Queueing Network Models For Multiple Class Broadband Integrated Services Digital Networks

Nelson Luis Saldanha da Fonseca

CENG Technical Report 94-25

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213)740-4579

December 1994

Queueing Network Models for Multiple Class Broadband Integrated Services Digital Networks

by

Nelson Luís Saldanha da Fonseca

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Computer Engineering)

December 1994

Dedication

To Evely, Tiago, Marilia and Nelson.

Acknowledgments

During the Ph.D. process I was fortunate to rely on the support of several good friends. I am deeply grateful to:

Professor John A. Silvester for his sound advice, unconditional support friendship, as well as for providing me with an excellent research environment. Dr. Silvester stood by my side throughout the whole process and I benefit a lot from his wisdom.

Professors Peter Danzig, Deborah Estrin, Kai Hwang, Victor O. K. Li and Dan Moldovan, my Ph.D. committee members, for their valuable suggestions, kindness and availability.

Evely Boruchovitch, my lovely wife, whose companionship makes life a wonderful experience, for always encouraging my professional career. Her love, heartening support and dedication are the key of my success. Evely was a unique friend during the Ph.D road. May God bless our fruitful journey.

Nelson Baptista da Fonseca and Marilia Saldanha da Fonseca, my parents, for their unconditional love which makes me strong and healthy enough to face life in an adult perspective. Their nurturing support are of paramount importance to all my accomplishments.

Lita Arcana, Bill Bates, Diane Demetras, Joe Lumunsad, Milly Montenegro, Regina Morton, Lucille Stivers and Mary Zittercob for their friendly administrative support.

My colleagues Dr. Anastassio Economides, Dr. Ram Khrisna, Dr. Arthur A. Y. Lin, Tekai Liu, Gilberto Mayor, Dr. Thomas Papavissilios, Dr. Stanley Wang Dr. Syu-Je Wang and Dr. John Yang for their friendly support along the way. Special thanks to Dr. Stanley Wang, my U.S.C. friend, for his support since my arrival, and Drs Arthur Li and Syu-Je Wang for their sound advice.

Professors Daniel Menascé and Daniel Schwabe for their enthusiastic support since the early days of my undergraduate studies, for believing on me and for being such good role models.

Dr. Sergio E. R. Carvalho, Cristina Pallazo, Dr. Angela B. Podkameni, Dr. Jose L. M. Rangel for their friendly pre and post departure support.

Dr. James A. Tweedie for taking care of the three of us in such a warm way.

Anna and Anoosh Hami, Elizabeth and Nelson Niremberg, Gilda Menasce, Cristiane Peixoto, Donald and Lizette Quarnstron, Marcus Silberman and Yasser Taima for their constant support as well as for making our life in U.S.A. much more pleasant.

Last but not least, Tiago Boruchovitch Fonseca, our adorable son for the unique joy he brought to me in the last stage of this dissertation.

Table of Contents

List of Tables

List of Figures

Abstract

The future communication network, Broadband Integrated Services Digital Network (B-ISDN), will carry voice, video and data in an integrated fashion. Different multimedia applications have different Quality of Service requirements. Several approaches are possible to cope with diverse QOS requirements. The key is to provide flexibility and efficiency without excessive complexity. We believe that a valuable tool towards this goal is the provision of multi-priority mechanisms.

Being able to estimate end-to-end performance is of paramount importance not only for the traffic control, but also for dimensioning the future B-ISDN. Efficient algorithms and approximate models for the analysis of computer / communication networks are available. However, such knowledge cannot be applied to the analyses of B-ISDN since it assumes that the networks flows are renewal processes.

This dissertation is concerned with both investigating appropriate queue ing network models for B-ISDN networks and with evaluating the impact of introducing a multiple class selective discard mechanism at the cell level. Concerning selective discard mechanism, we: i) introduce an algorithm for solving a queue with N priority classes in N steps , ii) prove a loss rate conservation law for non-renewal processes and iii) evaluate the impact of choosing different push out policies. Regarding queueing networks, we: i) define a framework for queueing networks with Markov modulated flows, and ii) extend this framework for priotized flows. Furthermore, we show how the queueing network framework can be used for the analysis of ATM networks.

Chapter 1

Introduction

1.1) The Objective of this Dissertation

The future communication network, Broadband Integrated Services Digital Network (B-ISDN), will carry voice, video and data in an integrated fashion. The new spectrum of multimedia applications will definitely change the way we communicate and live.

The advent of fiber optics has brought new technological constraints and changes in network switching and the protocol hierarchy [1], [2]. A new multiplexing scheme called Asynchronous Transfer Mode (ATM) [3] has been adopted by CCITT to be the standard for B-ISDN networks. ATM uses packet switching with fixed length packet (cells) and it offers connection and connectionless services. *Statistical multiplexing* was chosen instead of a reservation scheme (such as Synchronous Transfer Mode-STM) due to its efficient bandwidth utilization when multiplexing bursty sources.

The dramatic decrease of transmission delay, due to high data rate, and consequently the increasing significance of propagation delay render ineffective mechanisms which rely on network feedback to regulate traffic flow. As a result, new congestion control schemes [4]-[7] are being investigated, among these we mention: Admission Control, Source Rate Control and Selective Discard.

Admission Control [8], [9] makes the decision whether a new connection should be accepted or not based on its traffic descriptors. Common descriptors are: average rate, peak rate, burst period duration, to name a few. The decision to accept a connection takes into consideration maintenance of the Quality of Service (QOS) for the existing connections, as well as the ability to provide the QOS requested by the new connection. The QOS is mainly expressed in terms of average delay and average loss probability. Other performance measures such as average number of consecutive losses or delay jitter may also be used.

Source Rate Control [10], [11] monitors whether the stream produced by a source conforms with the declared traffic parameters. In case of violation, some policies reject the violating cells while others just mark them. In an overflow situation the marked cells are the first to be discarded.

Selective Discard [12], [13] drops cells in an overflow situation according to the cell's priority level. The cell's priority level may be either dynamically or statically assigned. The priority level expresses the cell's relevance, for instance, in a hierarchical video codec, cells with decreasing priority are less significant to the picture quality. Priorities may be assigned to all cells from a source or to substream of cells from a source. A selective discard mechanism is completely defined by the buffer management and by the push out policies. The buffer management policy establishes which priority classes can occupy a specific buffer slot, and the push out policy defines which cell is discarded in an overflow situation.

Different multimedia applications have different Quality of Service requirements [17], [18]. For example, a file transfer may tolerate large average delays, however it is very sensitive to loss. In contrast, a voice conversation may tolerate some degree of cell loss but it is sensitive to delay. Several

approaches are possible to cope with diverse QOS requirements. At one extreme, network resources can be dimensioned to satisfy the most demanding application. This solution clearly leads to network underutilization. At the other extreme, the network can be designed with complex control mechanisms to accommodate different requirements. The key is to adopt a balance - providing flexibility and efficiency without excessive complexity. We believe that a valuable tool towards this goal is the provision of multi-priority mechanisms, which give the flexibility needed to improve the utilization of network resources.

The integration of voice, video and data signals in a single network has brought many interesting problems to telecommunication traffic theory. The traffic stream from a multimedia source is strongly correlated and this non-renewal nature represents a major departure from the Poisson / renewal world [19]-[22]. Consequently, researchers have developed methods for the analysis of queues with multimedia traffic input such as: i) fluid flow model [23]-[26]; ii) generating function approach [27], and iii) Matrix Analytical method. In this dissertation, we choose to use Markov Modulated Processes (Matrix Analytical) to model multimedia traffic [28], [29]. Markov modulated processes have sucessfully been applied to the analysis of voice, video and data sources and several procedures to compute the parameters of a Markov modulated process given a specific configuration of sources have been defined [30]-[34].

Being able to estimate end-to-end performance is of paramount importance not only for the traffic control, but also for dimensioning the future B-ISDN. Efficient algorithms and approximate models for the analysis of computer / communication networks are available. However, such knowledge cannot be applied to the analyses of B-ISDN since it assumes that the networks flows are renewal processes. In B-ISDN, both the external traffic and the internal flows, will be correlated. Developing models for queuing networks with

non-renewal flows is a challenge that needs to be addressed in order to better understand B-ISDN networks.

This dissertation is concerned with both investigating appropriate queueing network models for B-ISDN networks and with evaluating the impact of introducing a multiple class selective discard mechanism at the cell level. We define a framework for the analysis of discrete time queueing networks with priotized Markov modulated flows. Moreover, we analyze in depth a multiplexer with a multiple class selective discard mechanism and Markov modulated input.

Concerning selective discard mechanism, we: i) introduce an algorithm for solving a queue with N priority classes in N steps [14], [15], ii) prove a loss rate conservation law for non-renewal processes and iii) evaluate the impact of choosing different push out policies [16]. Regarding queueing networks, we: i) define a framework for queueing networks with Markov modulated flows [35], and ii) extend this framework for priotized flows. Furthermore, we show how the queueing network framework can be used for the analysis of ATM networks.

1.2) Dissertation Outline

This dissertation is organized as follows. In chapter 2, we discuss the major models of voice, video and data sources. We also briefly describe Markov modulated processes in continuous and in discrete time. In chapter 3, we: i) introduce selective discard mechanisms, ii) compare different push-out policies, iii) prove a loss rate conservation law, and iv) present an efficient algorithm for solving queues with multiple class selective discard mechanisms.

