Performance Analysis of a Multicast Switch

Jing Fang

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

March 1993

ISBN 0-315-84654-2

ABSTRACT

Performance Analysis of a Multicast Switch

Jing Fang

A performance analysis of a multicast switch, with no fanout splitting service discipline, is presented. Among several proposed multicast service disciplines, no fanout splitting service keeps the integrity of all the copies in a selected packet by ensuring that all packet's copies will be transmitted in the same time slot. Moreover, when the packet fanout is not too high compared to the switch size, the performance achieved under this service discipline compares very well with that of the other disciplines. In addition it has a simplicity in the hardware implementation. First in this thesis, the distribution of the number of packets being successfully transmitted is derived under the assumption of a fixed number of packets contending during a particular slot and a random packet selection policy. Then this result is extended to the steady-state case through a Markov Chain analysis. It is shown that transmitted packets have a smaller size than the average size of the contending packets. Thus the blocked packets(left over from the previous slots) have a larger size. By taking into account this blocking effect through an equilibrium observation, both packet and copy throughput are determined, and the final results are expressed in terms of the incoming traffic only. From these, the best operation point of the system is found. As a result of copy generation, the effective load which is offered to the switch will be much higher than the initial input load, making the switch subject to a severe bottleneck. As a solution, speedup of the switch is studied in this thesis to increase

the throughput. The analysis is carried out based on a variation of the random packet selection policy. This enables us to obtain a closed form solution to the distribution of number of packets chosen during a slot, by using a discrete time birth process modeling of the packet selection. The effect of speedup on the output contention is determined. Finally we have run some simulation tasks to confirm the study of our theoretical analysis and the results obtained showed very good agreement between both.

ACKNOWLEDGEMENTS

I would like to express sincere appreciation and gratitude to my thesis supervisor, Dr. M. Mehmet-Ali, for his constant guidance and support during the entire preparation of the thesis and for his many precious suggestions and encouragement in the work.

Also I would like to express my deep gratitude to my family for the opportunity they provided me to further my study abroad and for their constant encouragement throughout my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF MAIN SYMBOLS

n	The Number of input(output) ports of a multicast switch.
α	The constant number of contending packets at the beginning of a slot.
PGF	Probability Generation Function.
RPS	Random Packet Selection policy.
$R\#$	Noninterfering packets set after #th choice under RPS policy.
p	The copy generation probability of a mixed packet(a packet from mixed fresh + old traffic).
p_f	The copy generation probability of a fresh packet.
p_o	The copy generation probability of an old(blocked) packet.
$\hat{P}_j$	The probability that a contending packet does not interfere with the jth chosen packet under RPS policy.
m_j	The number of outputs selected by the first j chosen packets.
n_j	The average number of copies of a jth chosen packet in the case without speeding up operation.
$\hat{T}_p$ ($\hat{T}_c$)	The packet (copy) throughput per slot given a constant number of contending input packets.
$P_j(i)$	The probability distribution of j packets being chosen given i contending input packets during a time slot.
W_j	The steady-state probability distribution of j packets being chosen during a time slot in the case of no speeding up operation.
$\check{W}_j$	The steady-state probability distribution of j packets being chosen during a time slot in the case of speeding up operation.
$p_{t\|m}$	The transition probabilities of having t old packets at an embedded point given m old packets at the previous embedded point in Markov Chain analysis.

Π_k	The probability of having k packets at an embedded point in Markov Chain analysis.
μ_f	The average number of copies in a fresh packet.
μ_o	The average number of copies in an old(blocked) packet.
μ_d	The average number of copies in an departing packet.
MRPS	Modified Random Packet Selection policy.
$\breve{P}_j$	The probability that a packet does not interfere with the first chosen j packets under MRPS policy.
$\breve{P}_j(i)$	The probability that j packets will be chosen by the end of the ith trial under MRPS policy.
$\breve{m}_j$	The number of unavailable outputs(or the outputs with number of copies greater than or equal to L) after j successful trials.
$\bar{\bar{n}}_j$	The average number of copies of a jth chosen packet in the case of speeding up operation
η_j	The probability that an output remains available after j successful trials in the case of speeding up operation.
T_p (T_c)	The steady-state packet (copy) throughput per slot.

Chapter 1

Introduction

Currently the study and realization of Broadband Integrated Services Digital Networks (B-ISDN) are receiving a great deal of attention, as these networks are expected to incorporate the most advanced technologies and new applications in telecommunications. There are two characteristics in Broadband ISDN which make them remarkably different from the traditional voice or data networks. The first characteristics is the capability of these networks to support many high bandwidth applications such as digitized video (about 50Mb/s for current standard and 150Mb/s for HDTV) and image communications(from 50Mb/s to 100Mb/s), which is made possible thanks to the high capacity of fiber optics transmission system. The second feature of Broadband ISDN is the presence of a single communication network which supports all services (voice, data, video and images) in an integrated and unified fashion.

None of these objectives could have been realized by the techniques used in the traditional communication networks such as those used in rigid circuit-switched voice networks and low-speed packet switched data networks. In response to the need of integrating the broadband services within a unified framework, the highly flexible and efficient Asynchronous Transfer Mode(ATM), defined and standardized by CCITT, has been recognized as the most appropriate switching technique for the transport layer integration. This packet switching mode is not only well suited for data applications involving bursty traffic but also capable of handling a wide diversity of data rates and latency requirements, resulting from the integration of services. Designing and implementing such a fast packet switch remains however a challenge. The ATM operation mode specifies fixed-size packets with 48 bytes of data and 5 bytes of control information each. Many line speeds are also specified with nominal rates equal to 155.52Mb/s (required for digitized HDTV) and 622.08Mb/s. A fast packet switch is thus required to handle rates of the order of 100,000 to 1000,000 packets per second per input line. The various architectures for such high speed packet switches have been studied and surveyed in [1]. Some of them have been implemented electronically using very large scale integrated circuit technology. Recent research on photonic switching shows a growing trend in using light as a potential means of implementing the switching operation in the near future. Based on these techniques, a wide range of multi-rate, multi-media and multi-point broadband applications will become a reality.

In this thesis, we study the performance of a multicast switch. A multicast switch is one of the building blocks in a broadband ISDN switch. It can support a multi-point communication with diverse applications in various areas such as distributed parallel computations, replicated databases for information processing, resource allocation for multiple servers, LAN bridging, commercial television distribution and voice and video teleconferencing.

1.1 A Multicast Packet Switch

In a high speed network, the task of a multicast switch is to transmit the information generated by one source to several destinations simultaneously. Many studies of multicast switches have been carried out with diverse objectives such as:

1. Investigate new switch architectures and designs.
2. Performance evaluation under different multicast service disciplines.

The switch architecture study and design invokes two functions which are needed for multicasting information: packet replication and switching. The mechanisms used to realize these functions must be both efficient and feasible for the broadband network implementation. Several switch architectures have been proposed[2-5]. The objective of performance analysis of a multicast switch is to determine the appropriate operational mode that will achieve the best performance index such as maximum throughput and minimum packet delay. This will clearly also depend on the multicast service disciplines which is employed.

The remaining of this chapter gives a brief review on multicast switch architectures and the performance of various service disciplines:

1.1.1 Multicast Switch Architecture

A multicast switch is required to provide a multipoint communication capability, namely transmission of information from one input of the switch to a set of specified output destinations. A multicast switch performs this function in two steps: first it replicates the source packets according to specific requests, then the packet copies are switched to the appropriate destinations. In a multicast switch (see Fig. 1.1), a serial combination of a copy network and a point-to-point switch(routing network) is considered to be sufficient for accomplishing this operation.

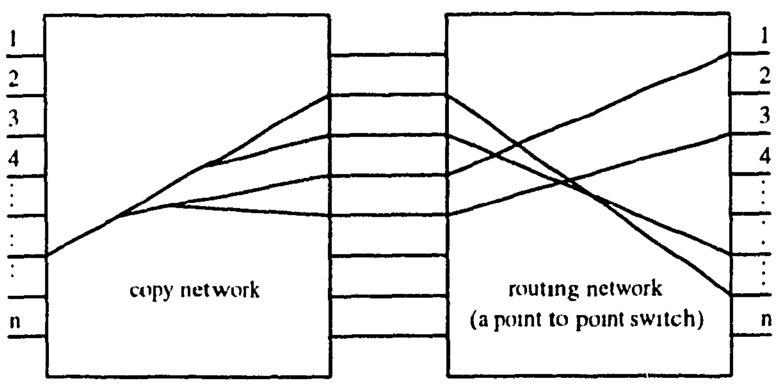


Figure 1.1: An nxn Multicast Packet Switch

The basic building block of both the copy and the routing networks is a Banyan network, which is an nxn multistage interconnection network that has been widely used as a point-to-point switch, for example in a routing network. A Banyan network has $N = \log_2 n$ stages, each stage contains $\frac{n}{2}$ nodes and each node is a 2x2 switch element(Fig. 1.2 illustrates this fact for the case n=8). The Banyan network has a self-routing property in the sense that all packets in the network are routed at each node according to the carried header information. The header of each incoming packet contains an n-bit destination address. A node at stage k transmits the packet either on the up-link or down-link according to whether the k^{th} bit is 0 or 1. This property is due to the topology design property in the Banyan network where the path from any input to any output is uniquely determined by the output address. A Banyan network is an internally blocking network by itself, that is two packets may arrive at the same node and attempt to go to the same output link at the same time. However, it can operate as a nonblocking network provided that input packets with distinct destination addresses are sorted in an ascending or a descending order. This nonblocking property can be realized by combining the Banyan network with a Batcher sorting network, the sorting network presents the compacted and sorted destination addresses of arriving packets at its outputs.

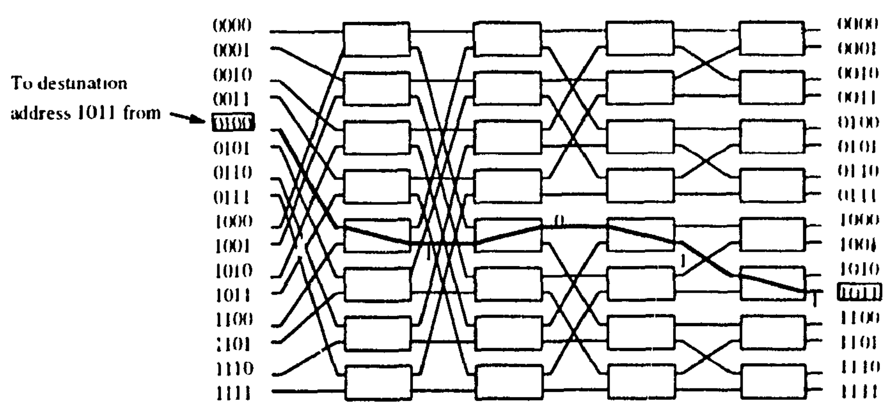


Figure 1.2: A Banyan Network with Self-routing Property

The self-routing and nonblocking properties of a Banyan network have been extended and used to design broadcast networks which are widely used for packet replications. Among the several copy networks which have been proposed, we can cite: *the Starlite copy network*, *the broadcast packet switch copy network* and *the nonblocking self-routing copy network.*

Next we discuss the packet replication algorithm used in each of these copy networks:

The *Starlite copy network*[6] uses a sorting network[7] and a broadcast network for the packet replication. The inputs to the sorting network are original source packets and the empty packets(which will be the copies of the source packet)generated by receivers. Initially the sorting network sorts the input packets according to their source addresses such that the original source packet and the associated empty packets with the same source address will appear contiguously at its outputs. Then, the subsequential broadcast network replicates the data in each source packet and inserts them into the empty data fields of the subsequent empty packets. A source packet is thus replicated into the required number of copies (see Fig. 1.3). The packet replication algorithm here is quite simple, however this scheme is not feasible for implementation in broadband networks due to the variable delay caused by buffering, multiplexing and switching in a multiple-hop connection.

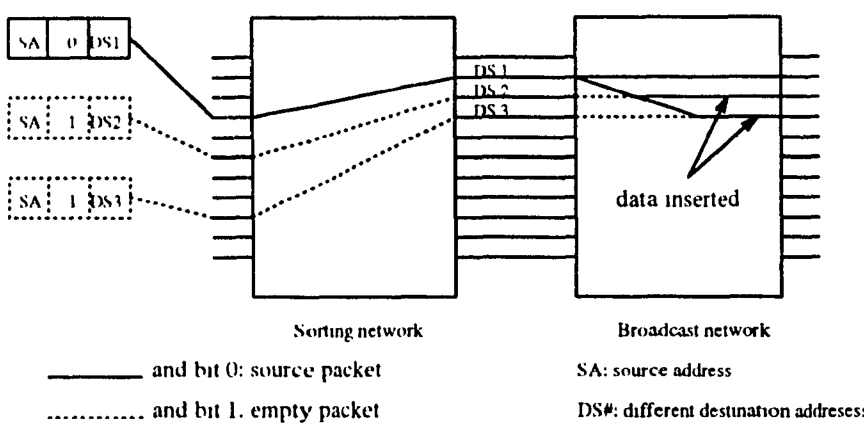


Figure 1.3: A Starlite Copy Network

The *broadband packet switch copy network*[5,9] consists of a Banyan network and a set of broadcast and group translators. In this copy network, the packet header contains two fields, a fan-out indicates the number of copies(CN) requested by a source packet and a virtual address known also as the broadcast channel number(BCN), which is used by the broadcast and group translators to determine the final destination of the copy packets in the routing network. The replication of packet is implemented by splitting the fanout in the Banyan network(see Fig. 1.4). When copies appear at the outputs of the network, the actual addresses of the copies in the routing network are determined through a table lookup method, using the broadcast and group translators, then they are switched to the destinations through the routing network. In this scheme, packets may experience internal blocking caused by the Banyan network and therefore buffers are required for every internal node to prevent packet loss and a random packet delay becomes unavoidable.

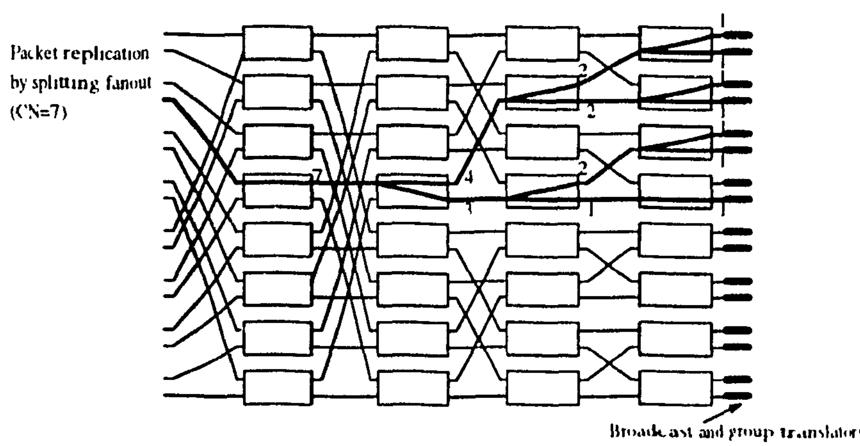


Figure 1.4: A Broadband Packet Switch Copy Network: packet replication by splitting the fanout in Banyan network

To overcome the drawbacks of variable delay or internal blocking encountered in the previously discussed copy networks, a *nonblocking, self-routing copy network* with constant latency has been studied by Lee[3]. This copy network adopts a broadcast Banyan network with switch nodes capable of packet replications and thus a packet arriving at each broadcast switch *node* can be either routed to one of the output links or replicated and sent out on both links. The broadcast Banyan network replicates packets according to a Boolean interval splitting algorithm, by which a packet header only requires an address interval (MIN, MAX) and an index reference(IR) pointing to the minimum address(MIN). The algorithm insures that a copy of the packet is routed to each broadcast Banyan network output address which is contained in this interval(see Fig. 1.5).

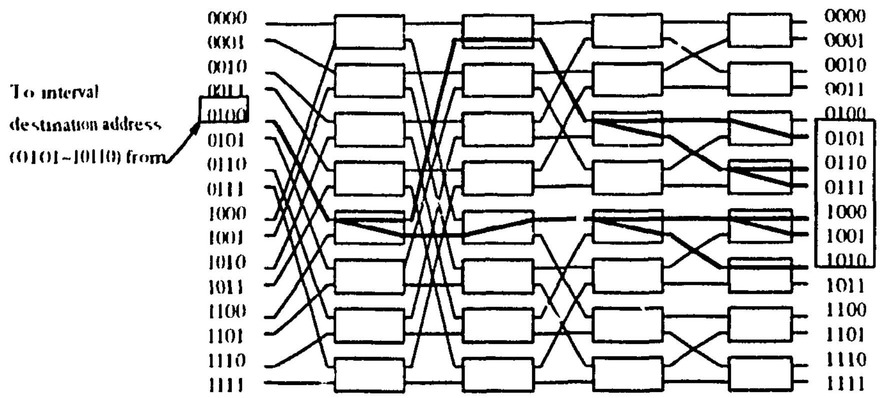


Figure 1.5: A Nonblocking Self-routing Copy Network: packet replication according to a Boolean interval splitting algorithm

In this copy network, because the link-independent nonblocking condition is satisfied by providing the monotonous address intervals of a set of inputs to the Banyan network, the packet replication algorithm therefore ensures contention-free internal paths of the broadcast Banyan switch when replicating copies. Also, in this algorithm, the routing decision at each node requires a low bound of two-bit header information, this reduces the complexity of broadcast switch nodes to the minimum. In addition in this copy network, an encoding process which provides the packet header information for replication is carried out by a running sum network and the dummy address encoders. A running adder network recursively adds up the copy numbers(CN) provided in the packet headers and subsequently the dummy address encoders utilize these running sums to replace the copy number field in each packet header with two new fields, namely a dummy address interval (MIN, MAX) and an index reference. At the outputs of the broadcast network, the actual addresses of all replicated packets are obtained by a trunk number translator. Following the completion of packet replication, switching of the copies based on their actual destination addresses is performed by a succeeding point-to-point nonblocking switching network. A block diagram shown in Fig. 1.6 illustrates the components of this copy network.

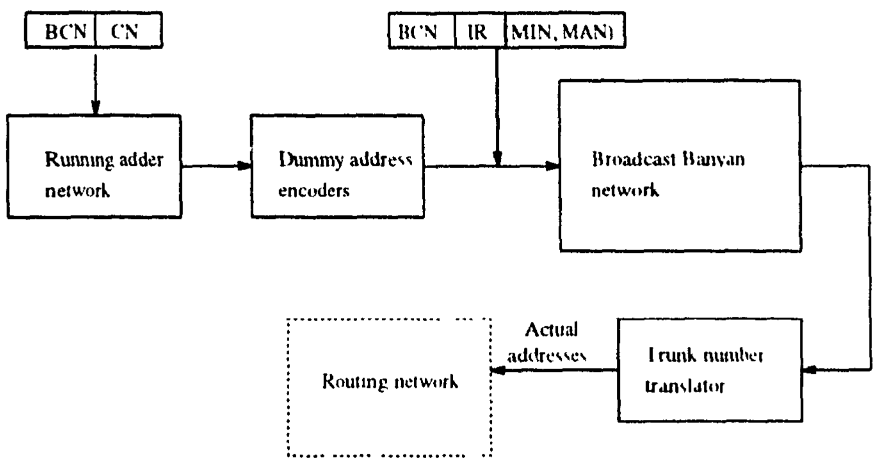


Figure 1.6: Components in a Nonblocking Self-routing Copy Network

For real time consideration, a multicast switch will also include a selector network[4] which will arbitrate among competing demands for copies so as to satisfy real-time requirements, such as a conflict-free operation at the switch's output.

