INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Optimal Buffer Allocation in ATM Switches by Effective Cell Loss

By

David S. Ahn

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

1998

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/28/98
Date

Chair of Examination Committee

1/29/98
Date

Executive Officer

Professor Tarek Saadawi

Professor Mitra Basu

Professor Srinibas Vemuru

Doctor Mokhtar Boukli

Supervisory Committee

The City University of New York

ABSTRACT

Optimal Buffer Allocation in ATM Switches by Effective Cell Loss

By

David S. Ahn

Advisor: Professor Myung J. Lee

One of the important issues in an ATM switch design is how to allocate the given buffer budget to ensure compliance with negotiated traffic contracts of each ATM connection. This problem owes mainly to the conflicting buffer requirements of different QoS parameters, such as cell loss, cell transfer delay, and cell delay variation. For example, in the event of network congestion, the cell loss due to buffer overflow can be reduced by adding more buffers to the connection. However the ensuing QoS guarantee for the connection may not be realized due to the excessive queueing delay accompanied by the increased buffer length. In this dissertation, an optimal buffer allocation in a generic nonblocking ATM switch is achieved for the required QoS guarantee of the connection by satisfying a single nodal QoS parameter called Effective Cell Loss (ECL). The ECL is an integral QoS parameter that properly incorporates both the cell loss due to buffer overflow and the cell loss due to excessive delay. Hence, an optimal buffer allocation is found by

minimizing ECL given the acceptable cell loss probability, the maximum cell delay tolerance, and the fixed buffer budget at an ATM switch. The immediate benefit of using ECL is that it is simple to quantify the conflicting buffer requirements of different QoS parameter in any network condition. One can simply choose a buffer allocation that satisfies the ECL requirement in order to guarantee all other QoS requirements. Some simulation results and numerical examples are presented to demonstrate the effective usage of ECL.

ACKNOWLEDGEMENT

I am deeply indebted to my academic advisor, Professor Myung J. Lee for sharing his insights, academic challenges, and brotherhood throughout my stay at City College. The completion of this dissertation wouldn't be possible without his guidance.

My sincere thanks to Professor Tarek Saddawi, Professor Ward Hindman and Professor Benjamin Liaw for their encouragement and financial assistance.

I would like to also thank the Electrical Engineering Faculty and the Examination Committee for their support.

Most of all, I would like to thank all of my family: my loveing parents, brothers and sisters, my son Michael, my daughter Michelle, and especially my wife, Jung-Eun. Without their constant prayer and sacrifice, I wouldn't be here today. May God bless all of you!

Finally, I would like to thank all of my friends and church members who unselfishly shared their love and encouragement.

This dissertation is dedicated to my loving wife, Jung-Eun.

Table of Contents

Figure Lists

Chapter 1 Introduction

Advent of high speed communication networks and a rapid advance in related technologies render many new exciting services and applications now available. Those services and applications, however, impose new challenges and certain requirements on the networks that have unmet before. A particular concern over a so-called Quality of Services (QoS) guarantee for various traffic including voice, video and real-time data with respect to an end-to-end user communication has been often addressed in recent literature. Evidently, future network should provide the flexibility to accommodate integrated traffic as well as the ability to incorporate rapid progress in technology.

As an ultimate solution to such needs, ITU-T (International Telecommunications Union - Telecommunication), formally known as CCITT (the International Consultative Committee for Telecommunications and Telegraphy), has adopted the Asynchronous Transfer Mode (ATM) technology for the Broadband Integrated Services Digital Network (B-ISDN) [1]. The name, ATM, is due to asynchronous multiplexing of cells at the edge of the network. ATM is also accepted as the technology to interconnect computers over ATM Local Area Networks (LAN) by the computer industry in the ATM Forum [2]. In brief, ATM is a technology that is introduced as a common protocol layer to support transport of multiple types of traffic in a broadband network in terms of a fixed-length (53-byte) packets called "cells" via packet switches along the connection-oriented path called "Virtual Circuit" (VC) [3, 4]. ATM, inherently a fast packet switching, also supports

connectionless-oriented services as found in the Internet [5]. Hence ATM is believed to have the advantages of both the circuit mode and the packet mode.

Motivation

ATM networks present a basic problem that networks are faced with the difficult task of satisfying the needs of connections requiring different QoS, but sharing the same physical resources, e.g., bandwidth and buffers. Different kinds of performance criteria should be considered simultaneously when ATM switching modes are designed.

One of the crucial concerns in ATM is whether it can support strictly delay constrained services of the future [6]. Cell delay in ATM can be characterized by propagation delay, fixed switching delay, packetization/depacketization delay and queueing delay at switches. All the delay parameters except the queueing delay are fixed, therefore, the maximum end-to-end cell delay would be derived by the sum of the maximum queueing delay at switches. Considering the time constraint imposed on real-time services, the queueing delay allowed at each ATM switch is small. In a typical ATM network scenario presented in [3], the maximum queueing delay allowed at each ATM switching node is less than a couple of milliseconds for voice service. In the worst case it could be as little as a couple of hundred microseconds, or less than 100 Cell Slot Times (CST:2.83 μsec cell transmission time at 150Mbps link). The cells experiencing delay beyond the time limit may become useless for end users. Consequently, at each node, a switch may be designed to handle those cells by two different methods: it discards them immediately

to ease downstream network resources, or it changes their priority bit and forwards to a next node. In the latter method, however, it still is not guaranteed that those forwarded cells reach their destinations in time set by the end users. The former method results in cell loss in addition to the cell loss due to buffer overflow [7]. Hence, the excessive queueing delays at switch nodes may jeopardize the whole development of ATM for delay constrained traffic.

In particular, such a problem is explicated by traffic characterized by nonuniform pattern. Two different types of traffic nonuniformities can be considered: spatial nonuniformity and temporal nonuniformity. The spatial nonuniformity is inherent in telecommunications and is determined by the difference at source, as affected by time, region and services. The temporal nonuniformity depends on the characteristics of cells , such as the cell arrival rate and the output port address embedded in the cell header. The time scale of the traffic nonuniformity is also of concern. For instance, the real-time traffic with a high peak rate and a long burst length can cause the short term nonuniformity. Output ports connected to destinations like a popular database attract more traffic than other output ports, and it may cause a semi-permanent nonuniform traffic pattern in a switch. Most switch designs are, however, based on the performance estimates obtained under the uniform traffic assumption [8]. Such estimates tend to be optimistic since traffic nonuniformity could result in increased switch congestion [9–13]. Accordingly, the performance prediction based on nonuniform traffic should make an integral part of

the switch analysis and design. Increasing buffer sizes at a switch might be a remedy for detrimental effects of nonuniform traffic pattern (i.e., excessive cell loss) [14], however, the benefit of adding buffers may be hindered by an excessive queueing delay associated with the increased buffer space.

Those concerns together call for an adequate performance measure that properly incorporates both the cell loss due to buffer overflow (LO) and the cell loss due to excessive queueing delay (LD). In this proposal, we introduce a quantitative performance measure called Effective Cell Loss (ECL) probability that provides the comprehensive cell loss probability of an ATM switch. This ECL requirement for each ATM switch can be set during a call-setup period, based on QoS requirement of traffic and current network status.

Furthermore, a fixed buffer budget is assigned at each switch. Buffers may be placed at the inputs, the outputs, or within the switching fabric, depending on the switch architecture. A contention among cells that destined to the same output address, a bursty arrival of cells at a particular port, or any form of nonuniform traffic pattern may cause sudden build-up of the queue. In order to quantify the performance of ATM switches, a proper analysis of switch with a finite buffer size at inputs and outputs is quintessential among many others. Many conventional queueing analyses of ATM switches have been done for the case with an infinite buffer size assumed for each input and output [10–16]. We present in Chapter 2, a performance analysis of ATM switch with a finite and an

arbitray buffer size allocated for each input and output is studied using matrix geometric solution technique [17].

The contention and the bursty arrival of cells may result in asymmetric use of available buffer in the switch, causing undesired performance degradation as well as inefficient use of system resources [18]. Evidently, some sorts of buffer sharings and controls among ports and resource management are anticipated at switches to improve buffer efficiency and enhance switch performance. Numerous buffer sharing schemes and control policies have been studied. In particular, Kamoun and Kleinrock [19] studied several buffer sharing schemes with an assumption of independent Poisson arrivals and exponential service times for uniformly distributed traffic. For instance, complete sharing in which an arriving packet is accepted if any buffer space is available, complete partitioning in which the entire buffer is permanently partitioned among the output ports, sharing with maximum queue lengths in which a limit on the number of buffers allocated to each output port is imposed for fairness, sharing with minimum allocation in which a minimum number of buffers is always reserved for each output port and the remaining buffers are shared between all output ports, and sharing with a maximum queue and minimum allocation which is a combination of previous two shemes. They obtained closed form expressions for the probability distribution of the buffer occupancy based on the fact that has a well-known product form solution.

Several buffer controlling shemes have been also proposed and studied to obtain the

existence and the structure of an optimal buffer sharing policy in the sense of minimum packet loss or maximum throughput. Foschini and Gopinath [20] first considered optimality within coordinate-convex policy class. This policy does not allow any pushout for the packets already admitted in the buffer. They proved for two ports case that the optimal coordinate-convex policy is to limit the queue length of output port to some fixed level m, such that $m1+m2 \geq B$, where B is the buffer size. Wei et al, [21] suggested a sharing policy that allows dropping of packets on demand basis, even if packets had been admitted in the buffer. This is called pushout policy which includes coordinate-convex policies as its subclasses. It is shown that pushout policy yields better throughput and lower packet losses than either the complete sharing or the complete partition policies. Cidon et al, [22] proved that, for a two-ported switch, the optimal policy is of pushout with threshold type in which the arrival is accepted whenever buffer is nonfull. Whenever it is full, an arrival from a preferred type is accepted by pushing out unpreferred type if the preferred type packets is below some threshold. It is worth to mention that they report an interesting and somewhat unexpected phenomenon. That is, while the overall improvement in loss probability of the optimal pushout with threshold policy over the optimal coordinate-convex policy is found to be relatively minor, a significant difference is observed when focusing on the loss probability of an individual output port. This finding alines with the basic concept of our buffer sharing algorithm presented in Chapter 3. Tassiulas et al, [23] compared three different classes of buffering policies: discarding

policy, pushout policy, and expelling policy. They showed that the expelling policy in which cells are droped or blocked irrespective of the system state such as threshold of each class is optimal.

As evidently seen from the above, most of studies have considered a single queue with different classes of traffic for different levels of priorities to be handled. However, an ATM switch is consisted of multi queues as to connect the multiple inputs to the multiple outputs. Consequently, finding an optimal buffer sharing and controlling schemes is not a simple matter. In Chapter 3, we attempt to find the optimal buffer allocation between inputs and outputs for an NxN nonblocking ATM switch derived by the ECL analysis. For example, the convex point on the graph that plots ECL performance with respect to different sized pair of inputs and outputs would provide the optimal buffer allocation to guarantee both cell loss and delay requirements of the user traffic. Under nonuniform traffic patter, we found that the output arrival intensity called "output pressure" felt by each output buffer may be a vital factor to optimize the usage of available buffers at output while improving system performance substantially. As a result, we present an algorithm that achieves the optimal buffer sharing at outputs.

The rest of the thesis is organized as follows: In Chapter 2, we describe system model we consider and analyze the performance of an input and output queueing NxN nonblocking ATM switch in terms of cell loss and cell delay which later intergrally incoporated into the unified performance measure, effective cell loss. We consider both

uniform and nonuniform traffic to the switch and present some extensive numerical examples depicting tradeoff among different system parameters. In Chapter 3, we propose and study a single nodal QoS parameter that properly integrate cell loss and cell delay at an ATM switch. By presenting numerical studies, we demonstrate the effect of ECL on choosing system parameters, such as speed-up factors and buffer sizes at input and output of the switch. In Chapter 4, we imply analytical results obtained in Chapter 2 and 3 to determine optimal buffer dimensioning. The system performance of a NxN noblocking ATM switch with input and output queueing is mostly affected by the output buffer performance which naturally guide out intuition to devise an optimal buffer sharing algorithm that provides an substantial improvement in cell loss performance. In Chapter 5, we conclude our study and discuss future improvement to be continued, especially on the foregoing research topic on the optimal buffer management of ATM switch.

Chapter 2 Performance Anaylsis of an ATM Switch

Section 2.1 Model Description

2.1.1 System Model

We consider a generic N x N nonblocking ATM switch shown in Figure 2.1. The buffer size at each input and output is arbitrary and denoted by K_i and L_j, for $1 \leq i, j \leq N$, respectively. The *First Come First Served* (FCFS) policy is assumed for queueing. Each input link is logically partitioned into time slots where each slot equals to one cell slot time (CST). The operation of the switch fabric is synchronized in time slots and no queue in the switch fabric is assumed. The switch has a speed-up factor (SF) equal to m, $1 \leq m \leq N$, where m denotes the number of cells that can be simultaneously delivered to each output port from multiple input ports. We adopt the *Queue Loss* transfer scheme within the switch fabric, which allows permanent cell loss for the cells arrived at outputs finding buffers full. Note that an ATM switch design based on the *Queue Loss* scheme is believed to be better than the *Back Pressure* scheme [24] because of its simple implementation for the same cell loss requirement.

Throughout the ananlysis, we focus our attention on the movement of a tagged cell which is arrived at a particular input i, and destined to a particular output j. We denote input and output by superscript $^\circ$ and $^\bullet$ respectively. For example, ϵ_i° and $\epsilon_j^\bullet$ denote the

cell loss probabilities occurred at the input i and the output j.

2.1.2 Traffic Model

We consider a general traffic: the arrival traffic intensities at individual inputs are nonuniform; the output address of each cell is independently and randomly assigned through a traffic routing probability matrix. The cell arrivals at each input queue are modelled by an independent Bernoulli process at slot units. Although this traffic model may not be an authentic representation of ATM traffic, Li [25] concluded from simulation results that the switch performance with a correlated input traffic is not much different from the one with the independent Bernoulli traffic arrival assumption.

We define the offered input traffic intensities to each input of the switch as a row vector Λ°,

$$\Lambda^{\circ} = (\lambda_1^{\circ}, \lambda_2^{\circ}, \cdots, \lambda_N^{\circ}) \tag{1}$$

where λ_i° represents the probability of cell arrival at the i–th input queue at each time slot.

The output address is assumed to be independently and randomly assigned through a traffic routing probability matrix $\mathbf{T}$

$$\mathbf{T} = [\, t_{ij} \,], \tag{2}$$

where $t_{i,j}$ is the probability for a cell at the input i to be routed to the output j with $\sum_{\forall j} t_{ij} = 1$.

With the *Queue Loss* scheme, the output traffic intensity is readily given by a row vector $\Lambda^{\bullet}$.

$$\mathbf{\Lambda^{\bullet}} = (\lambda_1^{\bullet}, \lambda_2^{\bullet}, \cdots, \lambda_N^{\bullet}) = \mathbf{\Lambda^{\circ}}\ \mathbf{diag}[\,1 - \epsilon_i^{\circ}]\mathbf{T} \tag{3}$$

where **diag**[$1 - \epsilon_i^{\circ}$] is a diagonal matrix with its ith diagonal element equal to $[1 - \epsilon_i^{\circ}]$ for $i = 1, 2, \ldots, N$. With the above notation the uniform traffic condition is summarized by

$$\lambda_i^{\circ} = \lambda \quad \text{and} \quad t_{i,j} = N^{-1}, \quad 1 \le i, j \le N. \tag{4}$$

2.1.3 Nonuniform Traffic Example

It is well understood from previous studies [9–14] that the traffic nonuniformity has an adverse effect on the switch performance. In numerical examples, we observe the effects of traffic nonuniformities on the cell loss performance of the switch with finite input and output queues for the speed-up factors greater than 1. In brief, we describe a simple bigroup model for numerical study, in which all inputs and outputs are divided into two statistically identical groups as shown in Figure 2.2. The sizes of the input groups are defined by N_1° and N_2° with arrival rate λ_1° and λ_2° respectively for $N_1^{\circ} + N_2^{\circ} = N$. Similarly, the sizes of the output groups are defined by $N_1^{\bullet}$ and $N_2^{\bullet}$ with arrival rate $\lambda_1^{\bullet}$ and $\lambda_2^{\bullet}$ respectively for $N_1^{\bullet} + N_2^{\bullet} = N$.

The arrival rates within each input group and output group are identical. The system offered load is then readily given by $\lambda = \frac{N_1^{\circ}}{N}\lambda_1^{\circ} + \frac{N_2^{\circ}}{N}\lambda_2^{\circ}$. The output group address

assignment is given by

$$T_g = \begin{bmatrix} \tau_{1,1} & \tau_{1,2} \\ \tau_{2,1} & \tau_{2,2} \end{bmatrix} \tag{5}$$

where $\tau_{i,j}$ is a output group addressing probability from an input group i to an output group j, for $i,j = 1,2$.

We define the input (output) nonuniformity by U° ($U^{\bullet}$), which is the ratio of input (output) traffic intensity given by $U^{\circ} = \frac{\lambda_1^{\circ}}{\lambda_2^{\circ}}$($U^{\bullet} = \frac{\lambda_1^{\bullet}}{\lambda_2^{\bullet}}$). We also define input(output) group size ratio as $d_i(d_o)$, where $d_i = \frac{N_1^{\circ}}{N_2^{\circ}}(d_o = \frac{N_1^{\bullet}}{N_2^{\bullet}})$. These traffic parameters, d_i, d_o, Tg, U°, and $U^{\bullet}$, completely describe the traffic nonuniformity of our bigroup example.

2.1.4 Output Contention Process

Consider a tagged cell newly joining HOL. Now it joins the contention with other cells in HOL for the transmission to output j. This contention results whenever there are more contending cells in HOL than the speed up m. The details of this process for m equal to one is thoroughly studied in [10, 13, 16]. In the case of $2 \leq m \leq N$ the switch fabric randomly selects up to m such cells to output j at each time slot. Thus the contention process to output j is characterized by

$$C_j' = (C_j - m)^+ + A_j \tag{6}$$

where the symbol $(\bullet)^+$ denotes the larger value of 0 or its arguments, and C_j', C_j and A_j are defined as follows:

C'_j : the number of cells destined to output j for the next time slot

C_j : the number of contending HOL cells destined to output j at the current time slot

A_j : the number of cells newly joining the contention in the current time slot, which will compete for the next time slot.

It has been proved that when N approaches infinity, the process A_j is characterized by a Poisson process at rate $\lambda_j^{\bullet}$[12, 16]. Accordingly, the output contention process C'_j forms an independent logical M/D/m queue, which is a Markov chain with its state represented by C_j. Let $a_{j,k} = \Pr(A_j = k)$ with

$$a_{j,k} = \frac{\left(\lambda_j^{\bullet}\right)^k \exp\left(-\lambda_j^{\bullet}\right)}{k!}, \text{ for } 0 \le k \le N \tag{7}$$

, where Pr(•) denotes the probability of (•). The state transition matrix for the arrival process A_j is given by

$$E_j^{\bullet} = \begin{bmatrix} a_{j,0} & a_{j,1} & a_{j,2} & a_{j,3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ a_{j,0} & a_{j,1} & a_{j,2} & a_{j,3} & \cdots \\ a_{j,0} & a_{j,1} & a_{j,2} & a_{j,3} & \cdots \\ 0 & a_{j,0} & a_{j,1} & a_{j,2} & \cdots \\ 0 & 0 & a_{j,0} & a_{j,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{8}$$

Also, let $p_{j,k}^{\bullet} = Pr\left(C_j = k\right)$, or in a row vector form $\mathbf{p}_j^{\bullet} = (p_{j,0}, p_{j,1}, p_{j,2}, \ldots)$, which can be derived from

$$\mathbf{p}_j^{\bullet}\left(\mathbf{I} - \mathbf{E}_j^{\bullet}\right) = 0 \tag{9}$$

with $\mathbf{p}_j^{\bullet}\mathbf{e} = 1$ where $\mathbf{e}$ is a unit column vector.

Consider the tagged cell destined to output j, which has just moved to the head of input queue i and found itself contending with k cells from other inputs. With the assumption that the contention process is already in equilibrium upon arrival of this tagged cell, the steady state probability of finding k such cells by the tagged cell will be $p^{\bullet}_{j,k}$. Thus, there are $k+1$ cells, including the tagged cell itself, contending for the output j in the next time slot. Based on the random selection policy, the probability of the tagged cell leaving the contention (or absorption probability) is $\delta^{\bullet}_{k,j} = 1 - \left[\frac{k+1-m}{k+1}\right]^{+}$ for $1 \le j \le N$, or a column vector form $\Delta^{\bullet}_{j} = \left(\delta^{\bullet}_{0,j}, \delta^{\bullet}_{1,j}, \delta^{\bullet}_{2,j}, \cdots\right)$.