In chapter 4, we describe a framework for queueing networks with Markov modulated flows. We introduce a procedure for modelling the output of a multiplexer with Markov modulated flow and show how to model splitting and joining. We also illustrate with examples of tandem and feed-forward networks. In chapter 5, we describe a framework for queueing networks with prioritized flow. Finally, in chapter 6, the conclusions of this dissertation are drawn.

Chapter 2
Multimedia Traffic Models

The integration of voice, video and data signals in a single network has posed many interesting problems to telecommunication traffic theory. The cell arrival stream from multimedia traffic is strongly correlated and this non-renewal nature represents a major departure from the Poisson/renewal world. In this chapter, we describe the characteristics and the major mathematical models for voice (section 2.1), video (section 2.2) and data (section 2.3) sources. In section 2.4, we provide the necessary background on Markov modulated processes to understand the rest of the dissertation.

2.1) Voice Sources

In [36] 1965, Brady conceived the *on-off* model for voice sources. In this model, a source alternates between on and off states and stays for a memory-less distributed periods of time in each state. The *on* state corresponds to a talkspurst period in a conversation, whereas the off state corresponds to a period of silence. Depending on which compression is used, cells are considered to be generated, either at a constant interval of time (no compression) or at exponentially (geometrically, if we consider discrete time) distributed intervals of time (compression). For instance, in a system with 64 Kbps PCM coding with activity detection and a 48 octect payload size, the constant arrival

rate would be 16 cells /msec. In the off (silence) state, no cells are generated.

The process generated by an *on-off* source is a renewal process with hyperexponential (hypergeometric) interarrival time. However, when we superimpose multiple voice sources, the resulting process is non-renewal. The correlated nature of the superposition process is caused by the fluctuations in the number of voice sources in the on state. In other words, the aggregate instantaneous arrival rate is modulated by the number of sources in the on state. In the aggregate process, interarrival times are nearly independent, but the cumulative effect of the small interarrival correlations over long periods leads to queueing behavior which dramatically differs from renewal systems [21], [22]. Methods to solve a queue with many on-off sources can be classified into two groups. In the first group, each source is represented exactly and the system is solved approximately. Stochastic Fluid Flow approximation and the generating function methods fall into this category [23]-[27]. An alternative approach is to consider an approximate representation of the aggregate arrival process and solve exactly the steady state system of equations (Figure 1). Markov modulated processes are normally used to model the aggregated process, especially the two state case [30]-[34].

The main idea of the Stochastic Fluid Flow Approximation is to decompose the solution of a constant service time queue with several on-off sources into two part [23]. These two partial results correspond to different time scales: burst and cell level. At the burst level, the input traffic is seen as a stream of fluid characterized by the flow rate. Formally, we solve a linear system of differential equations to obtain the buffer occupancy. The cell level is represented as a D/1/1 queue. The fluid flow model has been extended to the finite buffer [25] and to heterogeneous sources input. The major drawback of the fluid flow approximation is that it overlooks the behavior at the cell level and gives poor results for large buffer size [31].

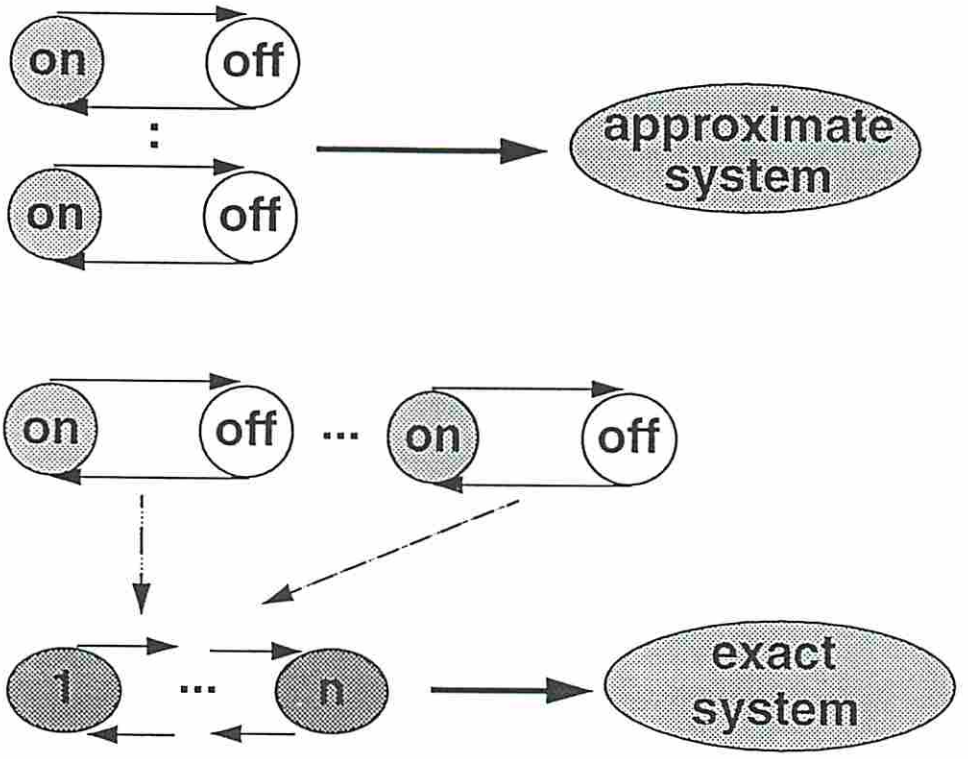

Figure 1: Two approaches for solving a system with several on-off sources

In the generating function method [27], the expression for the generating function of the queue occupancy is derived in tensor form. Then, the system characteristic function is decomposed for the evaluation of the roots. Finally, the set of linear equations for the boundary probability terms is solved. The generating function method allows the analysis of systems in which the sources are characterized by different time scales (heterogenous source). The main limitation of this method is the geometric complexity growth (memory requirement) of the boundary equations as a function of the number of traffic types (different time scale).

In the second group of methods, the arrival process is represented as a two state Markov Modulated Poisson Process (MMPP) [30]-[33]. In a Markov modulated process, the arrival rate depends on the state of an underlying

Markov chain [28], [29]. Markov modulated processes are of special interest because their flexibility allows us to set the arrival rate and the transition probabilities in each state so that we can model the correlation pattern of a traffic stream. The two state case has been widely used in the modelling of multimedia traffic due to its low computational complexity for and the ease of setting the parameters. The original work of Heffes opened avenues for many different models [19]. In each of the models, different properties of the arrival process are matched with the corresponding properties of the two state MMPP. Heffes and Lucantoni matched the i) arrival rate, ii) the variance to the mean arrivals in some interval, iii) long term variance to mean ratio of the number of arrivals, iv) the third moment of the number of arrivals in some time interval [30]. Nagarajan et al. considered two system states: underload and overload. The overload state corresponds to periods in which the arrival rate exceeds the system capacity. They matched the expected number of arrivals, the derivative of the expected value and the variance of the number of arrivals in both states [33]. Baiocchi et al. observed that the absorption time of the overload state has a phase type distribution. They computed the asymptotic decay rate which is the dominant eigenvalue of the rate transition matrix of the Markov chain corresponding to the underload state. The four parameters of the two state MMPP are computed as a function of the asymptotic decay rate [31]. Wang and Silvester [32] used the concept of underload/overload periods and determined the four MMPP parameters by computing an upperbound on the residence time in each period. Their procedure provides accurate results, has a low computational demand and can easily incorporate video, voice and data. They also derived a similar procedure for discrete time [32].

2.2) Video Modelling

Image Coding requires a large amount of information which makes video sources a major consumer of the network bandwidth. The amount of information in a coded video signal depends on the activity in the captured scene and on the compression scheme used. The major characteristics of a video stream can be specified by two types of correlation (short-term and long-term).

One compression technique to reduce the high bandwidth demand is to encode just the difference between two frames. Thus, results in a variable bit rate on a frame by frame basis. Short term correlations correspond to small fluctuations in bit rates (or uniform activity level) and usually last for a few hundred milliseconds. Long term correlations correspond to changes in the activity level (scene change) and last for a few seconds. The average and peak rate of a video stream typically vary in the range of 5 to 15 Mbits/s and 10 to 40 Mbits/s [37], [38].

Some authors have addressed the problem of predicting the bit rate generated by video codecs. Rodriguez-Dagnino et al. showed a relationship between the bit rate and several indices grouped into three categories: histogram information, spatial correlation and temporal correlation. They proposed a linear prediction model that can be used to estimate the behavior of several codecs. Pancha and El Zarki [39] studied the MPEG coding algorithm [40]. They analyzed the impact of the interframe to intrafame ratio and the quantizer scale on the distribution of cells generated per frame. This work is specially important given the applicability of MPEG algorithms for video service from HDTV to multimedia communications.

Analytical models for video source have also been defined. An autoregressive model predicts the occurrence of the next random event based on the

occurrence of the previous events [41]. The activity level within a scene varies very little and so does the bit rate. Autoregressive models are potentially good candidates to predict the bit rate within a scene or in a video without scene changes. More elaborate models can be derived by combining linear autoregressive functions (AR (p) models). Autoregressive models are typically used to fit the autocorrelation function, but they cannot generally fit the interarrival time marginal distribution. The transform expand sample (TES) model tries to overcome this problem [42]. TES models are composed of two stochastic process the parameters of which are derived from measurement and from the autocorrelation function. Although Autoregressive models are useful in simulation experiments, solving a queue with an autoregressive model as input is not analytically tractable.