To sum up, a multicast switch basically consists of three subnetworks which are internally nonblocking as well as self-routing: A selector network arbitrates the real-time requirements from incoming traffic, with a copy network replicating packets according to specific requests and a switching network routing the packet copies to their final destination addresses.

1.1.2 Multicast Service Disciplines

Multicast service disciplines determine the way the packets are to be served by the multicast switch, which also includes the way the conflicting copies are to be served. As in a unicast switch, the HOL(Head of Line) blocking occurs in a multicast switch when more than one packet in the head of the input queues are destinated to the same output. However, and in contrast to the unicast case, conflicts among HOL packets are more likely to occur since each packet may generate more than one copy. In the sequel, the term "fanout" will refer to the number of copies generated by a packet. One of the important aspects in studying multicast switches is to examine their performance under different service disciplines in order to achieve the best operational mode. Several queueing disciplines have been proposed and a brief review of them is given below:

a). *Sequential Service*

The sequential service is one of the two extreme cases in the HOL copy selection process, where at most one copy of a multicast packet is allowed to contend per slot. The destinations of a packet are served sequentially, and hence the transmission of a multicast packet takes at least as many slots as the total number of its copies(or fanout). A multicast switch which implements this type of service can be viewed as a unicast switch whose performance has been evaluated in [9,10], giving a maximum throughput of 0.586.

b). *Multicast Service with no Fanout Splitting*

This scheme, known also as the one-shot service[11], constitutes the other extreme case in the HOL copy selection process. Here, it is compulsory that all the destinations of a multicast packet are served in the same slot. A packet fails to be transmitted if one of its copies loses contention during a slot. This scheme keeps the integrity of all the copies by ensuring that all packet's copies will be transmitted in the same slot instead of reassembling them at the outputs as was the case for the sequential service discipline. Besides its simplicity in terms of hardware implementation, this scheme can also achieve a very good performance under low fanout condition.

c). *Multicast service with fanout splitting*

A compromising approach in the HOL copy selection is to split the fanout of a multicast packet. This allows a HOL packet to send as many copies as possible, depending on the availability of outputs. Multicast service with fanout splitting allows an increase in the output utilization(and hence a maximum throughput) and improves the packet delay performance. In addition, two subservices may be generated depending on how those copies, which are destined to the same output, are served:

.*A Random HOL service*: Here, and among the many copies which are destined to an output, only one of them is chosen randomly during a slot. Performance analysis of this service discipline has been done in [12].

.*A FCFS HOL service*: In this case, the copies destined to the same output are served according a *first come first serve*(FCFS). Compared to all the previously discussed disciplines this scheme maintains the best performance in terms of switch saturation throughput, fairness and delay performance [13]. However it requires additional complexity in the control structure of the switch.

d). *Multicast service with and without fanout splitting*

This is the combination of the last two service approaches. It was proposed by Chen[14] and called *revision scheduling*. With this scheme and during any slot, the packets are first selected according to the no fanout splitting discipline, then the remaining packets will contend for the rest of idle outputs according to the fanout splitting discipline. This service discipline maximizes the number of packets being transmitted and also achieves the best output lines utilization compared to other disciplines.

In the study of above mentioned service disciplines, different mathematical models and the related analysis approaches are used. In the notable first work by Hayes et. al.[12], performance analysis of a multicast switch was carried out under the assumptions of a random HOL service discipline(with fanout splitting) and an input port buffering. Where by modeling the system as an independent set of $M/G/1$ queues, they were able to derive the packet service time distribution and therefore the delay and throughput performances of the system were determined. One among the major problems specifically encountered in the multicast switch analysis, is the derivation of the conflicting HOL copy(or called contending traffic) distribution, which in [12] was characterized by means of a residual distribution. In the latter work, Chen[14] carried a performance analysis of multicast switch under different service disciplines. In the one-shot service discipline(which corresponds here to the no fanout splitting) considered in [14], the description of the copy distribution was modified according to the encountered total life distribution, and the copy generation mechanism was assumed to follow a binomial distribution with a modification, namely through a normalization factor $\frac{1}{1-P_r(k=0)}$, which ensures that non-zero number of copies will be generated by a packet. Of the two input access schemes(the random selection scheme and the cyclic priority scheme) mainly used for this service discipline, the cyclic priority scheme[15] allows the HOL packets to make the reservations of output ports according to their priority levels, rather than contend for the outputs randomly. This access scheme assigns N levels of priority to the N input ports at the beginning of each transmission slot, and rotates the assignment cyclically from slot to slot. An input port with priority level 1 can get its HOL packet(or in terms of all its copies) through the switch with probability 1 if its queue is non-empty. Then, the second priority port gains the access for delivering the HOL packet if it has one. However, this packet will get through the switch only if all its copies are contention free with the first priority packet, otherwise it will stay and wait for the next slot. Each input port is thus checked in the order of their priority levels up to N. It has been shown by a large number of simulations that, the delay-throughput performances of a cyclic priority scheme and the random selection scheme have no much difference. In the study of another service discipline referred to as *revision scheduling* in the above, the no fanout splitting and fanout splitting disciplines have been combined together. In which case, the contending packets are first chosen according to the one-shot discipline until all remaining packets are interfered with the chosen packets. Then it allows the individual copies of the remaining packets to contend for the remaining idle output ports. This combination access scheme mitigates the HOL blocking by enabling copies to go through system as many as possible, which insures a full use of output lines. Finally, Chen has proposed a general unified mathematical model for the the performance analysis under these different service disciplines, by using matrix-geometric techniques. The matrix-geometric technique provides a tool for the analysis of an infinite state space Markov Chain and has become an effective tool for evaluating the performance of complex queueing systems.

The implementation of each of the proposed service disciplines has brought up another topic of research. For a given service discipline, the corresponding output contention resolution algorithm provides a means of selecting contention-free packets to go through the switch. So far, several algorithms have been proposed for point to point transmission, such as the re-entry network[6], the three-phase algorithm[9] and the reservation-based contention solution algorithm[16]. However, with some modifications, these algorithms are applicable for the case of multi-point transmission. In [14], an improved output contention resolution algorithm named Cyclic Priority Input Access Scheme, derived from the ring reservation-based method, with the rotated priority assignment was proposed and implemented with a combinational logic circuit. It has been shown that this scheme possesses the features of being simple, fast, compact and also reliable. Another approach for the contention resolution algorithm in a multicast switch is to formulate the problem as an optimization problem and then solve by using a neural network method as also proposed in [14]. In this scheme, a set of selected packets not only is contention-free but also have a maximal throughput.

1.2 A Description of Analysis

In this thesis, we study the performance of the multicast service discipline with no fanout splitting, under a random packet selection policy. This service discipline has the following advantages:

(1) Comparable throughput performance to multicast service with fanout splitting in the case of low fanout.

(2) No buffering is required between the copy network and the routing network.

(3) The capability of switching all the copies of a packet within the same slot. This will be significant for certain applications such as distributed processing, where it is required to preserve consistency of the global data.

(4) Since each packet may generate one or more copies, the output load is greater than the input load. As a result, both the output lines and the switch may become bottlenecks. The former may be solved by operating the output lines faster than the input lines, or providing more output ports. The bottleneck problem at the switch may be solved by speeding up the operation of the switch fabric versus input lines. As will be seen this service discipline leads itself easily to the speed up operation.

1.2.1 Outline of the thesis

Following this introductory chapter, in Chapter 2, a detailed description of the Random Packet Selection Policy is given. A recursive form for the distribution of the number of chosen packets under this policy as well as the throughput are derived. These results are conditioned on a constant number of packets contending during a slot. Next, in Chapter 3 the steady-state distribution of the number of chosen packets is determined through an embedded Markov Chain analysis. Where the contending packets in any slot will be a mix of fresh and old(blocked) packets. It is shown that the old packets have larger fanout than the fresh packets, and the copy distribution of the fresh and old packets are related to each other through an equilibrium observation. The steady-state distribution and throughput are therefore expressed as a function of the fresh packets only, with the effect of blocking traffic(residual traffic) being embedded in. This constitutes a major contribution of this thesis. Chapter 4 studies the effect of the switch speed up operation on the system throughput. First, an equivalent implementation of the random packet selection policy is discussed, and this implementation is studied through a discrete time birth process which gives an alternative solution for the distribution of the number of chosen packets during a time slot. The advantage of this solution is its close form and this enables the study of the speed up in a multicast switch. Finally a conclusion and some suggestions for the future work are given in chapter 5.

Chapter 2

Random Packet Selection Policy and the Distribution of the Number of Chosen Packets

In a multicast switch performance analysis, the input packet selection policy determines the number of packets that are transmitted during a slot time. Therefore different selection policies for the input packets will result in different throughputs. In this chapter we describe a Random Packet Selection Policy which will be used throughout our analysis. First, we explain the Random Packet Selection Policy and its operation, then we derive an expression for the distribution of the number of packets being successfully transmitted under this policy.

The distribution of the number of packets being successfully transmitted derived in this chapter is conditioned on a given number of contending packets (denoted by α) at the inputs during a particular slot. In our derivation, we also derive the distribution of number of copies in a successfully transmitted packet, and theoretically show that it has a smaller size than the average of input packets. Based on the distribution of the number of transmitted packets, we will find out the mean value for the transmitted packets as well as the average number of copies that these packets generate. We will finally determine the throughput, conditioned on constant number of contending packets during a particular slot.

2.1 Random Packet Selection(RPS) Policy

RPS policy is the generalization of the corresponding policy in unicast switching[9][10]. Here we assume a constant number of contenting packets(α) at the inputs. During a slot, a series of selection rounds is performed in which packets will be chosen according to the RPS policy. This policy operates as follows:

In the first round of the selection, a packet is chosen randomly among the contending (α) packets, following that, those of the remaining contending packets interfering with the already chosen packet are eliminated from further contention and discarded, while noninterfering packets remain. We denote this set of remaining *noninterfering* packets after the first choice by $R1$. In the second round of the selection among $R1$, a second packet is randomly chosen, and again following that, those of remaining contending packets in $R1$, interfering with this second chosen packet, are discarded from $R1$, while the noninterfering packets remain. These remaining *noninterfering* packets form another subset denoted by $R2$, where $R2 \subset R1$. Packets in $R2$ will compete for random selection in the next round. The same process will keep on going until no more noninterfering packets are left after a certain selection round. For example, if after j rounds we get $Rj=\{\phi\}$, then the selection is terminated at jth round and the total number of packets chosen is j. As a result, all those chosen packets form a subset of the contending input packets, in which their copies destined to the outputs have no conflict with one another. These chosen packets therefore will be successfully transmitted in that slot. A simple illustration for this process(with α=4 input contending packets) is shown in Fig. 2.1 and Fig. 2.2:

Here we assume an 8x8 switch where there are four contending packets distributed at input ports 1, 3, 5 and 8, and named packet(a), (b), (c) and (d) respectively. Each contending packet is shown as the set of all copies it generates (see Fig. 2.1). A sequence of eight squares enclosing a digit "0" or "1" indicates a contending packet copy generation status for the outputs. We let the sequence of squares from right to left correspond to the output from port 1 to port 8; and digit "1" indicates a copy generated and destinated to the corresponding output port, while digit "0" means no copy is generated to that output. For example, a contending packet at input port 3 or packet (b) generates 3 copies to output ports 2, 6, and port 7.

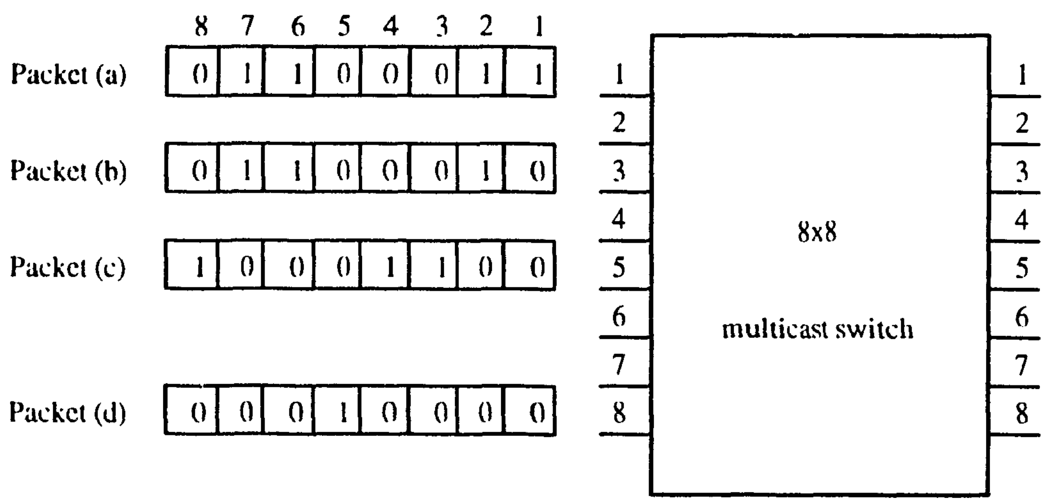


Figure 2.1: An Example of Packet Selection Process under RPS Policy (for α=4)

The RPS policy chooses one packet at a time from the remaining *noninterfering* packets set. At the beginning of the selection, $R0 = \{packets(a), (b), (c), (d)\}$. We assume packet(b) is chosen at the first round, and the selected outputs by this packet are port 2, 6 and port 7 as shown in row 1 of Fig. 2.2.

For the remaining contending packets, only packet(a) has output conflict with the chosen packet(b) at port 2 and thus it is discarded. The remaining *noninterfering* packet set after the first choice is therefore as $R1 = \{packet(c),(d)\}$. At the second selection round among $R1$, it is assumed that packet(c) is chosen and its copies are delivered to those empty outputs left in row 1. As packet(d) does not interfere with the second chosen packet, it will stay as another set of noninterfering remaining packets, giving $R2 = \{packet(d)\}$. The outputs selected in two rounds of selection are shown in row 2.

At the third round of selection, the last one packet(d) is picked up and its copies will be deliveried to those empty outputs left over after the last selection. The selection terminates as $R3 = \{\phi\}$. The total number of chosen packets is 3, and the total number of outputs being selected (or having copies) during that slot is 7 as shown in row 3.

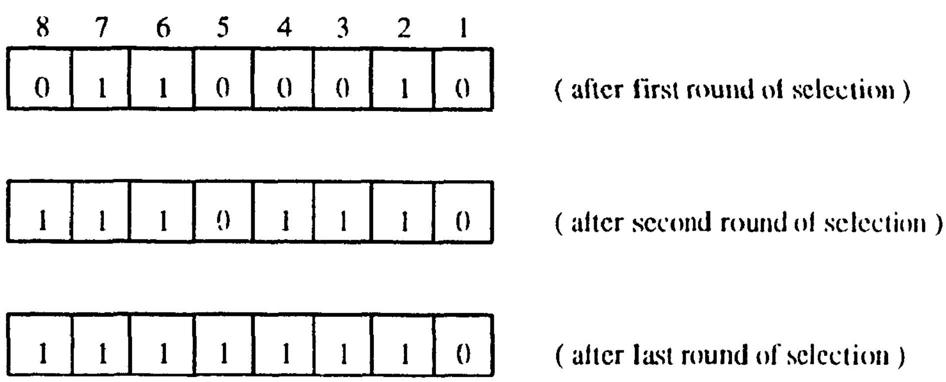


Figure 2.2: An Example of Packet Selection Process under RPS Policy (outputs occupancy)

2.2 The Distribution of the Number of Chosen Packets

2.2.1 Modeling and Definitions

We consider a multicast switch with n inputs n outputs, and also we make the following assumptions:

1. A slotted timing of fixed duration is provided for each input multicasting packet.
2. The number of contending packets at the beginning of a particular slot is assumed to be constant, equal to α, where $\alpha \leq n$.
3. Each packet generates a copy to each of the output ports according to an independent *Bernoulli* process with probability p where

$$\begin{cases} p & \text{a copy generated to an output} \\ 1-p & \text{no copy generated to an output} \end{cases}$$

   This results in a *binomial* distributed number of copies generated for each packet.

Although the copy generation process has a disadvantage that a packet may not generate any copy at all, we start with this assumption to carry out our analysis for the purpose of emphasizing the approach used.

For clarity, we shall mention that in the following analysis, the term "chosen packet" refers to "successfully transmitted packet", while the term "the jth chosen packet" means "the packet of jth choice". In addition the words "generating of a copy to an output" and "selecting an output" are used for the same meaning.

The following additional notation is mainly used in the mathematical derivations in our problem:

r_j: the number of outputs which remain unselected following the choice of the jth packet.

$R_j(z)$: PGF(Probability Generation Function) of the distribution of r_j.

n_j: the number of outputs selected by the jth packet.

$N_j(z)$: PGF of the distribution of n_j.

m_j: the total number of outputs selected by the first chosen j packets.

$M_j(z)$: PGF of the distribution of m_j.

$\tilde{U}_j$: the total number of copies that a contending packet interferes with the jth chosen packet.

$\tilde{U}_j(z)$: PGF of the distribution of $\tilde{U}_j$.

$\tilde{P}_j$: the probability that a contending packet will not interfere with the jth chosen packet.

q_{jk}: the probability distribution of k noninterfering packets left after the jth choice.

$P_j(\alpha)$: the probability distribution of j packets being chosen given α contending input packets during a slot.

$\hat{T}_p$ ($\hat{T}_c$): the packet (or copy) throughput per port given α contending packets during a particular slot.

2.2.2 Analysis

In the RPS policy, we need to know how many selection rounds are required to terminate the packet selection process during a given slot duration. The total number of packets being chosen is determined by the number of selection rounds since only one packet is chosen at each round. The purpose of the following analysis is to derive the distribution of the number of selection rounds(or the number of packets being chosen) during a slot. To do so, we first determine the probability that a contending packet will not interfere with a chosen packet and then we derive the distribution of the number of noninterfering packets remaining after each choice.

