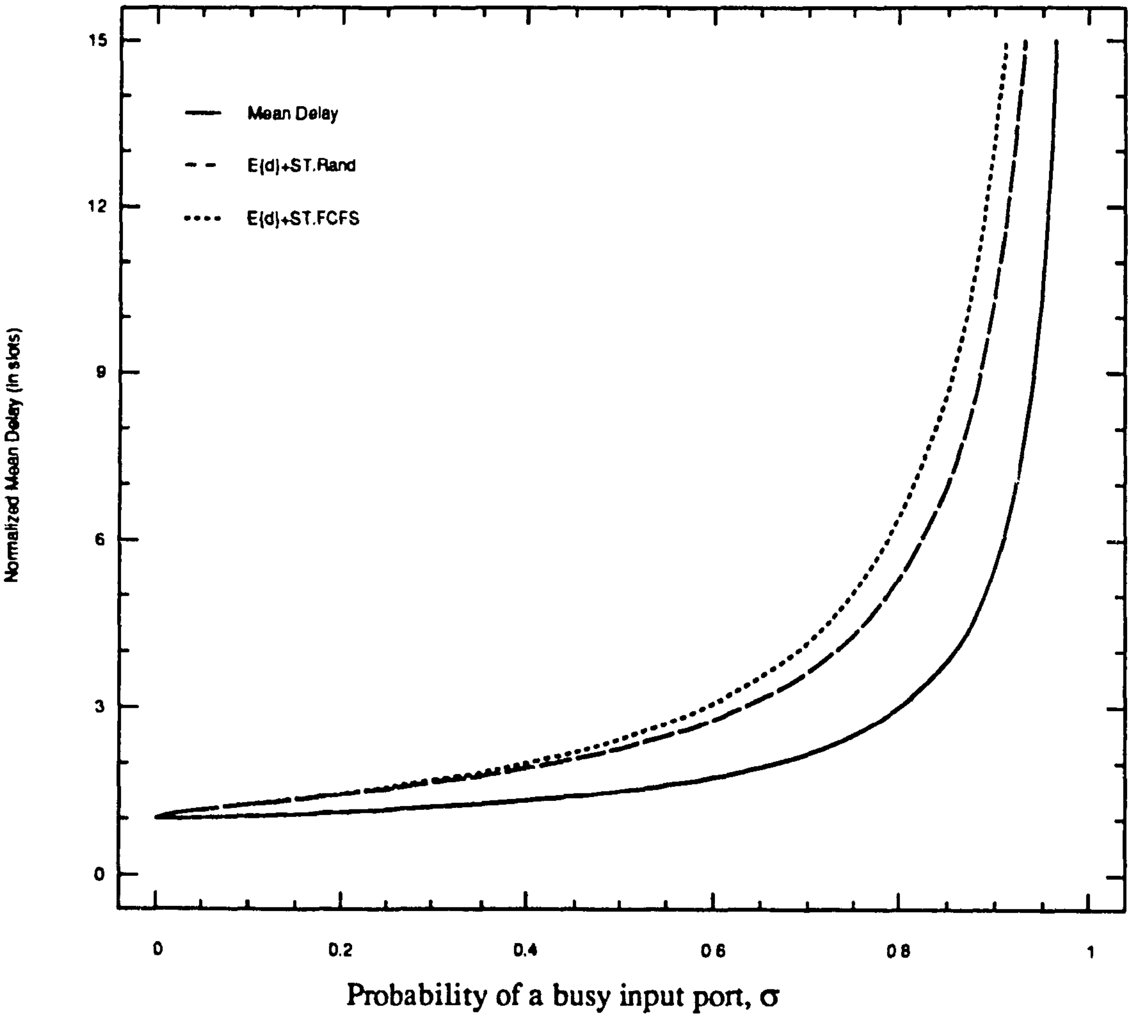


Figure 4.5 Theoretical bounds of the worst case expected delay, and the mean delay.

4.2 Buffer Size

The delay results of the previous section were based on the assumption that the buffer size is infinite at each of the input ports, whereas in reality the buffer size is finite. The buffer size may be approximated by the average number of packets in the system defined by the Pollacezek-Khinchin (P-K) formula [21].