Maglaris et al. [37] conceived a model for video without scene change. This model was later extended by Sen et al. [38] to incorporate scene changes. In both models they assumed that the cells are stored in a pre-buffer and are presmoothed before transmission. In the case of no scene change, the transmission rate is modeled as a continuous time birth death process. In other words, the transmission rate varies only between adjacent discretized values. The discretized value of the transmission rate as well as the transmission rate of the birth and death Markov chain are computed by using the exponential autocorrelation pattern. The transmission rate for video with scene changes is modelled by a two-dimensional Markov chain where one dimension corresponds to the activity within a scene and the other dimension refers to scene change. In this chain, transitions are only allowed between neighboring states. The transmission rate Markov chain can be viewed as the superposition of *on/off* minisources where the arrival rate on the *on* state is a unit of the discretized transmission rate. This analogy led the authors to solve a queue with their source model by applying the fluid flow approach. We observe that the trans-

mission rate Markov chains are Markov Modulated processes similar to that described before.

Markov modulated models for video sources have also been derived. Blondia and Casals [34] defined a discrete time Markov modulated process based on the first two moments of the arrival stream and on the autovariance pattern. Wang and Silvester defined a two-state D-BMAP which can easily incorporate video sources [32].

2.3) Data Sources

Data sources have historically been modelled as memoryless process (Poisson, Bernoulli). This type of process can be easily incorporated in a Markov modulated process by adding the data source arrival rate to each state dependent arrival rate.

Recent measurement investigations revealed the existence of long range slow decaying variance, dependence and self-similar (fractal) properties in ethernet traffic. In processes with slow decaying variances, if we observe the sequence of random variables which denote the number of cell arrivals in observation intervals, and compute arithmetic means from m samples of this sequence, we note that the variance of these means have a smaller decreasing rate than does the common inverse of the sample size decaying rate. In processes at long range dependences, the autocorrelation function decreases with a rate which is smaller than exponential. A self-similar process exhibits structural similarities across a wide range of time scales. In other words, it has the same traffic characteristics at the cell level, at the burst level, etc... In a self-similar process, as the time scale increases, there is only an indistinguish-

able or a small difference in the arrival pattern. In contrast, in a non self-similar process, the arrival pattern changes. Leaband et al. have questioned the validity of common employed burstiness measures, and currently used traffic models. It is important to bear in mind that the research on self-similar traffic is in its infancy, and its applicability in modelling B-ISDN traffic is under study, so we have restricted our study to independent / uncontrolled sources which results in a Poisson (geometric) assumption for data traffic.

2.4) Markov Modulated Processes

Markov modulated processes have gained popularity in the field of communications network traffic modeling due to their flexibility in capturing the behavior of voice, video and data source [30]-[33]. A Markov modulated process is a process in which the arrival rate depends on the state of an underlying Markov chain. In this dissertation, we use Markov modulated process to model multimedia traffic and the flows in ATM networks. Markov modulated processes can be defined either in continuous time (BMAP, MAP, MMPP) or in discrete time (D-BMAP, D-MAP), and it can incorporate batch arrivals (BMAP or D-BMAP) or just single arrivals (MAP, MMPP, D-MAP). In subsections 2.4.1 and 2.4.2, we detail the Markov modulated process in continuous time (BMAP) and in discrete (D-BMAP), respectively.

2.4.1) The Batch Markovian Arrival Process

In [28], Neuts defined a versatile point process which is the predecessor of the Batch Markovian Arrival Process [29]. Actually, these two processes are

equivalent. In a BMAP at any continuous time t, a batch may arrive. The elements of matrix D_n $(d_{ij})_n$ gives the probability of having n arrivals and the underlying Markov chain going from state i to state j. Clearly, we have that: $D = \sum_{n=0}^{\infty} D_n$ is the underlying Markov generator. The arrival rate of a BMAP is given by:

$$\lambda = \Pi \sum_{k=1}^{\infty} kD_k\bar{e}$$

where $\bar{e}$ is the unit column vector.

In order to solve a BMAP/D/1/K, we need to look at the queue at embedded departure times and define the state (n_i, J_i) where n_i is the number of cells in the system (queue+server) and J_i is the phase of the underlying process at the k^{th} departure. The transition probability matrix of the embedded renewal process (n_i, J_i) is given by:

$$T = \begin{bmatrix} B_0 & B_1 & B_2 & \cdots & \sum_{n=K}^{\infty} B_n \\ A_0 & A_1 & A_2 & \cdots & \sum_{n=K}^{\infty} A_n \\ 0 & A_0 & A_1 & \cdots & \sum_{n=K-1}^{\infty} A_n \\ 0 & 0 & A_0 & \cdots & \sum_{n=K-2}^{\infty} A_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sum_{n=1}^{\infty} A_n \end{bmatrix}$$

where the $(i, j)^{th}$ element of the submatrix A_n, denoted by $a_{n,ij}$, is the probability that the B-MAP makes a transition from state i to state j and n arrivals occur.

Similarly, the $(i, j)^{th}$ element of B_n is the probability that n cells arrive and that the state of the B-MAP is j at the end of a service time given that an idle period followed the last departure and the state of the B-MAP was i at that departure. For a constant service time queue with MMPP arrivals (i.e. single arrivals), the matrices A_n can be computed as [45]:

$$A_n = \sum_{j=0}^{\infty} e^{-\theta\tau} \frac{(\theta\tau)^j}{j!} R_n^{(j)}$$

where $R_n^{(j)}$ is recursively computed by:

$$R_n^{(j)} = \theta^{-1} R_{n-1}^{(j-1)} \Lambda + R_n^{(j-1)} \left(I + \theta^{-1} (D - \Lambda) \right)$$

$$R_0^{(0)} = I$$

$$R_n^{(0)} = 0 \quad for \quad (n \geq 1)$$

Λ is a diagonal matrix in which the elements of the diagonal are the state-dependent arrival rates and $\theta^{-1} = \max \{(-D_0)_{ii}\}$

The queue length distribution at departure times are computed by solving $X\,T = X$ and $X\bar{e} = 1$, and the queue length distribution at any arbitrary time is given by:

$$y_0 = -\lambda\, x_0\, D_0^{-1}$$

$$y_{i+1} = y_i\, D_i + I\,(x_i - x_{i+1})\, D_0^{-1}$$

The superposition of two BMAP processes, A and B, with m_a, m_b states and n_a, n_b maximum batch size is also a D-BMAP, process C, with $m_c = m_a \times m_b$ states and $n_c = n_a + n_b$ maximum batch size. The matrix $D_k^{(c)}$ with elements $(d_{ij})_k$ defines the probability of going from state i to state j and having a batch arrival of size k and can be computed by [46], [47]:

$$D_k^{(c)} = \sum_{q=0}^{min(n_1,\, k)} D_q^{(a)} \otimes D_{k-q}^{(b)}$$

2.4.2) The Discrete Time Batch Markovian Arrival Process

The Discrete time Batch markovian Arrival process is the discrete time version of the batch Markovian Arrival process and it was introduced by Blondia. In a D-BMAP a batch may arrive at any discrete time *t*. The elements of the matrix D_n $(d_{ij})_n$ give the probability of having *n* arrivals and the underlying Markov chain going from state *i* to state *j*. Clearly, we have that $D = \sum_{n=0}^{\infty} D_n$ is the underlying Markov generator. The arrival rate of a BMAP is given by:

$$\lambda = \Pi \sum_{k=1}^{\infty} kD_k\bar{e}$$

where $\bar{e}$ is the unit column vector.

In order to solve a D-BMAP/D/1/K, we need to look at the queue at embedded departure times and define the state (n_i, J_i) where n_i is the number in the system and J_i is the phase of the underlying process at the k^{th} departure. The transition probability matrix of the embedded renewal process (n_i, J_i) is given by:

$$T = \begin{bmatrix} B_0 & B_1 & B_2 & \cdots & \sum_{n=K}^{\infty} B_n \\ A_0 & A_1 & A_2 & \cdots & \sum_{n=K}^{\infty} A_n \\ 0 & A_0 & A_1 & \cdots & \sum_{n=K-1}^{\infty} A_n \\ 0 & 0 & A_0 & \cdots & \sum_{n=K-2}^{\infty} A_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sum_{n=1}^{\infty} A_n \end{bmatrix}$$