In our analysis, we will focus on the remaining number of unselected outputs instead of the selected ones, as they are easier to be kept track of. Here selected/unselected output means there is/is not a copy destinated to it from the chosen input packets. The number of selected outputs can always be obtained by subtracting the number of unselected outputs from the total number of output ports(n). As we have defined, a packet generates a copy to each of the outputs with probability p according to a *Bernoulli* trial, let us define two random variables $\tilde{x}_i$ and x_i for this process as follows:

$\tilde{x}_i$ is the outcome of a *Bernoulli* process for output i,

$$\tilde{x}_i = \begin{cases} 1 & \text{with probabilty } p \\ 0 & \text{with probability } 1\text{-}p \end{cases}$$

Clearly, $\tilde{x}_i$'s are independent identically distributed random variables with probability generating function (PGF)

$$\tilde{X}(z) = \tilde{X}_i(z) = pz + 1 - p \tag{2.1}$$

For convenience, we also define a random variable x_i as the complement of $\tilde{x}_i$,

$$x_i = 1 - \tilde{x}_i$$

with a PGF given by

$$X(z) = X_i(z) = (1-p)z + p \tag{2.2}$$

At the beginning of the selection process, the number of available outputs r_0, and its PGF $R_0(z)$ are:

$$r_0 = n \tag{2.3}$$

$$R_0(z) = z^n \tag{2.4}$$

Since the first chosen packet will select each of n available outputs with probability p, the unselected outputs r_1 following the first choice and its PGF $R_1(z)$ are given by:

$$r_1 = \sum_{i=0}^{r_0} x_i \tag{2.5}$$

where r_0 is independent of x_i, thus

$$R_1(z) = E[z^{r_1}] = R_0(z)|_{z=X(z)} = [1-(1-p)(1-z)]^n \tag{2.6}$$

Similarly, the outputs available for the jth choice are those still unselected following the $(j-1)$th choice. Thus, the remaining unselected outputs r_j following the jth choice is related to r_{j-1} by:

$$r_j = \sum_{i=0}^{r_{j-1}} x_i \tag{2.7}$$

from the independence of r_{j-1} and x_i, the PGF of r_j is:

$$R_j(z) = E[z^{r_j}] = R_{j-1}(z)|_{z=X(z)} \tag{2.8}$$

This formula relates the PGF of remaining number of unselected outputs following the jth choice to that of following the $(j-1)$th choice. The repeated application of Eq.(2.8) with the initial condition in Eq.(2.4) results in:

$$R_j(z) = [1-(1-p)^j(1-z)]^n \tag{2.9}$$

Since the jth chosen packet performs a *Bernoulli* trial for each of the available r_{j-1} outputs, the number of outputs selected by the jth packet n_j, and its PGF $N_j(z)$ are:

$$n_j = \sum_{i=1}^{r_{j-1}} \tilde{x}_i \tag{2.10}$$

and

$$N_j(z) = E[z^{n_j}] = R_{j-1}(z)|_{z=\tilde{X}(z)} \tag{2.11}$$

or:

$$N_j(z) = [1 - p(1-p)^{j-1}(1-z)]^n \tag{2.12}$$

The average number of outputs selected by the jth chosen packet $\bar{n}_j$ is:

$$\bar{n}_j = \frac{dN_j(z)}{dz}|_{z=1} = np(1-p)^{j-1} = \mu(1-p)^{j-1} \tag{2.13}$$

where $\mu = np$ is the average number of copies generated by an input packet. As may be seen, only the first chosen packet has the same average number of copies as a typical packet, while the size of the subsequently chosen packets drops.

To find out the total number of outputs selected by the first j chosen packets, let us define m_j as the total number of outputs selected by the first j chosen packets. Here m_j equals to the total number of output ports minus the number of unselected outputs after the jth choice. That is:

$$m_j = \sum_{i=1}^{j} n_j = n - r_j \tag{2.14}$$

and its PGF

$$M_j(z) = E[z^{m_j}] = E[z^{n-r_j}] = z^n E[z^{-r_j}] = z^n R_j(z^{-1}) \tag{2.15}$$

by substituting for $R_j(z)$ from equation(2.8), we have:

$$M_j(z) = [z - (1-p)^j(z-1)]^n \tag{2.16}$$

The average number of total outputs selected by the first j chosen packets, $\bar{m}_j$, is then given by:

$$\bar{m}_j = \frac{dM_j(z)}{dz}|_{z=1} = n[1-(1-p)^j] \tag{2.17}$$

From now on we are ready to apply the RPS policy to determine the distribution of the number of chosen packets in a particular slot with constant number(α) of contending input packets. As we discussed in section 2.1, following each packet selection, those of remaining contending packets interfering with the chosen packet will be eliminated from further contention and be discarded, while the noninterfering packets remain and compete in the next selection round. The same process will continue until the number of noninterfering packets left over after certain selection rounds becomes zero. Let us further define

q_{jk} as the probability distribution of number of noninterfering packets left after the jth choice.

Thus the probability $P_j(\alpha)$ that "j and only j packets will be chosen among a given number of contending packets(α) during a slot" shall be given by:

$$P_j(\alpha) = q_{j0}\prod_{i=0}^{j-1}(1-q_{i0}) \tag{2.18}$$

where q_{j0} is the probability that zero non-interfering packets will remain following the jth choice. To get q_{j0}, we need to know q_{jk}. This constitutes the main complement of our analysis for the RPS policy case. The probability distribution q_{jk} can be computed in a manner, as explained below:

In z-domain, the PGF of a random sum of random variables is applied repeatedly in our derivation. First we shall determine the probability that "a contending packet will not interfere with the jth chosen packet", we denote this by $\tilde{P}_j$.

We define u_l as follows,

$$u_l = \begin{cases} 1 & \text{if a contending packet interferes with the } l\text{th copy} \\ & \text{of the } j\text{th chosen packet} \\ 0 & \text{otherwise} \end{cases}$$

$$u_l(z) = pz + 1 - p \tag{2.19}$$

having known the number of copies in the jth chosen packet n_j, and its PGF $N_j(z)$ in equation (2.12), the number of copies that a contending packet interferes with the jth chosen packet $\tilde{U}_j$ is given by:

$$\tilde{U}_j = u_1 + u_2 + \cdots + u_{n_j} = \sum_{i=0}^{n_j} u_i$$

from the independence of n_j and u_i, the PGF of $\tilde{U}_j$ is:

$$\tilde{U}_j(z) = E[z^{\tilde{U}_j}] = N_j(z)|_{z=u_l(z)}$$

or substituting from (2.12) and (2.19):

$$\begin{aligned} \tilde{U}_j(z) &= [1 - p(1-p)^{j-1}(1 - pz - 1 + p)]^n \\ &= [1 - p^2(1-p)^{j-1}(1-z)]^n \end{aligned} \tag{2.20}$$

Then $\tilde{U}_j(0)$ corresponds to the zero number of copies that a contending packet interferes with the jth chosen packet, namely the probability $\tilde{P}_j$. Therefore we have:

$$\tilde{P}_j = \tilde{U}_j(0) = [1 - p^2(1-p)^{j-1}]^n \tag{2.21}$$

Having $\hat{P}_j$, we shall next determine the total number of contending packets which will not interfere with the jth chosen packet. For this purpose, we further define w_l as:

$$w_l = \begin{cases} 1 & \text{if a contending packet does not interfere with} \\ & \text{the } j\text{th chosen packet} \\ 0 & \text{otherwise} \end{cases}$$

then the distribution of w_l, and its corresponding PGF is given by:

$$w_l = \begin{cases} 1 & \text{with probability } \hat{P}_j \\ 0 & \text{with probability } 1 - \hat{P}_j \end{cases}$$

$$w_l(z) = \hat{P}_j z + 1 - \hat{P}_j \tag{2.22}$$

In the following, a recursive form, for the probability distribution of "k remaining noninterfering packets after the jth choice(q_{jk})", is derived from the two relationships considered in the packet selection process:

Let k_j' denotes the number of packets available for the jth choice ($k_j' \geq 1$), and q_{jk}' and $Q_j'(z)$ denote the probability distribution and the PGF of k_j' respectively

Let k_j denotes the number of noninterfering packets left over after the jth choice, and q_{jk} and $Q_j(z)$ denote the probability distribution and the PGF of k_j respectively.

Since:

$$k_j = w_1 + \ldots + w_l + \ldots + w_{k_j'-1} \tag{2.23}$$

the minus one in the last subscript accounts for the jth choice. Then the relationship between the PGFs of distribution of k_j and k_j' from this random sum equation is given by:

$$Q_j(z) = z^{-1}Q'_j(z)|_{z=w_l(z)} \tag{2.24}$$

Where multiplication by z^{-1} takes into account that the jth choice reduces the number of packets by one. Then, a second relationship which relates $q_{(j-1)k}$ and q'_{jk}, namely the distribution of number of noninterfering packets left after the $(j-1)$th choice and the distribution of number of available packets for the jth choice, is found and given by:

$$q'_{jk} = \frac{q_{(j-1)k}}{1 - q_{(j-1)0}}, \qquad k \geq 1$$

In the above, we have normalized $q_{(j-1)k}$ with a factor $\frac{1}{1-q_{(j-1)0}}$, which denotes the probability that after the $(j-1)$th choice there will be at least one noninterfering packet available for the jth choice to take place.

In terms of their corresponding PGFs, we have:

$$\begin{aligned} Q'_j(z) &= \sum_{k=0}^{n} z^k q'_{jk} \\ &= \sum_{k=1}^{n} z^k \cdot \frac{q_{(j-1)k}}{1 - q_{(j-1)0}} \\ &= \frac{1}{1 - q_{(j-1)0}}[Q_{(j-1)}(z) - q_{(j-1)0}] \end{aligned} \tag{2.25}$$

By substituting for $z = w_l(z) = \tilde{P}_j z + 1 - \tilde{P}_j$ into the equation (2.24) and then $Q'_j(z)$ with equation (2.25), a recursive form is finally found as:

$$Q_j(z) = \frac{z^{-1}}{1 - q_{(j-1)0}}[O_{(j-1)}(z) - q_{(j-1)0}]|_{z=\tilde{P}_j z+1-\tilde{P}_j} \tag{2.26}$$

with a given number of contending packets(α), the initial distribution q_{0k} and its PGF $Q_0(z)$ are given by:

$$q_{0k} = \begin{cases} 1 & k=\alpha \\ 0 & \text{otherwise} \end{cases}$$

and

$$Q_0(z) = z^{\alpha} \tag{2.27}$$

Therefore the equation (2.26), which relates the PGF of the distribution of remaining noninterfering packets after the $(j-1)$th to jth choice, will enable us to calculate q_{jk} recursively. For each j (j=1,2,$\cdots$, α), q_{jk} is the corresponding coefficient of z^k in $Q_j(z)$. Having q_{jk}, the probability distribution of number of chosen packets $P_j(\alpha)$ in equation (2.18) can be then computed.

2.3 Throughput Performance

From the distribution of the number of chosen packets, the mean value of number of chosen packets and the average number of total copies generated from these packets are found. In the following we give the throughput performance in terms of both successfully transmitted packets and generated copies per port during a slot.

We defined,

packet throughput:

$$\hat{T}_p = \frac{\text{the average number of packets chosen during a slot}}{\text{number of input ports}} = \frac{1}{n}\sum_{j=1}^{n} j P_j(\alpha) \tag{2.28}$$

By conditioning on the event that "there are j and only j packets being chosen during a slot", the average number of total copies generated by j chosen packets is equal to the average number of total outputs selected in j choices, which is given in equation (2.17) as:

$$\bar{m}_j = n[1 - (1-p)^j]$$

Therefore copy throughput is given by:

$$\hat{T}_c = \frac{\text{the average number of copies generated during a slot}}{\text{number of output ports}} = \frac{1}{n}\sum_{j=1}^{n} \bar{m}_j P_j(\alpha) \quad (2.29)$$

by substituting for $\bar{m}_j$,

$$\hat{T}_c = 1 - \sum_{j=1}^{n}(1-p)^j P_j(\alpha) \quad (2.30)$$

2.4 Numerical Results

The numerical results of this chapter are shown in the figures 2.1 ~ 2.3 and explained as follow.

Figure 2.1 presents the average number of copies in the jth chosen packet, $\bar{n}_j$, as a function of j as given in (2.13). $\bar{n}_j$ is plotted for various values of p(copy generation probability) and for a switch size of n=32. The starting point of the curves corresponds to the average number of copies in the first chosen packet which equals to the average number of copies generated by the input packets, $\mu = np$. Following that all the curves decrease monotonically, indicating that the chosen packet size is getting smaller one by one. The larger the copy generation probability p, the faster is the decay of chosen packet size. As a result, short packets pass through the switch system easier with long packets being left behind.

Figure 2.2 presents the probability that "a contending packet will not interfere with the jth chosen packet", $\hat{P}_j$, as a function of j given in (2.21). Results are plotted for different values of p and a switch size of n=32. As may be seen, all the curves are monotonically increasing. This is expected because the size of the chosen packet drops down as the packet selection goes on. As a result, with increasing j, the probability that a contending packet will not interfere with the jth chosen packet is getting higher. Clearly, because $\hat{P}_j$ is low for initial values of j, most of the contending packets are discarded in the early rounds.

Figures 2.3a and 2.3b present the packet throughput per port, $\hat{T}_p$ and the copy throughput per port, $\hat{T}_c$, both as a function of p. The number of contending packets and the switch sizes are α=n=8 and α=n=16. In Fig. 2.3a, as may be seen the packet throughput goes down as p gets larger, due to the increasing interference among larger contending packets, and in the limit the throughput approaches $\frac{1}{n}$ which corresponds to the choice of a single large packet. In Fig. 2.3b, from the copy throughput curves we can see that, for very small values of p, though the packet throughput is very high, copy throughput is very low due to very small size of a chosen packet. As p increases, $\hat{T}_p$ drops down while $\hat{T}_c$ rises, reaching a maximum because of the increasing number of copies in a chosen packet. This will be a convenient operating point for the system since both $\hat{T}_p$ and $\hat{T}_c$ are relatively high. Then following this maximum, $\hat{T}_c$ drops slightly and then increases linearly, and in the limit $\hat{T}_c \rightarrow 1$ where single packet which generates copies to most of the outputs is chosen.

Results of throughput shown in this chapter are valid when system is operating in the saturation case and discarding blocked packets in each slot.

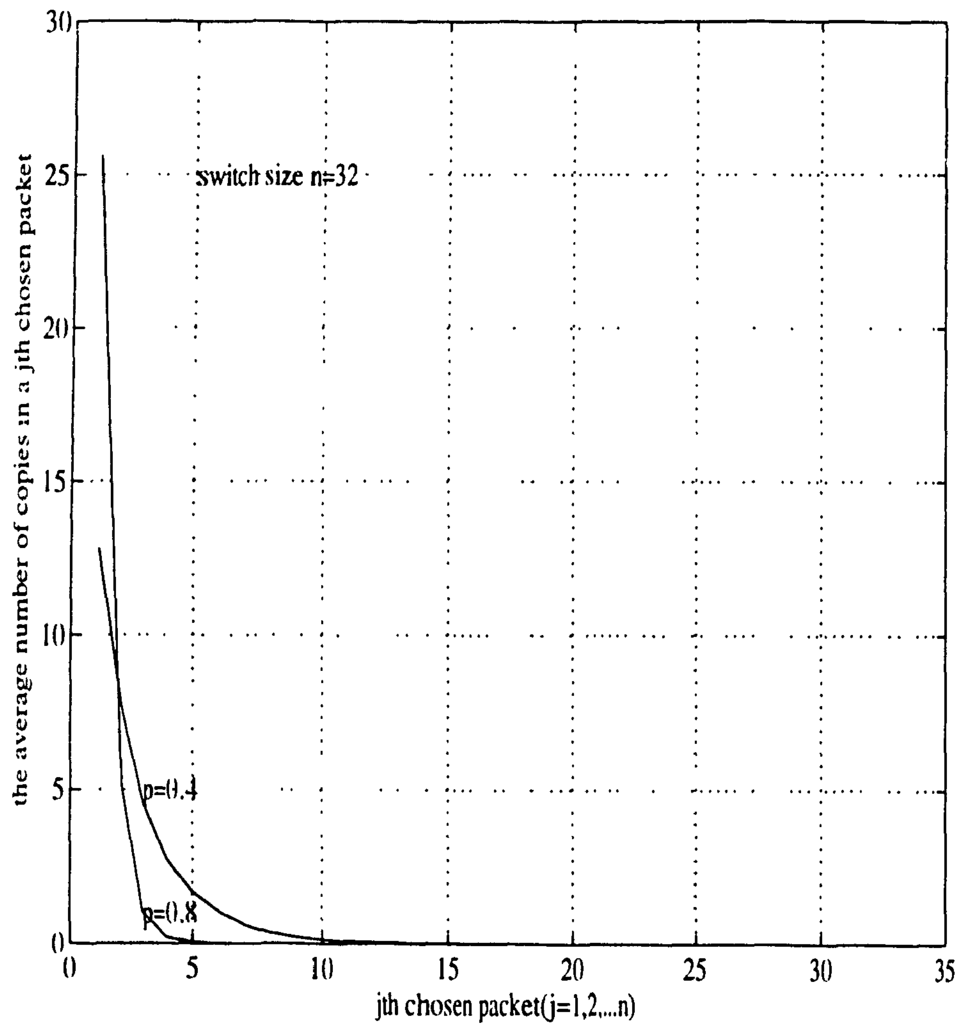


Fig. 2.1 The average number of copies in a j th chosen packet as a function of j from Eq.(2.13), for different values of copy generation probability p=0.4, 0.8 and switch size of n=32.

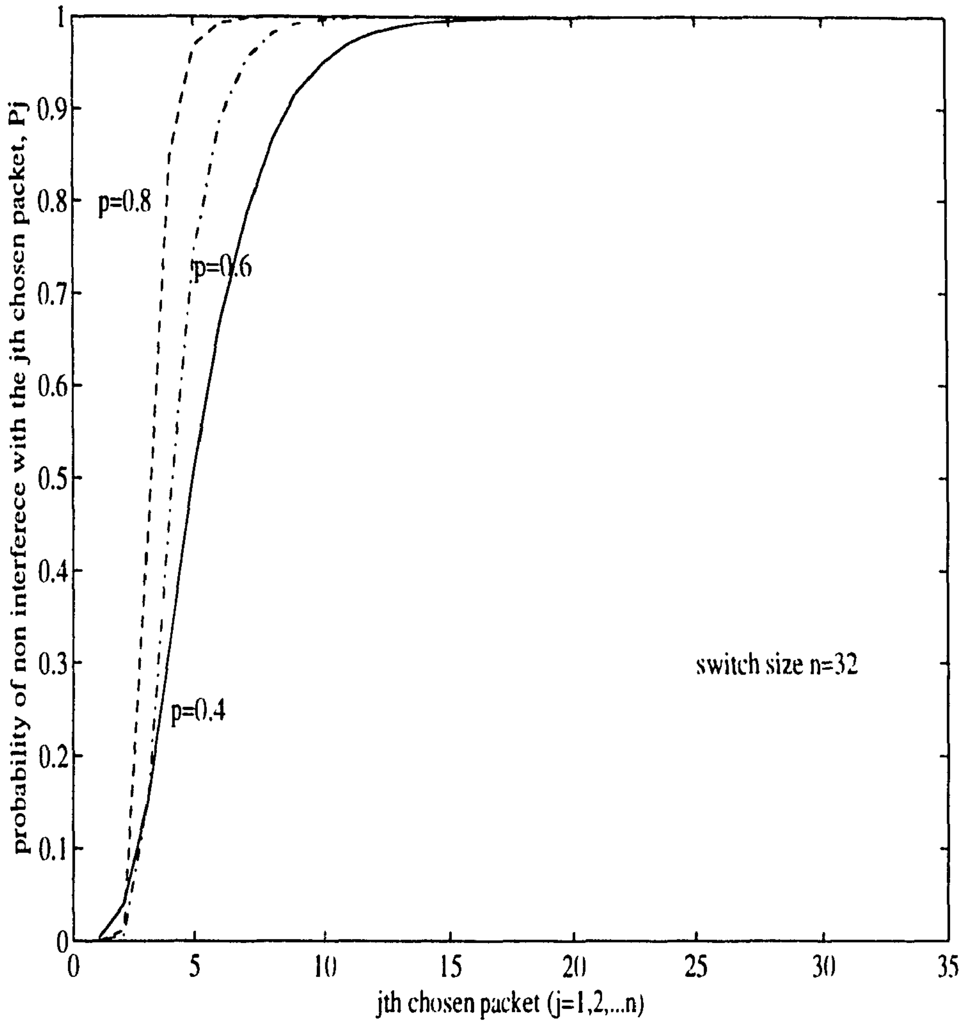


Fig. 2.2 Probability that a contending packet will not interfere with jth chosen packet from Eq.(2.21), as a function of j, for different values of copy generation probability p=0.4, 0.6, 0.8 and switch size of n=32.