Assuming that $Q(z)$ is the z-transform of distribution of the number of customers in the system and $M(s)$ is the Laplace transform of the message service time, then using the P-K formula we have

$$Q(z) = \frac{M(n\lambda - n\lambda z)(1-\rho)(1-z)}{M(n\lambda - n\lambda z) - z} \tag{4.22}$$

Further we note that for a constant service time, the Laplace transform is

$$M(s) = e^{-ms} \tag{4.23}$$

Replacing the above(4.23) into (4.22) we obtain

$$Q(z) = \frac{e^{-m(n\lambda - n\lambda z)}(1-\rho)(1-z)}{e^{-m(n\lambda - n\lambda z)} - z} \tag{4.24}$$

We had also defined at steady state that $\rho = n\lambda m$ then $Q(z)$ will be

$$Q(z) = \frac{e^{-\rho(1-z)}(1-\rho)(1-z)}{e^{-\rho(1-z)} - z} \tag{4.25}$$

So now we may use the inverse transform of $Q(z)$ to approximate the queue size for a given load. The above transform however may not be easily found, so numerical methods such as the Fast Fourier Transforms (FFT) would be particularly useful to find the distribution function of the buffer length.

Using this distribution, a buffer size may be chosen such that the probability of buffer overflow is acceptable.

4.3 Delay Analysis of Output Queueing

In the following section we study the delay/throughput results of the switch using the output queueing model and compare it those found using the input access scheme. This comparison is due to the superior performance of output queueing as opposed to the other queueing architectures [14].

With output queueing all arriving traffic to the input ports are immediately routed to the proper outputs where they are stored in the buffer to be transmitted (Figure 4.6).

The switch fabric in this case, operates n times faster than the inputs/output lines so that all the packets addressed to the same output are routed to the destination outputs, hence avoiding the HOL blocking of the ordinary input queueing model. Moreover, at each of the output queues packets are transmitted on a FCFS basis.

By placing the buffers at the outputs and operating the switch n times faster, we assure that all the packets arriving to the input ports are sent through the switch which makes the switch utilization(throughput) maximum for any switch size and loading conditions. This compares to the throughput results of the Input Access Scheme for practical switch sizes (n>64), but at the cost of operating the switch n times faster.

For an nxn switch, if the arrivals to the inputs are Poisson(for large n) with rate Λ and that it is equally likely that the messages are destined to any of the outputs with probability $\frac{1}{n}$, then for a tagged output this results in a total arrival rate of $n\left(\frac{\Lambda}{n}\right) = \Lambda$.

Assuming an infinite buffer and large n (n>32), we may approximate the number of customers in the system by an M/D/1 model [14] similar to the one explained in the delay analysis of the input access scheme. For finite values of n (n<32) and buffer length, discrete time Markov chains may be used to find the number of packets in the tagged queue (derivations in [14]).

Denoting σ as the probability that a tagged output queue is busy, then the expected waiting time, $\overline{w}$, using the M/D/1 approximation is

$$\overline{w} = \frac{\sigma\overline{m}}{2(1-\sigma)} \tag{4.26}$$

The message transmission time is the one slot that takes to transmit the chosen packet from the queue, then the total expected delay is

$$\overline{d} \quad = \overline{w} + \overline{m} \tag{4.27}$$

$$= \frac{\sigma\overline{m}}{2(1-\sigma)} + \overline{m} \tag{4.28}$$

$$= \frac{\sigma}{2(1-\sigma)} + 1 \tag{4.29}$$

$$= \frac{(2-\sigma)}{2(1-\sigma)} \quad \text{(slots)} \tag{4.30}$$

where $\sigma = \Lambda m$ which is identical to (4.9).

This results in the same values of expected delay as the Input Access Scheme for (n>64), so for the larger switches, the two schemes would yield to identical delay/throughput results.

But operating the switch fabric n times faster for large n, may not be so practical in terms of implementation. As a result it is suggested in [22] to speedup the switch L times for L=1,...,n and corresponding delays and packet loss probabilities are obained.

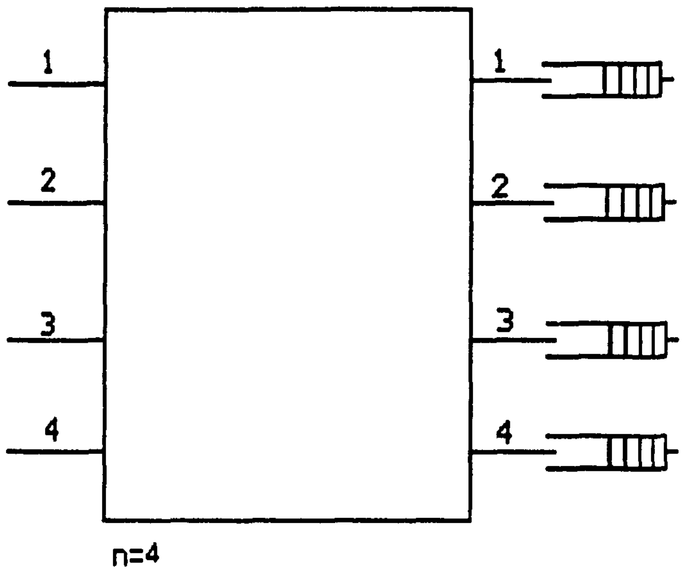


Figure 4.1 Illustration of the output queueing model for a switch size n=4.