The $(i, j)^{th}$ element of the submatrix A_n, denoted by $a_{n,ij}$, is the probability that the D-BMAP makes a transition from state i to state j and n arrivals occur. Similarly, the $(i, j)^{th}$ element of B_n is the probability that n cells arrive and that the state of the D-BMAP is j at the end of a service time given that an idle period followed the last departure and the state of the D-BMAP was i at that departure. For a queue with constant service time equal to one discrete time unit, we have that:

$$A_n = D_n$$

$$B_n = (I - D_0)^{-1} \sum_{j=0}^{n} D_{j+1} A_{n-j}$$

The superposition of two D-BMAP processes, A and B, with m_a, m_b states and n_a, n_b maximum batch size is also a D-BMAP, process C, with $m_c = m_a \times m_b$ states and $n_c = n_a + n_b$ maximum batch size. The matrix $D_k^{(C)}$ with ele-

ments $(d_{ij})_k$ defines the probability of going from state i to state j and having a batch arrival of size k can be computed as [46], [47]:

$$D_k^{(c)} = \sum_{q=0}^{min(n_1,\, k)} D_q^{(a)} \otimes D_{k-q}^{(b)}$$

We define the D-BMAP$^{[c_1,\ldots,c_N]}$ process. The D-BMAP$^{[c_1,\ldots,c_N]}$ is a D-BMAP process in which the generated elements (cells) belong to one of the c_n groups, $n = 1,..,N$. In this dissertation, we assume that the cells are independently classified into these groups and that the components of $P_m = (p_1,\ldots, p_n)$ give the probability that cell belongs to the n^{th} group when the underlying Markov chain is in state m. In a D-BMAP$^{[c_1,\ldots,c_N]}$, the probability of having w arrivals belongings to the n^{th} group and the underlying process going from state i to state j is given by:

$$\sum_{T=w}^{\infty} \sum_{t_1=0}^{T-w} \cdots \sum_{t_{n-1}=0}^{z_{n-1}} \sum_{t_{n+1}=0}^{z_{n+1}} \cdots \binom{T}{t_1, \ldots, t_N} p_1^{t_1} \times \ldots \times p_n^{w} \times \ldots \times (d_{ij})_T$$

where $z_i = T - w - \sum_{j=1}^{i} t_j$

Having introduced basic notions and notations, we proceed to the analysis of selective discard mechanism.

Chapter 3

Multiple Class Selective Discard Mechanism

In this chapter, we study a multiple class selective discard mechanism. Initially, we define selective discard mechanism (section 3.1) and compare different push out policies (section 3.2). We proceed by proving a loss rate conservation law (section 3.3) which is the theoretical background for the analysis of a queue with multiple levels of priority. We then introduce an efficient procedure for the computation of the loss rate per class in a multiple class queue (sections 3.4 and 3.5). Finally, we illustrate the advantages of introducing a multiple class selective discard mechanism at the cell level of B-ISDN networks (section 3.6).

3.1) Selective Discard Mechanism

In a statistical multiplexer, whenever the incoming stream of cells exceeds the storage capacity, loss occurs. A multiple class selective discard mechanism discards cells in an overflow situation according to their priority level. Whenever a cell finds the buffer full and there is an enqueued cell with lower priority, the lower priority cell is discarded in order to release a buffer for the arriving cell. It is important to notice that the loss phenomenon always

occurs when statistical multiplexing is used in a network carrying multimedia traffic. Multimedia traffic is bursty and the only way of guaranteeing no loss is to allocate resources according to the sources' peak rate. However, this solution leads to undesirable network underutilization.

When we consider selective discard, there are two crucial aspects: i) what the preemptive access (discard / push out) policy is, and ii) how the buffer space is shared. A push out policy determines which of the lowest priority enqueued cells is the one to be dropped. The main push out policies are [57]:

i) First-In-First-Drop (FIFD) - chooses the cell closest to the head of the queue,

ii) Last-In-First-Drop (LIFD) - chooses the cell closest to the tail of the queue,

iii) Random (RAND) - Any cell is dropped with equal probability.

The choice of a specific push out policy impacts the performance of each priority class. In section 3.2, we describe a comparative simulation experiment of the three policies defined above.

A buffer sharing policy specifies which buffer slots can be occupied by which priority class. The main buffer sharing policies are [48], [49]:

i) Complete Sharing - all buffer spaces are accessible to all customers;

ii) Complete Partitioning - Each class of customers has its own individual queue

iii) Partial Sharing - cells of class k can only enter the queue if there are fewer than T_k cells of any class in the system;

iv) Sharing With Maximum Queue Length - There is a maximum limit for the number of customers of each class;

v) Sharing With a Minimum Allocation - There is a minimum amount of buffer space allocated to each class of customers.

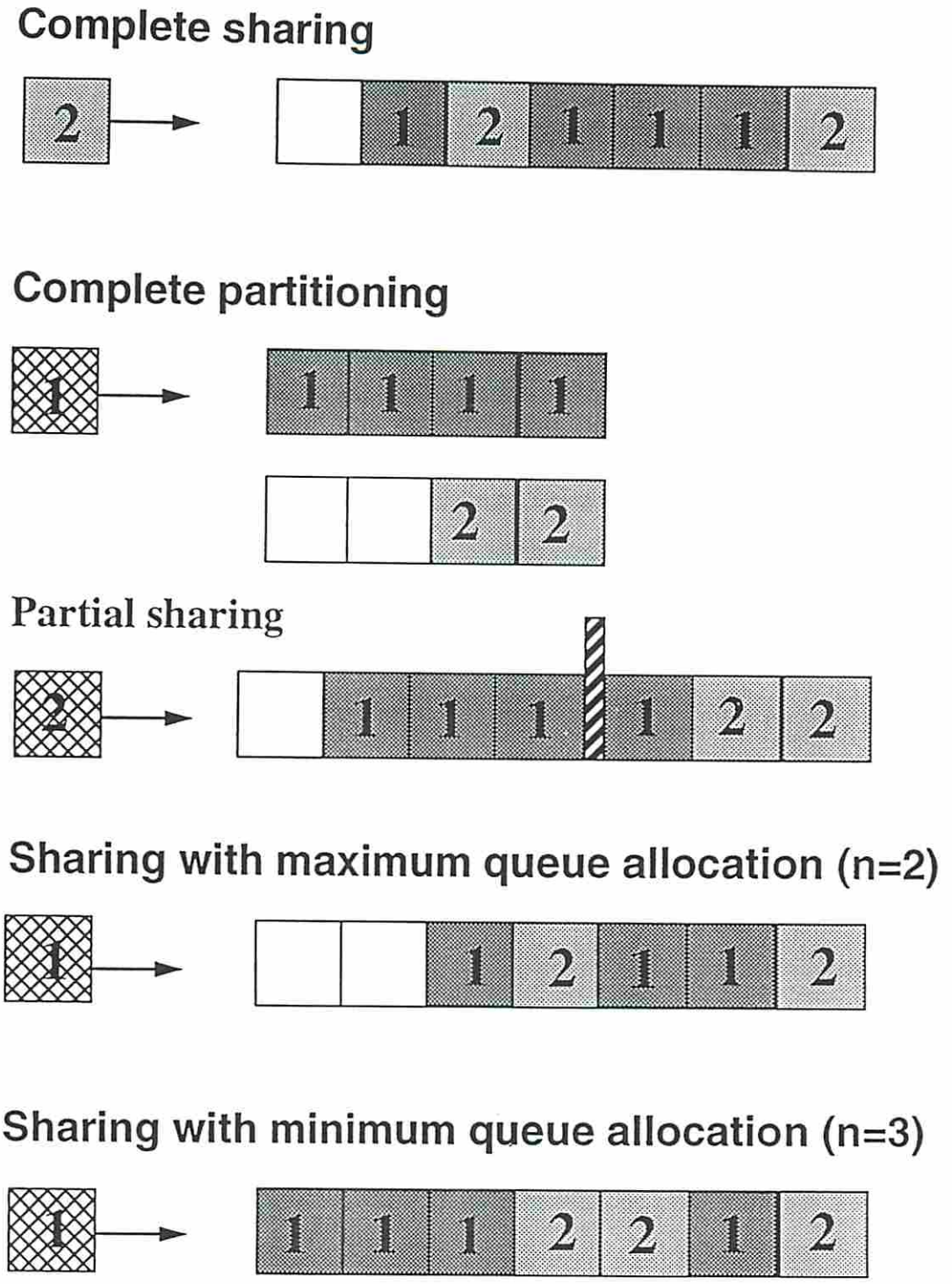

Figure 2: Examples of buffer sharing policies

A queueing scheme defines the disciplines for sharing the server and the buffer space. A queueing scheme is considered work-conserving if [50]: i) no server is left idle when the buffer is not empty, ii) customers are lost when the buffer is full, iii) all classes of customers have the same service requirement, iv) served customers are immediately removed from the system and v) no work (i.e service requirement) is created or destroyed by any particular employed rule. In other words, a work conserving scheme fully utilizes the system resources. According to this definition, Complete Sharing and the Sharing with a Minimum Allocation buffer policies are work-conserving buffer policies.

Selective discard has been extensively studied for systems with two priority levels [51]-[60]. The studies differ in: i) the arrival assumption (Poisson [53], Geometric [54], [55], and batch arrivals [55]), ii) the service time (exponential [53], deterministic [55] and general) and iii) buffer organization (complete sharing and partial sharing). To the best of our knowledge no model for systems with multiple levels of priority has been developed.