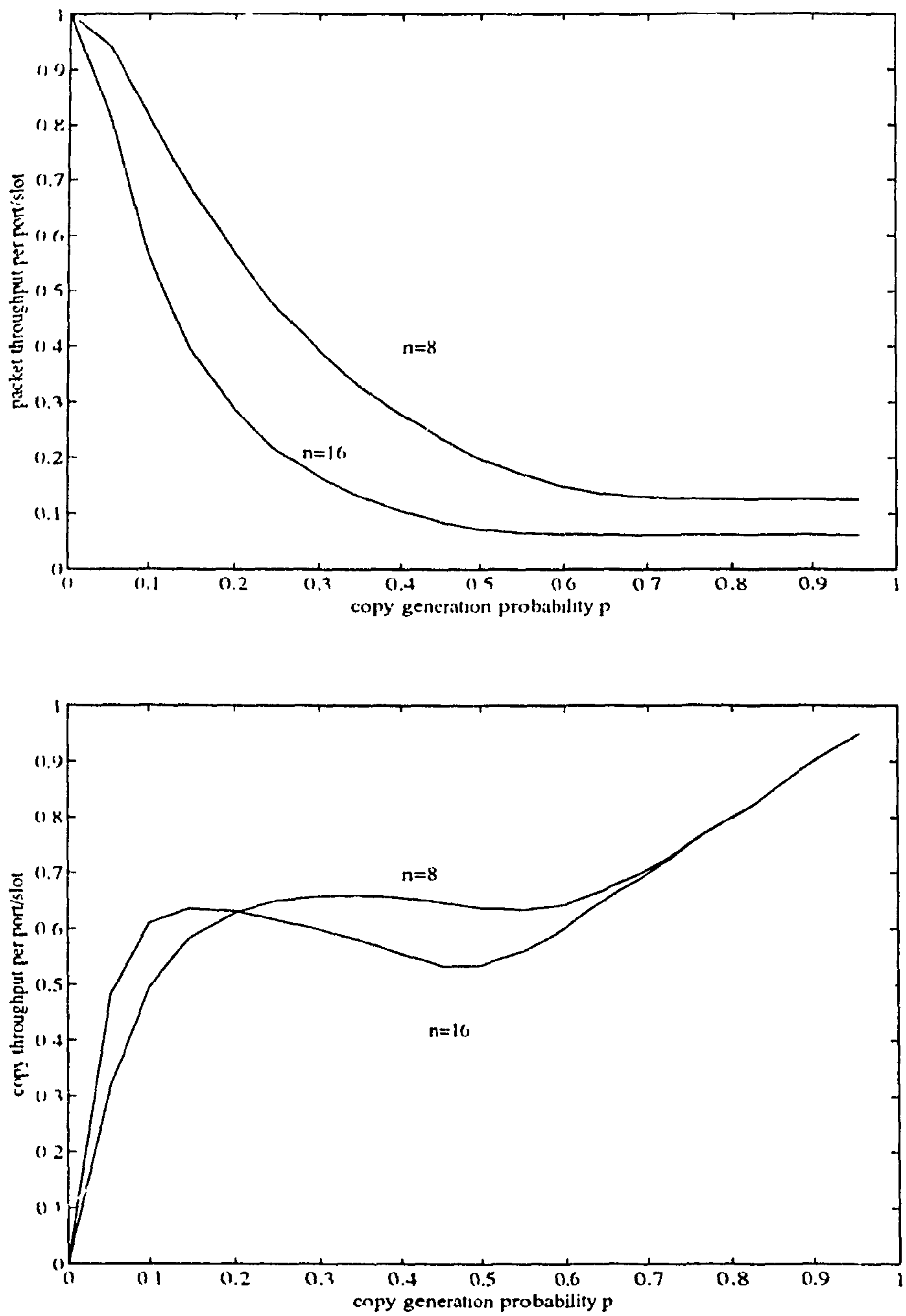


Fig. 2.3. Packet throughput and copy throughput against copy generation probability p, with no. of contending packets equal to switch size n=8, 16.

Chapter 3

The Steady-State Distribution of Number of Packets Chosen with Heterogeneous Contending Packets

During normal operation with the blocked(interfering) packet left from the previous slots being taken into account, the total number of contending packets varies from slot to slot, except when the system becomes saturated. This is in contrast with our previous case in chapter 2 where the distribution of chosen packets was determined for a fixed number of contending packets in a particular slot. In the first section of this chapter, we derive the steady-state distribution of the number of successfully transmitted packets by applying a discrete time Markov Chain analysis. In section 3.2, we decompose the steady-state traffic into new and old traffic and present an approximation for the copy generation probability of the mixed traffic. To take into account the blocked packet's effect on the distribution of the number of transmitted packets and hence the steady-state throughput performance, an equilibrium equation for the average size of transmitted packets is derived. As a result, both the distribution of the number of packets being transmitted and the throughput can be expressed as a function of fresh incoming traffic only. The numerical results are presented in section 3.3, and finally the simulation environment and the accuracy of the approximations are described.

3.1 A Markov Chain Modelling and Analysis for the Distribution of the Number of Chosen Packets

In steady-state the number of packets being blocked in each slot is a stochastic process which can be described in terms of a discrete-time Markov Chain. In the Markov Chain modeling, we place the embedded points at the end of the slots. Packets at an embedded point will be referred to as old packets and they correspond to those packets who have lost contention in the previous slot. These old packets will wait in their inputs for contention in the next slot. It is assumed that an idle input at an embedded point will generate a fresh packet with probability ρ at the beginning of the next slot. These newly generated packets, together with the old packets will be contending for the random selection during that slot. The chosen packets are transmitted, while the blocked packets are left behind as old packets at the next embedded point. By this Markov Chain analysis, we will be able to determine the distribution of the number of old packets at an embedded point. This will enable us to determine the number of existing idle inputs, and consequently the number of new packets generated from these inputs at the beginning of the next slot, will be derived. Our objective behind this analysis is to determine the steady-state distribution of number of the packets being successfully transmitted during any slot(W_j). For this, we need to find the probability distribution of number of the total contending packets at the beginning of a slot:

Some additional notation is defined as follows:

$p_{t|m}$: transition probabilities of having t old packets at an embedded point given m old packets at the previous embedded point.

$\nu(f)$: probability of having f new packets arriving at the beginning of a slot.

$\tau(i)$: probability that a sum of i (new and old) packets will be contending during a slot.

Π_k: probability of having k packets at an embedded point.

W_j: steady-state probability that j packets will be chosen during a slot.

A transition probability matrix P and a row vector Π for the probability distribution of the number of old packets at an embedded point are respectively defined as:

$$P = \begin{bmatrix} p_{0|0} & p_{0|1} & p_{0|2} & \cdots & p_{0|(n-1)} \\ p_{1|0} & p_{1|1} & p_{1|2} & \cdots & p_{1|(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{t|0} & \cdots & p_{t|m} & \cdots & p_{t|(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{0|(n-1)} & p_{1|(n-1)} & p_{2|(n-1)} & \cdots & p_{(n-1)|(n-1)} \end{bmatrix}_{(n-1)*(n-1)} \tag{3.1}$$

$$\Pi = [\Pi_0 \Pi_1 \ldots \Pi_k \ldots \Pi_{n-1}] \tag{3.2}$$

If we assume that the number of contending packets in any slot is always greater than zero , then the number of chosen packets will be at least one. This explains why in (3.2) the maximum number of old packets is (n-1) at an embedded point.

To determine the transition probabilities, $p_{t|m}$, we first determine the following conditional probabilities:

(1). $\nu(f|m)$=Pr{f new arrivals | m old packets at an embedded point}

Clearly, there will be (n-m) empty inputs given m old packets at an embedded point. Since each empty input generates a new packet with probability ρ in a slot, this results in a *binomial* distribution of $\nu(f|m)$:

$$\nu(f|m) = \begin{pmatrix} n-m \\ f \end{pmatrix} \rho^f (1-\rho)^{n-m-f} \tag{3.3}$$

(2). $\tau(i|m)$=Pr{i packets are contending | m old packets at an embedded point}

Since given m old packets at an embedded point and the total number of contending packets is i, then $(i-m)$ of them will constitute the newly generated packets. Therefore we have:

$$\tau(i|m) = \begin{cases} \nu(i-m|m) & i \geq m \\ 0 & i < m \end{cases} \tag{3.4}$$

Therefore, the transition probability, $p_{t|m}$, that "there are t packets at the present embedded point given there were m packets at the previous embedded point " is given by:

$$p_{t|m} = \begin{cases} \sum_{i=t+1}^{n} P_{i-t}(i)\tau(i|m) & t > 0 \\ \sum_{i=0}^{n} P_i(i)\tau(i|m) & t = 0 \end{cases} \tag{3.5}$$

where the above derivation was based on the fact that, given that there are m old packets at an embedded point and a total number of i contending packets in a slot, then $(i-t)$ packets have to be chosen and transmitted in order to leave t at the next embedded point. In addition in (3.5) $P_{i-t}(i)$ is the probability that $(i-t)$ packets will be chosen from a total number of i contending packets in a slot and, from (2.18), it is given by:

$$P_j(i) = q_{j0} \prod_{h=0}^{j-1} (1 - q_{h0}) \qquad 1 \leq j \leq i \tag{3.6}$$

where instead of α, the number of contending packets becomes i.

Finally $p_{t|m}$ is determined in equation (3.5) by summing over all possible values of i, $i \geq t$. Following the determination of the transition probability matrix P, the steady-state distribution of number of old packets at an embedded point may be found by solving the vector equation given by:

$$\Pi = \Pi P$$

Once this is done, the steady-state distribution of the number of packets chosen during a slot, W_j, can be found as follows:

$$W_j = \begin{cases} \sum_{k=0}^{n-1} \sum_{i=j}^{n} P_j(i) \tau(i|k) \Pi_k & j > 0 \\ \tau(0|0)\Pi_0 & j = 0 \end{cases} \tag{3.7}$$

where

$$\tau(i|k) = \begin{cases} \begin{pmatrix} n-k \\ i-k \end{pmatrix} \rho^{i-k}(1-\rho)^{n-i} & i \geq k \\ 0 & i < k \end{cases}$$

The $W_j's$ are derived as follows:

First recall that $P_j(i)$ is the probability that j packets will be chosen given a total of i contending packets. Next $\tau(i|k)$ gives the probability that i packets will be contending given k old packets at an embedded point. Thus we uncondition $P_j(i)$ with respective to i using $\tau(i|k)$ and then with respective to k using Π_k.

Once the $W_j's$ are found, the packet and copy throughput per slot are then obtained using Eq.(2.28) and Eq.(2.30), respectively, with $P_j(\alpha)$ replaced by W_j, ie:

$$T_p = \frac{1}{n} \sum_{j=1}^{n} j W_j \tag{3.8}$$

and

$$T_c = 1 - \sum_{j=1}^{n} (1-p)^j W_j \tag{3.9}$$

At this point we shall note, that the steady-state distribution and the throughput results are both function of the mixed traffic, old and new packets. In the next section, we will study the decomposition nature of the input traffic.

3.2 Traffic Decomposition at Steady-state

At steady-state, the traffic at the switch inputs consists of a mixture of two kinds of packets: new incoming packets and old packets left over from the previous slots. The copy generation characteristics of new and old packets is not the same, and this affects the throughput of the system. Therefore, a closer look into this traffic becomes necessary. In this section, we first give an approximation for the copy generation process for a packet belonging to the mixed traffic. Then we express the steady-state throughput as a function of only new incoming traffic through an equilibrium equation.

3.2.1 An Approximation of the Copy Generation Process for Mixed Traffic

As it has been shown in chapter 2, the overall average size of transmitted packets is smaller than the average of the contending packets. As a result, the packets left behind(old packets) will be larger packets and the copy generation probability to each output from these packets will become higher. If we assume that an incoming packet(fresh packet) and blocked packet(old packet) each generates a copy to an output with probability p_f and p_o, respectively, then we must have $p_o > p_f$. Thus the mixed packets have heterogeneous packet length distributions (packet length is in terms of number of copies in a packet). In the following, the copy generation process for the mixed traffic will be approximated, and its accuracy will be shown.

Next, we introduce some notations:

$b(z)$: PGF of the distribution of the number of copies generated by an input packet without making a distinction between the fresh and old packets.

$b_f(z)$: PGF of the distribution of the no. of copies generated by a fresh packet.

$b_0(z)$: PGF of the distribution of the no. of copies generated by an old packet.

p_f: probability that a fresh packet generates a copy to an output.

p_o: probability that an old packet generates a copy to an output.

p: probability that a mixed packet generates a copy to an output.

The corresponding PGFs of the number of copies generated by a fresh packet and an old packet are respectively given by:

$$b_f(z) = (p_f z + 1 - p_f)^n \tag{3.10}$$

$$b_o(z) = (p_o z + 1 - p_o)^n \tag{3.11}$$

Next let us assume that the number of contending packets in a slot, α, which is the sum of fresh packets, f, and old packets, o, is constant. In the policy under study, during each selection round one of the remaining contending packets is chosen randomly and this is analogous to sampling without replacement(with two types of objects: fresh and old packets). This sampling can be described by a classical model: There are two kinds of balls with different colors (red and black) in a urn. R red balls and B black balls. One ball is drawn at random each time without replacing it. Assuming we draw n balls without replacing them, then the probability of the event that "j red balls are chosen" is given by the *hypergeometric* distribution[17]. This distribution approaches a *binomial* distribution as the total number of balls approaches infinity (T=R+B $\rightarrow \infty$). Thus for large T, the probabilities of drawing a red ball or a black ball at each trial approach constants given by $\frac{R}{T}$ and (1-$\frac{R}{T}$) respectively. Let a fresh packet corresponds to a red ball and an old packet to a black ball, and define p_s and (1-p_s) as the probabilities of randomly choosing a fresh and an old packet respectively at each selection round. Assuming large α, then p_s is given by $\frac{f}{\alpha}$. Therefore, the PGF of distribution of the number of copies in a chosen packet will be given by:

$$b(z) = p_s b_f(z) + (1 - p_s) b_o(z) \qquad where \quad p_s = \frac{f}{\alpha} \tag{3.12}$$

or

$$b(z) = \frac{f}{\alpha}(p_f z + 1 - p_f)^n + (1 - \frac{f}{\alpha})(p_o z + 1 - p_o)^n \tag{3.13}$$

The above distribution will be approximated by a *binomal* distribution by assuming that each mixed packet will generate a copy to an output with probability p, which n be expressed in terms of p_f and p_o as follows:

$$p = \frac{f}{\alpha} p_f + (1 - \frac{f}{\alpha}) p_o \tag{3.14}$$

Therefore the new PGF of the distribution of the number of copies generated by a mixed packet at the input becomes:

$$B(z) = (pz + 1 - p)^n \tag{3.15}$$

This binomial distribution, B(z), is the approximation for the distribution in equation (3.13) and both of them have the same average number of copies, namely:

$$\mu = np = n[\frac{f}{\alpha}p_f + (1 - \frac{f}{\alpha})p_o] \quad (3.16)$$

Clearly, the above definition of p is consistent with its prior use in this thesis, therefore the results which have been derived so far remain valid. The accuracy of this approximation is shown in Table 3-1 and the throughput performance based on this approximation matches very well the simulation results as illustrated in Fig. 3-2~3-4. Clearly, because the number of fresh and old packets and their total ($\alpha = f + o$) are random variables, these quantities have to be replaced with their averages. As a result, equation (3.14) becomes,

$$p = \frac{f}{\alpha}p_f + (1 - \frac{f}{\alpha})p_o \quad (3.17)$$

3.2.2 An Equilibrium Consideration on the Transmitted Packet Size

In steady-state, a mixed packet and a new packet have different average sizes. In this section, we show the relationship between them through an equilibrium consideration. Based on that relation, we will be able to determine the throughput as a function of only new incoming traffic. The *Bernoulli* copy generation process gives the average size of an incoming fresh packet(np_f), of an old packet(np_o) and of a mixed packet (np). We first look to equation (3.17) which relates these three quantities. Among the three copy generation probabilities p_f, p_o and p, only p_f is an external parameter, the other two are internal and unknown. However, from the steady-state equilibrium consideration, all packets arriving at the inputs will finally go through the switch to their corresponding outputs. Therefore the average packet size, observed from both inputs and outputs should be identical. The input average is given by $\mu_f = np_f$, while the departing packet average size(μ_d), is determined as follows: Conditioning on the event that "j and only j packets will be transmitted in a slot ", and from the previous analysis in chapter 2, the average size of j departing packets is:

$$(\mu_d|j) = \frac{\sum_{i=1}^{i=j} \bar{n}_i}{j} \tag{3.18}$$

or

$$(\mu_d|j) = \frac{\bar{m}_j}{j} \tag{3.19}$$

where m_j is the total number of copies generated by j chosen packets which from (2.17) is given by:

$$\bar{m}_j = \sum_{i=1}^{i=j} \bar{n}_i = n - r_j = n[1 - (1-p)^j] \tag{3.20}$$

Clearly, the above result is in terms of p, the copy generation probability of mixed packets. Unconditioning $(\mu_d|j)$ with respective to j using the probability distribution of the number of chosen packets, W_j, the average size of a departing packet is then given by:

$$\mu_d = \sum_{j=1}^{n} \frac{\bar{m}_j}{j} \frac{W_j}{1 - W_0} \tag{3.21}$$

where the above normalization takes into account that at least one packet needs to be chosen during a slot in order that we can talk about its size. Substituting for $\bar{m}_j$ gives:

$$\mu_d = n \sum_{j=1}^{n} \frac{1 - (1-p)^n}{j} \frac{W_j}{1 - W_0} \tag{3.22}$$

Finally equating $\mu_o = \mu_f$ gives p_f in terms of p as:

$$p_f = \sum_{j=1}^{n} \frac{1 - (1-p)^j}{j} \frac{W_j}{1 - W_0} \tag{3.23}$$

where the steady-state distribution, W_j, has been solved for every value of p in section 2.1. This equation expresses the relationship between the copy generation probabilities of incoming and mixed packets. In the previous section, we have calculated the distribution of the number of packets chosen as well as the throughput T_p and T_c by varying p in the range [0, 1]. From (3.23), for each value of p, the distribution and the throughput can be expressed as a function of the external parameter p_f.