4.4 Application of Delay Results to *B-ISDN*

Using the results of the previous section we may now study how different classes of periodic traffic are effected by the switch delay and if critical, the measures that must be taken for those classes of services.

In the delay analysis of the switch, we assumed that the normalized throughput per input port is one for all loading conditions and large n (n>64), hence a single packet is chosen from the set of available packets in every slot.

Also, in the Input Access Scheme under consideration, the packets arriving to an input are not served on a FCFS basis, and as it was discussed in the previous section the mean delay variations are bounded by thoses of FCFS as the lower bound, and random order of service as the upper bound. Based on these variation bounds, the worst expected delay was shown to be the sum of the mean delay and the standard deviation for each case as shown in Figure 4.5.

Next we analyze the effect of delay for the class of symmetric and periodic traffic to the switch. The Broadband services which we consider range from the low bit rate services such as Voice at 64 kbps, Data/Teleconferencing at 1 Mbps, to the higher rates for full motion video at 45 Mbps.

Besides the input traffic we must also consider the maximum port speed of the switch that has a great effect on the delay. For a broadband packet switch, current technology supports a port speed of approximately 150 to 280 Mbps [23,7,1,2]. Furthermore it is anticipated that a port speed of around 600 Mbps may be achievable by the early 1990's, with economic implementation by the mid-nineties [7]. In this example however, we assume a port speed of 140 Mbps.

Now assuming an operating load of 0.6 for the switch, the worst case expected packet delay is given by the mean delay plus its standard deviation for the random order of service, as illustrated in Figure 4.5 which translates into 1.75+1.32=3.08 slots. Now assuming that a packet consists of 48 bytes of information field and 6 bytes of header, and that the port speed is set at 140 Mbps, then the slot duration will be 3.09 μsec.

Now for voice, the sampling rate is at 8000 samples/sec or 125 μsec/sample, we get the associated packetization delay of 6 msec. This means that for voice the packet interarrival time is about 1941 slots which is far less than the switch delay of 3.08 slots. So for voice packets, the switch may practically be operated under any loading condition (<0.99).

For Data or Teleconferencing at 1 Mbps, the packetization delay will be 384 μsec and the packet interarrival time is about 124 slots. Again for an operating load of 0.6, the switch delay of 3.08 slots is much less than the packet interarrival time. Reference to the worst case expected delay results (Figure 4.5) illustrates that the switch delay for Data packets is not critical even for higher loads (<0.98).

Finally accommodating Video at around 45 Mbps we obtain a packetization delay of 8.53 µsec or an interarrival time of 2.76 slots. We observe that the switch delay is quite critical in this case, and as a matter of fact the switch delay at a load of 0.6 which which is 3.08 slots will cause Video packets to bundle up. It is shown in Figure 4.5 that the maximum operating load for Video traffic is about 0.55.

However as the load increases, this delay will be large enough to cause on overflow in the buffers. Therefore we must provide a mechanism to control the variable delay to preserve the periodicity of the traffic. One way would be to assign a priority for the time critical traffic to reduce the variable delay and the neural network would easily lend it self to this. The priority may be implemented by setting the corresponding row of the input matrix to zero except for the element corresponding to the output of the packet to be given priority and the neural network will do its best to serve this packet during the next slot.

In the event of high priority packets contending with each other from all inputs, there would be a single packet in every row of the input matrix and the throughput would decrease to that of input queueing with HOL blocking.

Transmission rate	Packet Interarrival time (slots)	σ_{max} for $\overline{d} + \sigma_{Random}$
64 Kbps	1941	0.99
1 Mbps	124	0.98
45 Mbps	2.76	0.55

Table 4.1 Traffic Analysis results for the Input Access Scheme

(Slot duration = 3.09 μsec., $\overline{d} + \sigma_{Random} |_{\sigma=0.6}$= 3.08 slots)

CHAPTER 5

CONCLUSION

The throughput/delay performance of a High-Speed packet switch is dependent on the Queueing architecture being considered. A number of such architectures have been proposed based on Input Queueing for its relative simplicity, and it has been shown that the throughput performance of the classical Input Queueing architecture is limited to 0.586 because of Head Of Line blocking.

The proposed Input Access Scheme presented in this thesis is a variation of the Input Queueing architecture in which the HOL blocking is avoided altogether by dedicating a single queue for each of the outputs at any input port of the switch. It was shown in the upper and lower bound analysis of chapter 3 that with this scheme, the throughput performance of large switches, that are of interest for B-ISDN, is in fact Ideal and equals the optimum performance characteristics of Output Queueing without the requirement of speeding up the switch n times faster.