3.2) Push Out Policies

The choice of a specific push out policy impacts the loss performance of each priority class since it influences the likelihood that a high priority cell finds an enqueued lower priority cell that it can push out[16]. For example, let us suppose that there are two low priority cells in the queue, one at the tail and the other at the head of the queue (Figure 3). An incoming high priority cell finds the buffer full. According to the Last In First Drop policy, the low priority cell at the tail will be dropped, and according to the First In First Drop policy the one

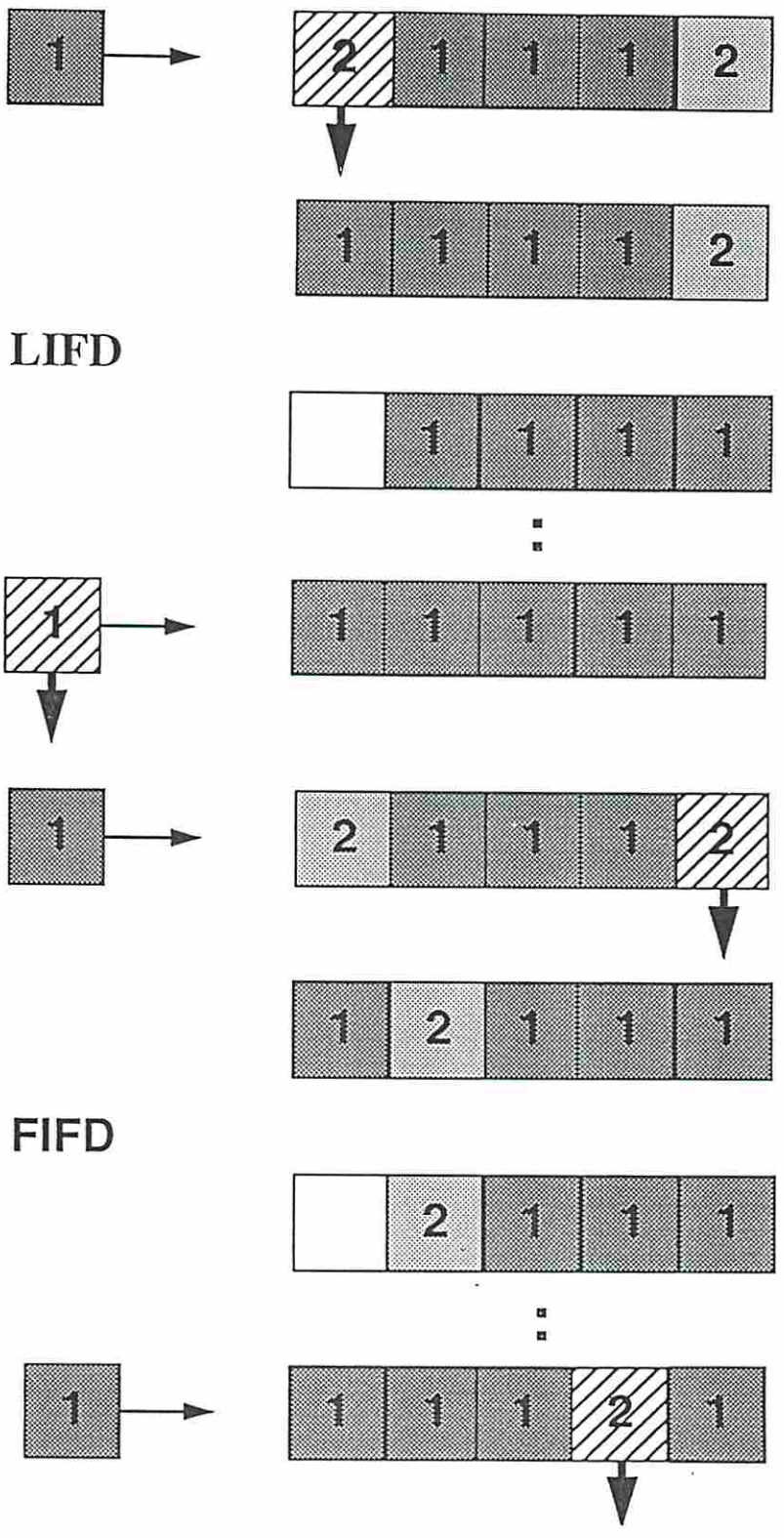

Figure 3: An example of the impact of different push out policies

at the head of the queue will be the one dropped. Sometime later, after several high priority arrivals, the buffer is full again and a high priority arrival occurs. In the LIFD, the cell at the head of the queue at the previous high priority arrival (which was not dropped) has already been transmitted, and consequently the high priority cell is lost. For FIFD however, the low priority cell at the tail of the queue at the previous high priority arrival was not dropped and can now be dropped to make space for the new high priority cell.

Although the probability of cell loss is an important metric in understanding service quality, it does not completely characterize the loss phenomenon [61], [62]. An understanding of the correlation of successive cells is also important, particularly for quality of real time signal reconstruction at the receiver. For instance, imagine a video stream composed of 1000 cells with a loss rate of 0.25, this could correspond to one cell lost in every four, or 250 consecutive cell losses. Depending on coding, the pictorial distortion in one case will be radically different from the other. In order to help understand this aspect of loss behavior, we study the distribution of the number of consecutive losses in addition to loss rate.

To compare different push out policies, we need a detailed event sequence. An exact analytical solution for a two priority system via Markov chain requires 2^K states where K is the buffer size. Thus, we use simulation to evaluate how the choice of a specific push out policy (First In First Drop, Last In First Drop, and Random) influences the loss rate and the average number of consecutive losses in a system with two levels of priority. In the simulation experiment, we represent both the high and the low priority classes as two-state Markov Modulated Poisson Process. We used 10^{-2} and 10^{-4} as transition rates to the two-sate MMPP which is consistent with practical values. In order to cover a range of buffer size, we varied it from 10 to 200. Variations on the high and low priority burstinesses (peak rate / mean rate) are also considered.

Regarding loss rate, we found that for the high priority class, the First-In-First-Drop policy always gives the lowest values. Last-In-First-Drop gives the highest values and the values given by Random are between these two. FIFD is the best policy for the high priority class because it always chooses the low priority cell which has stayed the longest period in the queue. Figure 4 illustrates an experiment where the low priority class load is fixed at 0.3 and the high priority load varied from 0.3 to 0.6, both burstinesses are equal to 2. The maximum observed difference between the high priority loss rate given by the Last-In-First-Drop and First-In-First-Drop policies is 50% and it happens for a loss rate of the order of 10^{-5}. The choice of a given policy does not have a significant impact on the low priority loss rate

In order to investigate how the results for different policies differ as a function of the high priority burstiness, we: i) fix both loads fixed at 0.3, ii) the low priority burstiness equals to 2.0 and iii) vary the high priority burstiness from 2.0 to 5.0 (Figure 5 and Table 2). We observed that as the high priority burstiness increases, so does the high priority loss rate and consequently the difference between the policies. The low priority burstiness has almost no influence on the differences in the high priority loss rate.

Regarding the average number of consecutive losses, we noticed that for the low priority class the Last-In-First-Drop policy gives smaller averages than does the First-In-First-Drop policy. This relationship can be understood by the following: in the Last-In-First-Drop policy a low priority cell that would be dropped by the First-In-First-Drop policy is eventually transmitted. This transmission breaks the loss burst into two smaller bursts. As the low priority probability increases, the frequency of long bursts also increases and an eventual LIFD transmission brings a huge difference in the average loss burst. In Figure 6, we show traces of the size of consecutive loss bursts for a high priority load of 0.6 and a low priority load of 0.3. Note that the maximum burst size for the

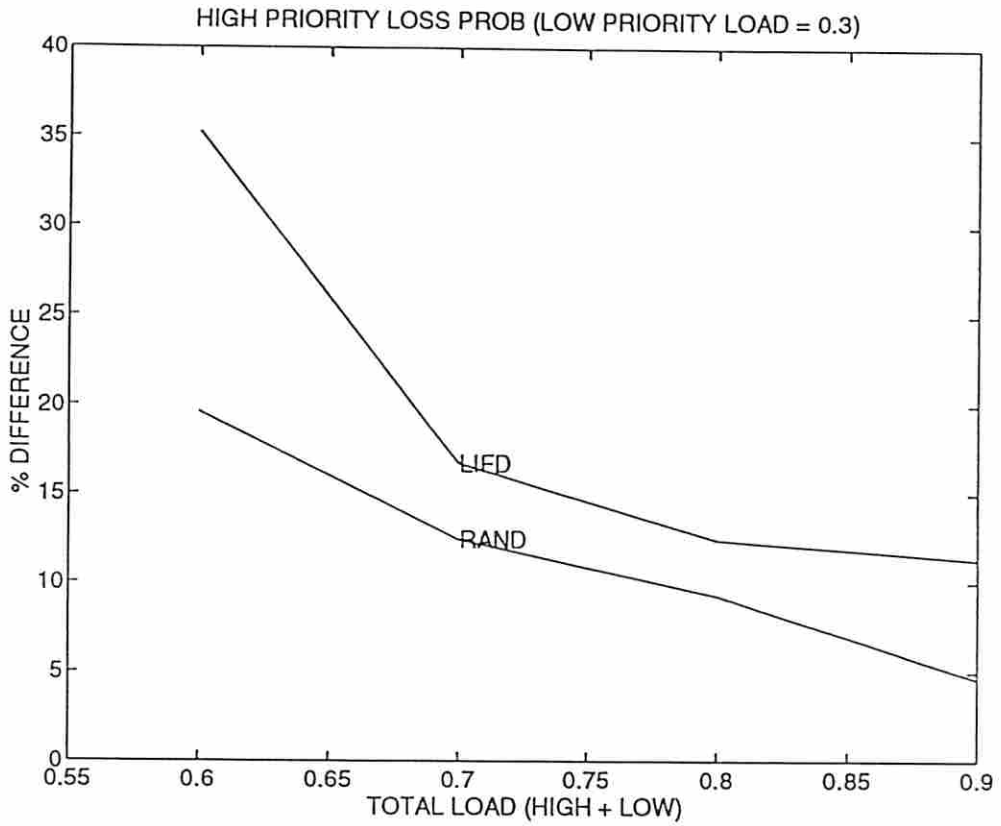

Figure 4: Percentage difference of the high priority loss rate due to different push out policies