In the context of the switch analysis, the cell service time is equivalent to the contention time for the tagged cell at HOL. This contention time can be characterized by the sojourn time in a transient Markov chain constructed from the stationary Markov chain $\mathbf{E}^{\bullet}_{j}$. The initial state of this transient Markov chain is assigned by the probability vector $\mathbf{p}^{\bullet}_{j}$, representing the current state of C_j when the tagged cell moves into HOL. Therefore the transition probability from state k to l is given by $\{[(k+1-m)/(k+1)]^{+}\}a_{j,l-k+1}$, where $a_{j,l-k+1}$ is the probability that the number of new arrivals destined to output j equals $l-k+1$.

Define $\mathbf{H}^{\bullet}_{j}$ as the state transition matrix of this transient Markov chain. Then $\mathbf{H}^{\bullet}_{j}$ is expressed by

$$\mathbf{H}^{\bullet}_{j} = \mathrm{diag}\left[\left\{\frac{k+1-m}{k+1}\right\}^{+}\right]\mathbf{E}^{\bullet}_{j}. \qquad (10)$$

Define $S^{\bullet}_{j}$ as an output contention time. The output contention time distribution of $S^{\bullet}_{j}$

has a discrete phase-type distribution represented by $\left(\mathrm{p}_j^{\bullet}, \mathrm{H}_j^{\bullet}\right)$ and is given by

$$s_j^{\bullet}(n) = \mathrm{p}_j^{\bullet}\left(\mathrm{H}_j^{\bullet}\right)^{n-1}\Delta_j^{\bullet} \quad \text{for } n = 1, 2, \ldots \tag{11}$$

where n represents CSTs spent at HOL until the transmission. The mean and the variance of $S_j^{\bullet}$ can be easily derived from the distributions obtained [12]. For example, the mean is given by

$$\overline{S_j^{\bullet}} = \mathrm{p}_j^{\bullet}\left(\mathrm{I} - \mathrm{H}_j^{\bullet}\right)^{-1}\mathrm{e}^{\mathrm{T}}. \tag{12}$$

where e^{T} is a transpose of the unit vector.

Section 2.2 Cell Loss Analysis

2.2.1 Input Queue Analysis

The input service time S_i° at input i can be identified as another phase-type distribution by the closure property of the phase-type distribution [Chap. 2, 17]. Since each cell at HOL maintains its originally assigned output address, its state transition matrix can be constructed as below.

$$\mathrm{H}^{\circ} = \begin{Bmatrix} \mathrm{H}_1^{\bullet} & 0 & \cdots & 0 \\ 0 & \mathrm{H}_2^{\bullet} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{H}_N^{\bullet} \end{Bmatrix} \tag{13}$$

The initial state of the input service time is assigned by a probability vector of vectors as

$$\mathrm{p}_i^{\circ} = \left(t_{i,1}\mathrm{p}_1^{\bullet}, t_{i,2}\mathrm{p}_2^{\bullet}, \cdots, t_{i,N}\mathrm{p}_N^{\bullet}\right). \tag{14}$$

The input service time s_i° is then characterized by $(\mathrm{p}_i^{\circ}, \mathrm{H}^{\circ})$. According to the above descriptions, the absorption of each individual output contention process is characterized

by $\mathrm{a} = vct\left[1 - \left(\frac{k+1-m}{k+1}\right)^{+}\right]$, which is a column probability vector with its k–th element equal to $\left[1 - \left(\frac{k+1-m}{k+1}\right)^{+}\right]$ for $k \geq 0$. Define

$$\Delta = \left(\underbrace{\mathrm{a}, \mathrm{a}, \mathrm{a}, \cdots, \mathrm{a}}_{N}\right)^{\mathrm{T}} \tag{15}$$

which gives the absorption probability column vector of each input service process. The initial state of the next cell service time, upon a current service time completion, is assigned by

$$\mathbf{S}_i = \Delta \mathbf{p}_i^{\circ}. \tag{16}$$

The service time at each input queue then has an independent distribution of phase-type, described by $(\mathbf{p}_i^{\circ}, \mathbf{H}^{\circ})$ at the i–th input. Since the cell interarrival time is geometric, each input queueing process is a Geom/PH/1/K process. It is a two-dimensional Markov chain process with its state constructed by levels and phases. Each level corresponds to a buffer size and each phase represents a service state [10]. Consider the transition from level l where all the phases on each level are superimposed. The transition from level l to itself consists of two possible events as shown in Figure 2.3. One is with no arrival and no departure in a slot so that only the service time phase transition occurs. This is represented by $(1 - \lambda_i^{\circ})\mathbf{H}^{\circ}$. Another event is with one cell arrival and one cell departure in the same slot, upon which the next cell service time is initiated. This is represented by $\lambda_i^{\circ}\mathbf{S}_i$. Define $\mathbf{A}_{i,1} = (1 - \lambda_i^{\circ})\mathbf{H}^{\circ} + \lambda_i^{\circ}\mathbf{S}_i$. Similarly, define $\mathbf{A}_{i,0} = (1 - \lambda_i^{\circ})\mathbf{S}_i$ and

$\mathbf{A}_{i,2}= \lambda_i^{\circ}\mathbf{H}^{\circ}$ as the transition matrices from the level l to the level l-1 and from the level l to the level l+1 respectively. This model then has the structure of a quasi-birth-death type with its transition matrix given by the following block-partitioned form [Chap.3, 17]:

$$\mathbf{P}_i = \begin{pmatrix} 1-\lambda_i^{\circ} & \lambda_i^{\circ}\mathrm{p}^{\circ}{}_i & 0 & \cdots & 0 \\ (1-\lambda_i^{\circ})\boldsymbol{\Delta} & \mathbf{A}_{i,1} & \mathbf{A}_{i,2} & \cdots & 0 \\ 0 & \mathbf{A}_{i,0} & \mathbf{A}_{i,1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \mathbf{A}_{i,0} & \mathbf{A}_{i,1} & \mathbf{A}_{i,2} \\ 0 & 0 & 0 & \mathbf{A}_{i,0} & \lambda_i^{\circ}\mathbf{S}_i+\mathbf{H}^{\circ} \end{pmatrix} \tag{17}$$

Let $\boldsymbol{\Pi}_i = (\Pi_{i0}, \boldsymbol{\Pi}_{i1}, \ldots, \boldsymbol{\Pi}_{il}, \ldots)$ be the steady state probability vector, where each element $\boldsymbol{\Pi}_{il}$ is a row vector except Π_{i0}. By definition, $\boldsymbol{\Pi}_{il}$ is the steady state probability vector for the i–th input queue equal to l at different phases. The balance equation is written as

$$\boldsymbol{\Pi}_i(\mathbf{P}_i - \mathbf{I}) = 0, \quad \boldsymbol{\Pi}_i\mathrm{e} = 1. \tag{18}$$

Similar to the matrix geometric solution form given in [10], the queue length distribution of an input queue with a finite size of $K \geq 2$ can be described by

$$\boldsymbol{\Pi}_{il} = \Pi_{io}\,\mathrm{p}_i^{\circ}\mathbf{R}_i\left(\mathbf{H}^{\circ}\mathbf{R}_i\right)^{l-1}, 1 \leq l \leq K \tag{19}$$

$$\Pi_{iK} = \lambda_i^{\circ}\Pi_{i0}\,\mathrm{p}_i^{\circ}\mathbf{R}_i\left(\mathbf{H}^{\circ}\mathbf{R}_i\right)^{K-2}\mathbf{H}^{\circ}\left(\mathbf{I}-\lambda_i^{\circ}\mathbf{S}_i-\mathbf{H}^{\circ}\right)^{-1} \tag{20}$$

where

$$\mathbf{R}_i = \lambda_i^{\circ}\left[\mathbf{I} - \left(1 - \lambda_i^{\circ}\right)\mathbf{H}^{\circ} - \lambda_i^{\circ}\mathbf{e}\mathbf{p}_i^{\circ}\right]^{-1}$$

$$\Pi_{i0} = \left[1 + \mathbf{p}_i^{\circ}\mathbf{R}_i\left[\sum_{i=l}^{K-l}\left(\mathbf{H}^{\circ}\mathbf{R}_i\right)^{l-l} + \lambda_i^{\circ}\left(\mathbf{H}^{\circ}\mathbf{R}_i\right)^{K-2}\mathbf{H}^{\circ}\left(\mathbf{I} - \lambda_i^{\circ}\mathbf{S}_i - \mathbf{H}\right)^{-1}\right]\mathbf{e}\right]^{-1} \tag{21}$$

Denoting q_{il} as the probability of the length of the i–th input queue equal to l, we obtain,

$$q_{il} = \mathbf{\Pi}_{il}\mathbf{e} \tag{22}$$

By definition, q_{iK} gives the cell loss probability at input i. We recall that the probability of this happening has been defined by ϵ_i^o. Consequently, we may obtain ϵ_i^o by iteration, starting with $\epsilon_i^{\circ} = 0, \forall\ i$, which corresponds to the case of infinite buffer size, the next value of ϵ_i^o is given by q_{iK} in equation (22) until the difference between them becomes smaller than the specified convergency interval. Therefore one can obtain an exact solution of the cell loss probability at each input. The overall cell loss probability in the system is then given by

$$\epsilon^{\circ} = \frac{\sum_{i=0}^{N}\epsilon_i^{\circ}\lambda_i^{\circ}}{\sum_{i=0}^{N}\lambda_i^{\circ}} \tag{23}$$

2.2.2 Output Queue Analysis

The arrival process to outputs is determined by the number of HOL contending cells and m. Consequently the arrival process to the output j can be characterized by a batch

arrival with size r at each time slot, for $0 \leq r \leq m$. Define O_j as the number of cells arriving at the output queue j in the current time slot. In steady state, the batch size distribution is given by $p^{\bullet}_{jr}$ evaluated from Equation (9). Let $o_{jr} = \Pr(O_j = r)$ in each time slot, then $o_{jr} = p^{\bullet}_{jr}$, where $o_{jm} = \sum_{k=m}^{N} p^{\bullet}_{jk}$. The Markov chain for the output queueing process can be constructed according to Equation (22) and o_{jr}, where each state represents the queue length of the output j. With the batch arrival, fixed output service time and the queue size L_j, the output queueing process can be viewed as a G/D/1/L_j. The governing equation for the output queueing process is given by

$$Q_j' = (Q_j - 1)^+ + O_j \tag{24}$$

where Q_j', and Q_j are defined as follows:

Q_j' the number of cells in the output queue j at the next time slot

Q_j the number of cells in the output queue j including one being served in the current time slot

Define the state transition matrix of the Markov chain for the output queueing process, $\mathbf{F}_j$, by

$$\mathbf{F}_j = \begin{bmatrix} o_{j0} & o_{j1} & \cdots & o_{jm} & 0 & \cdots & 0 & 0 & 0 \\ o_{j0} & o_{j1} & \cdots & o_{jm} & 0 & \cdots & 0 & 0 & 0 \\ 0 & o_{j0} & \cdots & o_{jm-1} & o_{jm} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & o_{j0} & o_{j1} & 1-\sum_{i=0}^{1} o_{ji} \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & o_{j0} & 1-o_{j0} \end{bmatrix}. \tag{25}$$

Similar to the input queueing analysis, the steady state probability of this Markov chain, $q^{\bullet}_{jl}$, $0 \le l \le L_j$ for $j = 1, 2, \cdots, N$, or a vector form $\mathbf{q}^{\bullet}_j = \left(q^{\bullet}_{j0}, q^{\bullet}_{j1}, \cdots q^{\bullet}_{jL_j}\right)$ can be obtained from

$$\mathbf{q}^{\bullet}_j(\mathbf{I} - \mathbf{F}_j) = \mathbf{0} \tag{26}$$

with $\mathbf{q}^{\bullet}_j \mathbf{e} = 1$.

The cell loss probability at output j due to buffer overflow is previously defined by $\epsilon^{\bullet}_j$. Define a blocking state as an output queue length state for which new arrivals may get blocked. Then, as shown in Figure 2.4, the blocking occurs whenever there are more cell arrivals than there is space available at the output, namely, when $r > L-n+1$ with r being the number of cell arrivals up to m, and n being the current buffer status. The $\epsilon^{\bullet}_j$ can be obtained as

$$\epsilon^{\bullet}_j = \frac{E\{Number\ of\ Lost\ Packets\ \ at\ output\ j\}}{Offered\ load\ at\ ouput\ j} = \frac{\sum_{a=0}^{m-2} \sum_{i=a+1}^{m-1} (i-a) o_{m-a} q_{L-m+1+i}}{\lambda^{\bullet}_j} \tag{27}$$

where $\lambda^{\bullet}_j = \sum_{i=1}^{N} \lambda^{o}_i (1 - \epsilon^{o}_i) t_{i,j}$. Notice that the number of cells lost at different blocking states may be different, and this fact is accounted in the weighting constant (i-a) and relevant limits in the summations in Equation (27).

The probability that a cell lost at output j was actually from input i is defined by ϵ_{ij}, and is described by

$$\epsilon_{i,j} = \epsilon^{\bullet}_j \left[\frac{\lambda^{o}_i (1 - \epsilon^{o}_i) t_{i,j}}{\lambda^{\bullet}_j} \right]. \tag{28}$$

If a cell which arrived at input i is lost in the switch, it either happens at the input $i(\epsilon_i^{\circ})$, or at the output$(\epsilon_{i,j})$. Consequently the cell loss probability of the cells from input i is derived by

$$\mathbf{P}_{iL} = \epsilon_i^{\circ} + \sum_{j=1}^{N} \epsilon_{i,j} = \epsilon_i^{\circ} + \sum_{j=1}^{N} \left[\epsilon_j^{\bullet} \left(\frac{\lambda_i^{\circ}(1-\epsilon_i^{\circ}) t_{i,j}}{\lambda_j^{\bullet}} \right) \right]. \tag{29}$$

Finally the total cell loss probability of the switch, P_T, is given by

$$P_T = \frac{\sum_{i=1}^{N} \mathbf{P}_{iL} \cdot \lambda_i^{\circ}}{\lambda} \tag{30}$$

where $\lambda = \sum_{i=1}^{N} \lambda_i^{\circ}$ denotes the total offered load of the switch.

2.2.3 Numerical Examples

In numerical studies, we are interested in observing the effects of different system and traffic parameters on the switch performance measured by cell loss probability, for both uniform and nonuniform traffic cases. The cell loss probability is computed by the iteration method in which the difference between two adjacent iterations is to be less than 10^{-16}. Convergence of the iteration is very fast and all the numerical solutions presented in this paper are obtained within 4 iterations. For instance, the convergence for the case with m=2, λ=0.8 and K=L=30 is obtained after 2 iterations with the cell loss probability equaling 9.28E-10. We assume that a cell loss probability of 10^{-9} or less is regarded as an acceptable measure for an ATM switch since it is desirable that the cell loss at a switch

should be kept lower than such losses elsewhere in the network [3]. As a result of the numerical examples, we address design trade-offs to be considered during switch design.

Uniform Traffic Case

The Effect of Speed-up Factor m As m increases, the number of cell arrivals at an output increases accordingly. In such cases, we found that the total cell loss probability is dominated by the output cell loss as m increases, as evidenced in Figure 2.5a. The reason is as follows: since multiple cells are delivered to an output by the switch fabric, fewer cells are waiting at HOL, resulting a smaller cell loss at each input. On the other hand, more cells will be lost at the output.

In Figures 2.5b and 2.5c, we compare the cell loss probability at various speed-up factors and the output queue size L by fixing the input queue size K at 10 and 25, for the offered load λ of 0.4 and 0.75 respectively. We choose λ=0.4 for the average load and λ=0.75 for high load of the switch. We can find the optimal m, where the optimal m is defined by the point that achieves the desired cell loss with a given input/output buffer sizes. For instance, the optimal m is obtained at 2 to guarantee the 10^{-9} cell loss probability requirement if system and traffic parameters are fixed at K=L=10 and λ=0.4. Another example: a selection of$m$=2, K=L=25 with λ=0.75 is sufficient to guarantee 10^{-9} cell loss probability. This is a significant result, since increasing$m$ by 1 means much higher cost involved in implementation of the switch fabric than increasing buffer size as discussed in the introduction. Notice also in Figure 2.5a and 2.5b that as m gets

larger, the cell loss performance decrease with the fixed buffer sizes. We expect, however, that if a larger output buffer size (L) is provided, the optimal cell loss will be found at larger m. For instance, m=3 achieves the minimum cell loss probability when L=18 as shown in Figure 2.5b. If the output buffer size is infinite, the larger the speed-up factor, the better the performance. The cell loss probability for various offered loads with m=2 and K=10, 20 and 30 are plotted in Figure 2.5d. One may choose any design parameters below the 10^{-9} cell loss line depending on design constraints such as complexity. For instance, K=L=30, is enough to meet 10^{-9} cell loss requirement even at λ=0.8. Therefore, our numerical study shows that for a moderate input/output buffer budget, choosing m=2 is good enough to meet our desired cell loss requirement for an offered load up to 0.8.

Effect of Queue Size and Optimal Buffer Allocation Let us first examine how different input/output queue sizes affect the cell loss performance for λ=0.4 and λ=0.75. In Figure 2.6a, we plot cell loss probability vs. various input/output queue sizes for m=2 and λ=0.4. First, it is noted that for fixed input buffer size K, the cell loss gets smaller as we increase output buffer L only until L reaches a certain value. For instance, with K=6 the cell loss decreases until L=10. After L=10, adding more output buffers results in negligible improvement. This result is due to the combination of adopting *Queue Loss* scheme and fixed m. If we increase K, this saturation point takes place at a higher L. The same tendency is observed in Figure 2.6b for λ=0.75. For instance, with K=20 the cell loss gets smaller until L=28, where the cell loss probability is at 3.5×10^{-10}. Beyond

L=28, performance improvement diminishes.

Secondly, we can find optimal buffer allocations for a fixed buffer budget from Figures 2.6a and 2.6b. For instance, there are some possible combinations to distribute a total buffer size of 16. In Figure 2.6a, equal distribution, i.e. K=L=8, gives a cell loss probability of 10^{-8}, while K=7 and L=9 achieves the optimum cell loss probability at 0.6×10^{-9}. We observe larger differences between K and L for increased load λ=0.75, as shown in Figure 2.6b.

For example, with the total budget of 48, K=L=24 gives 10^{-9} cell loss probability, while K=21 and L=27 achieves the optimum cell loss probability at 2.2×10^{-10}. These observations suggest that for *Queue Loss* scheme we need to allocate more buffers at output than input. It is noted that our result for λ=0.4 is slightly different from the previous result in [13], which claims that the equal distribution is the optimal allocation. However, for high load λ=0.75, the difference between the equal allocation and our result becomes markedly larger.

Nonuniform Traffic Case

The Effect of Input Nonuniformity We examine the effect of increased input nonuniformity U° with respect to the speed-up factor m as plotted in Figure 2.7a. We set K=L=10, $d_i=d_o=1$, uniform T_g and λ= 0.4. Note that m=2 is the optimum choice for given buffer sizes. We observe higher cell losses as U° increases regardless of the speed-up. This says that a switch with given parameters can not meet the cell loss requirement

of 10^{-9} when U° > 30. In addition, for $m \geq 3$, the effect of increased U° is not much. However, for m=2, the effect of increased U° is large, yet it still is the only case that satisfies the cell loss requirement.

The effect of increasing U° with respect to various K and L is shown in Figure 2.7b and 2.7c respectively. Given m=2, it can be seen from Figure 2.7b that the effect of increasing U° on the cell loss performance becomes negligible for input queue size K>14. By contrast, this is not true for the case in Figure 2.7c. Increase in output queue size L alone cannot absorb the effect of U°. Hence, it suggests that the effect of U° can be absorbed by increasing input queue sizes.

The Effect of Group Size We examine the effect of different group size ratios on cell loss performance. Five different cases are chosen for comparison: (1) U° =1, d_i=1, d_o=1, uniform traffic; (2) U° =3, d_i=1, d_o=1); (3) U° =3, $d_i=\frac{3}{7}$, d_o=1; (4) U° =3, d_i=1, $d_o=\frac{3}{7}$; (5) U° =3, $d_i=\frac{3}{7}$, $d_o=\frac{3}{7}$. Also, we use uniform output group addressing in all five cases. In Figure 2.8a, cell loss probability is plotted with respect to the speed-up factors. Obviously, the uniform traffic case achieves the best performance. Case (2) is the same as uniform traffic except input group 1 is three times as loaded as input group 2. The cell loss increases a little, but not much. If we change the input group size ratio from 1 to 3/7(case 3), the cell loss slightly increases due to the increased traffic intensity at input group 1. This means that the performance is affected moderately by nonuniform input group size ratio. Now we change the output group size ratio from 1 to 3/7 (case 4). This

means that, while the traffic load to output group 1 stays the same, the number of ports assigned to output group 1 becomes smaller. This case 4 abruptly degrades the cell loss performance, which is the worst among the five cases. In case 5, we change the input group ratio from 1 to 3/7. Then the degradation in case 4 is alleviated a little, because changing the input group ratio smoothes out the total nonuniformity. As in previous examples with increased speed-up factor, we observed a similar result: increasing speed-up factor worsens the performance. It is clear at this point that nonuniformity in output group size ratio affects the cell loss performance significantly more than that of input group size ratio.