As may be seen equation (3.17) relates p, p_f, p_0. Since we have expressed p in terms of p_f, we can also determine p_0 in terms of p_f which will show us that $p_0 > p_f$(see Fig. 3.1).

3.3 Numerical Results

First, Table 3.1 presents some numerical results concerning the accuracy of the binomial approximation for the mixed packet copy distribution in equation (3.13). As may be seen, the two probability distribution are quite close.

Fig. 3.1 presents the average packet length of three kinds of traffic: fresh packet, old packet and mixed packet from equations (3.23) and (3.17). The curve confirms our theoretical finding that an old packet has a larger average length than a fresh packet. This result was also observed through simulation in [14].

Fig.3.2~3.4 present the steady-state packet and copy throughput per slot as a function of the fresh packet copy generation probability p_f, for different switch sizes(n=8, 16, 32). Results are plotted for different values of the load parameter (ρ). Also shown are the corresponding simulation results. As may be seen the packet throughput drops down as p_f increases, because of the increasing interference among the packets, and has the asymptotic value of $\frac{1}{n}$. The copy throughput starts with low value due to very small packet sizes, but increases with p_f reaching a maximum then dropping slightly and then once again rising to one. It appears that the best operating point will be in the middle, since both the packet and copy throughputs are significantly high. As may be seen, the theoretical and simulation results agree with each other though the agreement is better for the packet throughput results. We also notice that the curve shapes for the small and large switch sizes are the same.

3.4 Simulation Environment Description and Accuracy of the Approximations

In order to interpret correctly the difference between the simulation and the theoretical results, we provide, in this section, a brief description of the simulation environment and the theoretical approximations in the study.

3.4.1 The Simulation Environment Description

1. *The Random Number Generator for Both Packet and Copy Generation*: Recall from our earlier analysis, that at each input port, a packet is generated according to a Bernoulli process with probability ρ, while at the output a copy is generated with probability p. Therefore in the simulation, we use two library functions(in C): srand48(seed) and drand48(), as a random number generator to create these two Bernoulli processes.

The function srand48(seed) takes one parameter "seed" for the random number generation; while with a given seed value, drand48() then generates an uniformly distributed random number r in the range of [0, 1]. For the input packet generation process, we take the event $(r_p \leq \rho)$ to indicate that there is a packet to be generated at an input, and $(r_p > \rho)$ to indicate the other case. The same indications with $(r_c \leq p)$ and $(r_c > p)$ are given to the copy generation at the outputs. We have provided independent values for "seed", each time a random number is generated.

2. *The Packet Selection Process and the Residual Traffic Processing*: To randomly choose a packet for transmission among the remaining non-interfering

contending packets, the random number generator is once again used. Assuming that at a certain round of the selection there are m available contending packets, multiply m by a random number r, $(r * m)$ will give a value which falls into one of m intervals of [i-1, i], where $i \in \{1, \cdots, m\}$. Then we take the *ith* packet among m as a random chosen packet. When the number of available contending packets reduces to zero, the selection is terminated and a new slot starts.

For those packets interfering with the chosen packet during a slot, they are referred to blocked packets. A blocked packet is kept at its input and will join the contention in the next slot. In the simulation, we carry the blocked packet from slot to slot until it is chosen and transmitted. According to the time duration that a blocked packet has stayed at its input, we have different ages for the blocked packets. Therefore our simulation took into account the effect of the residual traffic as in the actual operation of the system.

3. *The Steady-state Conditions*:

After certain long running time, the simulation may reach a steady-state. We used the following ways to confirm it:

(a). When we increase the number of running slots, both the packet throughput and the copy throughput are not changing any more.

(b). We consider that the system reaches equilibrium when it reaches the steady state, in which case, any packet that arrives to the switch will eventually depart from it. Based on this and in our simulation, when the condition (a) is satisfied, we observed that the total number of packets generated at the inputs was equal to the total number of packets departed from the outputs.

To obtain more accurate results about the steady-state performance, we have started collecting statistics after 10^3 slots in order to eliminate the transient period.

3.4.2 Accuracy of the Approximations

Two approximations were made in our steady-state performance analysis of the multicast switch. These are:

(a). For the mixed traffic(blocked+new packets) in steady-state, the copy generation process of a mixed packet was approximated by a binomial distribution over the outputs with a parameter p, where $p = p_s p_f + (1 - p_s)p_o$.

(b). We ignored the fact that different copy generation probabilities ($p_{o1}, p_{o2}, \cdots$, $p_{oj}, \cdots$) exist for different ages of blocked packets. Instead we used a single parameter p_o to characterize the average of copy generation probabilities of all ages of blocked packets. This approximation has simplified the analysis which, otherwise may become intractable, as we have to take into account a large set of different parameters, p_{oj}(where $j = 1(slot), 2(slot), \cdots$).

In addition, a simple calculation can show us that, when the switch size $n \to \infty$ (i.e. n=256) and at the case of low fanout($\mu, \mu = np_f$), the binomial distribution of copy generation process has a nearly zero probability of $P_r(k = 0)$, this means the affection of the probability that a packet may generates zero copy can be neglected.

At the end of this chapter we shall mention that, our approach of Markov Chain analysis and the equilibrium consideration, for the determination of the steady-state distribution of the number of transmitted packets as well as the throughput with taking into account of the blocking effect on the switch performance, does not depend on a particular packet selection policy or a particular packet copy generation process. In other words, this approach is applicable to other packet selection policies such as a variation of the RPS policy that we will discuss in the next chapter.

Table 3-1. Mixed packet copy distribution $P_1(k)$ and its approximating Binomial distribution $P_2(k)$ for specific values of p_s, p_f, p_o and n=8.

$p_s = 0.2$	$p_f = 0.6$	$p_o = 0.8$
k	$P_1(k)$	$P_2(k)$
0	0.00013	0.00001
1	0.00164	0.00028
2	0.00918	0.00309
3	0.03211	0.01957
4	0.08315	0.07748
5	0.17318	0.19629
6	0.27669	0.31079
7	0.28635	0.28119
8	0.13758	0.11130

$p_s = 0.2$	$p_f = 0.2$	$p_o = 0.3$
k	$P_1(k)$	$P_2(k)$
0	0.07967	0.07222
1	0.22523	0.22469
2	0.29590	0.30582
3	0.23266	0.23786
4	0.11808	0.11563
5	0.03918	0.03597
6	0.00823	0.00700
7	0.00100	0.00078
8	0.00005	0.00001

$p_s = 0.4$	$p_f = 0.2$	$p_o = 0.3$
k	$P_1(k)$	$P_2(k)$
0	0.10170	0.08992
1	0.25281	0.25275
2	0.29533	0.31081
3	0.21119	0.21841
4	0.10003	0.09592
5	0.03168	0.02696
6	0.00646	0.00474
7	0.00077	0.00048
8	0.00004	0.00002

$p_s = 0.4$	$p_f = 0.2$	$p_o = 0.25$
k	$P_1(k)$	$P_2(k)$
0	0.127177	0.123574
1	0.294398	0.295293
2	0.304318	0.308715
3	0.183305	0.184427
4	0.070260	0.068861
5	0.017513	0.016455
6	0.002766	0.002458
7	0.000252	0.000210
8	0.000010	0.000008

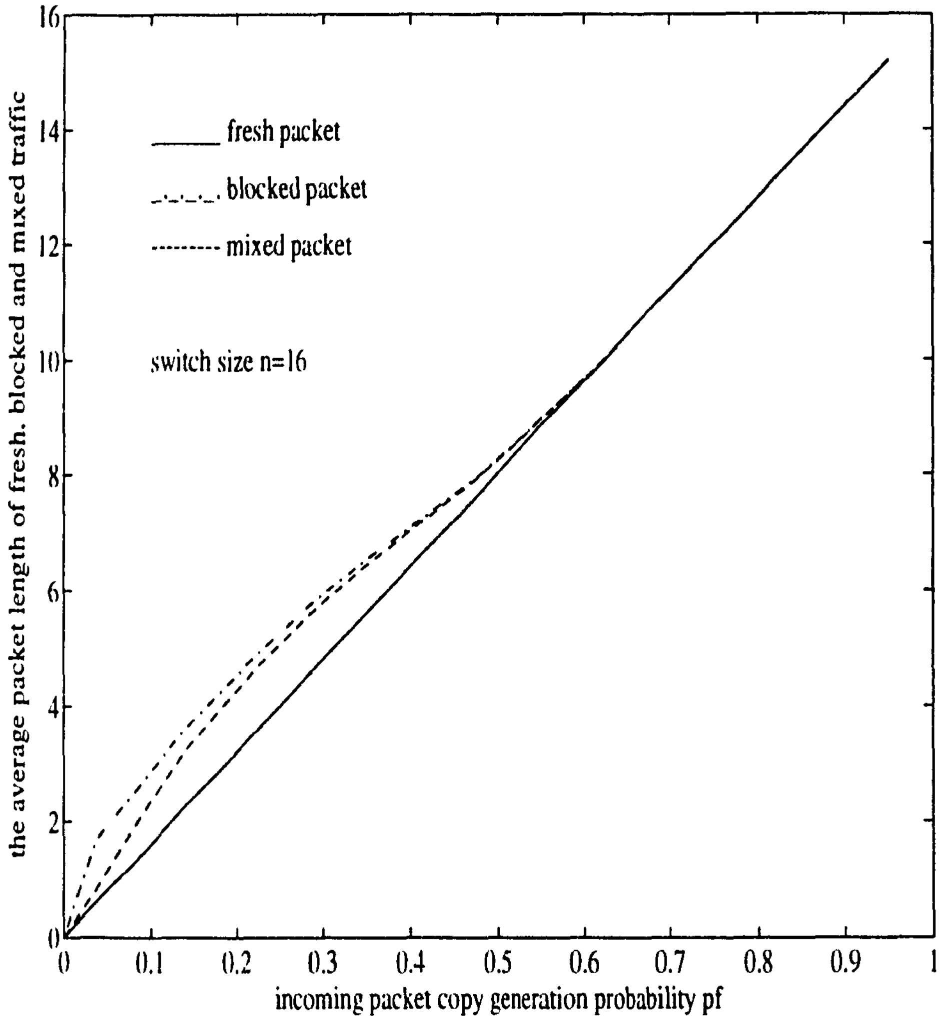
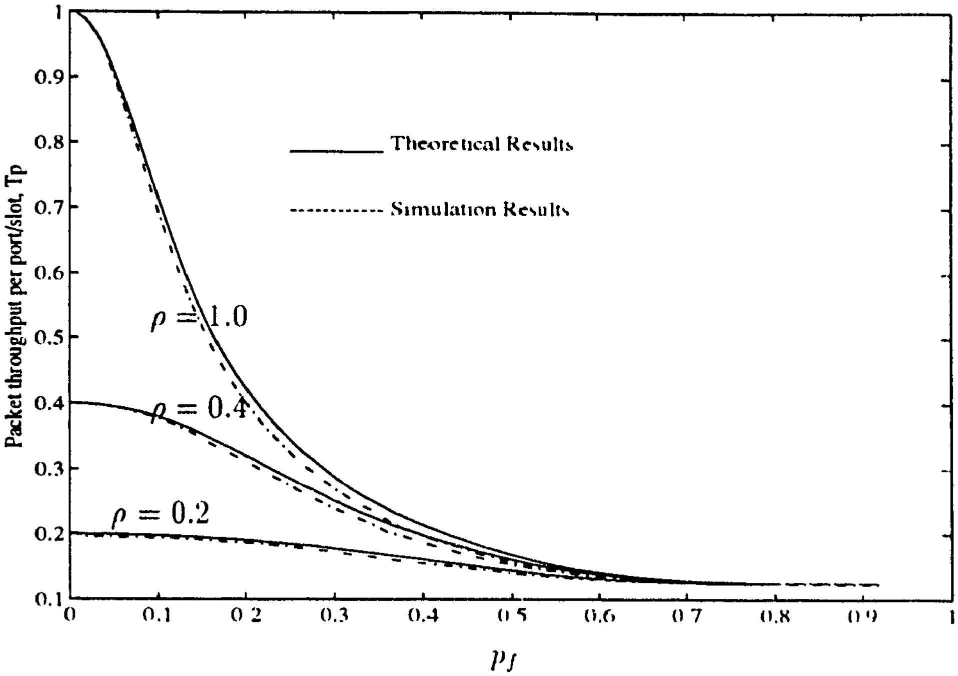
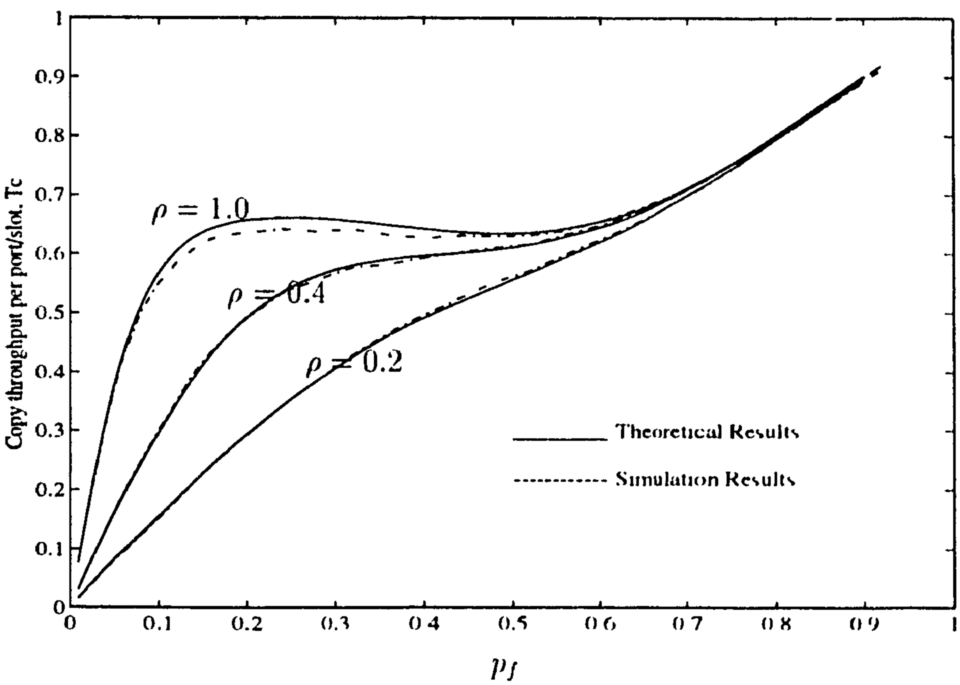


Fig. 3.1 The steady-state average packet lengths of three kinds of traffic (fresh, blocked and mixed packet) as a function of the incoming packet copy generation probability p_f, with load ρ=1 and switch size n = 16.

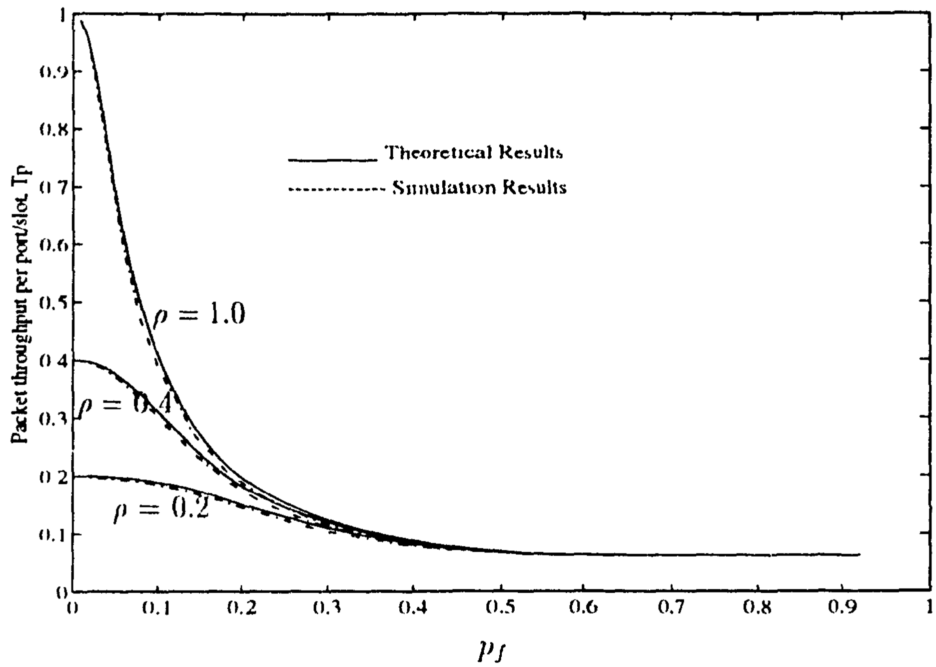


—a—

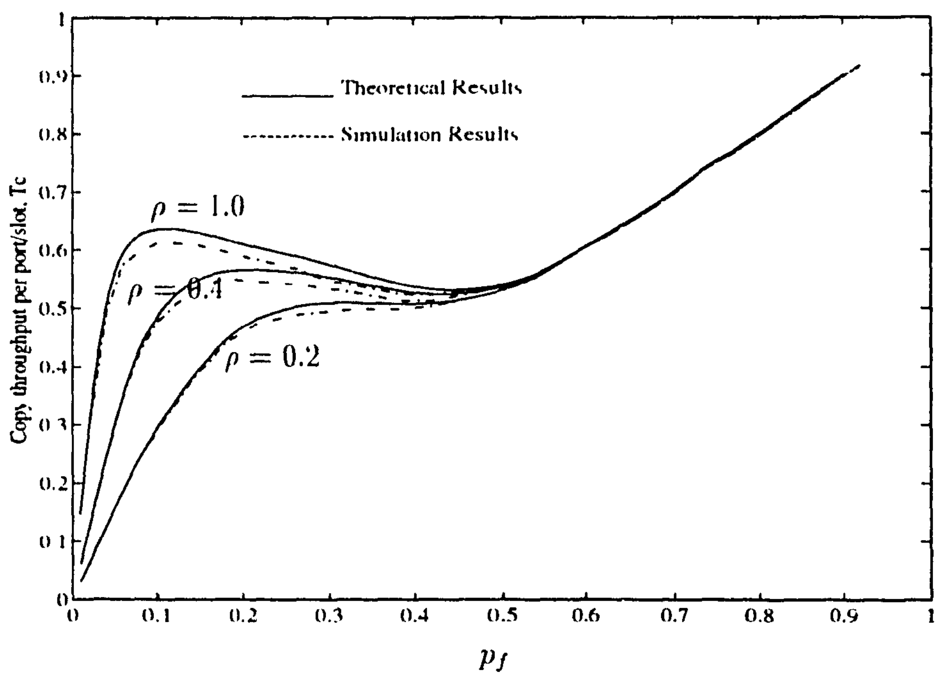


—b—

Fig. 3.2 The steady-state packet throughput and copy throughput per port/slot against fresh packet copy generation probability p_f, for different values of the load ρ, and switch size of n=8.