FIFD	RAND	LIFD
2.58933E-5	3.09683E-5	3.83220E-5
3.81373E-4	4.28701E-4	4.45138E-4
1.8581E-3	2.03072E-3	2.30591E-3
5.36053E-2	5.60872E-2	5.96604E-2

Table 1: High priority loss rates for different push out policies

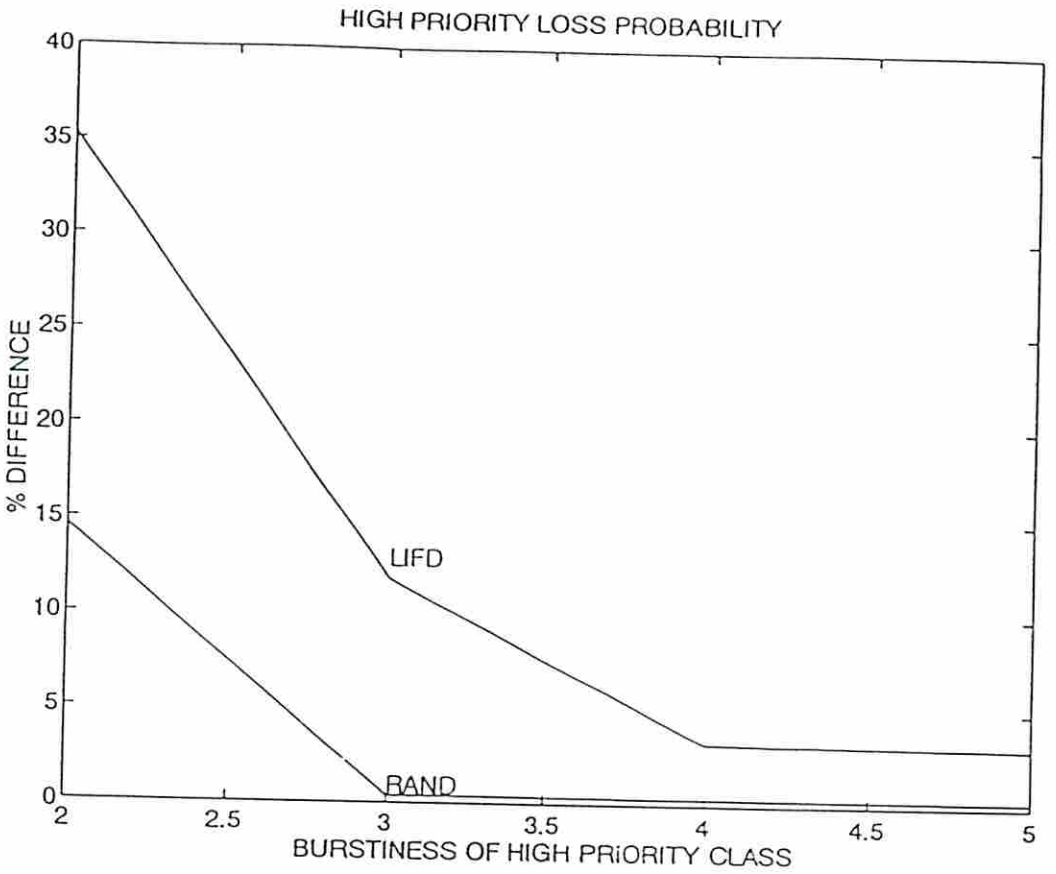

Figure 5: Percentage difference of the loss rate as a function of the high priority burstiness

FIFD	RAND	LIFD
2.5893E-5	2.96737E-5	3.50258E-5
1.34479E-3	1.35017E-3	1.50455E-3
7.14522E-3	7.17023E-3	7.37958E-3
1.36214E-2	1.36649E-2	1.40504E-2