Figure 2.8b and 2.8c show the effect of group size ratios with respect to the input/output buffer sizes. As noticed previously, performance improvement by increasing input buffer size saturates much earlier than that of output buffer size.

The Effect of Output Group Addressing Consider the effect of T_g on the cell loss performance. In practice, the T_g is determined by the destination address embedded in the cell header. Let us take a case with λ=0.4, m=2, L=K=10, d_i=d_o=1 and $T_g = \begin{bmatrix} g_1 & 1-g_1 \\ 0.4 & 0.6 \end{bmatrix}$, where g_1 is the group addressing probability from input group 1 to output group 1, and varies from 0.2 to 0.8. For comparison, the input nonuniformity, U° is varied as 1, 3 and 10 respectively as shown in Figure 2.9.

It is interesting to see the valleys, which give the minimum cell loss probabilities for individual U°'s. For instance, as we changeg_1 from 0.2 to 0.8, the valley for U°=1

takes place wheng_1=0.6. At this valley the output addressing matrix $T_g = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$. With this T_g, the input and output traffic intensities become equal, offering the same performance as uniform traffic. Observe also that at the valley U^* turns out to be 1. Since U^* defines the output traffic intensity ratio, it is a function of input traffic intensity U° and output addressing probability matrix T_g. Thus, the output traffic nonuniformity can be used as a collective nonuniformity parameter, reflecting the total degree of nonuniformity of a switch. We observe degraded performance for the increased input nonuniformities, U°=3 and U°=10. At the valleys the respective U^* for U°=3 and U°=10 become 1. In general, since input nonuniformity may be absorbed by increased input buffers, the degradation of the switch performance is primarily attributed to the nonuniform T_g.

Section 2.3 Delay Analysis

2.3.1 Input Queueing Delay Analysis

The service time of the input i, S_i°, can be identified by another discrete phase-type distribution due to the closure property of the phase-type distribution [17, Chap.2]. Since each cell at HOL maintains its originally assigned output address, its state transition matrix can be constructed as shown in Equation (13). The initial state of the input service time is assigned by a probability vector p_i°, as given in Equation (14). The input service time distribution $s_i^\circ(n)$ is then characterized by a phase-type distribution represented by (p_i°, H°) and given by

$$s_i^\circ(n) = \sum_{j=1}^{N} t_{ij} s_j^\bullet(n) \tag{31}$$

The derivation for the input queueing delay distribution follows. Suppose that the tagged cell has just arrived at the input i and enters *l-th* level counted from HOL. Due to FCFS queueing policy, it can only move to *(l-1)th* level in the next CST, only if the cell at HOL is successfully transmitted during the current CST. Thus, the sojourn time of the tagged cell in *l-th* level is the service time of the HOL cell, S_i°. Notice that the sojourn time of the tagged cell at *(l-1)th* level is S_i° again because the selection process for the present HOL cell is identical (due to random selection policy) from the one for the previous HOL cell. Therefore, the sojourn time of the tagged cell at each level is independent and identically distributed, and is represented by S_i°.

Viewing a HOL as a server for an input queue, we define W_{il}° as a waiting time of the tagged cell from its arrival time at *l-th* level until it moves into HOL, given by

$$W_{il}^{\circ} = \underbrace{S_i^{\circ} + S_i^{\circ} \cdots + S_i^{\circ}}_{l-1} \quad \text{for} \quad 2 \le l \le K_i . \tag{32}$$

Correspondingly, the waiting time distribution $w_{il}^{\circ}(n)$, is derived by *(l-1)*fold convolution of the input service time distribution $s_i^{\circ}(n)$, given by

$$w_{il}^{\circ}(n) = \underbrace{s_i^{\circ}(n) \circledast s_i^{\circ}(n) \circledast \cdots \circledast s_i^{\circ}(n)}_{(l-1)fold} \quad \text{for} \quad 2 \le l \le K_i . \tag{33}$$

It is yet to be determined which level the tagged cell enters upon its arrival. In the steady state, the probability that the tagged cell enters *l-th* level is q_{il-1}°, and the input queueing delay distribution for the tagged cell can be derived by q_{il-1}° . $w_{il}^{\circ}(n)$ and $s_i^{\circ}(n)$. This

is viewed as a branching process depicted in Figure 2.10. Note that q_{iK}° represents the cell loss probability at input i. We define a random variable D_{ij}° as the input queueing delay experienced by the tagged cell from its arrival till absorption to the switch fabric(leaving HOL contention). Hence, the queueing delay of the tagged cell at input i is given by

$$D_{ij}^{\circ} = q_{i0}^{\circ} S_i^{\circ} + \sum_{l=2}^{K} q_{il-1}^{\circ} \left(W_{il}^{\circ} + S_i^{\circ} \right) \text{ for } i = 1, \cdots, N. \tag{34}$$

Considering the independence between W_{il} and S_i° as explained earlier, the input queueing delay distribution is readily derived as,

$$d_{ij}^{\circ}(n) = q_{i0}^{\circ}\, s_i^{\circ}(n) \; + \sum_{l=2}^{K} q_{il-1}^{\circ}\, w_{il}^{\circ}(n) \circledast s_i^{\circ}(n). \tag{35}$$

2.3.2 Output Queueing Delay Analysis

The arrival process to outputs and the state transition matrix of the Markov chain for the output queueing process are given in the previous section.

Successfully admitted tagged cell now spends a random waiting time until its departure from the output queue j. We define $D_{ij}^{\bullet}$ as the queueing delay at the output queue j and $d_{ij}^{\bullet}(n)$ as the corresponding output queueing delay distribution in terms of n CSTs.

The derivation of the output queueing delay distribution for the tagged cell, $d_{ij}^{\bullet}(n)$, is similar to that of the input's. It is simplified to find the distribution of the levels that the tagged cell enters upon its arrival, since the service time of each output queue is fixed at one CST.

In steady state, the levels into which the tagged cell enters is determined by output queue status and the number of admitted cells in the batch size r. To compute the probability that the tagged cell enters n–th level, we consider the following. Suppose that the output queue j is occupied by $n-1$ cells before the batch arrival including the tagged cell, and its probability is represented by $q^{\bullet}_{jn-1}$. Upon the batch arrival, the probability that the tagged cell enters n–th level is $\frac{1}{r}$. for $1 \leq r \leq m$, and the same probability is applied for other admitted cells. In this manner, the tagged cell may occupy any one of r levels from n–th level. By considering all the possibilities that the tagged cell ends up occupying n–th level, the delay distribution of the tagged cell at the output j can be computed.

For example, if L_j=5 and m=3, the probability that the tagged cell enters the first position in the output queue j (to be served in one CST), is given by $q^{\bullet}_{j0}(o_{j1}+\frac{1}{2}o_{j2}+\frac{1}{3}o_{j3})$. In other words, the output queue j is empty and the number of admitted cells, including the tagged cell, can be one, two, or three depending on the batch size. Likewise, the probability that the tagged cell enters the second position (to be served in two CSTs) is given by $q^{\bullet}_{j0}(\frac{1}{2}o_{j2}+\frac{1}{3}o_{j3})+q^{\bullet}_{j1}(o_{j1}+\frac{1}{2}o_{j2}+\frac{1}{3}o_{j3})$, where the first term indicates the cases in which the tagged cell ends up entering the second position although the output queue is empty and the second term indicates the case when the output queue j is occupied by one cell before the batch arrival. If all such possibilities are listed, the probability that the tagged cell enters n–th level can be summarized by

$$\sum_{l=(n-m)^+}^{n-l} q_{jl}^{\bullet} \sum_{k=n-i}^{m} \frac{o_{jk}}{k} \quad \text{. for} \quad n = 1, 2, \ldots, L \text{ (CSTs).}$$

The tagged cell admission to the output j is conditioned on two facts: there are cell arrivals at the output j $(1 - O_{j0})$ and the tagged cell is admitted $(1 - \epsilon_{ij})$. Hence, the output queueing delay distribution for the tagged cell is derived as,

$$d_{ij}^{\bullet}(n) = \frac{1}{(1 - o_{j0})(1 - \epsilon_{ij})} \left(\sum_{l=(n-m)^+}^{n-l} q_{jl} \sum_{k=n-i}^{m} \frac{o_{jk}}{k} \right) \text{ . for } n = 1, 2, \ldots, L \text{ (CSTs).} \tag{36}$$

Note that n starts from 1 because cells must wait at least one CST to be served according to the governing equation, Eq. (18).

2.3.3 Total Queueing Delay Analysis

In addition to the random delays at input and output, the tagged cell experiences a fixed transmission delay in the switch fabric. The switching time depends on the switching architecture. For example, it would be the cell self-routing time spent through a multi-staged Batcher-Banyan type of switch fabric, or the system read-write time in a Shared-Buffer type of switch fabric. We use D_{sw} as the fixed delay caused by the switch fabric. LetD_{ij} be the total delay experienced by the tagged cell from its arrival instant at the input$i$ till the time instant of a successful departure from output j. It is readily given by

$$D_{ij} = D_{ij}^{\circ} + D_{ij}^{\bullet} + D_{sw} \tag{37}$$

In order to derive corresponding total queueing delay distribution, we claim that input queueing delay and output queueing delay for the tagged cell are independent.

Theorem: In a switch with the *Queue Loss* scheme and the random selection policy among contending cells, D_{ij}° *and* $D_{ij}^{\bullet}$ *are independent* for any cells having (i, j) input and output port address pair.

Proof The independence between D_{ij}° and $D_{ij}^{\bullet}$ is self-evident by the imposed conditions:*Queue Loss* and random selection policy among contending cells. A quantitative proof is provided here by showing that the total queueing delay distribution equals the convolution of input and output queueing delay distributions forD_{ij}° and $D_{ij}^{\bullet}$ via Monte Carlo computer simulation. The conditions for the Monte Carlo computer simulation are given as follows: 150× 150 switch size, uniform traffic load of 0.8, speed-up factor of 2, input buffer size of 15 and output buffer size of 15. 15 million cells are input to the switch for each subrun and a total of six separate subruns are performed to obtain 95% confidence interval. Let $D_T = D_{ij}^{\circ} + D_{ij}^{\bullet}$, and $d_t(n)$ be the corresponding probability density function. We separately obtain $d_{ij}^{\circ}(n), d_{ij}^{\bullet}(n)$ and $d_t(n)$ from the computer simulation as follows. Among the cells arriving at the input i, we time-stamp those cells destined to the output$j$. We measure the input queueing delays for those cells with (i, j) port address when they are selected and successfully admitted into the output j. The measured and normalized statistic is $d_{ij}^{\circ}(n)$. Similarly we time-stamp those cells upon their arrival at the output j and measure the output queueing delay when they depart

from the switch. This provides $d^{\bullet}_{ij}(n)$. These are shown in Figure 2.11a with 95% confidence intervals. We also separately measure the total delay for those cells from arrival instant at the input i till the departure instant from the output j. This provides $d_t(n)$. Now we convolve $d^{\circ}_{i}(n)$ and $d^{\bullet}_{j}(n)$ and compare the convolution result with the separately measured $d_t(n)$, as shown in Figure 2.11b. The convolution result is confined within 95% confidence intervals from the measured total queueing delay distribution. Therefore, D°_{ij} and $D^{\bullet}_{ij}$ are independent since the convolution of input and output queueing delay distribution equals the total queueing delay distribution.■

Since D°_{i} and $D^{\bullet}_{j}$ are independent, the total queueing delay distribution of the tagged cell is given by

$$\begin{aligned} d_{ij}(n) &= d^{\circ}_{ij}(n) \circledast d^{\bullet}_{ij}(n) \circledast d_{sw}(n) \\ &= d^{\circ}_{ij}(n) \circledast d^{\bullet}_{ij}(n) \circledast \delta(n - D_{sw}). \end{aligned} \tag{38}$$

which is the convolution of $d^{\circ}_{ij}(n)$ and $d^{\bullet}_{ij}(n)$, shifted by D_{sw} in time.

Chapter 3 Effective Cell Loss

Section 3.1 A Single Nodal QoS Parameter

In this section, we study the effect of excessive cell delay under a certain time requirement. We focus on the quantitative analysis of cell loss due to excessive queueing delay (LD).

Say that the maximum allowable delay at a switch node is T CSTs. Any cells experiencing delay longer than T are assumed to be useless. Enforcing such time requirement at the switch is yet to be determined. We can think of a simple and intuitive way to implement it on the switch. For example, the input port controller time-stamps arriving cells, while the output port controller checks the elapsed time of each cell before the transmission to the output link by stripping the time-stamps. If the time spent in the switch exceeds T, the output port controller immediately drops it. Such a cell loss probability can be obtained by summing the tail portion of the total delay distribution after T. We denote $\epsilon^{\bullet}_{D_{ij}}$ the cell loss probability due to excessive delay (DLP) for cells with (i, j) input and output port address pair. This is illustrated in Figure 2.12, with an arbitrary delay distribution and T.

Then, the loss probability of the cells arriving at the input i due to excessive delay is given by

$$\mathbf{P}_{iD} = \sum_{j=1}^{N} \epsilon^{\bullet}_{D_{ij}}. \tag{39}$$

In the previous section, we found the loss probability of cells from the input i due to buffer overflow, represented by P_{iL}, as given in Equation (29). Hence the Effective Cell Loss probability of the cells with (i, j) input and output port address pair, P_{iE}, of the switch is given by

$$P_{iE} = P_{iL} + P_{iD} = \epsilon_i^{\circ} + \sum_{j=1}^{N} \left(\epsilon_{ij} + \epsilon_{D_{ij}}^{\bullet} \right). \tag{40}$$

Finally the total effective cell loss probability of the switch, P_E, is given by

$$P_E = \frac{\sum_{i=1}^{N} P_{iE} \cdot \lambda_i^{\circ}}{\lambda} \tag{41}$$

where $\lambda = \sum_{i=1}^{N} \lambda_i^{\circ}$ denotes the total offered load of the switch.

Section 3.2 Numerical Study of the Effect of ECL

In the following, we examine the effect of choosing different system parameters on ECL performance of a switch via some numerical examples. In particular, we investigate the effect of increasing buffer sizes and speed-up factors, different buffer allocations between inputs and outputs, and the nonuniform traffic pattern given the maximum delay allowed at a switch (T). Throughout the examples, the switch is assumed to satisfy 10^{-9} ECL requirement for a given input to output path within a maximum delay T set by an end user. Note that, in the end user's point of view, the ECL performance of a particular path from an input to an output is more important than the overall ECL performance of the switch. This is evidenced by different nonuniform examples in the following subsections.

For the method of numerical analysis, Neuts [17, p51] has derived a Markov matrix generator to convolve two discrete phase type distributions. It is, however, easier to take the direct convolution with the input service time distribution vectors since it is computationally simple compared to handling matrices. As a check, we have examined both the Neuts' algorithm and the direct convolution, and obtained exactly the same results.

3.2.1 Model Verification by Queueing Delay Distribution

In this subsection, we first verify our analysis by Monte Carlo computer simulation written in C.

Uniform Traffic Case With uniform traffic assumed, a set of switch conditions is considered: a switch size of 100 × 100, speed-up of 2, input and output buffer size of 15 each, and given load of 0.8. The total delay distribution of a cell travelling from a particular input i to a particular output j is obtained by both numerical analysis and simulation in which the simulation covers a range of probability from 10^{-6} to 10^{-1}. During each simulation, approximately 15 million cells have been generated.

As shown in Figure 2.13, the simulation and the numerical result matches very well for the ranges of interest (the tail distribution for the ECL computation), except our analysis underestimates the queueing delay for less than 3 CSTs.

Nonuniform Traffic Case For nonuniform examples, the input buffer size of 50 and the output buffer size of 80 with an offered load of 0.47 are considered.

For the comparison, the switch inputs and outputs have been divided into two groups each. Individual inputs and outputs belonging to the same group are assumed to be statistically independent and identical. Specifically, each input group has 50 inputs, the output group 1 has 70 outputs, and the output group 2 has the remaining 30 outputs. The probability traffic matrix is $T = \begin{bmatrix} 0.4 & 0.6 \\ 0.4 & 0.6 \end{bmatrix}$. This moderate nonuniform traffic conditions result in the congestion for the traffic destined to the output group 2, regardless of input loads. Also, the maximum throughput of the switch is substantially reduced due to the load imbalance at the outputs even if the load to the inputs are balanced. With the given system conditions, the maximum throughput of the switch is found to be only 0.53, based on the analysis in [12]. Thus, the offered load of 0.47 is 90% of the maximum throughput. There is, however, nothing much that a switch can do to prevent such severe reductions in throughput due to nonuniform traffic.

Both analysis and simulation results are plotted for input, output and total queueing delay distributions of the two groups, and shown in Figure 2.14a, 2.14b, and 2.14c respectively. Those figures attest that analysis and simulation result matches extremely well. As shown in Figure 2.14a, the cells destined to the output group 2 (congested group) experience substantially larger delay than the cells destined to the output group 1 (underload group). This phenomenon gets intensified in output queueing delay distributions as observed in Figure 2.14b. The total delay, as shown in Figure 2.14c, is dominated by the output delay. In particular, the cell delay for the underload group is concentrated

within 40 CSTs, whereas the cell delay for the congested group widely spreads out up to 82 CSTs and then sharply falls. The break point at 82 CST is resulted from L=80 plus 2 additional CSTs due to the input/output governing Equations (6) and (24). It is evident from Figure 2.14 that, with the given switch and traffic conditions, the cell delay for the congested group is mainly due to output queueing delay, whereas the cell delay for the underload group is largely dependent on the input queueing delay.

3.2.2 The Effect of Increasing the Buffer Size

In general, increasing the buffer size reduces cell loss due to buffer overflow (LO). However, previous figures indicated that adding buffers may cause longer queueing delay for heavy traffic, which could be detrimental to the delay constrained traffic. In this section, we investigate the effect of increasing the buffer size on the switch cell loss performance with respect to ECL performance.

Uniform Traffic Case Consider a set of input/output buffer sizes to satisfy the cell loss probability of the order of 10^{-9} for an offered load of 0.8. Based on the analysis in [9], the input buffer size of 30 and the output buffer size of 28 achieves the cell loss probability due to buffer overflow (LOP) of 7.3699×10^{-9}. The LOP is further reduced by increasing the output buffer size (L_j) as shown in Figure 3.1a by a solid line. For instance, with the output buffer size L greater than or equal to 32, the switch satisfies LO requirement. The LO improvement becomes quickly saturated when L_j increases beyond 35. Although the input buffer size is fixed at 30 for this numerical example,

further improvement is possible if the input buffer size is increased [9]. On the same figure, the cell loss probability due to excessive delay (LDP) is plotted with arbitrarily given delay constraint, Ts. As shown, a stringent delay constraint easily gives rise to intolerable cell loss performance due to LD.

The ECL performance is shown on Figure 3.1b. We observe that all the input/output buffer size pairs satisfying LO requirement of 10^{-9} in Figure 3.1a, now become unusable if the given delay constraint is less than 40 CSTs (approximately 120 μsec.). Whether this delay constraint is realistic or not depends on, among many others, QoS requirements set by the end users during the VC setup. Note that the improvement of ECL performance gained by adding buffers saturates after a certain output buffer size for a given T. This suggest that, given input buffer size, a minimum output buffer size satisfying QoS requirement can be found using our analysis. For example, in Figure 3.1b, choosing K=30 and L=34 is good enough to satisfy QoS requirements of $T \leq$ 40CSTs and LOP $\leq 10^{-9}$ for the offered load=0.8.

Nonuniform Traffic Case Keeping the same nonuniform conditions used, L_j is varied from 50 to 130 for the offered load 0.47, and LOP for both traffic groups has been evaluated. For example, when $L_j = 80$, LOP $= 1.5394 \times 10^{-8}$ for the congested group whereas LOP $= 0$ for the underload group. Since both LOP and LDP for the underload group are negligible ($\ll 10^{-9}$), only the congested group is considered for LOP and LDP.

Figure 3.2a shows both LOP and LDP versus output buffer size at congested output

group 2. As expected, LOP is decreased by adding more output buffers as shown by a solid line. LDP, however, is rapidly increasing as T is decreasing. Hence, the ensuing delay increase is no longer negligible. For instance, if the given delay constraint T is 90 CSTs (approximately 270 μsec.) for K=50 and all possible values of L, DLP alone becomes greater than 10^{-8}, which falls short of the cell loss requirement. Such rapid degradation in LDP saturates when $T < L$. This attests that, for m >1, cell loss due to excessive delay (LD) occurs mostly by the output queueing delay rather than the input queueing delay in which HOL contention time is included.

Figure 3.2b effectively facilitates the trade-off study between LOP and LDP. In particular, the ECL probabilities (ECLP) are plotted with respect to buffer size at output group 2. Suppose the delay constraint T is 90 CSTs. The ECL continues to decrease until $L_j = 80$. Further increase in buffer size begin to worsen the ECL performance. This buffer size 80 is a point where the gain from LO is offset by the loss resulted from LD. Thus, using Figure 3.2b, we can find an optimal set of input/output buffer sizes that achieves the minimum ECL for a given T at a switch. In addition, the ECL probabilities for different values of delay constraint T are plotted in Figure 3.2b.