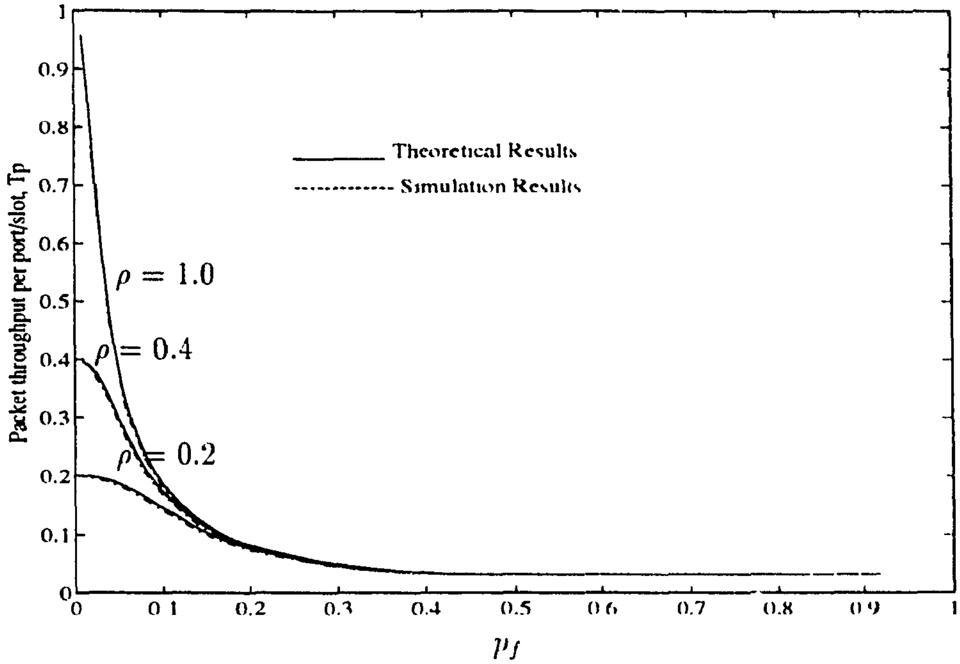


—*a*—

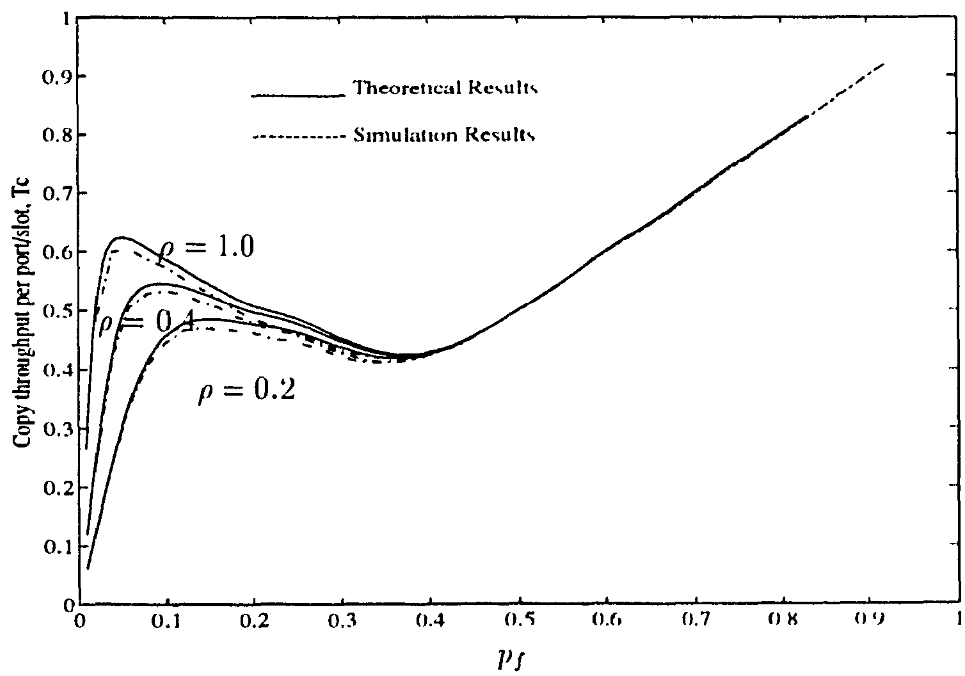


—*b*—

Fig. 3.3 The steady-state packet throughput and copy throughput per port/slot against fresh packet copy generation probability p_f, for different values of the load ρ, and switch size of n=16.



—a—



—b—

Fig. 3.4 The steady-state packet throughput and copy throughput per port/slot against fresh packet copy generation probability p_f, for different values of the load ρ, and switch size of n=32.

Chapter 4

Speedup Effect Analysis under a Modified Random Packet Selection(MRPS) Policy

In this chapter, the throughput performance of a multicast switch with speed up operation is studied under a MRPS policy. The MRPS policy, is a variation of the original random packet selection policy discussed in chapter 2, and it provides an important approach in the study of speed up operation of a multicast switch. This chapter is organized as follows: In the next section, we start by describing the MRPS policy and its operation. In section 4.2, we derive the distribution of the number of packets being successfully transmitted for different speed up rates. The steady-state throughput performance will be studied in section 4.3 and some numerical results are presented in section 4.4. In the study of speed up operation, we still assume that all the copies of a selected packet must be served in the same time slot.

For the sake of comparison, we here give a brief review of the effect of speed up operation in a unicast switch.

In an input queued unicast switch, the Head-Of-Line(HOL) packets contend for the output in each time slot. Only one among the conflicting HOL requests(namely those packets with the same output address) is served during a slot. The rest are blocked and queued at the input, where they may retry in the next slot. This kind of blockage, termed HOL blocking, limits the throughput of an input queued packet switch to 58.6%. This upper bound, known also as the saturation throughput, was derived under the assumption that the inputs are saturated. To reduce HOL blocking, several solutions are proposed, one approach is to increase the internal speed of the switch fabric relative to the speed of input/output lines. Consequently the slots may have more than 58.6% occupancy, and queueing is shifted from the inputs to the outputs. In [18], the maximum throughput of a unicast switch with speed up operation has been studied. It is assumed that the switch fabric operates L times faster than the input/output lines. Thus L HOL packets conflicting with each others may be switched to the common output in the same slot. Under the assumption of an infinite number of buffers at the inputs as well as at the outputs, the maximum throughput has been shown to be 0.8845 and 0.9755 for $L=2$ and $L=3$, respectively. This constitutes a significant improvement over the maximum throughput of switch without speedup(0.586).

The speed up operation is more important for multicast switching than for unicast switching, as a result of copy generation. In a multicast switch, the load presented to the switch fabric is multiplied by the packet fanout. This makes the switching fabric a severe bottleneck and limits the higher throughput achievable. Though a fanout splitting discipline provides a higher throughput than the no fanout splitting case considered here, it is still totally inadequate to solve the problem. Therefore, it comes clearly that speedup of the switch operation will become a good solution in both service cases.

The scheme by which one can evaluate the effect of speedup on the throughput performance in a multicast switch is different from that in an unicast switch, due to the stochastic copy generation process at the outputs from each input packet. The original Random Packet Selection(RPS) policy and its corresponding analysis given in chapter 2 is not appropriate for studying the speed up operation in a multicast switch. In the following, a different implementation of the RPS policy, known as the Modified Random Packet Selection(MRPS) policy is considered. This particular policy provides us an important approach to characterize ' output contention of the switch which is the main issue in the study of speed up operation. In our analysis, MRPS policy is modelled as a discrete time birth process. This modeling enables us to derive a difference equation, which describes the dynamics of packets selection in the system with speed up operation. The solution of the equation will give the distribution of the number of packets being chosen during a slot, for the different speed up rate L. From there, we will be able to study the steady-state throughput performance by applying the results from chapter 3.

The following will be the subjects of our discussion in the remaining of this chapter:

· The description of MRPS policy

· A *birth process* modeling of MRPS policy and the derivation of distribution of number of packets chosen

· Steady-state throughput

· Numerical results on speed up operation

4.1 Modified Random Packet Selection(MRPS) Policy

The MRPS policy describes the same protocol as of the previous RPS policy, but in a different way. With a given number of contending input packets at the beginning of a slot, the MRPS policy will perform as many selection rounds(or selection trials) as the number of packets. At each trial, a contending packet is randomly picked up and examined to be chosen or discarded, depending on whether it is interfering with any of the packets chosen in the earlier trials. A chosen packet will be transmitted during the slot. In the first trial of the selection, a packet is picked up randomly from among total α contending packets and it is kept as the first chosen packet. In the second trial, another packet is picked up randomly among the remaining contending packets, however this packet is kept only if it does not interfere with the first chosen packet and is otherwise discarded. Similarly, in the every following trials, a picked up packet is kept as a chosen packet if it does not interfere with the previous chosen packets. The process continues until the contending packets are exhausted. As a result, all the chosen packets form a subset of the contending input packets, in which their copies destined to the outputs have no conflict with each others. These chosen packets therefore will be successfully transmitted in that slot.

As we can see, this MRPS policy differs from the RPS policy in a way the interfering packets are discarded: The MRPS policy discards the interfering packets one at a time only if the picked contending packet is not successful; while the RPS policy discards interfering contending packets cumulatively after each choice. In practice, the RPS policy may be preferred for its implementation efficiency. Fig. 4.1 illustrates a simple example of the packet selections under the MRPS policy (with the number of contending input packets $\alpha=4$):

Where we assume that four contending packets named (a), (b), (c) and (d) are distributed at the inputs of a 8x8 multicast switch, as shown in the previous figure 2.1, where each packet is indicated by the set of copies it generates. The copies of a packet are arranged in a sequence of squares from left to right that corresponds to the outputs from port 1 to port 8. The digit "1"/"0" in a square indicates whether there is/is not a copy addressed to the corresponding output port. Next, as we know the MRPS will consist of four trials, we assume that a random sequence of picked

A random selection sequence:	Ouput ports 1 to 8 destined by copies of each packet : 8	7	6	5	4	3	2	1	Trial number	Selection results :
Packet (b)	0	1	1	0	0	0	1	0	1	success
Packet (d)	0	0	0	1	0	0	0	0	2	success
Packet (c)	1	0	0	0	1	1	0	0	3	success
Packet (a)	0	1	1	0	0	0	1	1	4	fail

Figure 4.1: An Example of Packet Selection Process under MRPS Policy (for α=4)

up packets in these trials(1, 2, 3, 4) are packet (b), (d), (c), and (a) respectively. In which, the first trial is always a success and the packet "b" becomes the chosen packet. Also the second trial is a success because of non interference between the packet "d" and "b". Similarly the third trial is also a success trial since the picked up packet "c" for the trial does not interfere with both packets "b" and "d". Finally, the fourth trial fails as the packet "a" interferes with packet "b" on the output ports 2, 6 and 7, and therefore is discarded.

In the next section, we apply this MRPS policy to carry out a derivation for the distribution of number of chosen packets during a slot.

4.2 Speedup Effect on the Distribution of the Number of Chosen Packets

4.2.1 Modeling and Definitions

In this section, we will adopt the same assumptions as those in chapter 2, namely we consider an nxn nonblocking multicast switch, where the total number of contending packets(HOL packets) at the inputs during a particular slot is a constant number, α, where $0 < \alpha \leq n$. A HOL input packet will generate a copy to each of the outputs with probability p, according to an independent *Bernoulli* process. As mentioned before, this results in a *binomial* distributed number of copies for each packet. Again we mention that the assumption of *binomial* distribution of copy generation is for the simplicity in the analysis which tends to illustrate the approach used.

We also assume that switch fabric is operating L times faster than the input/output lines(where $1 \leq L \leq n$), and thus an output can receive up to L copies during a slot. We assume an infinite number of buffers are placed at the inputs as well as at the outputs.

Some further notation is defined in addition to those in chapters 2 and 3:

$\check{P}_j$: probability that a packet will not interfere with the first j chosen packets.

$P_j(i)$: probability that j packets will be chosen by the end of the ith trial.

$P_j(z)$: PGF of the probability distribution of $P_j(i)$.

$\check{W}_j$: the steady-state probability distribution of the number of chosen packets during a slot.

4.2.2 A Discrete Time Birth Process Modelling for the MRPS Policy

As explained before in the MRPS policy, we perform as many *Bernoulli* trials as the number of contending packets(α) in the inputs, and in each trial at most one packet is chosen. Thus the total number of packets being chosen during a slot depends on the dynamics of packet selection among α trials. We will show in the below, that this MRPS policy can be modeled as a discrete time birth process, and a difference equation derived from it will be able to describe the dynamics of the system.

In each *Bernoulli* trial, a contending packet is picked up and the packet is chosen only if it does not interfere with any of the previously chosen packets, otherwise it is discarded. Clearly, the success of a *Bernoulli* trial depends on the number of packets chosen in the previous trials. We let $\check{P}_j$ denotes the success probability and $(1 - \check{P}_j)$ denotes the probability of failure (the subscript j indicates there are j chosen packets before the present trial). Also, we have $P_j(i)$ which denotes the probability of the number of packets being chosen by the end of the ith trial.

$P_j(i)$=Pr{j packets will be chosen by the end of the ith trial}

To illustrate the dependence of packet selections on $\check{P}_j$ or $(1 - \check{P}_j)$, we give an example which shows the event $P_4(6)$ that "4 packets will be chosen by the end of the $6th$ trial" in Fig. 4.2: The horizontal axis gives the trial numbers 1, 2, ..., 6. The letter "S" or "F" above each trial number indicates whether that trial is successful or not. The first row underneath the horizontal axis shows the probability of success in each trial and the second row shows the number of packets chosen by the end of that trial. From Fig. 4.2, a picked up packet in the 1^{st} trial is always the chosen packet with probability $\check{P}_0$=1, and $P_1(1)$ indicates one packet is chosen by the end of the first trial; In the 2^{nd} trial, the picked up packet fails to be chosen with probability (1-$\check{P}_1$), then $P_1(2)$ denotes the number of chosen packets remains one at the end of the first two trials. The following 3^{rd} and 4^{th} trials are both successful with probabilities $\check{P}_1$ and $\check{P}_2$ respectively, as the numbers of the chosen packets before the 3^{rd} trial and the 4^{th} trial are 1 and 2 accordingly. And $P_2(3)$ and $P_3(4)$ each indicates the number of packets being chosen by the end of the 3^{rd} and the 4^{th} trials. After that, the 5^{th} trial fails($P_3(5)$) again and the 6^{th} trial succeeds($P_4(6)$) with the corresponding probabilities shown. As we can see, $\check{P}_i$ varies after each successful trial.

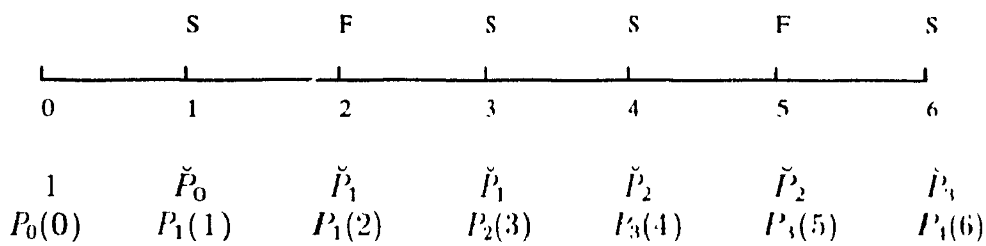


Figure 4.2: An Illustration of Choosing j Packets by the End of the ith Trial with j=4, i=6

From the above illustration, a discrete time *birth process* can be used to model the number of packets chosen during a slot in the way that the trial number is taken as the discrete time and the birth as the successful choice of a packet. It is clear that the population size at any time corresponds to the number of packets chosen by the end of that trial. The birth coefficients are independent of time and are function of population size j, given by $\check{P}_j$ (j=1,2,..., α-1). Next we give a difference equation describing the dynamics of this system, namely:

$$P_j(i+1) = (1-\check{P}_j)P_j(i) + \check{P}_{j-1}P_{j-1}(i) \qquad 1 \le j \le i \tag{4.1}$$

This equation relates the state of the system at the end of $(i+1)$st trial to that of the ith trial. Choosing j packets by the end of $(i+1)$st trial may come from two mutually exclusive events. The first event is having j packets chosen by the end of the ith trial and the failure of choosing a packet at the $(i+1)$st trial. The second event is having $(j-1)$ packets at the end of the ith trial and the success of choosing a packet at the $(i+1)$st trial. The difference equation thus determines $P_j(i+1)$ as the sum of the probabilities of these two events, with zero initial condition:

$$P_j(0) = \begin{cases} 1 & j = 0 \\ 0 & j > 0 \end{cases} \tag{4.2}$$

In the difference equation (4.1), we first need to determine the birth coefficient $\check{P}_j$, the probability that a packet will not interfere with the first j chosen packets in the earlier trials, then we can solve for $P_j(i)$.

4.2.3 The Determination of Output Contention and the Distribution of the Number of Chosen Packets

As one can see, the birth coefficient $\check{P}_j$ actually reflects the degree of output contentions of the switch under speed up operation. In this section, we first present our derivation of $\check{P}_j$, j=1, 2, . . . , α-1. Then, with these birth coefficients, we will solve the difference equation for the distribution of the number of chosen packets in a particular slot.

We have assumed that the switching fabric is operating L times faster than the input/output lines, and a maximum number of L copies can be accepted by an output in each slot. As in the analysis of chapter 2, we keep track of the remaining number of available outputs instead of the occupied ones (the term "available outputs" here refers to the outputs which have less than L number of copies destined to them, and are available for the next selection). For this purpose, we define a random variable I_j, which indicates the availability status of an output after j successful trials(this corresponds to successful choices of j packets).

$$I_j = \begin{cases} 1 & \text{if an output remains available after } j \text{ successful trials} \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

Also, we modify the definition of "an interfering packet" from now on to denote a packet with any of its copies destined to the outputs which have been selected by the copies of L chosen packets in the earlier trials.

First, we look at the case of L=1: the PGF of distribution of number of outputs available after j successful trials, already derived in chapter 2, is given by:

$$\begin{aligned} R_j(z) &= [1-(1-p)^j(1-z)]^n \\ &= [1-(1-p)^j+z(1-p)^j]^n \quad (4.4) \\ & \quad (4.5) \end{aligned}$$

Since the above is the PGF of a *binomial* distribution, the inversion of it gives,

$$\begin{aligned} r_j(k) &= Pr\{k \text{ outputs available after } j \text{ successful choice}\} \\ &= C_n^k[(1-p)^j]^k[1-(1-p)^j]^{n-k} \quad (4.6) \\ & \quad (4.7) \end{aligned}$$

where C_n^k denotes the permutation of having k available outputs from the total number of n outputs.