Table 2: High priority loss rate x high priority burstiness

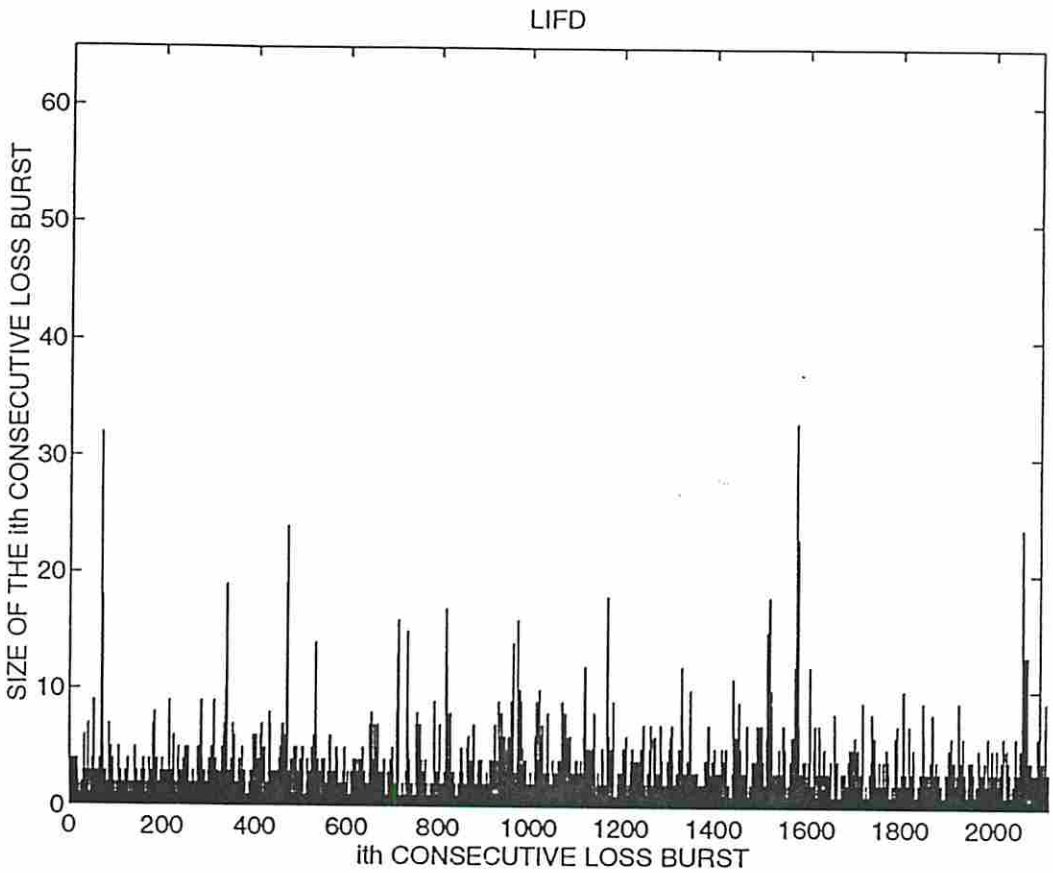

Figure 6a: Trace of burst size for LIFD

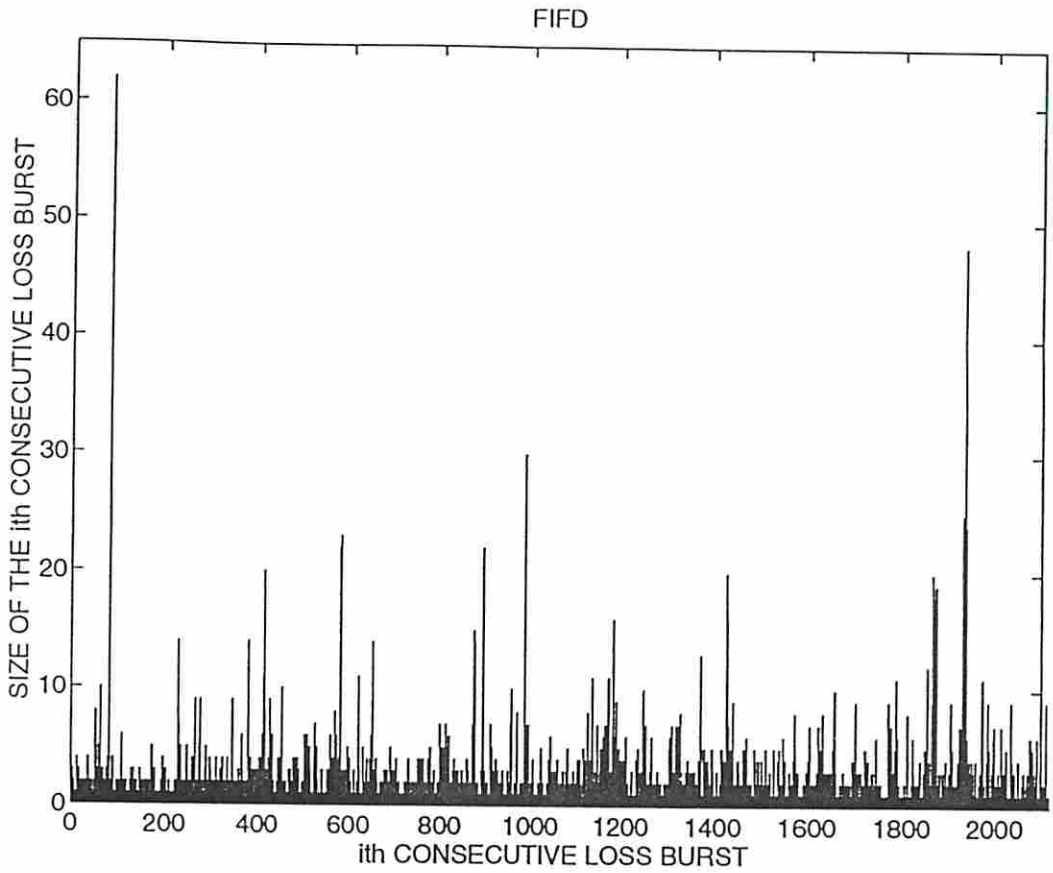

Figure 6b: Trace of loss burst size for FIFD

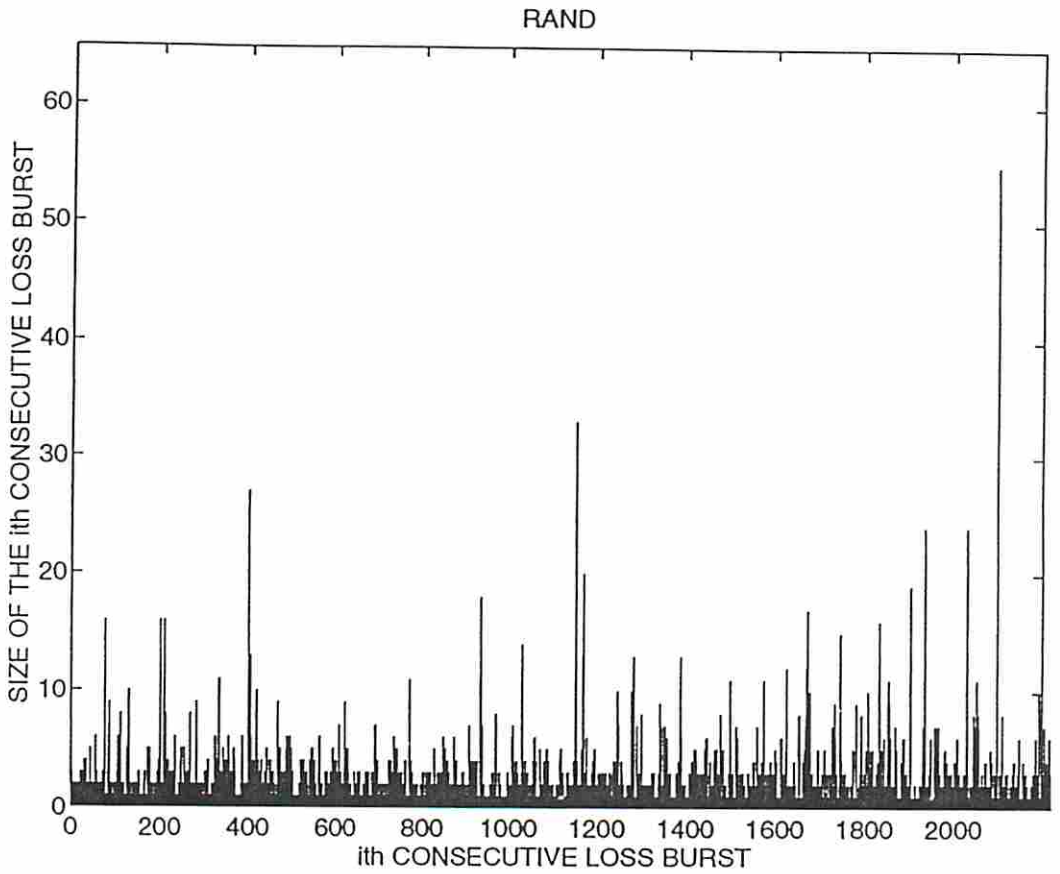

Figure 6c: Trace of loss burst for RAND
Figure 6: Trace of loss burst size

LIFD is half the value for FIFD. Note also that smaller loss burst is much more frequent in LIFD than is in FIFD. Random again shows an intermediate behavior between the two policies. Figures 7 and Table 3 show the percentage difference of the average number of consecutive losses for the traces displayed in Figure 6. Most of the losses are individual cells which give a small value for the average. The impact of the push-out policies on the high priority number of consecutive loss is not as significant as it is for the low priority case.

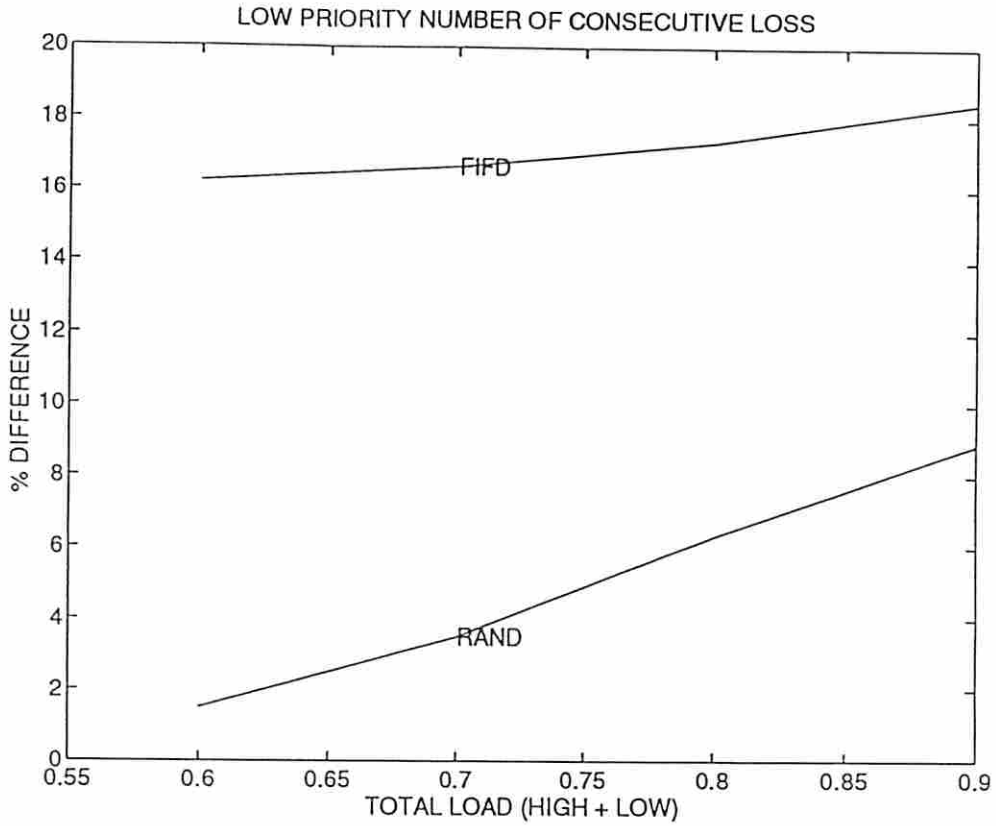

Figure 7: Percentage difference of the average number of consecutive loss

LIFD	RAND	FIFD
1.20808	1.2262	1.40427
1.26283	1.30702	1.47309
1.35323	1.43848	1.58801
1.35719	1.47797	1.60786

Table 3: Average number of consecutive losses

Although differences were found for the loss rate, they do not imply in a decrease of an order of magnitude. We notice that the numerical differences do not change for a fixed loss rate value when we varied the buffer size. Our study did not investigate loss rates lower than 10^{-9} due to computational limita-

tions. It would be interesting to pursue future investigations based on rare event simulation which could deal with lower values for the loss rate. We also found that when the low priority loss rate is high, the Last-In-First-Drop plays a major role in the decrease of the number of consecutive losses and depending on the coding may have a decisive impact on the quality of real-time signal reconstruction at the receiver. Choosing a policy between First-In-First-Drop and Last-In-First-Drop is a trade-off between improving Quality of Service requirements: high priority loss rate and low priority average number of consecutive loss. Random offers intermediate performance quality between the aforementioned policies.

3.2) A Loss Rate Conservation Law

In this section, we introduce a loss rate conservation law which is the theoretical background for an efficient computational solution of a queue with multiple class selective discard mechanism. In [50], Clare and Rubin introduced a loss probability conservation law for work-conserving slotted queues with *i.i.d.* arrivals and multiple class selective discard mechanism. The law establishes that the product of the aggregated loss rate by the aggregated arrival rate equals the per class summation of the product of the loss rate by the arrival rate. Jeon and Viniotis [63] derived a similar law for queues with Markov Modulated Poisson Process. Their law states a relationship between the arrival rate and the loss rate conditioned on the state of the process at the beginning of busy periods. Although insightful, Jeon and Viniotis' law has a limited applicability given that the related measures are not usual descriptors of a system. Clare and Rubin's law is specially important because it allows the computation

of the loss rate per class with low computational complexity. However, Clare and Rubin's law cannot be applied to a queue with non-renewal arrivals. The main reason for this restriction is that in a non-renewal process we cannot relate time averages to steady-state statistical averages. In other words, the long term ratio between the number of losses and the number of cells does not converge to the definition of probability. Whenever we apply the concept of probability, we assume that we pick a random cell form the universe of cells and check if it will be lost or not. In a correlated process (non-renewal), the loss of a cell depends on the past loss history; it is not a random event. Actually, when trying to guarantee minimum Quality of Service, we are interested in the fraction of lost cells (loss rate) and not exactly in the loss of a particular (randomly selected) cell (loss probability). Our loss rate law establishes that the product of the aggregate loss rate by the aggregate arrival rate in a work conserving queue is equal to the summation of the per class product of the loss rate by the arrival rate. In other words,

$$\lambda R = \sum_{n=1}^{N} \lambda_n R_n$$

where: N is the number of priority classes,

R is the aggregate loss rate,

R_n is class n loss rate,

λ is the aggregate arrival rate,

λ_n is class n arrival rate.