Chapter 4 Optimal Buffer Management

Section 4.1 Buffer Dimensioning

The switch design requires a proper buffer dimensioning. The optimal buffer allocation between input and output based on internal cell transfer scheme, speed-up factors, traffic pattern and load, and the QoS requirement has been studied [9, 13]. The studies show that approximately equal allocation of the available buffers between input and output achieved the best LO.

With stringent delay constraint at each switch, however, such results may not be applicable any more. In this section, using ECL analysis, we show how one can obtain optimal buffer allocation that satisfies both LO and LD requirements.

4.1.1 Uniform Traffic Case

As an example, a total buffer budget of 70 (K+L=70) is chosen for a particular input to output path. By varying the buffer sizes between input and output for given speed-up factors and the offered load=0.8, we have plotted the effective cell loss probabilities (ECLP) for different *T*s as shown in Figure 4.1, where LOP is drawn with bold solid line. Different values of *T*s resulted in different plots of ECLP.

If LO performance is considered only, increasing speed-up factor results in the optimum buffer allocation at a large output buffer size. For instance, for m=2, a minimum LOP=3.95×10^{-11} can be obtained at K=33 and L=37. For m=3, a smaller LOP=6.82×10^{-13} can be obtained at K=13 and L=57. For larger m, even smaller LOP is

possible by allocating most of available buffers to the output. This is intuitively correct since increasing speed-up factor to a total input size, N, results in an output queueing switch.

Such trend is no longer true when the delay constraint T is imposed as shown in Figure 4.1, where the minimum ECLP points are indicated by ■, ▲, and ○ for m=8, 3 and 2 respectively. It is clearly demonstrated that as the delay constraint gets tighter, the ECL performance gets worse. For instance, with the total buffer budget 70 and m=3, a switch can satisfy the cell loss requirement10^{-9} by allocating K=13 and L=57. However, if the delay constraint T = 45 CST, all different buffer allocation fails to meet such cell loss requirement. In general, a large speed-up responds more sensitively to T. For example, significant jumps from LOP plot (bold solid line) to ECLP plot for T=40 CSTs (solid line) are observed for large speed-ups (m=3, 8) whereas a small jump is observed for m=2. The optimal allocations can be different if the system conditions change. It is interesting to see that the minimum ECLP is found at L=T-3 for m>2 while it is found at L=57 for m=2.

4.1.2 Nonuniform Traffic Case

To show the effect more vividly, we consider another extreme nonuniform case. Each input group has 50 inputs as before, but the output group 1 now has 10 outputs and the output group 2 has the remaining 90 outputs. The probability traffic matrix $T = \begin{bmatrix} 0.8 & 0.2 \\ 0.7 & 0.3 \end{bmatrix}$. This may represent a network condition for a very popular database

during its peak hours. The offered load of 0.128 already results in a severe cell loss for the congested group due to buffer overflow.

In order to absorb the detrimental effect by the nonuniform traffic condition, we have increased the total buffer size for each input-output path to 200. With the given buffer budget, only m=2 satisfies ECL requirement as shown in Figure 4.2. Again the minimum ECLP is found at L=T-3 for all cases considered. For example, choosing m=2, K=23 and L=177 with T=180 CSTs (509 μsec) provides the minimum ECLP=5.3×10^{-11}.

Section 4.2 Output Buffer Sharing Algorithm

Designing switches based on the uniform traffic condition is risky in a real traffic environment. Even with conservative design, unexpected nonuniform traffic can seriously congest a switch. It is therefore important to develop appropriate control schemes to alleviate the degradation by the traffic nonuniformity. An effort at the switch input to alleviate the input nonuniformity is sought by the introduction of a randomization/distribution network [26]. With this randomizer, input nonuniformity may no longer be a serious problem requiring increased cost in the switching fabric. Yet, a persistent problem is the output nonuniformity, for instance, caused by nonuniform output addressing probability due to nonuniform nature of *Virtual Channel Identifier*(VCI)'s of the ATM header. Then, a natural question arises: can we smooth out the traffic nonuniformity felt at individual outputs? In answering this question, we develop an algorithm for an effective buffer sharing with respect to the intensity of incoming traffic.

Let us revisit Figure 2.9. As stated earlier, with U°=1 the achieved cell loss probability is minimum when U* becomes 1. U*=1 means $\lambda_1^* = \lambda_2^*$. In Figure 2.9, when U*=1, $\lambda_1^* = \lambda_2^*$=0.4. Now we change the input traffic nonuniformity U° to 3 and to 10. As the case of U°=1, wherever the minimum cell loss probability occurs, the output traffic intensities $\lambda_1^* and \lambda_2^*$ become equal at 0.4. This observation suggests that whenever output traffic intensity at each output is identical, the cell loss is minimal. Below we extend this observation to an efficient output buffer sharing algorithm to alleviate the performance degradation by traffic nonuniformities.

The proposed algorithm is an effort to make the *Output Traffic Pressure* felt at individual outputs identical, where the *Output Traffic Pressure*(OTP) is defined as the ratio of output arrival traffic intensity to output buffer space allocated to each output. The OTP is then described by $\frac{\lambda_j^*}{L_j}$, with L_j being the output buffer size at output port *j*. We allocate more buffers, taken out from lightly loaded outputs(or with low traffic pressure), to the outputs with large traffic intensities so that the corresponding OTP become smaller, while OTP's for the lightly loaded outputs increase. As a result, all outputs experience the same OTP, which gives the optimal cell loss performance with given buffer budget. Notice that with the fixed output buffer size L with uniform traffic of λ, the OTP for each output is represented by λ /L.

Let us consider the optimal buffer allocation for each output. Traffic nonuniformity patterns change with time. Efficient buffer allocation should consider this dynamic nature

of traffic nonuniformity. In order to capture dynamic nonuniform traffic patterns perceived at outputs, we define an observation time interval T in slot time at output port controllers (OPC's). The detailed implementation is beyond the scope of this paper and we simply assume T is known to us. During T, each OPC measures the number of cells arrived, including those lost, and computes $E_T\left\{\lambda_j^{\bullet}\right\}$, where $E_T\{\bullet\}$ denotes the expectation. Let us assume that a shared buffer manager collects the information on $E_T\left\{\lambda_j^{\bullet}\right\}$ from all OPC's every T time slot.

Assume the total output buffer budget is equal to NL, with L being the output buffer size for uniform traffic condition. Let $\overline{\lambda} = \frac{1}{N}\sum_{j=1}^{N} E_T\left\{\lambda_j^{\bullet}\right\}$. The optimal buffer allocation algorithm for output j is then given by

$$L_j = \frac{L\, E_T\left\{\lambda_j^{\bullet}\right\}}{\overline{\lambda}}. \tag{42}$$

This algorithm is optimal because the OTP felt at each output will be the same constant given as $\frac{\overline{\lambda}}{L}$. The maximum possible performance to be obtained by the algorithm is always bounded by the uniform traffic condition case as evidenced in previous figures. Obviously, in order to improve the performance more, it will be at the expense of the increased total output buffer size. Additional buffer allocation to each output is immediately given by

$$L_j = \frac{L(1+\alpha)E_T\left\{\lambda_j^{\bullet}\right\}}{\overline{\lambda}} \tag{43}$$

where α reflects the additional buffers to be allocated. Notice in Equation (28) that the value of L_j becomes a decimal fraction in general. With this decimal fraction, the OTP calculated for each output will be identical. In practice, however, we need to take the

nearest integer value for the real buffer allocation, defined by $int\{L_j\}$. Thus, ensuing buffer allocation will be suboptimal.

Apply now the OTP based buffer allocation to the bigroup example used in this section. The nonuniform parameters used: U°=1, $T_g = \begin{pmatrix} 0.4 & 0.6 \\ 0.3 & 0.7 \end{pmatrix}$, λ=0.4, $d_i = 1.5$. $d_o = 3.0$. By this condition, output group 2 will be heavily congested. Output traffic intensities are evaluated at $\lambda_1^{\bullet} = 0.25$, $\lambda_2^{\bullet} = 0.625$.

For uniform traffic condition with λ=0.4 and K=L=10, the cell loss performance is again shown to be the best as exemplified in Figure 4.3. The allocation in Equation (42) will be reduced to

$$L_j = \frac{10\lambda_j^{\bullet}}{\lambda}. \tag{44}$$

where $E_T\left\{\lambda_j^{\bullet}\right\}$ is approximated by $\lambda_j^{\bullet}$. The buffer sizes for output groups 1 and 2 are 5.714 and 14.286, respectively. With these values, the OTP's are the same at 0.04375. After taking the nearest integer, however, the buffer sizes for group 1 and 2 become 6 and 14, which result in unequal OTP values at 0.0417 and 0.0446. At m=2, the cell loss probability without buffer sharing is placed at 4.44×10^{-7}. By output buffer sharing, the cell loss probability is reduced to 5.85×10^{-9}, which shows significant improvement with two orders of magnitude difference. As mentioned earlier, further improvement is possible by increasing total output buffer size. In Figure 4.3, we have shown the curves for a 10% buffer increase (equivalent to 1 cell size at each output) and a 20% increase (equivalent to 2 cell size), respectively. Using Equation (43), we obtain buffer

allocations. For the 10% increase, 6 for group 1 and 16 for group 2, and for the 20% increase, 7 for group 1 and 17 for group 2 are the resulting allocations. Observe that with the 10% buffer increase, the cell loss probability becomes 3.44×10^{-9}, and for the 20% increase the cell loss probability is 2.07×10^{-10}. Notice that the cell loss performances with the 20% increased buffer size becomes very close to that of the uniform traffic case.

Chapter 5 Conclusion and Discussion

ATM networks present two basic problems. One is that networks are faced with the difficult task of satisfying the needs of connections requiring different QoS, but sharing the same physical resources, e.g., bandwidth and buffers. Different kinds of performance criteria should be considered simultaneously when ATM switching modes are designed. Applications of control policies providing several performance levels to different traffic classes is also anticipated. Another is that performance objective values may differ greatly between media, as shown by the example for voice and data. This second problem leads to the need for a service quality control scheme, e.g., a priority scheme, that enhances efficiency of resource sharing among different media.

It has been understood that a trade-off exists between cell loss and delay in an ATM network: increasing buffer size of a switch reduces cell loss probability but results in a longer queueing delay. As more of future network applications impose real-time constraint, it is an important issue for an ATM switch to guarantee its service for delay sensitive applications. A longer queueing delay at a switch can not be overlooked. We have provided a quantitative analysis of the cell loss caused by excessive queueing delay. We have analyzed and derived the queueing delay distribution for a typical input and output port address pair in a generic nonblocking ATM switch. The Effective Cell Loss has been introduced as a unified performance measure that properly incorporates both cell losses: one due to buffer overflow and another due to excessive queueing delay.

The future traffic to an ATM switch may not always be uniform. In the numerical examples, we have shown a severe performance degradation at a switch, caused by various nonuniform traffic patterns. This implies that any switch designs with uniform traffic assumption may be too optimistic for realistic traffic. In particular, the dominant nonuniformity is shown to be the output group address probability matrix, which is determined by the destination address embedded in the cell header. We also found that such nonuniformity results in a poor performance guarantee for delay constrained traffic, mainly caused by excessive output queueing delay. One thing is certain that the optimal cell loss performance is achieved whenever output traffic intensities at individual outputs become identical. Based on this finding, we provide an algorithm for an effective output buffer sharing. This algorithm based on the OTP finds the optimal buffer allocation for outputs, where the OTP at each output beomes equal. A significant improvement on the cell loss performance is obtained using this algorithm.

Although our proposed buffer sharing algorithm is effective and simple to implement for one class of traffic, future ATM networks must be able to handle different classes of traffic that genetically convey different requirements of QoS. Such requirements calls for a better buffer management at ATM switches. In the foregoing research, we propose to continue to derive optimal buffer management scheme that would offer solutions to guarantee different QoS requirements as well as to improve buffer efficiency.

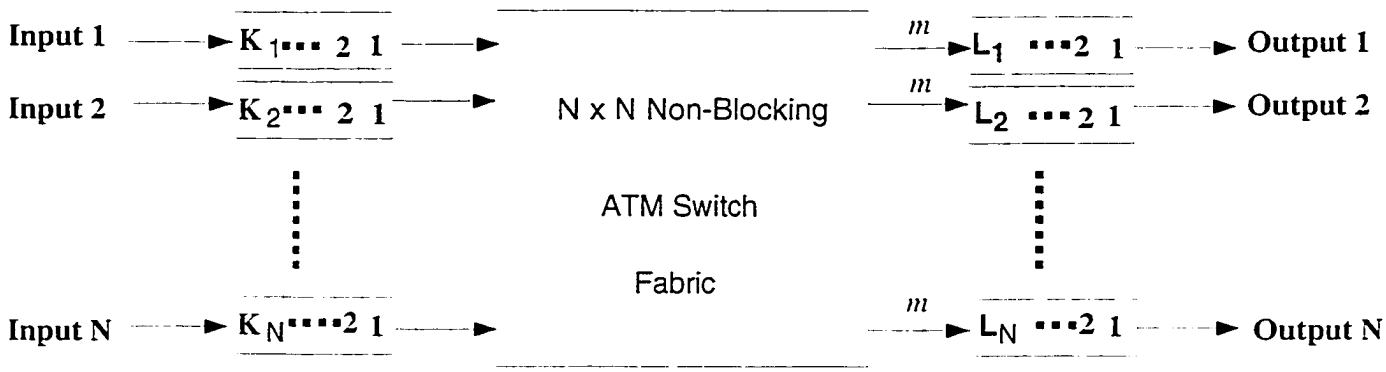

Figure 1 . N x N Non-blocking ATM switch with a Speed-up Factor, ***m***.

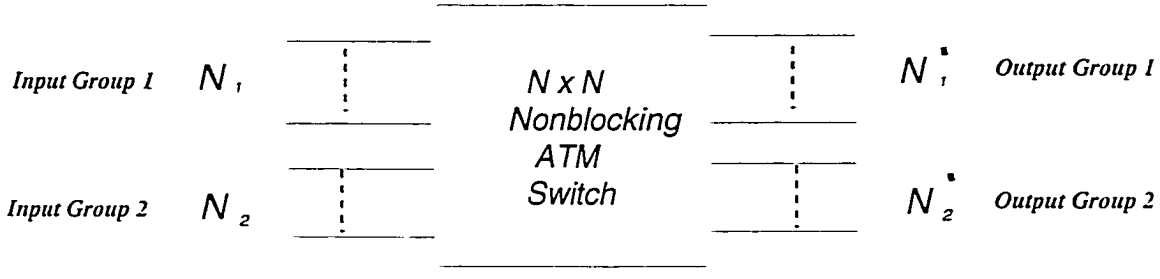

Figure 2.2. Nonuniform example given by two statistically identical groups

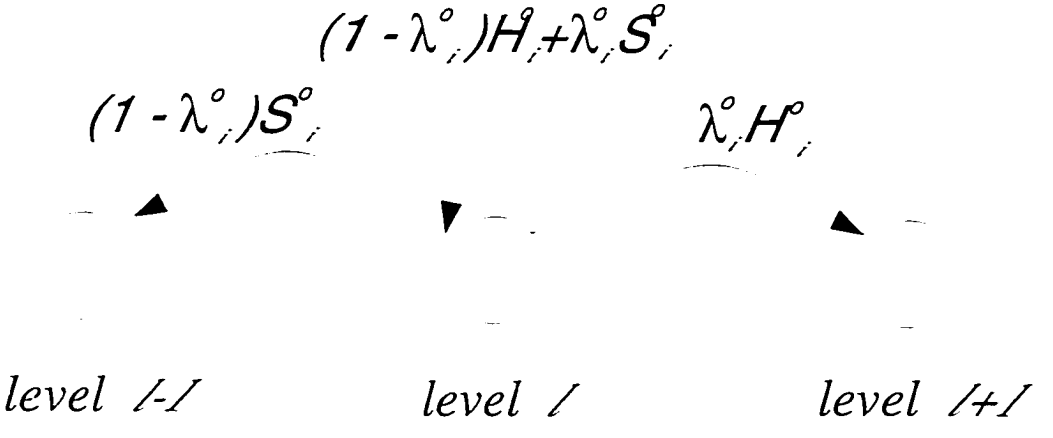

Figure 2.3. Level transition diagram from level l at input i for $l = 1, \ldots, K$

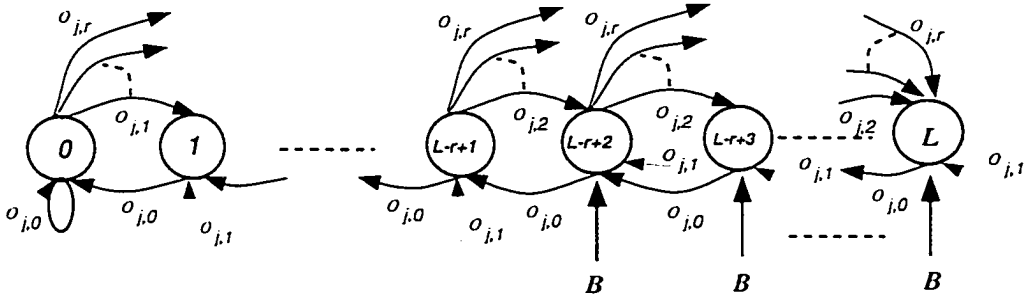

Figure 2.4. State transition diagram of output queueing process with blocking states marked by ***B*** for $r = 0, 1, \ldots m$.

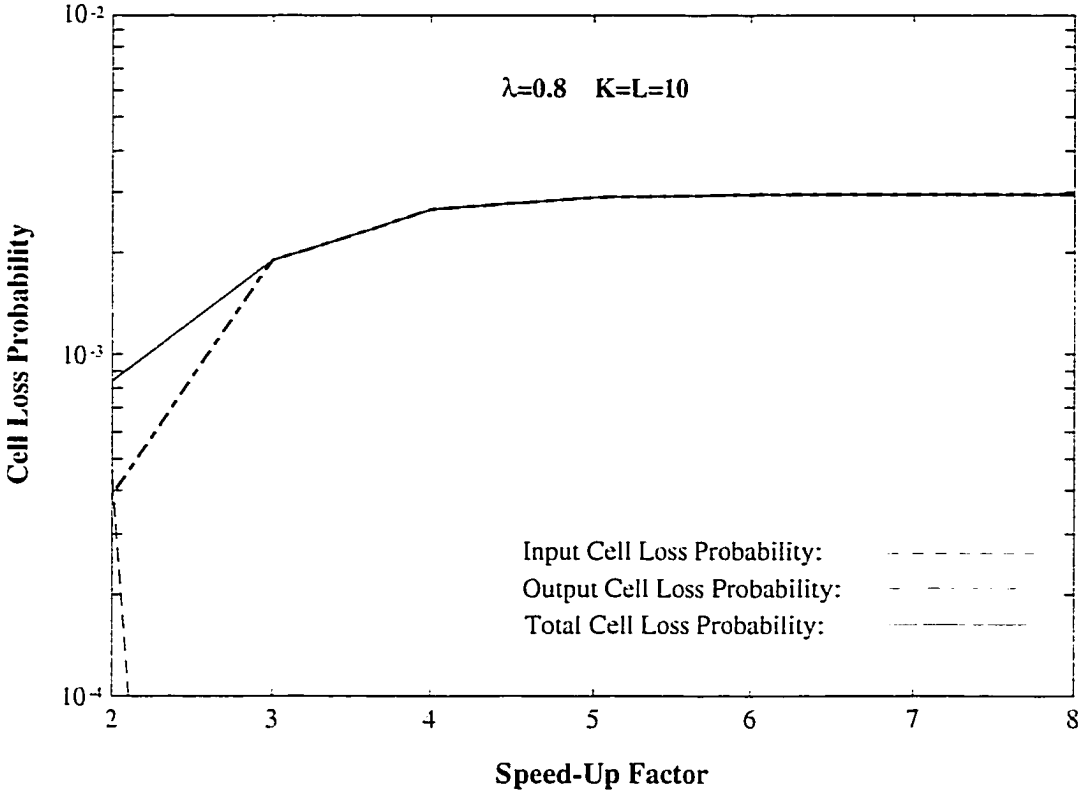

Figure 2.5a. Total Cell Loss contributed by input/output cell loss

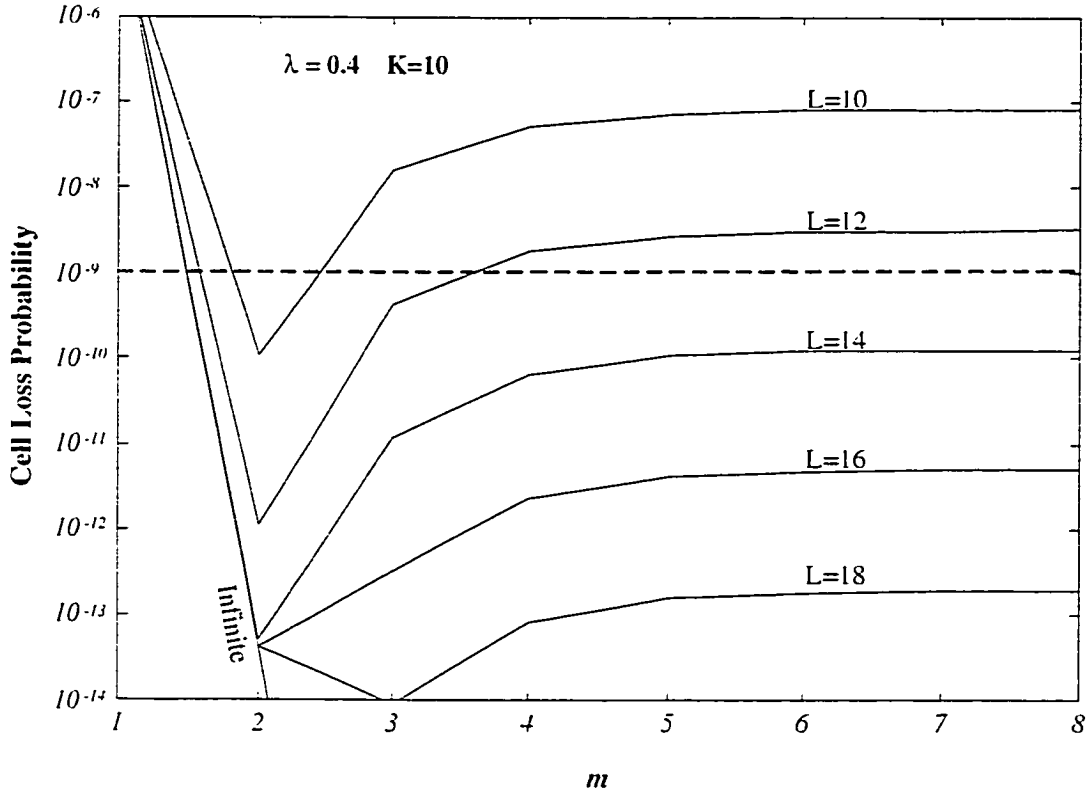

Figure 2.5b. Effect of increased *m*.