This distribution indicates that:

$$I_j = \begin{cases} 1 & \text{with probability } (1-p)^j \\ 0 & \text{with probability } 1-(1-p)^j \end{cases} \quad (4.8)$$

The above shows that the available outputs after j successful trials are those not being selected by the copies of any j chosen packets.

This may be extended to the case of $L > 1$ as follows: the available outputs for the $(j+1)$th chosen packet are those having less than L number of copies destined to them after j successful trials. The status of an output after j successful trials may be given as:

$$I_j = \begin{cases} 1 & \text{if an output is selected less than } L \text{ times by the } j \text{ chosen packets} \\ 0 & \text{if an output is selected } L \text{ times by the } j \text{ chosen packets} \end{cases} \tag{4.9}$$

Next, we determine the probability distribution of I_j.

There are two cases to be considered:

1). $j < L$

$$I_j = \begin{cases} 1 & \text{with probability } 1 \\ 0 & \text{with probability } 0 \end{cases} \tag{4.10}$$

2). $j \geq L$

As each chosen packet generates a copy to each output with probability p, the number of copies generated to an output by the j chosen packets is *binomial* distributed.

Let:

A_m=Pr{m copies will be generated to an output by j chosen packets}

Let η_j denotes the sum of those probabilities for $0 \leq m < L$; ie:

$$\eta_j = \sum_{m=0}^{L-1} A_m = \sum_{m=0}^{L-1} C_j^m p^m (1-p)^{j-m} \tag{4.11}$$

thus we have:

$$I_j = \begin{cases} 1 & \text{with probability } \eta_j \\ 0 & \text{with probability } 1 - \eta_j \end{cases} \tag{4.12}$$

Hence, the distribution of the number of available outputs after the jth choice is given by:

$$\breve{r}_j(k) = \begin{cases} 0 & 0 \le k < n, j < L \\ 1 & k = n, j < L \\ C_n^k \eta_j^k (1 - \eta_j)^{n-k} & 0 \le k \le n, j \ge L \end{cases} \tag{4.13}$$

The PGF of $\breve{r}_j(k)$ is:

$$\breve{R}_j(z) = \begin{cases} z^n & j < L \\ [1 - \eta_j(1 - z)]^n & j \ge L \end{cases} \tag{4.14}$$

Since the jth chosen packet performs a *Bernoulli* trial on each of the available outputs after$(j-1)$ successful trials, the PGF of the number of copies in the jth chosen packet will be given by:

$$\breve{N}(z) = \breve{R}_{j-1}(z)|_{z=\tilde{X}(z)} \tag{4.15}$$

where $\tilde{X}(z)$ is the PGF of $\tilde{x}_i$, which as defined in chapter 2, is the outcome of a *Bernoulli* trial for copy generation of a packet to an output i. ie:

$$\tilde{x}_i = \begin{cases} 1 & \text{with probability p} \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

Then we have: $\tilde{X}_i(z) = \tilde{X}(z) = pz + 1 - p$.

and:

$$\breve{N}_j(z) = \begin{cases} \breve{R}_{j-1}(z)|_{z=\tilde{X}(z)} = z^n|_{z=pz+1-p} & j < L \\ \breve{R}_{j-1}(z)|_{z=\tilde{X}(z)} = [1 - \eta_{j-1}(1 - z)]^n|_{z=pz+1-p} & j \ge L \end{cases} \tag{4.17}$$

or:

$$\breve{N}_j(z) = \begin{cases} (pz + 1 - p)^n & j < L \\ [1 - \eta_{j-1} p(1 - z)]^n & j \geq L \end{cases} \tag{4.18}$$

the average number of copies in the jth chosen packet, $\bar{\breve{n}}_j$, therefore is given by:

$$\bar{\breve{n}}_j = \frac{d\breve{N}_j(z)}{dz}\Big|_{z=1} = \begin{cases} np & j < L \\ n \cdot \eta_{j-1} p & j \geq L \end{cases} \tag{4.19}$$

Next, we shall determine $\breve{P}_j$, the probability that a packet picked up for a trial following the jth chosen packet will not interfere with the j chosen packets. This is equivalent to the probability that a packet picked up generates no copies to any of those outputs already selected by L of the j chosen packets. As one can observe, the first L packets picked up for trial can not interfere with each other and they will be chosen, therefore $\breve{P}_j$=1 for $j < L$. For $j \geq L$, $\breve{P}_j$ will be determined as follows:

We first define:

$\breve{m}_j$ as the total number of outputs which has been selected by L of the first j chosen packets. These outputs are unavailable for the afterward selections. $\breve{m}_j$ can be obtained by subtracting $\breve{r}_j$ from the total number of output ports(n):

$$\breve{m}_j = n - \breve{r}_j, \tag{4.20}$$

Let $\breve{M}_j(z)$ be the PGF of $\breve{m}_j$, then we have:

$$\breve{M}_j(z) = E[z^{\breve{m}_j}] = E[z^{n-\breve{r}_j}] = z^n E[z^{-\breve{r}_j}] = z^n \breve{R}_j(z^{-1}) \tag{4.21}$$

$$\breve{M}_j(z) = [z - \eta_j(z - 1)]^n \tag{4.22}$$

Further let $\breve{U}_j$ denote the number of copies generated to the m_j outputs by a packet picked up for the trial after the choice of j packets, and $\breve{U}_j(z)$ be the PGF of $\breve{U}_j$, then:

$$\breve{U}_j(z) = \breve{M}(z)|_{z=\hat{X}(z)}, \tag{4.23}$$

or

$$\breve{U}_j(z) = [pz + 1 - p - \eta_j(pz + 1 - p - 1)]^n \tag{4.24}$$

In particular, $\breve{U}_j(0)$ gives the probability that a packet picked up for trial generates no copies to m_j outputs, this is the birth coefficient $\breve{P}_j$ in the difference equation(4.1). Hence,

$$\breve{P}_j = \breve{U}_j(0) = \{1 - p[1 - \eta_j]\}^n \tag{4.25}$$

This $\breve{P}_j$, a factor which reflects the degree of output contentions of the switch under speed up operation, is plotted as a function of j for different values of speed up rate L in the numerical section.

With these birth coefficients $\breve{P}_j$ (j=1,2,...,α-1), we are now able to solve the difference equation given in (4.1) for the distribution of the number of chosen packets in a particular slot. The rest of this section presents the solution to it. The difference equation is rewritten here for reference:

$$P_j(i+1) = (1 - \breve{P}_j)P_j(i) + \breve{P}_{j-1}P_{j-1}(i) \tag{4.26}$$

Let:

$$P_j(z) = \sum_{i=0}^{\infty} P_j(i)z^i \tag{4.27}$$

Taking the PGF of the both sides of the above equation gives:

$$\sum_{i=0}^{\infty} P_j(i+1)z^i = (1-\breve{P}_j)\sum_{i=0}^{\infty} P_j(i)z^i + \breve{P}_{j-1}\sum_{i=0}^{\infty} P_{j-1}(i)z^i \tag{4.28}$$

Next, by applying the z-transform property

$$f_{n+1} \Leftrightarrow \frac{1}{z}[F(z) - f_0]$$

the above equation becomes:

$$\frac{1}{z}[P_j(z) - P_j(0)] = (1-\breve{P}_j)P_j(z) + \breve{P}_{j-1}P_{j-1}(z) \qquad j \geq 1 \tag{4.29}$$

by substituting for $P_j(0) = 0$ for $j \geq 1$ from Eq.(4.2), we get:

$$P_j(z) = \frac{z\breve{P}_{j-1}P_{j-1}(z)}{1 - z(1-\breve{P}_j)} \qquad j \geq 1 \tag{4.30}$$

This is a recursive formula for $P_j(z)$, which may be evaluated by using the initial distribution $P_0(i)$. $P_0(i)$ and hence its PGF $P_0(z)$ are determined from equations (4.1) and (4.2) as:

$$P_0(i) = \begin{cases} 1 & i = 0 \\ 0 & i \geq 1 \end{cases} \tag{4.31}$$

and

$$P_0(z) = \sum_{i=0}^{\infty} P_0(i)z^i = 1$$

Then by iterating the equation (4.30) with $P_0(z)$, we get:

$$P_j(z) = \frac{z^j}{\breve{P}_j}\prod_{k=1}^{j}\frac{\breve{P}_k}{[1-(1-\breve{P}_k)z]} \tag{4.32}$$

The product expression in the (4.32) may be replaced with a summation by applying the partial fraction expansion method[19], giving,

$$P_j(z) = \frac{z^j}{\breve{P}_j} \sum_{k=1}^{j} \frac{A_k}{1-(1-\breve{P}_k)z} \tag{4.33}$$

where

$$\begin{aligned} A_k &= (z-z_k) \prod_{k=1}^{j} \frac{\breve{P}_k}{[1-(1-\breve{P}_k)z]} \Big|_{z=z_k=\frac{1}{(1-\breve{P}_k)}} \\ &= \breve{P}_k \prod_{x=1 x\neq k}^{j} \frac{\breve{P}_x}{[1-(1-\breve{P}_x)z]} \Big|_{z=\frac{1}{(1-\breve{P}_k)}} \\ &= \breve{P}_k \prod_{x=1 x\neq k}^{j} \frac{(1-\breve{P}_k)}{1-\frac{\breve{P}_k}{\breve{P}_x}} \end{aligned} \tag{4.34}$$

(4.35)

Now equation (4.33) may be inverted with a transform property

$$\frac{A_k}{1-(1-\breve{P}_k)z} \Leftrightarrow A_k(1-\breve{P}_k)^i$$

giving:

$$P_j(i) = \frac{1}{\breve{P}_j} \sum_{k=1}^{j} \breve{P}_k(1-\breve{P}_k)^{i-j} \prod_{x=1 x\neq k}^{j} \frac{(1-\breve{P}_k)}{1-\frac{\breve{P}_k}{\breve{P}_x}} \qquad 1 \leq j \leq i \tag{4.36}$$

Since we have assumed a constant number of contending packets(α) per slot, substituting α for i gives,

$$P_j(\alpha) = \frac{1}{\breve{P}_j} \sum_{k=1}^{j} \breve{P}_k(1-\breve{P}_k)^{\alpha-j} \prod_{x=1 x\neq k}^{j} \frac{(1-\breve{P}_k)}{1-\frac{\breve{P}_k}{\breve{P}_x}} \qquad 1 \leq j \leq \alpha \tag{4.37}$$

This is the solution of the difference equation given in (4.1). $P_j(\alpha)$ gives the distribution of number of chosen packets during a slot for a given α contending packets at the inputs. For values of $\alpha \leq 16$, the computation is carried out in a straightforward manner, but beyond this limit some finite precision problems occur.

Next, from Eq.(2.28) the packet throughput per slot is given by:

$$\hat{T}_p = \frac{1}{n}\sum_{j=1}^{n} j \cdot P_j(\alpha) \tag{4.38}$$

and the copy throughput will be determined as follows: The total average number of copies generated by j chosen packets is given by $(\sum_{i=1}^{j} \breve{n}_i)$. Multiplying this by $\breve{P}_j(\alpha)$ and summing over possible values of j gives the overall average number of copies generated, which is $\sum_{j=1}^{n}(\sum_{i=1}^{j} \breve{n}_i) \cdot P_j(\alpha)$. Then the copy throughput per slot is given by:

$$\hat{T}_c = \frac{1}{n}\sum_{j=1}^{n}(\sum_{i=1}^{j} \breve{n}_i) \cdot P_j(\alpha) \tag{4.39}$$

substituting for $\breve{n}_i$ from Eq.(4.19), yields:

$$\begin{aligned} \hat{T}_c &= \frac{1}{n}[\sum_{j=1}^{L}(j \cdot np) + \sum_{j=L+1}^{n} \sum_{i=1}^{j}(n \cdot \eta_{i-1}p)]P_j(\alpha) \\ &= [\sum_{j=1}^{L}(j \cdot p) + \sum_{j=L+1}^{n} \sum_{i=1}^{j}(p \sum_{m=0}^{L-1} C_{i-1}^{m} p^{m}(1-p)^{i-1-m})] \cdot P_j(\alpha) \end{aligned} \tag{4.40}$$

4.3 The Steady-state Distribution of the Number of Chosen Packets and Throughput

The steady-state distribution of the number of chosen packets and the throughput are determined in the same way as the one was described in chapter 3. We let W_j denote the distribution of the number of chosen packets in the steady-state. The

Markov chain analysis in section 3.1 remains valid for determining $\breve{W}_j$. Then, replacing $P_j(\alpha)$ by $\breve{W}_j$ in Eq.(4.38) and Eq.(4.40) gives the packet and copy throughputs at the steady-state as:

$$T_p = \frac{1}{n}\sum_{j=1}^{n} j\breve{W}_j \tag{4.41}$$

and

$$T_c = [\sum_{j=1}^{L}(j \cdot p) + \sum_{j=L+1}^{n}\sum_{i=1}^{j}(p\sum_{m=0}^{L-1} C_{i-1}^{m} p^{m}(1-p)^{i-1-m})]\frac{\breve{W}_j}{1-\breve{W}_0} \tag{4.42}$$

However, as stated before, the switch's traffic in the steady state consists of fresh as well as old packets. T_p and T_c are function of mixed traffic parameter p, which is the copy generation probability of a mixed packet. To express T_p and T_c in terms of the new incoming traffic parameter p_f only, we follow the approach of section 3.2 and reapply the steady-state equilibrium equation which states that the average number of copies in a departing packet, μ_d, equals to the average number of copies in an incoming fresh packet, μ_f. We know that the average size of a fresh incoming packet is $\mu_f = np_f$, while for the average size of a departing packet, μ_d, we assume that j packets have been chosen during a slot, then the average number of copies in a departing packet is given by:

$$(\mu_d|j) = \frac{\sum_{i=1}^{j}\acute{n}_i}{j} \tag{4.43}$$

unconditioning μ_d with respective to j gives:

$$\mu_d = \sum_{j=1}^{n}(\frac{\sum_{i=1}^{j}\acute{n}_i}{j})\frac{\acute{W}_j}{1-\acute{W}_0} \tag{4.44}$$

by substituting for $\acute{n}_i$ from equation (4.19) and solving for $\mu_f = \mu_d$ we obtain

$$p_f = [\sum_{j=1}^{L} \frac{(j \cdot p)}{j} + \sum_{j=L+1}^{n} \frac{\sum_{i=1}^{j} (\eta_{i-1} p)}{j}] \frac{\breve{W}_j}{1 - \breve{W}_0} \quad (4.15)$$

With this relationship between p_f and p, the steady-state throughput can be plotted as a function of the incoming traffic p_f as shown in the Fig. 4.3 and Fig. 4.4.

4.4 Numerical Results

In this section we provide some numerical results for the purpose of illustration.

Fig. 4.1 shows the equivalence of the random packet selection policies(RPS and MRPS policies) by plotting the distribution of number of packets chosen in a particular slot for both policies for a constant number α of contending packets. (α=n=16).

Fig. 4.2 shows the probability $\breve{P}_j$ from equation (4.25), that a contending packet will not interfere with any of the outputs which were selected by L of the first j successful trials, as a function of j at p=0.1. Results for different values of speed up rate L (L=1,2,3,4,5) are plotted. As may be seen the probability $\breve{P}_j$, as a parameter which characterizes the degree of the output contention in the speed up operation, decays more slowly as L increases. This is due to the fact that the speed up operation of a switch reduces the output conflicts. In addition, the fact that an output can receive up to L copies during a slot is shown in the curves, where the $\breve{P}_j'$s remain at 1 until j exceeds L-1.

Fig. 4.3 ~ 4.4 present the packet and the copy throughputs at the steady-state for different values of speed up rate L, on switch sizes n=8, 16 respectively. Curves are plotted against fresh packet copy generation probability p_f, with input load $\rho = 1.0$. Also shown are the simulation results. As we can see, both copy and packet throughputs exhibit a significant improvement with increasing rate of speed up, but in a nonlinear fashion. Results also show a very good agreement between the theoretical study and the simulation verification(which is taken in the same environment as described in section 3.4.1, except we have redefined the output conflicts as the case when there are more than L packets destined to the same output).

In a multicast switch, the speed up operation shows a drastic improvement on the throughput performance, as it allows more than one HOL packets destined to the same output to be served in the same time slot and hence reduces the HOL blocking. Compared to the unicast switch, the reduction in the output conflicts plays a more important role than the decrease in the HOL blocking(packets behind the HOL packet can not access to the idle outputs), on the improvement of the throughput performance in a multicast switch with speed up operation under no fanout splitting service discipline. This can be seen from the curves shown in the Fig. 4.3 and Fig. 4.4, where T_p and T_c each increases nearly equally with each value of L.

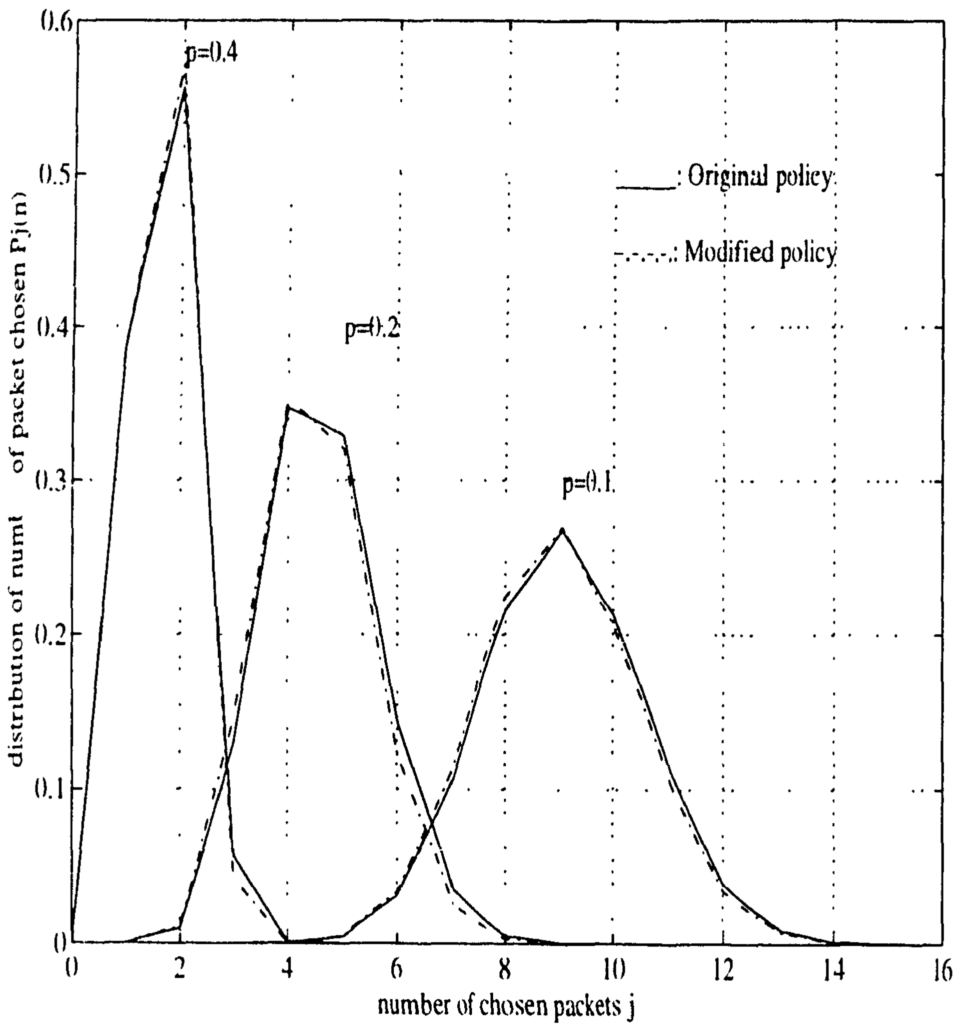


Fig. 4.1. Comparision of the results of distribution of number ot packet chosen under two random selection policies: Original policy and Modified policy, for different values of copy generation probability p=0.1, 0.2, 0.4 and switch size of n=16.