Moreover, the delay analysis of chapter 4 has demonstrated that the mean packet delay in the switch is minimal and identical to the case of Output Queueing. Also variations of the mean packet delay are bounded and application of the worst case expected delay to B-ISDN has shown desirable results for Voice and Data, although for Video a priority scheme was suggested for higher operating loads.

The fact that the Input Access Scheme may be implemented using Neural Networks as indicated in [16] makes it a viable solution for High-Speed packet switching.

References

[1] J. Y. Hui, E. Arthurs, "A Broadband Packet Switch for Integrated Transport", IEEE J. Select. Areas In Commun., Vol. SAC-5, Oct. 1987, pp.1264-1273.

[2] H. Suzuki, T. Takeuchi, F. Akashi, T. Yamaguchi, " Very High Speed and High Capacity Packet Switching for Broadband ISDN", IEEE J. Select. Areas In Commun., Vol.6, No.9, Dec. 1988, pp.1556-1563.

[3] P. E. White, "The Broadband ISDN- The Next Generation Telecommunication Network", ICC'86, Vol.I, pp.385-388.

[4] J. Y. Hui, "Network, Transport, and Switching Integration for Broadband Communications", IEEE Network, Mar. 1989, pp.41-45.

[5] P. O'Reilly, "The Case for Circuit Switching in future wide Bandwidth Networks", in Proc. ICC 88, Philadelphia, Jun. 1988, pp.899-904.

[6] J. S. Turner, L. F. Wyatt, "A Packet Switch Network Architecture for Integrated Services", Globecom'83, Vol.I, pp.45-50.

[7] H. Ahmadi, W. E. Denzel, C. A. Murphy, E. Port, "A High Performance Switch Fabric for Integrated Circuit and Packet Switching", Infocom 88, New Orleans, Mar. 1988, pp.1A.2.1.-1A.2.10.

[8] M. De Prycker, " Definition of Network Options for the Belgian ATM Broadband Experiment", IEEE J. Select. Areas In Commun., Vol.6, No.9, Dec. 1988, p.1538.

[9] C. Wu, T. Feng, "On a Class of Multistage Interconnection Networks", IEEE Trans. on Computers, Vol.C-29, No.8, Aug. 1980, pp.694-700.

[10] P. Newman, "A Fast Packet Switch for the Integrated Services Backbone Network", IEEE J. Select. Areas In Commun., Vol.6, No.9, Dec 1988, pp.1468-1476.

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

[12] D. M. Dias, J. R. Jump, "Packet Switching Interconnection Networks for Modular Systems", Computer, Dec. 1981, pp.43-52.

[13] R. J. McMillen, "A Survey of Interconnection Networks", Globecom'84, Atlanta, Vol.I, Nov. 1984, p.107.

[14] M. Hluchyj, M. J. Karol, "Queueing in High-Performance Packet Switching", IEEE J. Select. Areas In Commun., Vol.6, No.9, Dec. 1988, pp.1590-1591.

[15] Y. Yeh, M. G. Hluchyj, A. S. Acampora, "The Knockout Switch: A simple, Modular Architecture for High-Performance Packet Switching", IEEE J. Select. Areas In Commun., Vol.SAC-5, No.8, Oct. 1987, pp.1274-1282.

[16] H. Tri Nguyen, "A Neural Network Implementation of an Input Access Scheme in a High Speed Packet Switch", Master's Thesis, To be published.

[17] M. Mehmet Ali, H. Tri Nguyen, "A Neural Network Implementation of an Input Access Scheme in a High Speed Packet Switch", Globecom'89, Vol.2, Nov. 1989, pp.1192-1195.

[18] J. Riordan, "An Introduction to combinatorial Analysis", J. Wiley, 1958, pp.164-168, pp.178-180.

[19] L. Kleinrock, Queueing Systems, Vol.I Theory, 1975, pp.190-191.

[20] J. W. Cohen, the single server queue, North-Holland series, 1982, pp.249-256, pp.439-443.

[21] L. Kleinrock, p.194.

[22] Y. Oie, M. Murata, K. Kubota, H. Miyahara, "Effect of Speedup in a Nonblocking Packet Switch", ICC'89, Vol.1, Jun 1989, pp.410-414.

[23] L. Hayward, A. Gottlieb, S. Jain, D. Mahoney, "CMOS VLSI Applications in Broadband Circuit Switching", IEEE J. Select. Areas In Commun., Vol. SAC-5, No.8, Oct.89, pp.1231-1240.