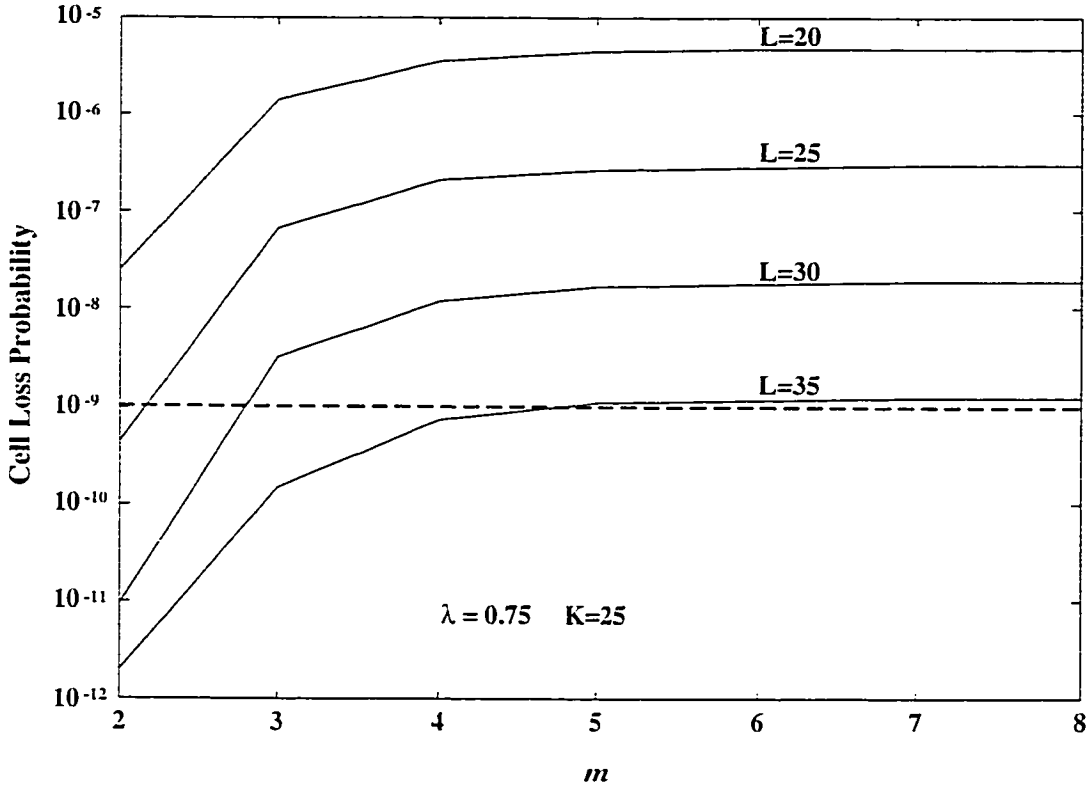

Figure 2.5c. Effect of increased queue size

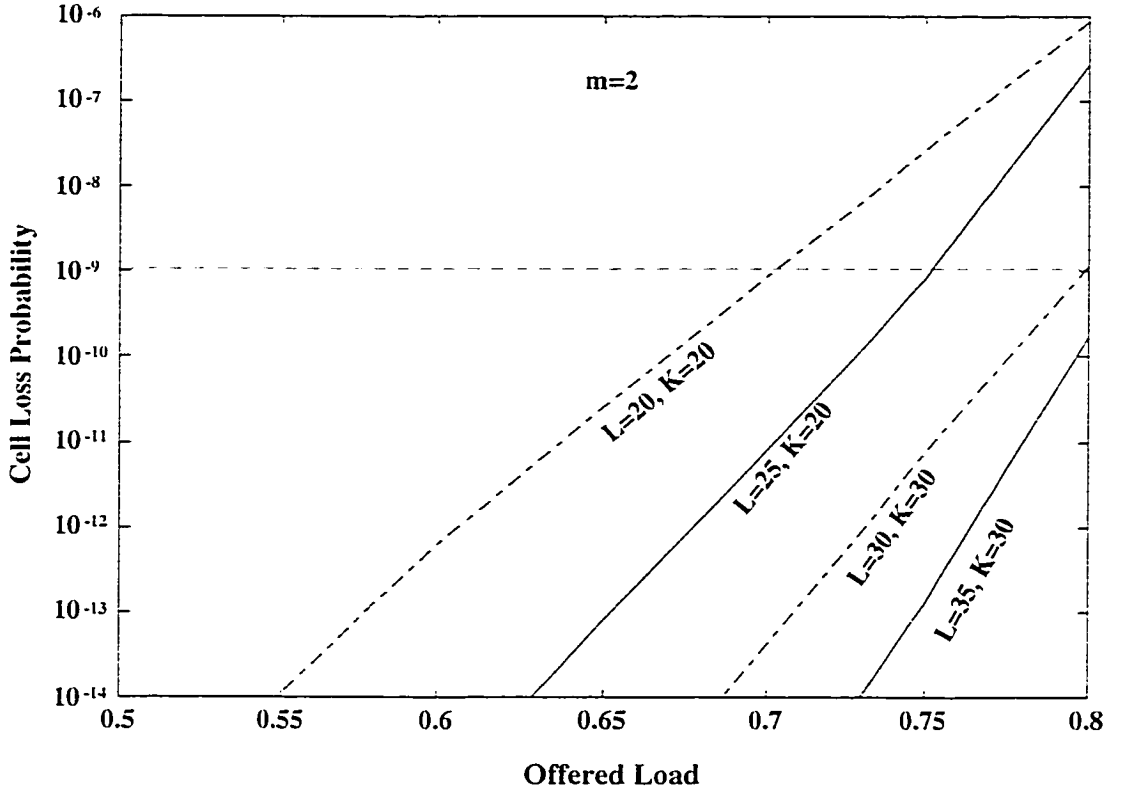

Figure 2.5d. CLP vs. increased queue size

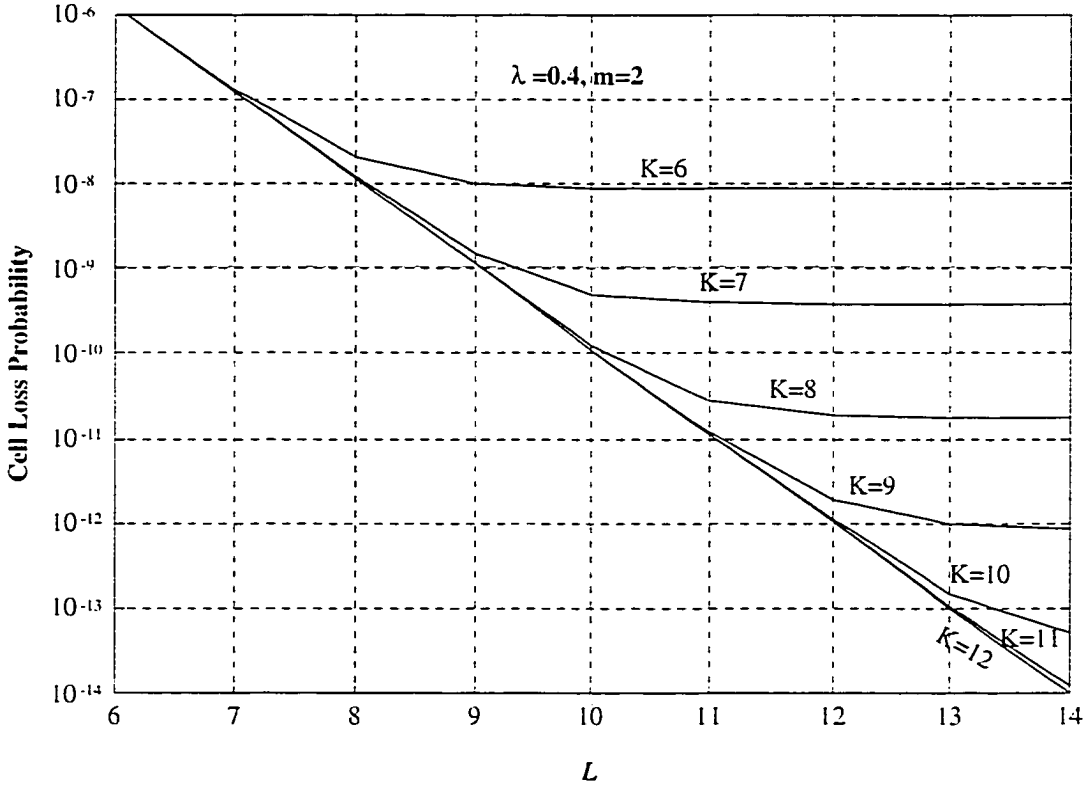

Figure 2.6a. CLP vs. different set of input/output queue size: $\lambda = 0.4$

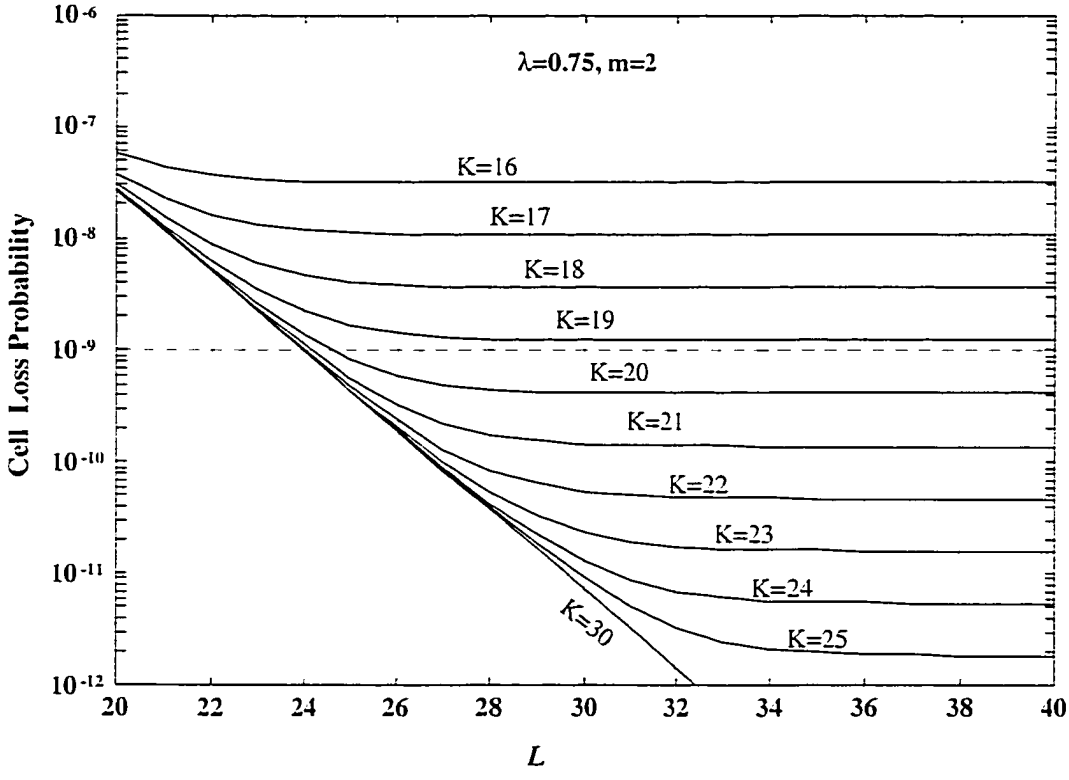

Figure 2.6b. CLP vs. different set of input/output queue size: $\lambda = 0.75$

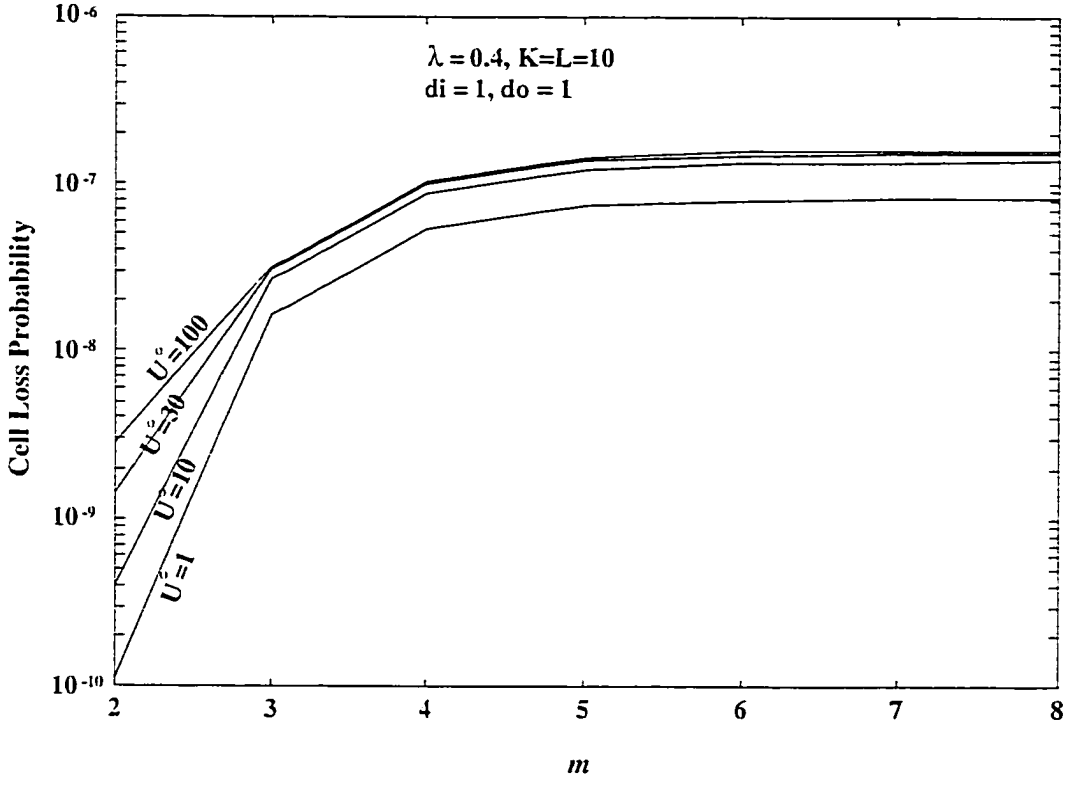

Figure 2.7a. Effect of input nonuniformity

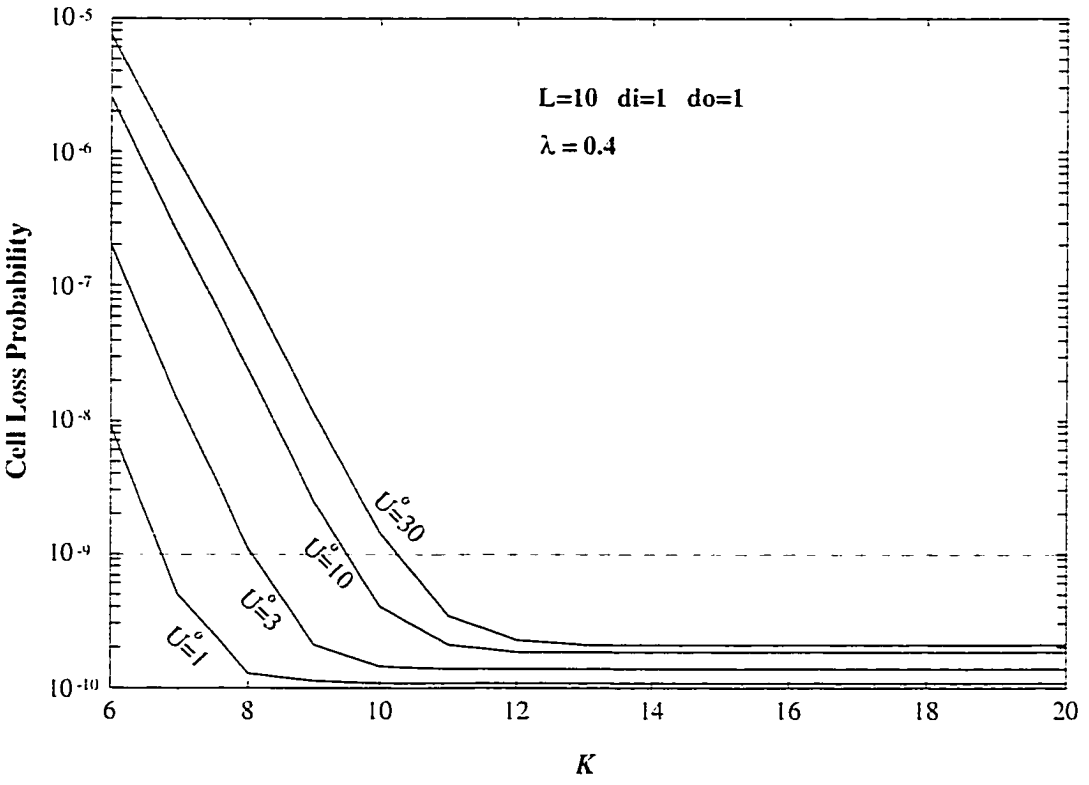

Figure 2.7b. CLP vs. input queue size with different input nonuniformity

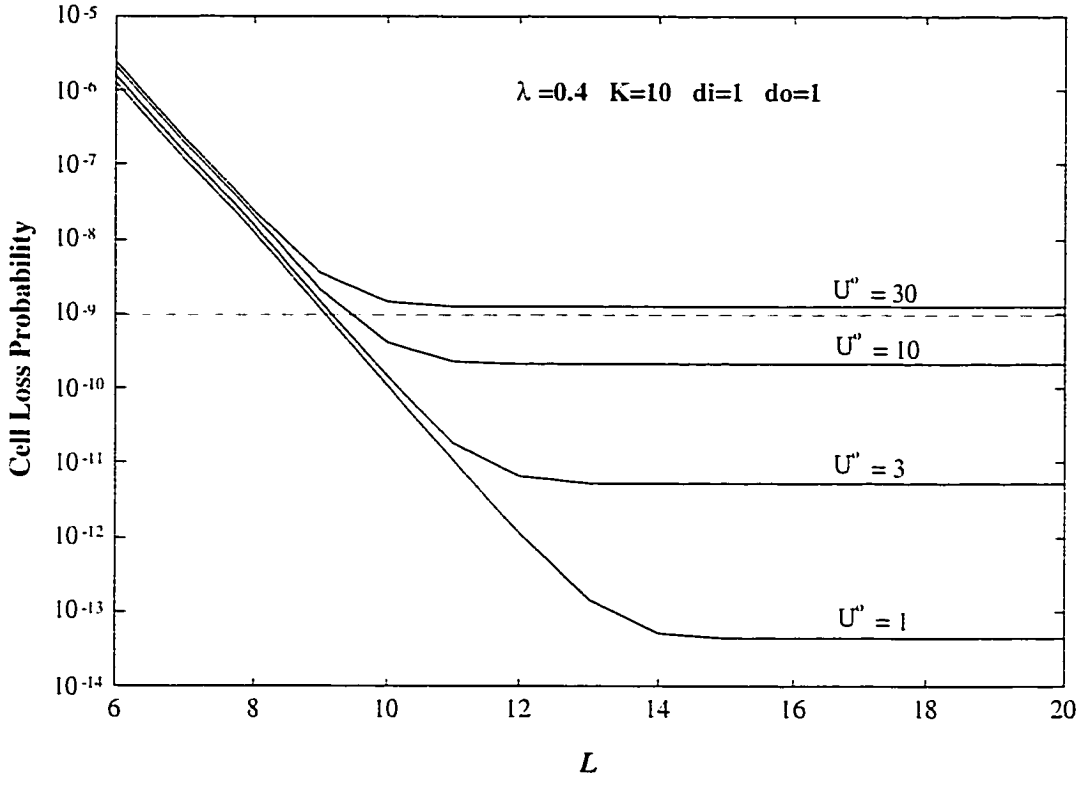

Figure 2.7c. CLP vs. output queue size with different input nonuniformity