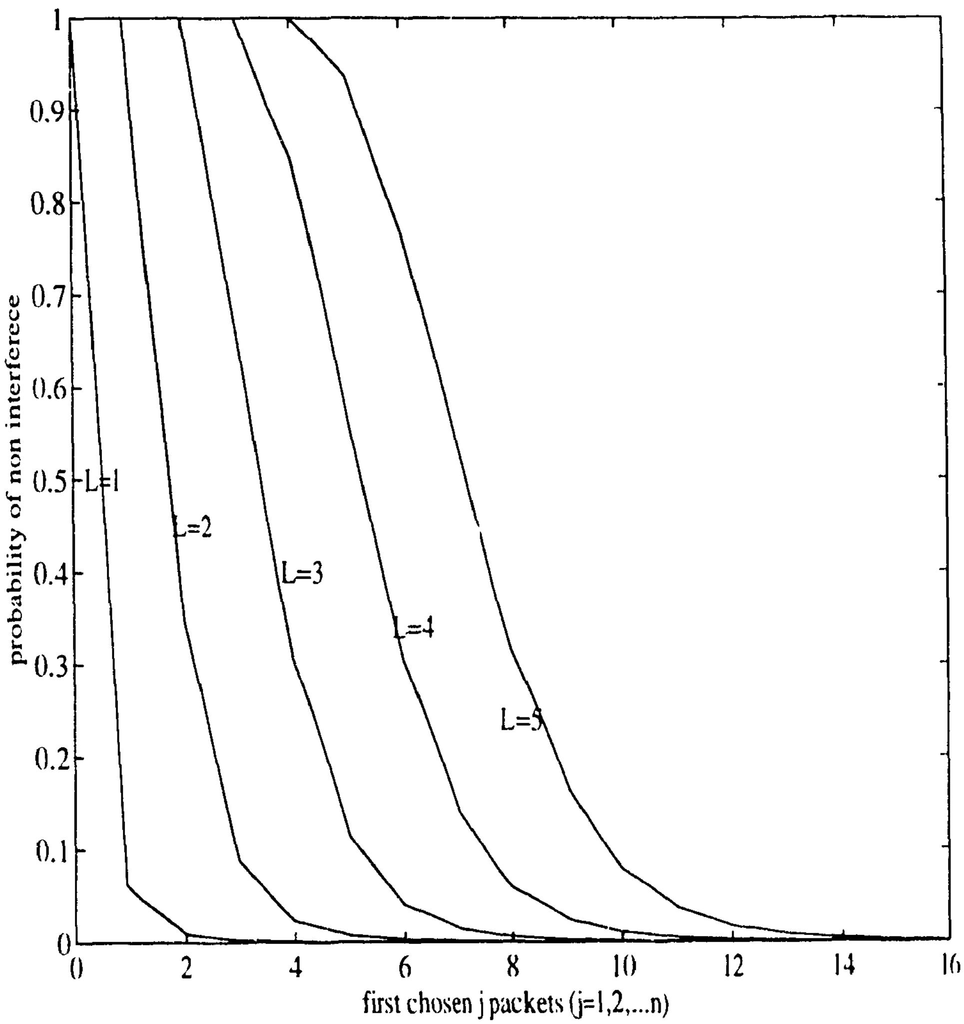
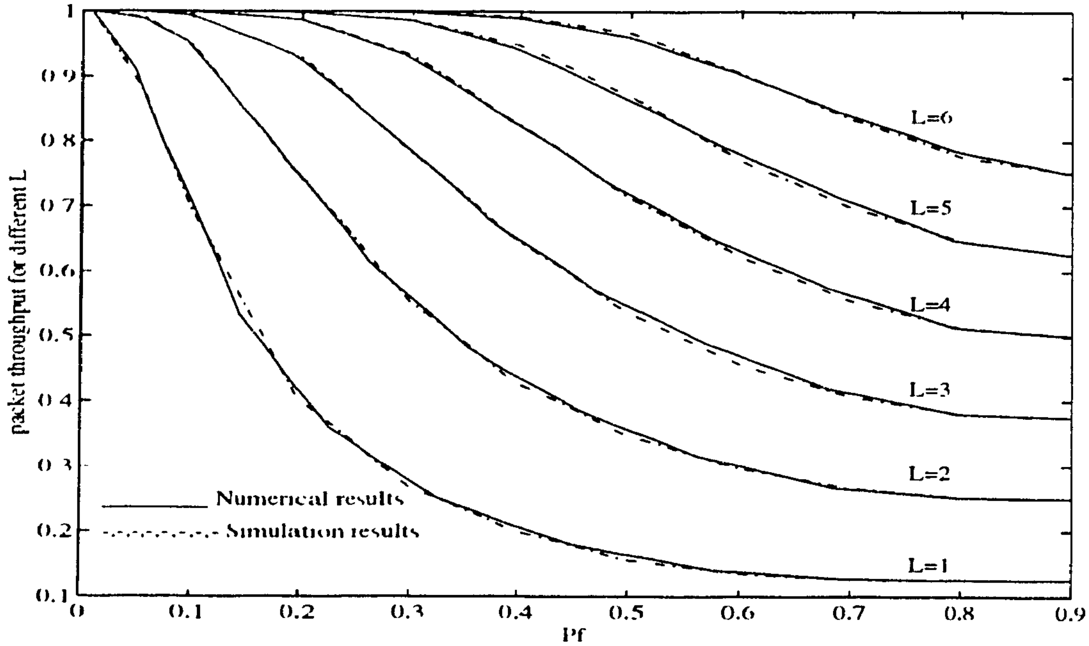
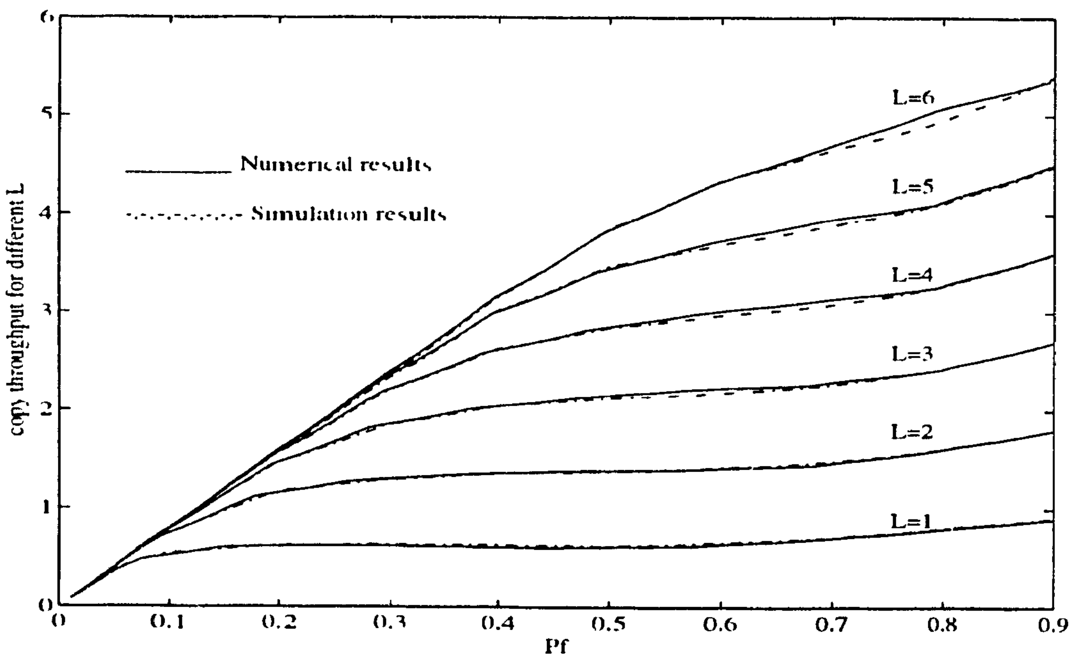


Fig. 4.2. Probability $\check{P}_j$ that a packet picked up for trial will not interfere with the first chosen j packets, with speedup rate L as a parameter copy generation probability, p=0.4, and switch size of n=16.

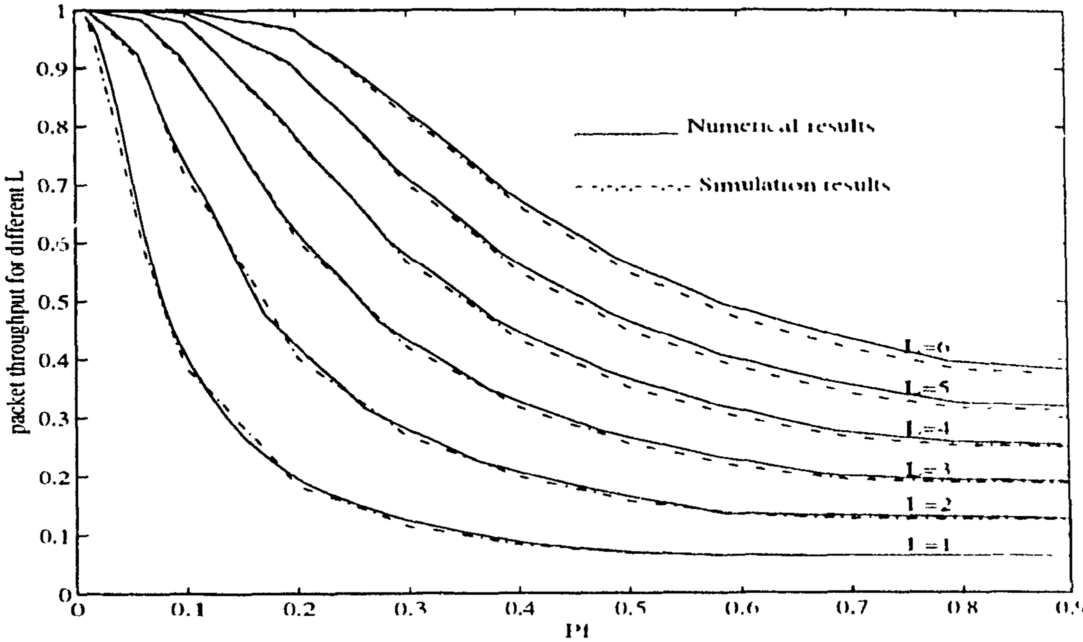


—a—

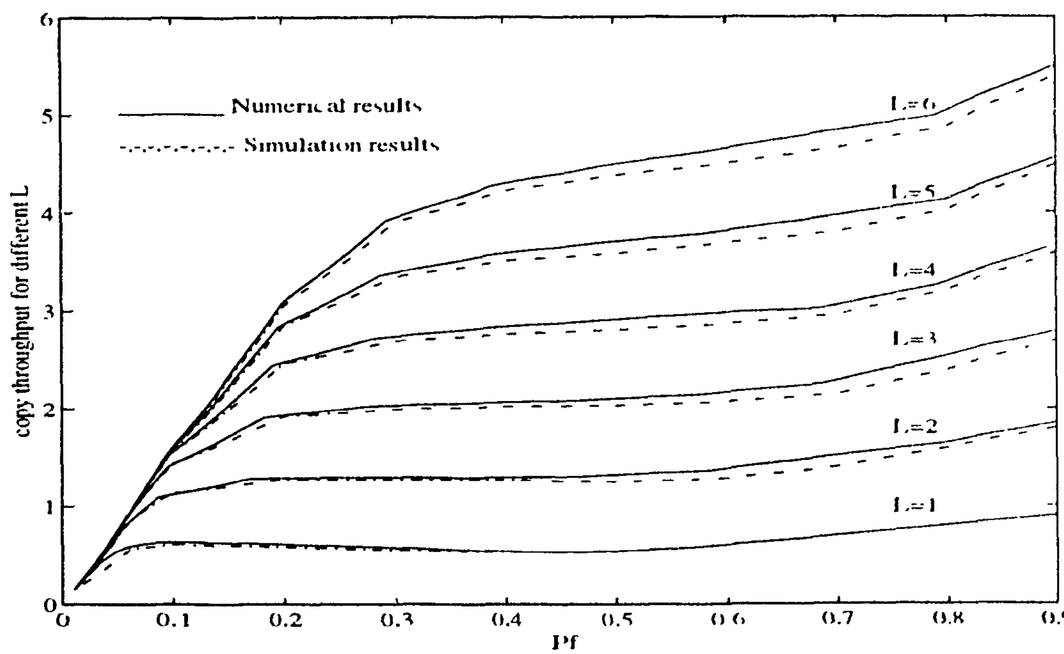


—b—

Fig. 4.3 The steady-state packet throughput and copy throughput per port/slot against fresh packet copy generation probability p_f, for different values of speedup rate L, at load $\rho = 1.0$, and switch size of n=8.



—a—



—b—

Fig. 4.4 The steady-state packet throughput and copy throughput per port/slot against fresh packet copy generation probability p_f, for different values of speedup rate L, at load $\rho = 1.0$, and switch size of n=16.

Chapter 5

Conclusion

In this thesis we presented a stochastic model for the performance analysis of a multicast switch under a no fanout splitting service discipline. First, we have derived the probability distribution of the number of chosen packets during a time slot based on the assumption of a random packet selection policy and a constant number of contending packets at the inputs. Then the study were extended to the steady-state through a Markov Chain analysis. The probability distribution of the number of copies in a transmitted packet was also determined accordingly. The results has shown that a transmitted packet has a size smaller than the average packet and therefore smaller packets go through the system easier than the larger packets. During any time slot, the contending packets will consist of a mix of fresh and old packets(blocked packets) which have different copy generation probabilities. These copy generation probabilities are related through a steady-state equilibrium consideration, which is a major finding in the analysis. That enables us to express the packet and copy throughputs as a function of the incoming traffic only through the copy generation probability, in which the blocking effect has been embedded. Subsequently the best operating point for the system was found. As a result of the copy generation, the load presented to the switch will be multiplied by a fanout factor. This causes a severe bottleneck in the switch. As a solution, we proposed the speed up operation of the multicast switch. The analysis of this operation was based on a variation of the random packet selection policy. Under this new policy, a closed form solution to the distribution of the number of packets chosen during a slot was found, and an important parameter which characterizes the output contention of the speed up operation was also determined. It is shown that the speed up operation will overcome the bottleneck in the switch. All the numerical results obtained from the analysis were verified through simulations and they were shown to be accurate enough over the entire range of switch load.

The main conclusion in this thesis is that the multicast service discipline with no fanout splitting provides a good solution for the multicast switch in the following aspects: It needs no buffers between the copy and routing networks; and requires no additional control structure in the switch. No fanout splitting service switches all the copies of a packet in the same slot and this is significant for certain applications such as distributed processing that needs to preserve consistency of the global data. As we have seen from the numerical results presented in chapter 3, the throughput performance in a low fanout case compares well with that of the other services with fanout splitting. Moreover, it has been shown that the throughput improvement could be achieved through speed up operation which is easily implemented by this scheme.

As a suggestion for the future work, we may combine both unicast and multicast traffic into the same switch, providing the multicast with a higher priority. The switch operation with mixed unicast and multicast traffic may be expected to achieve a better output utilization and hence a better system throughput performance.

References

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

[2] J. S. Turner, *"Design of an Integrated Service Network"* IEEE J. Select. Areas Commun., Vol. SAC-4, pp. 1373-1380, Nov. 1986.

[3] T. T. Lee, *"Non-blocking Copy Network for Multicast Packet Switching"* IEEE J. Select. Areas Commun. Vol. SAC-6, pp. 1455-1467, Dec 1988.

[4] T. T. Lee, R. Boorstyn and E. Arthurs, *"The Architecture of a Multicast Broadband Packet Switch"* Proc. IEEE INFOCOM'88, New Orleans, LA, pp 1-8.

[5] J. S. Turner, *"Design of a Broadband Packet Switching Network"* Proc. IEEE INFOCOM'86, pp. 667-675.

[6] A. Huang, S. Knauer, *"Starlite: a wideband digital switch"*, Proc. of Globecom'84, pp.121-125, Atlanta, GA., Dec. 1984.

[7] K. E. Batcher, *"Sorting Networks and Their Applications"* AFIPS Proc. Spring Joint Comput. Conf., 1968.

[8] M. K. Mehmet-Ali, J. Fang, *"Performance Analysis of a Multicast switch"* Proc. 16th Biennial Symposium on Commun., Kingston, Ontario, pp. 238-241, May 1992.

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

[10] M. J. Karol, M. G. Hluchyj, S. P. Morgan, *"Input Versus Output Queueing on a Space-Devision Packet Switch"* IEEE Trans. on Commun., Vol. 35, No. 12, pp. 1347-1356, Dec. 1987.

[11] X. Chen, J. F. Hayes, M. K. Mehmet Ali, *"A One Shot Access Scheme for a Multicast Switch"* Proc. of CCECE'91, pp. 7.2.1-7.2.4, Quebec, Sept. 1991.

[12] J. F. Hayes, R. Breault and M. K. Mehmet-Ali, *"Performance Analysis of a Multicast Switch"* IEEE Trans. on Commun., Vol. 39, No. 4, pp 581-587, April 1991.

[13] J. Y. Hui, T. Renner, *"Queueing Strategies for Multicast Packet Switching"*, Proc. of GLOBECOM'90, pp.1431-1437, San Diego, CA, Dec. 1990

[14] X. Chen, *"Multicast Packet Switch"* Ph.D's Thesis. 1992.

[15] X. Chen, J. F. Hayes, M. K. Mehmet Ali, *"Performance Analysis of Cyclic-Priority Input Access Method for a Multicast Switch"*, Proc. IEEE Infocom'91, pp. 1189-1195, Miami, April 1991.

[16] B. Bingham, H. Bussey, *"Reservation-based contention resolution mechanism for Batcher-banyan packet switches"*, Electronics Letters, 23rd, Vol.24, No.13, pp.772-773, June 1988.

[17] W. Feller, *"An introduction to Probability Theory and its Applications"*, Vol.-1, Wiley, New York, 1971.

[18] , 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.

[19] , L. Kleinrock, *"Queueing Systems"*, Vol. 1: Theory, John Wiley and Sons, New York, 1975.