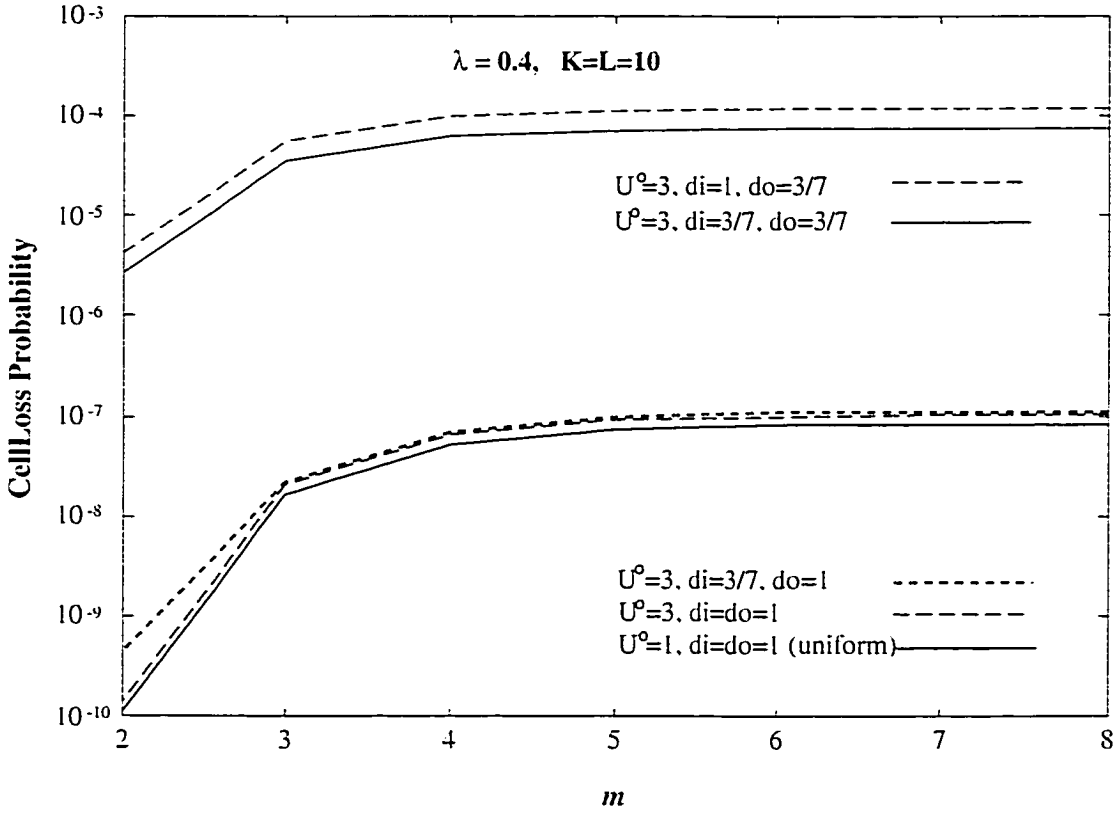

Figure 2.8a. Effect of group size ratio

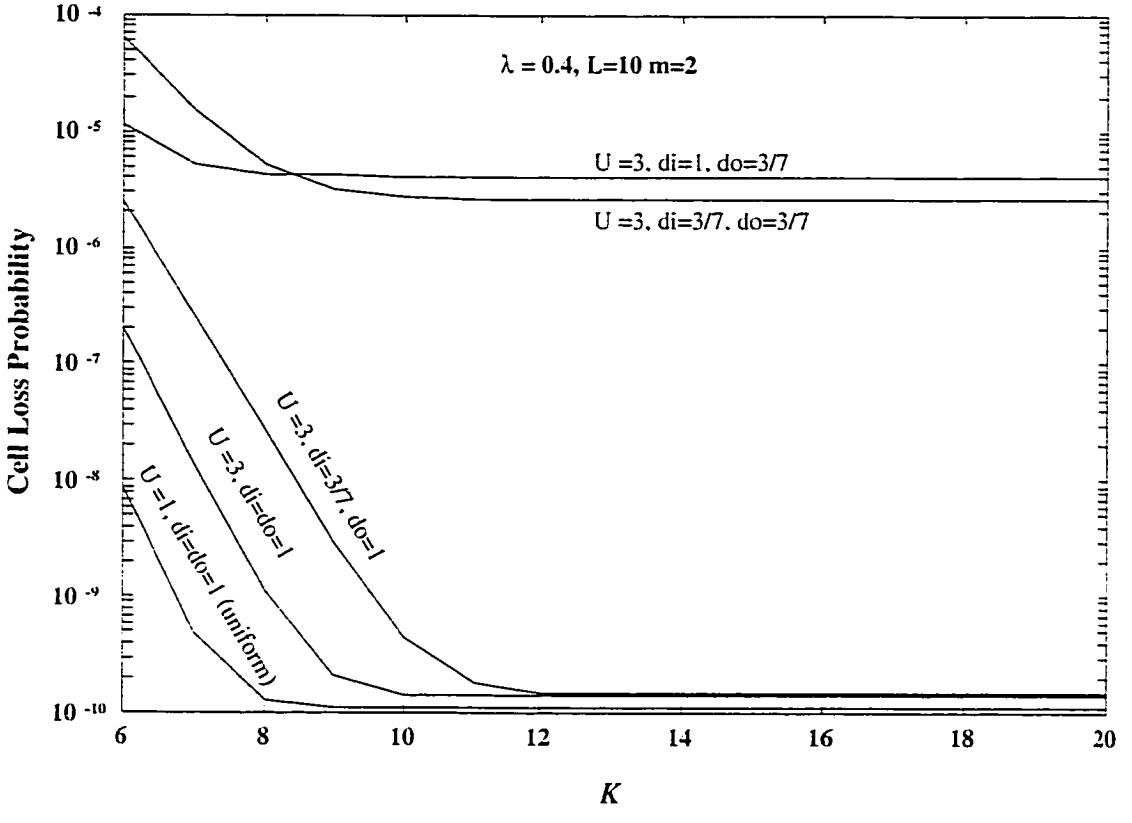

Figure 2.8b CLP vs. input queue size with different group size ratio

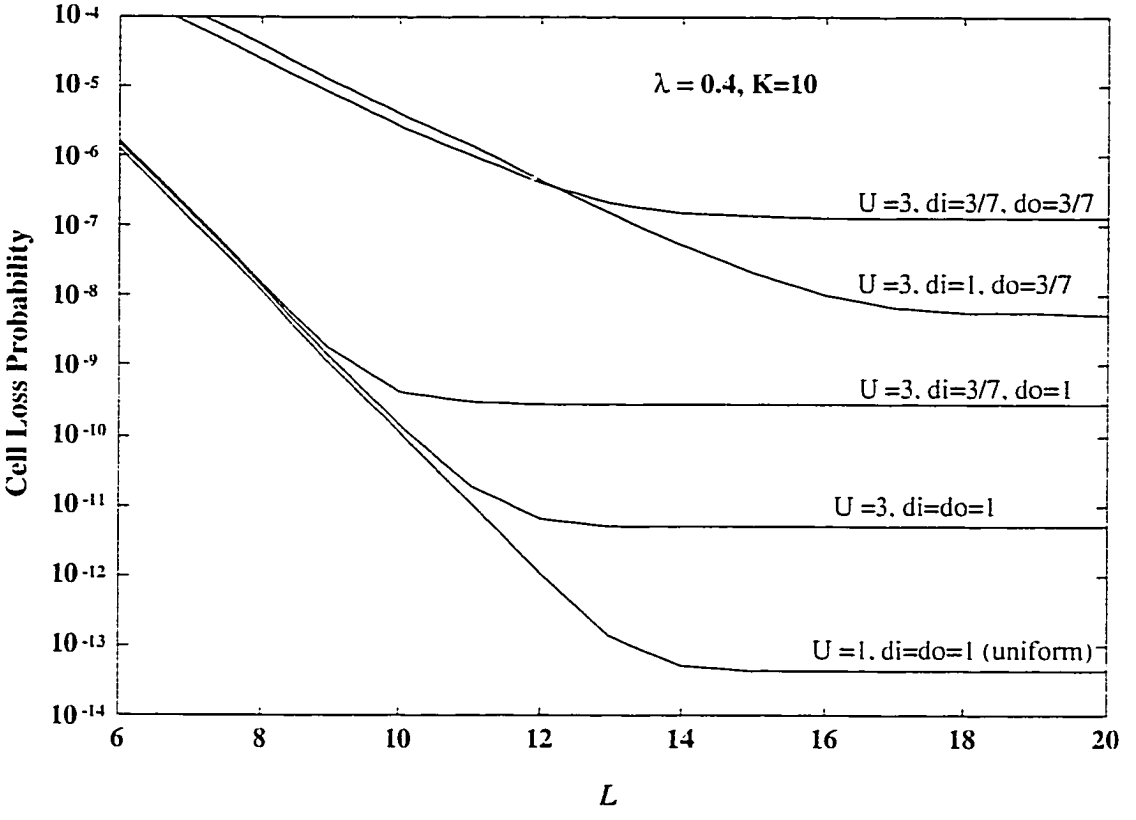

Figure 2.8c. CLP vs. output queue size with different group size ratio

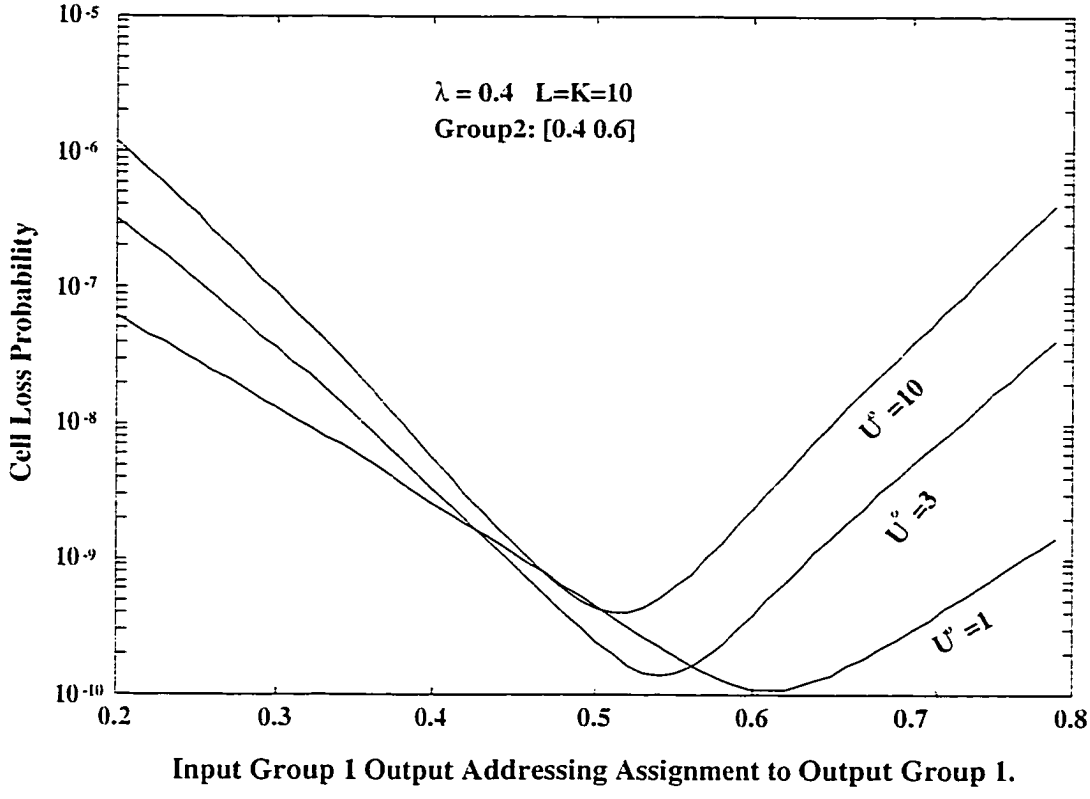

Figure 2.9. Effect of output group addressing assignments

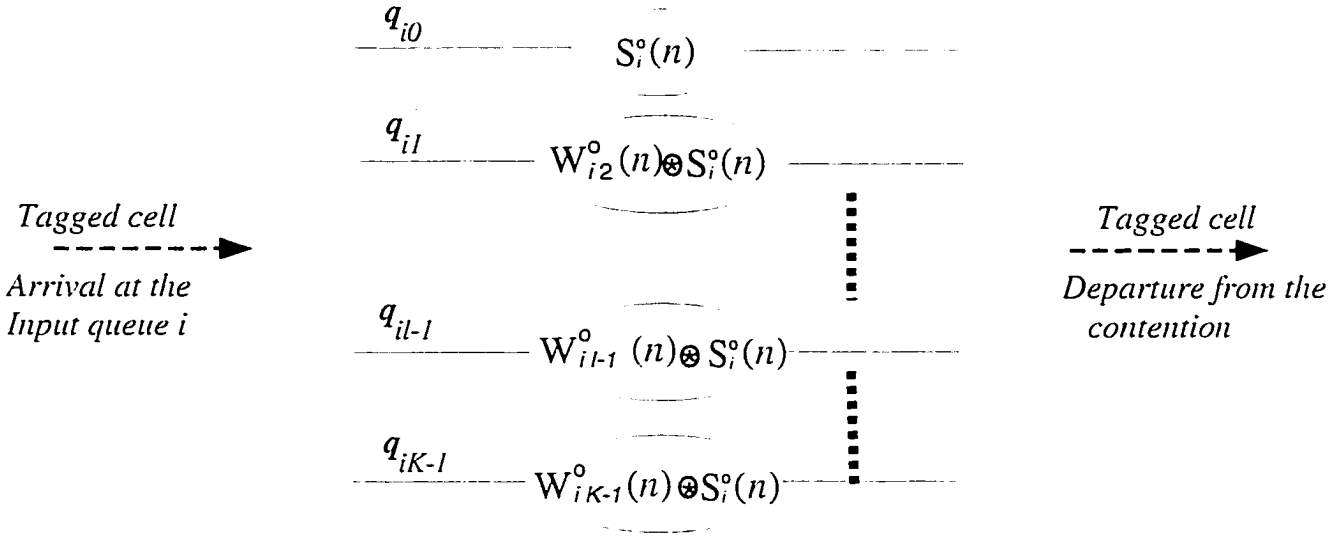

Figure 2.10. The input queueing delay process at the input queue *i*.

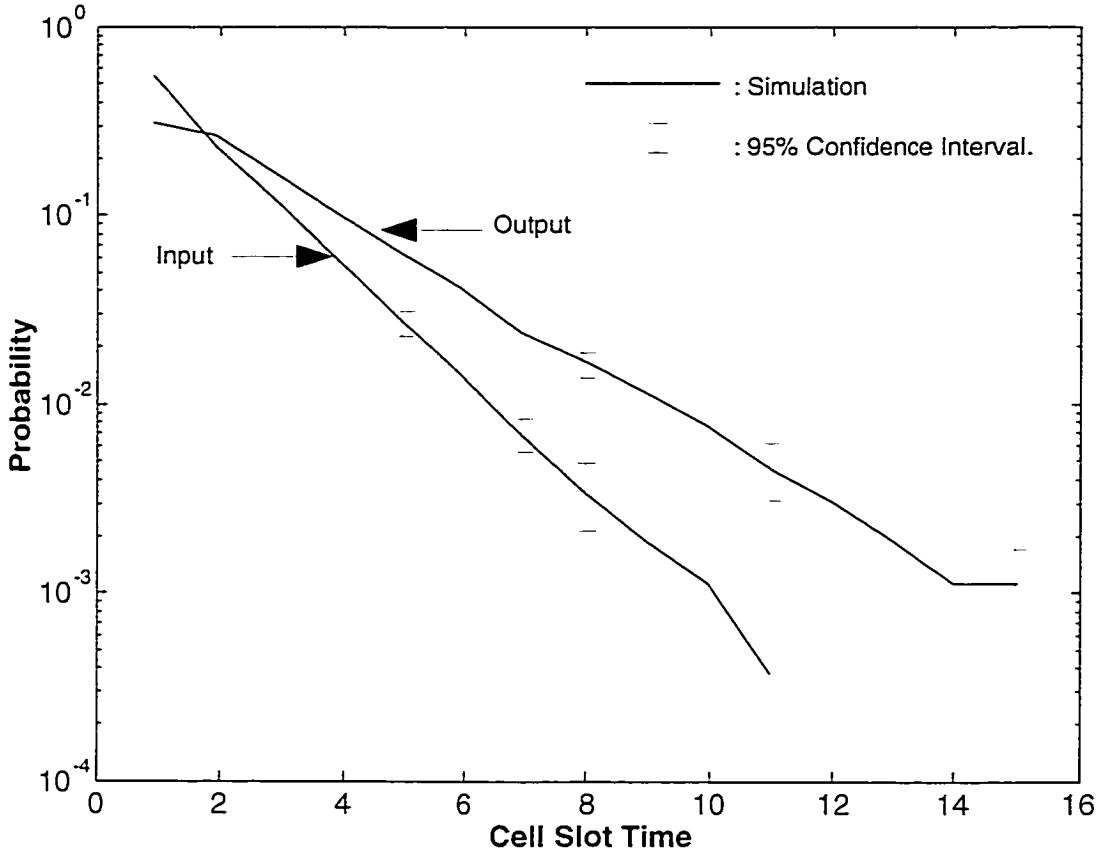

Figure 2.11a. Input and Output Queueing Delay Distribution

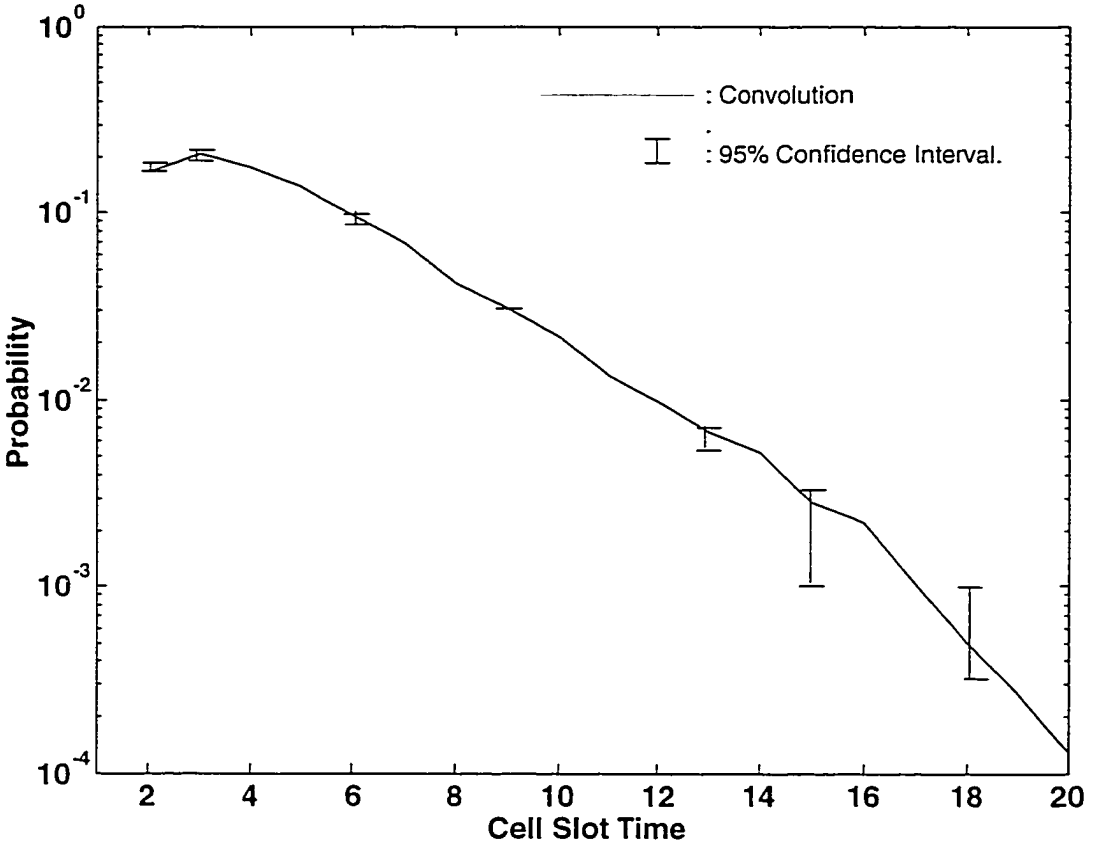

Figure 2.11b. Total Queueing Delay Distribution.

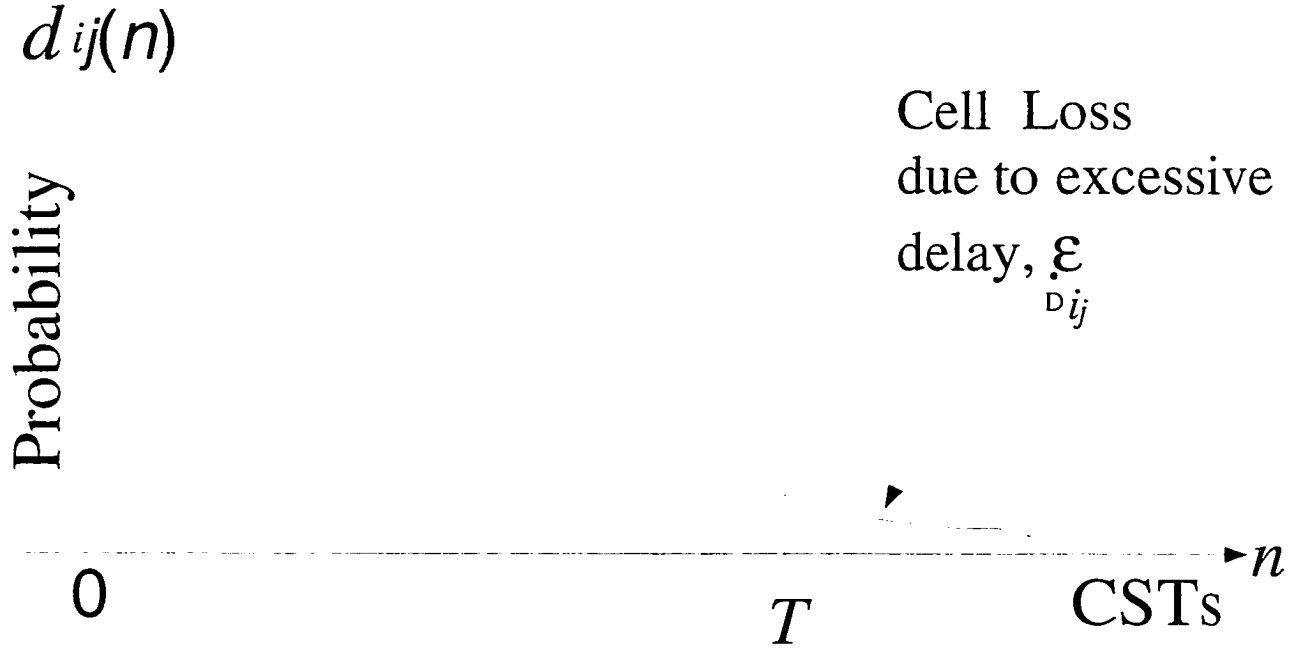

Figure 2.12. CLP due to an excessive delay of time constraint traffic.

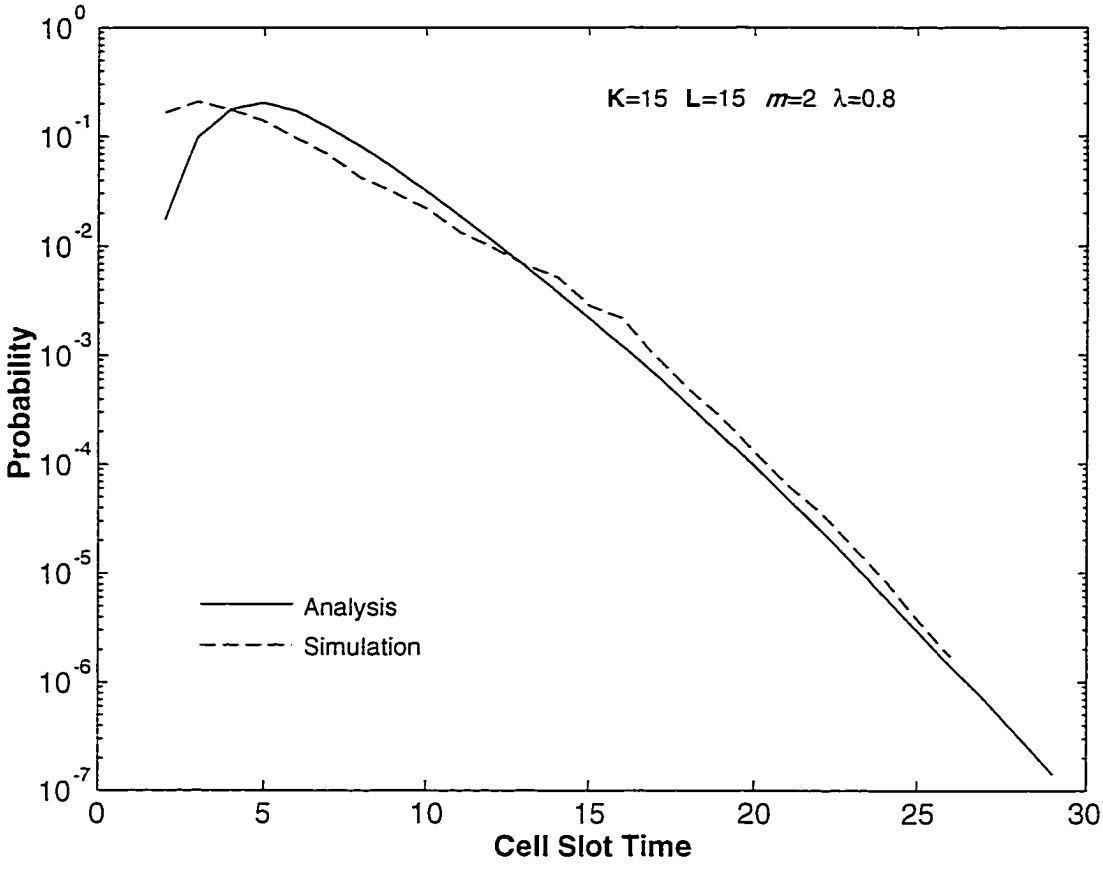

Figure 2.13. Total Delay Distribution: Uniform Traffic.

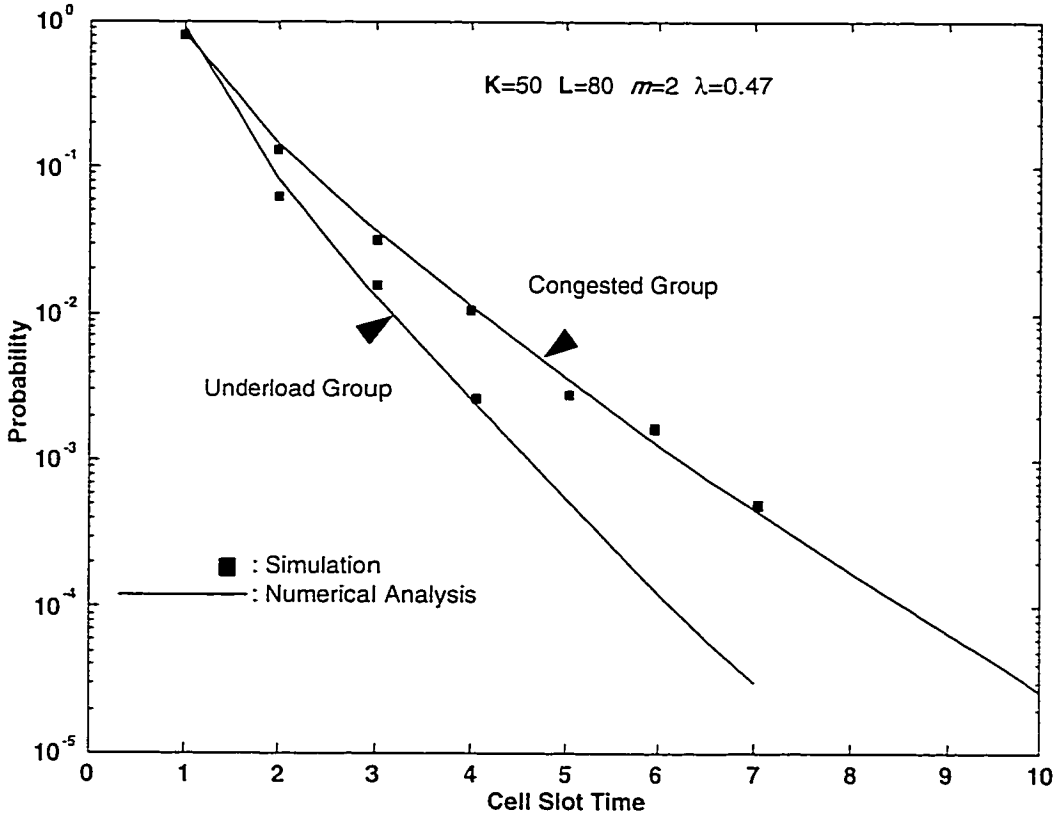

Figure 2.14a. Input queueing delay distribution: Nonuniform traffic case

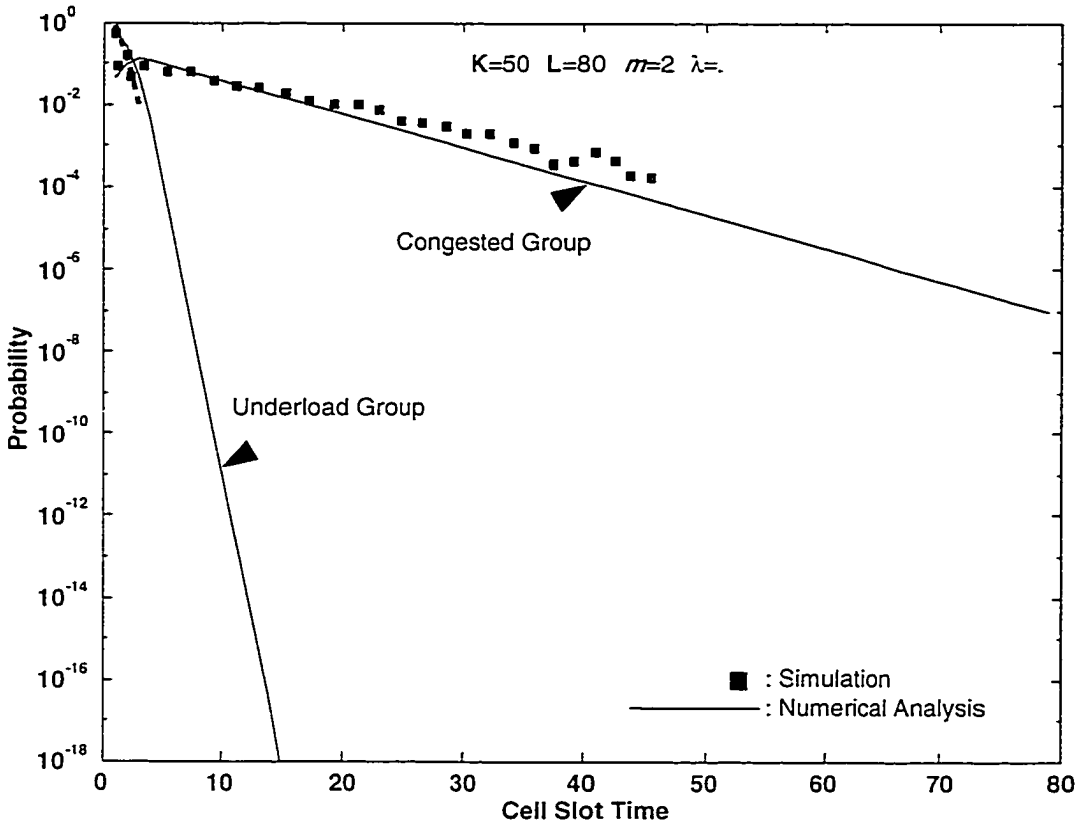

Figure 2.14b. Output queueing delay distribution: Nonuniform traffic case

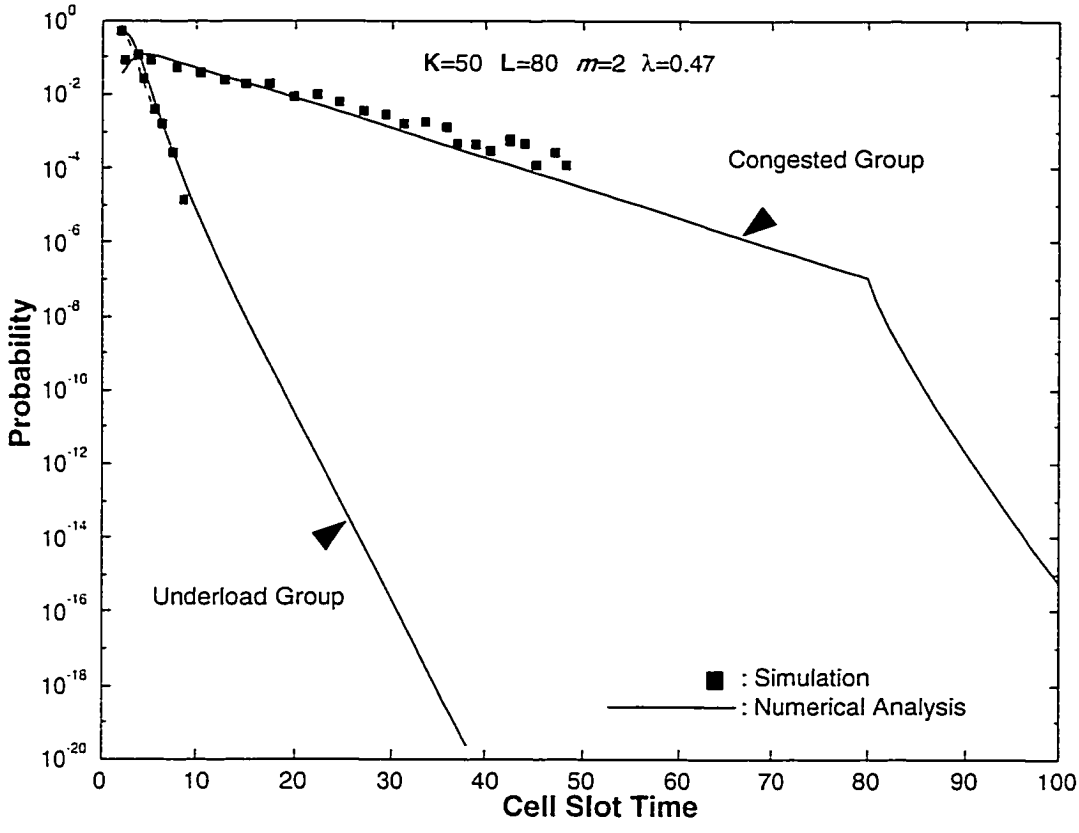

Figure 2.14c. Total queueing delay distribution: Nonuniform traffic case

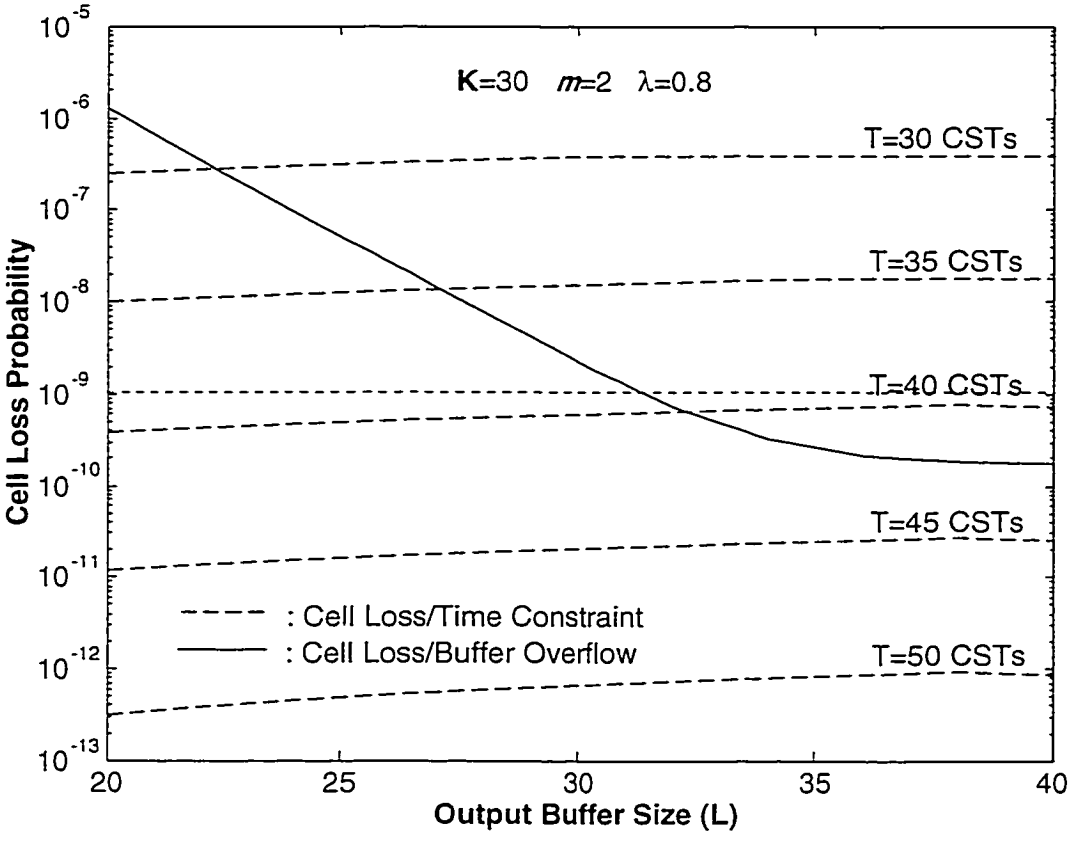

Figure 3.1a. CLP vs. L: Uniform traffic case

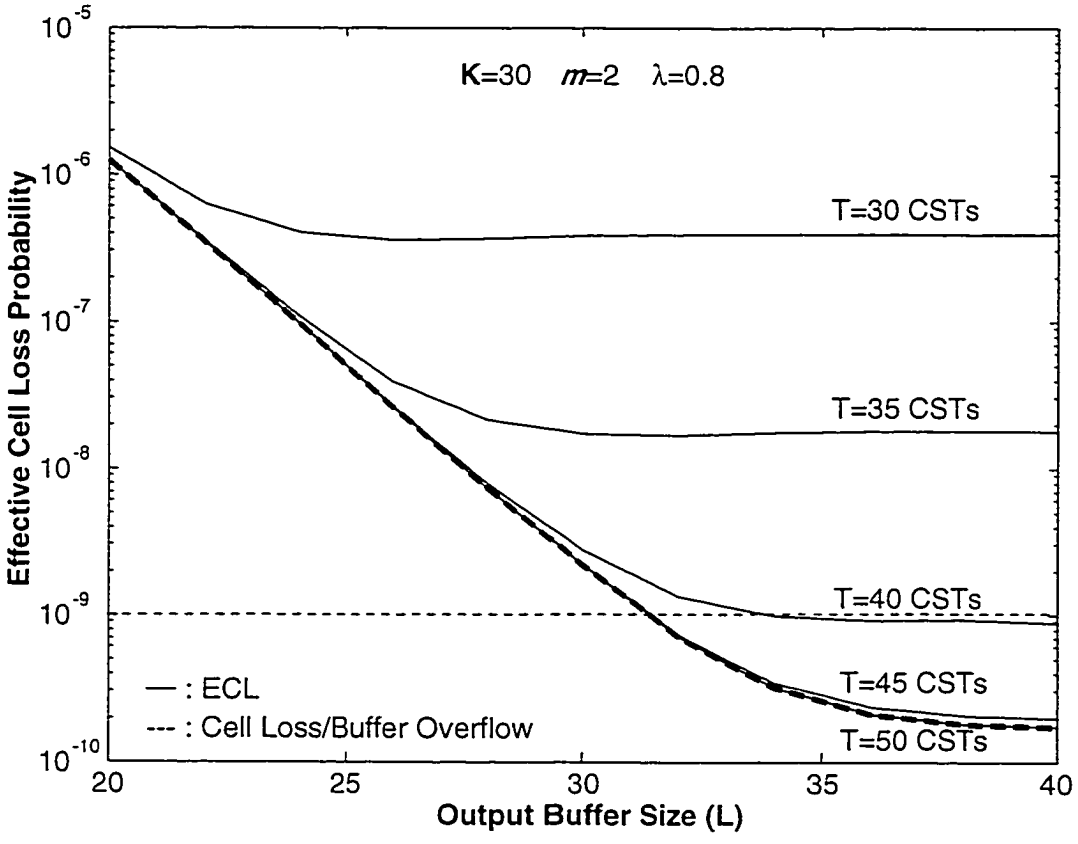

Figure 3.1b. ECLP vs. L with given T: Uniform Case

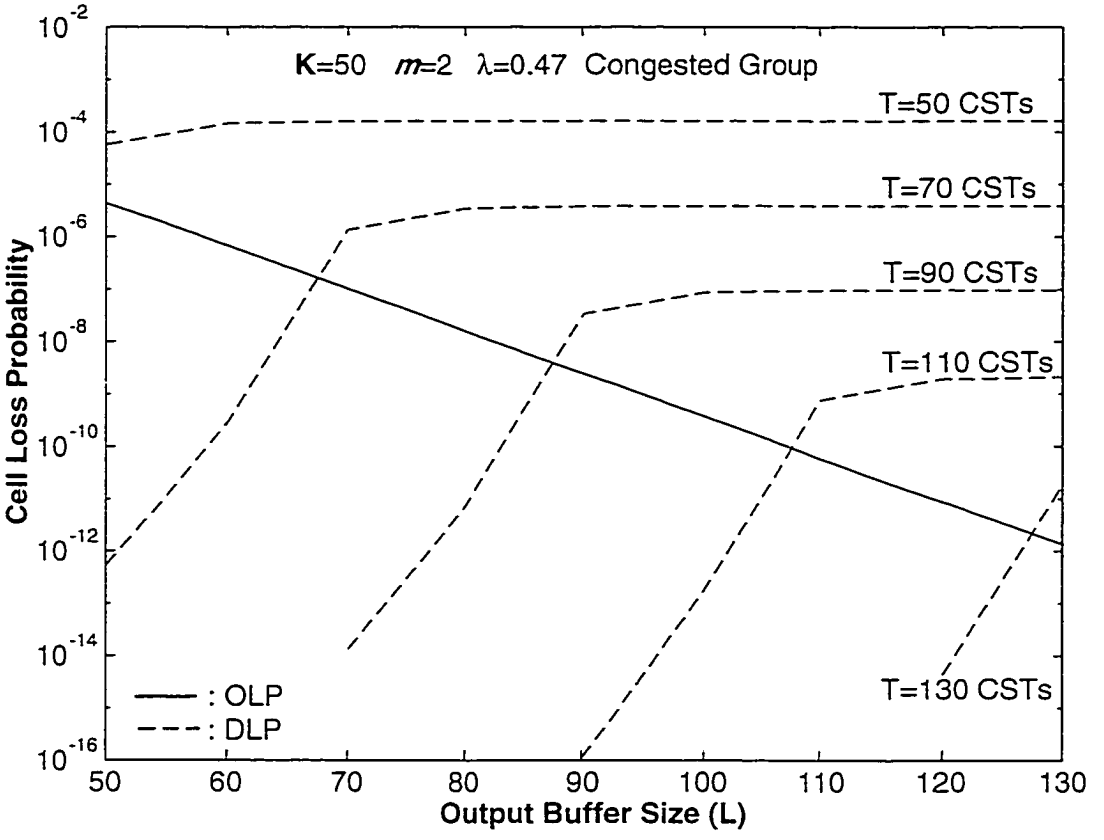

Figure 3.2a. CLP vs. L: Nonuniform Traffic Case

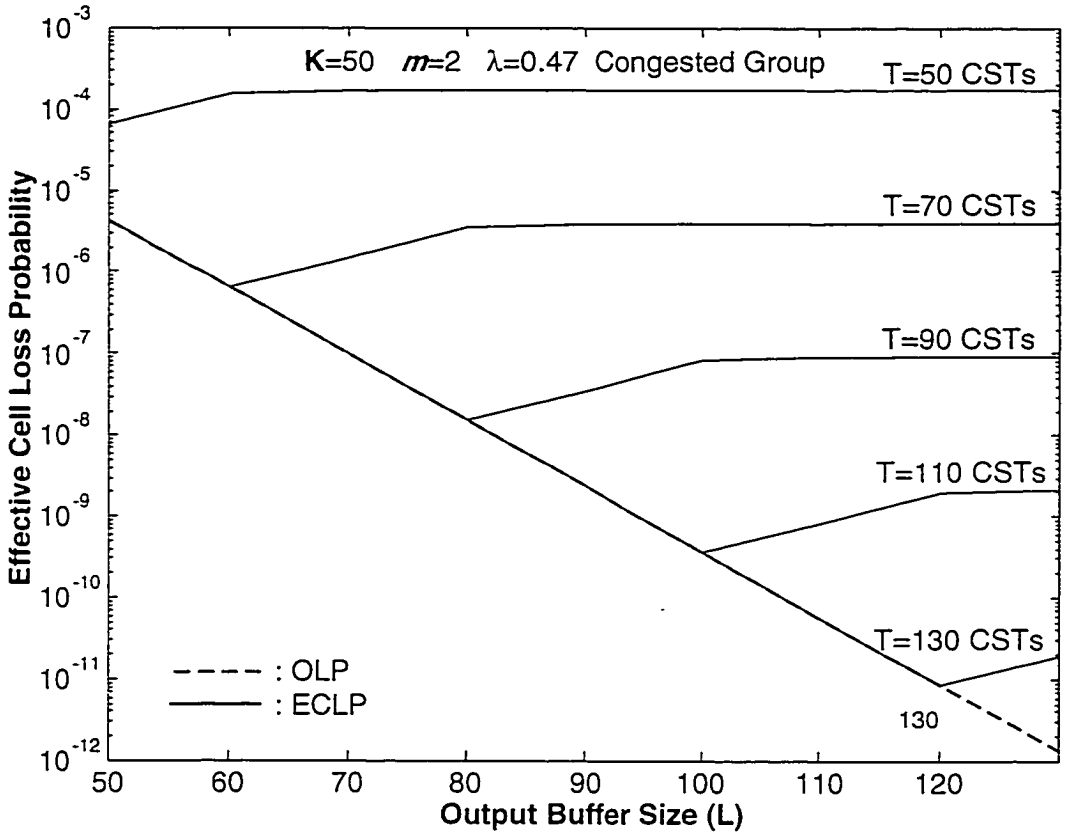

Figure3.2b. ECLP vs. L with given T: Nonuniform case

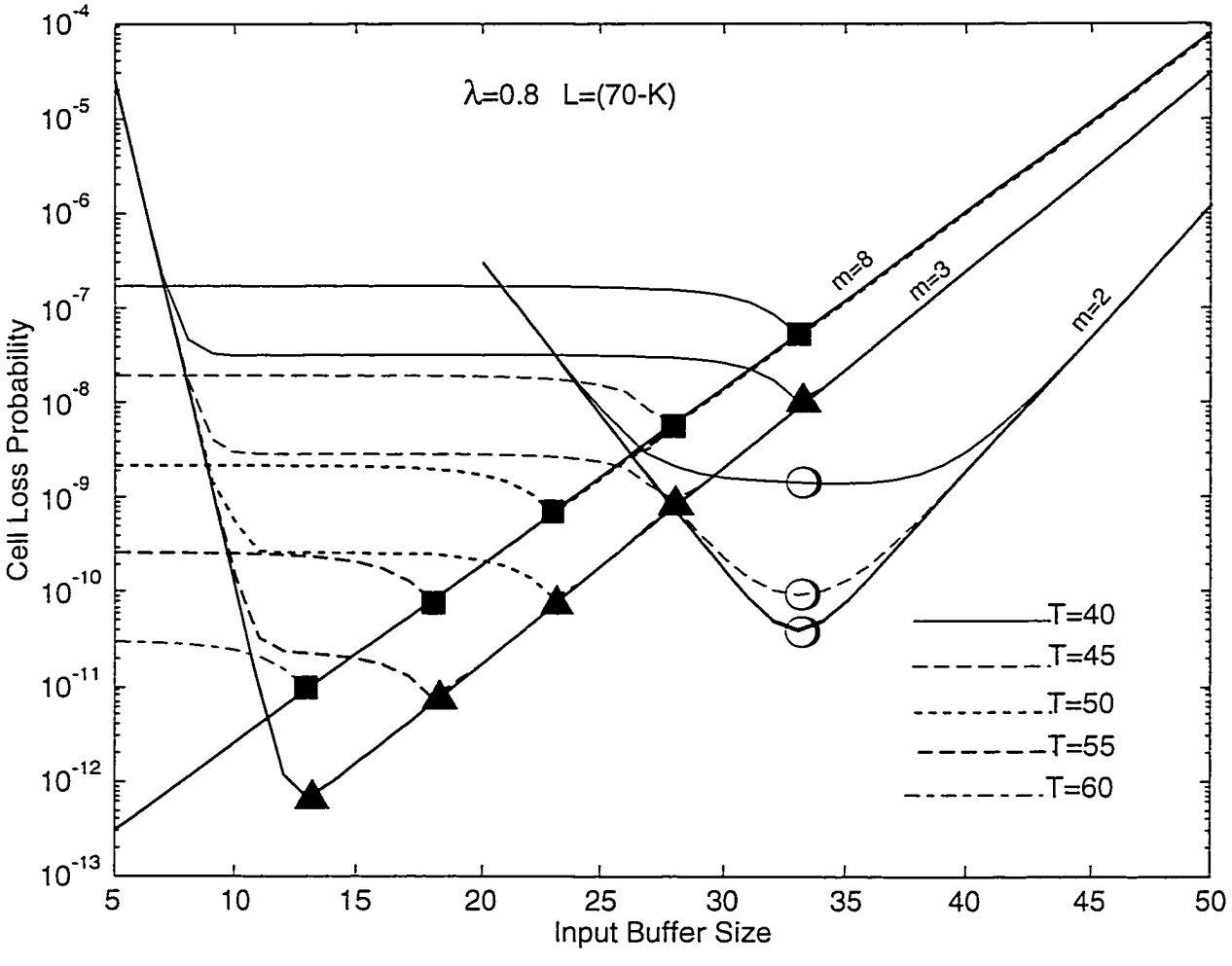

Figure 4.1. Optimal Buffer Allocation w.r.t. ECLP : Uniform Traffic Case

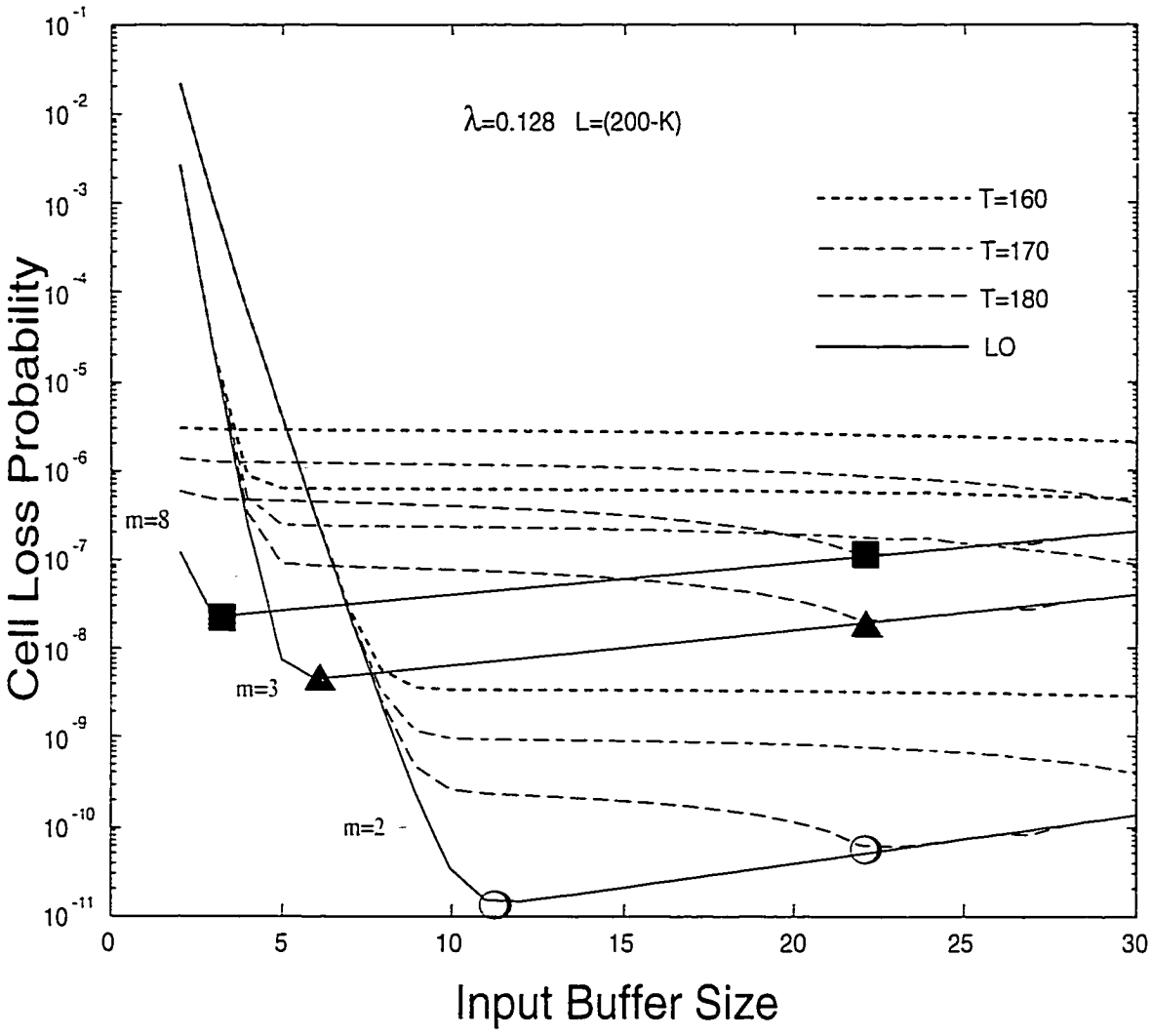

Figure 4.2. Optimal Buffer Allocation w.r.t ECLP: Nonuniform Traffic Case

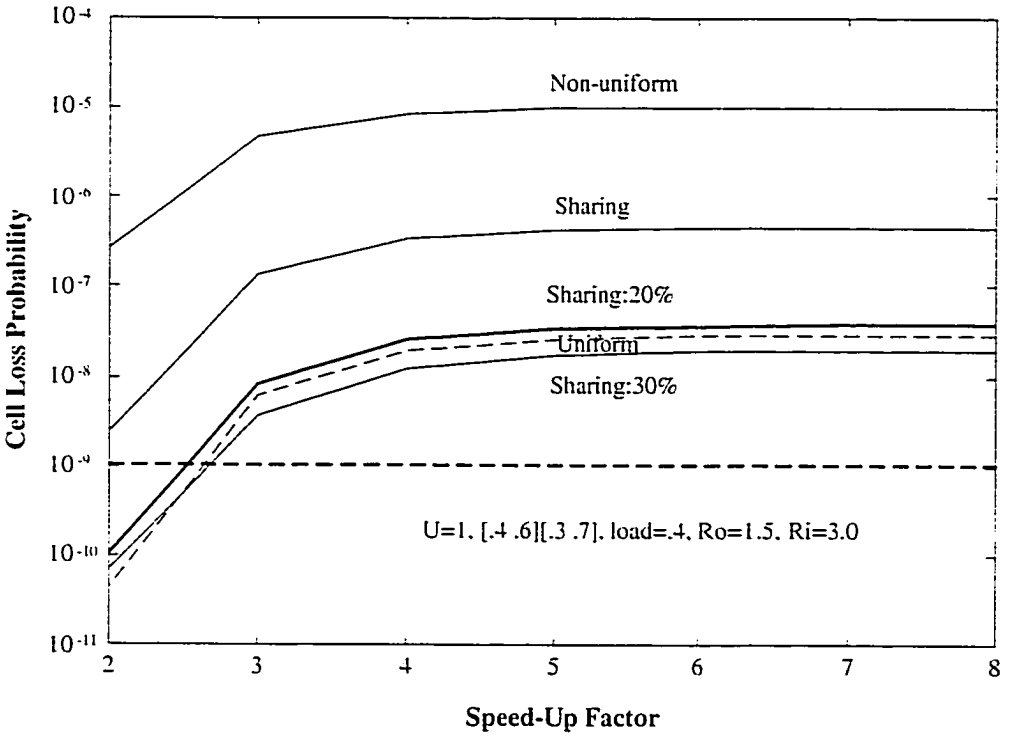

Figure 4.3. Improvement by output buffer sharing.

REFERENCES

[1] CCITT, Draft Recommendation I.361, "ATM Layer Specification for B-ISDN," Geneva, Jan. 1994.

[2] ATM Forum, LAN Emulation SWG Drafting Group, "LAN Emulation over ATM Specification - Version 1.0," Jan. 1995.

[3] M. de Prycker, *Asychronous Transfer Mod: solution for broadband ISDN*, Ellis Horwood Limited, 1991.

[4] H. Saito, *Asychronous Transfer Mode*, Arctech House, 1994.

[5] M. Laubach, "Classical IP and ARP over ATM," IETF RFC 1577, Apr. 1994.

[6] D. Dubois, N. Georganas and E. Horlait, "A QoS Selector for Multimedia Applications on ATM Networks," *Proceedings of IEEE ICC'94*, New Orleans, May 1994.

[7] D. S. Ahn and M.J. Lee, "Effective Cell Loss Analysis of a Nonblocking ATM switch with Nonuniform Traffic," *Proceedings of IEEE ICC'95*, Seattle, June 1995.

[8] T. G. Robertazzi, ed., *Performance Evaluation of High Speed Switching Fabric and Networks*, New York: IEEE Press, 1993.

[9] D.S. Ahn and M.J. Lee, "Cell Loss Analysis and Design Trade-Offs of Nonblocking ATM Switches with Nonuniform Traffic," *IEEE/ACM Trans. on Networking*, vol. 3, no. 2, pp. 199-210, April 1995.

[10] M.J. Lee and S.-Q. Li, "Performance Trade-Offs in Input/Output Buffer Design for a Nonblocking Space-Division fast packet switch," *Int. J. Digital Analog Commun. Syst.*, vol. 4, pp. 21-31, 1991.

[11] M.J. Lee and S.-Q. Li, "Performance of a Nonblocking Space-Division Packet Switch in a Time Variant Nonuniform Traffic Environment", *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1515-1524, Oct. 1991.

[12] S.-Q. Li, "Nonuniform traffic analysis on a nonblocking space division packet switch," *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 21-31, July 1990.

[13] J.S.-C. Chen and T.E. Stern, "Throughput analysis, optimal buffer allocation, and traffic imbalance study of a generic nonblocking packet switch," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 439-449, Apr. 1991.

[14] D.S. Ahn and M.J. Lee, "Packet Loss Analysis of Nonblocking ATM Switches with Nonuniform Traffic and Performance Improvement by Output Buffer Sharing," *Proceedings of IEEE GLOBECOM'95*, Houston, Nov. 1993.

[15] I. Illiadis and W. E. Denzel, "Analysis of Input and Output Queueing," *IEEE Trans. Commun.*, vol. 41, no. 5, pp. 731-740, May 1993.

[16] M.J. Karol, M.G. Hluchyj and S.P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans. Commun.*, vol. COM-35, no. 12, pp. 1347-1356, Dec. 1987.

[17] M.F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach.* Baltimore, MD: John Hopkins Univ. Press, 1981.

[18] J. W. Causey and H. S. Kim, "Comparison of Buffer Allocation Schemes in ATM Switches: Complete Sharing, Partial Sharing, and Dedicated Allocation," *Proceedings of IEEE INFOCOM'94*, Montreol. June 1994.

[19] F. Kamoun and L. Kleinrock, "Analysis of shared finite storage in a computer network node environment under general traffic condition," *IEEE Trans. Commun.*, vol.28, no. 7, July 1980.

[20] G. J. Foschini and B. Gopinath, "Sharing memory optimally," *IEEE Trans. Commun.*, vol. 31, no. 3, Mar. 1983.

[21] S. X. Wei, E. J. Coyle, and M. T. Hsiao, "An optimal buffer management policy for high-performance packet switching," *Proceedings of IEEE GLOBECOM'91*, Dec. 1991.

[22] I. Cidon, L. Georgiadis, R. Guerin, and A. Khamisy, "Optimal Buffer Sharing"*IEEE JSAC*, vol. 13, no. 7, Sept. 1995.

[23] L. Tassiulas, Y. Hung, and S. Panwar, "Optimal buffer control during congestion in an ATM network node" *Proceedings of IEEE GLOBECOM'93*, Houston, Nov. 1993.

[24] A. Pattavina, "Nonblocking Architectures for ATM Switching," *IEEE Commun. Mag.*, vol. 31, no. 2, pp. 38-48, Feb. 1993.

[25] S.-Q. Li, "Performance of a Nonblocking Space-Division Packet Switch with Correlated Input Traffic," *IEEE Trans. Commun.* , vol. 40, no. 1, pp. 97-108, Jan. 1992.

[26] G. E. Daddis, Jr. and H. C. Torng, "A Tasonomy of Broadband Integrated Switching Architectures," *IEEE Commun. Mag.*, vol. 27, no. 5, pp. 32-42, May. 1989.