In order to prove out conservation law, we observe the queue at the beginning of each busy period. These instants are regenerative points. Thus, we can use the Strong Law of Large Numbers for Markov Chains [45]. Let us first introduce some notation and definitions:

Notation:

L - the expected number of lost cells,

A - the expected number of arrivals,

L_n - the expected class n number of lost cells,

A_n - the expected class n number of arrivals,

A (0, t) - number of arrivals in the interval [0, t],

A_n (0, t) - class n number of arrivals in the interval [0, t],

L^j - the expected number of lost cells during busy periods with starting state j,

A^j - the expected number of arrivals during busy periods with starting state j,

L_n^j - the expected class n number of lost cells during busy periods with starting state j,

A_n^j - the expected class n number of arrivals during busy periods with starting state j,

$L^{k,j}$ - the number of lost cells during busy periods with starting state j up to the k^{th} busy period,

$A^{k,j}$ - the number of arrivals during busy periods with starting state j up to the k^{th} busy period,

$L_n^{k,j}$ - class n number of lost cells during busy periods with starting state j and up to the k^{th} busy period,

$N^{k,j}$ - the number of busy periods with starting state j up to the k^{th} busy period,

P^j - long term rate of occurrence of busy periods with starting state j,

m - number of states of the B-MAP process.

Definition 1: The overall loss rate and the loss rate per class are respectively given by:

$$R = \frac{L}{A} \qquad R_n = \frac{L_n}{A_n}$$

Definition 2: The overall and the per class arrival rate are respectively given by:

$$\lambda = \lim_{t \to \infty} \frac{E[A(0,t)]}{t} \qquad \lambda_n = \lim_{t \to \infty} \frac{E[A_n(0,t)]}{t}$$

Proposition 1:
$$R = \frac{L}{A} = \lim_{k \to \infty} \frac{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} L^{k,j} \right) \times \frac{N^{k,j}}{k}}{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} A^{k,j} \right) \times \frac{N^{k,j}}{k}} \qquad (1)$$

Proof: By the Strong Law of Large Numbers, we have that the following limits exist

$$\lim_{k \to \infty} \frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} L^{k,j} = L^j \quad \text{and} \quad \lim_{k \to \infty} \frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} A^{k,j} = A^j$$

Note that by assuming a work-conserving queue, the losses (overall and per class) and arrivals during a busy period depend only on the phase of the Markov modulated process at the beginning of the busy period (starting state). Consequently these losses and arrivals are independent of the losses and arrivals during a busy period with another starting state. In addition, the cell losses and arrivals during two busy periods with the same starting state are *i.i.d.*. Given that L^j and $L^{\tilde{j}}$ are independent, for $j, \tilde{j} = \overline{1, m}$ and $j \neq \tilde{j}$, and that $\lim_{k \to \infty} N^{k,j}/k$, we have that:

$$L = \sum_{j=1}^{m} L^j \times P^j \quad \text{and} \quad A = \sum_{j=1}^{m} A^j \times P^j$$

Proposition 2: $\lambda R = \sum_{n=1}^{N} \lambda_n R_n$

Proof: Given that $L^{k,j} = \sum_{n=1}^{N} L_n^{k,j}$ (1) can be rewritten as:

$$R = \lim_{k \to \infty} \frac{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} \sum_{n=1}^{N} L_n^{k,j} \right) \times \frac{N^{k,j}}{k}}{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} A^{k,j} \right) \times \frac{N^{k,j}}{k}} =$$

$$\lim_{k \to \infty} \frac{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} L_1^{k,j} \right) \times \frac{N^{k,j}}{k}}{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} A^{k,j} \right) \times \frac{N^{k,j}}{k}} + \ldots + \frac{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} L_N^{k,j} \right) \times \frac{N^{k,j}}{k}}{\sum_{j=1}^{m} \left(\frac{1}{N^{k,j}} \sum_{k=1}^{N^{k,j}} A^{k,j} \right) \times \frac{N^{k,j}}{k}}$$

We know that when $t \to \infty$, $k \to \infty$ where t denotes time and k denotes the number of busy periods. We also know that as a consequence of the system being stable (arrival rate < service rate) and the buffer space being finite, we have that both the duration of a busy period, and the number of arrivals and losses during a busy period are finite. Given that each busy period is finite and that in the long-run $t \to \infty$ each state of the underlying Markov Chain is visited an infinite number of times, we have an infinite number of busy periods with the same starting state. Thus, (2) can be rewritten as:

$$\lim_{t,\,k\to\infty} \frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} L_1^{k,j}\right)\times\frac{N^{k,j}}{k}}{\frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,}} A_1^{k,j}\right)\times\frac{N^{k,j}}{k}}{t}} \times \frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,}} A_1^{k,j}\right)\times\frac{N^{k,j}}{k}}{\frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,}} A^{k,j}\right)\times\frac{N^{k,j}}{k}}{t}} + \ldots$$

$$\frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} L_N^{k,j}\right)\times\frac{N^{k,j}}{k}}{\frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,}} A_N^{k,j}\right)\times\frac{N^{k,j}}{k}}{t}} \times \frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} A_N^{k,j}\right)\times\frac{N^{k,j}}{k}}{\frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} A^{k,j}\right)\times\frac{N^{k,j}}{k}}{t}}$$

by definition, we have that:

$$\lim_{t,\,k\to\infty} \frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} A^{k,j}\right)\times\frac{N^{k,j}}{k}}{t} = \frac{E[A(0,t)]}{t} = \lambda$$

$$\lim_{t,\,k\to\infty} \frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} A_n^{k,j}\right)\times\frac{N^{k,j}}{k}}{t} = \frac{E[A_n(0,t)]}{t} = \lambda_n$$

by a similar rationality used to prove (1), we can show that:

$$\lim_{t,\,k\to\infty} \frac{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,j}} L_n^{k,j}\right)\times\frac{N^{k,j}}{k}}{\sum_{j=1}^{m}\left(\frac{1}{N^{k,j}}\sum_{k=1}^{N^{k,}} A_n^{k,j}\right)\times\frac{N^{k,j}}{k}} = R_n$$

Thus, we can rewrite (2) as:

$$R = \frac{R_1\lambda_1}{\lambda} + \frac{R_2\lambda_2}{\lambda} + \ldots + \frac{R_N\lambda_N}{\lambda}$$

or

$$\lambda R = \sum_{n=1}^{N} R_n\lambda_n$$

❑

3.4) The Multiple Class Buffer Priority Algorithm

The main problem in computing the loss rate per class in a queue with multiple levels of buffer priority is that the complexity grows as a function of the number of priority levels. For instance, if we use a *n* dimension Markov chain for a *n* class system it is O (K^n) where *K* is the number of buffers and *n* is the number of classes.

In order to reduce the complexity of the solution, we use the loss rate conservation law introduced in section 3.3. Let us show how the conservation law can be applied. Let us suppose we have an original queue with *N* priority levels and a shadow system with two priority levels which solution can be eas-

ily obtained. Initially, we solve the shadow system by considering its high priority class as being the highest priority class of the original *N* class system and its low priority class as being the aggregation of the lowest *N-1* classes in the original system. By doing so, we are able to compute exactly the loss rate of the highest priority class of the original system. At a second step, we solve the shadow system by considering its high priority class as being the aggregation of the two highest priority classes of the original and its low priority class as being the aggregation of the *N-2* lowest priority class of the original system. Given that the shadow high priority loss rate is the aggregation of the loss rate of the two highest classes of the original system and that we know the exact class value of the original highest priority loss rate, we can derive the exact value of the second highest class loss rate by applying the conservation law. We can continue this process by defining a different mapping of the original *N* classes into the two shadow classes. Using the rationality explained before, we developed a procedure called the Multiple Class Buffer Priority Algorithm (MCBP) for computing the loss rate per class in a queue with multiple priority levels.

The MCBP algorithm avoids the complexity growth of the exact solution of the *N* class system by solving *N-1* two priority class systems [15]. Each of these two priority class systems corresponds to a different splitting of the original *N* classes. The algorithm starts by mapping the highest priority class of the *N* class system into the high priority superclass (aggregation of classes) of a two class system and the other *N-1* superclasses into the low priority class. At step *i*, the highest *i* priority classes are mapped into the high priority class of a two class system and the remaining *N-i* are mapped into the low priority superclass. The individual class loss rates of the original *N* class system are computed from the loss rate of the *N-1* two class systems. The MCBP algorithm is similar to a multiple class algorithm developed by Clare and Rubin [50], but

their work assumes that a high priority class has both buffer and service priority over the low priority class.

Let us introduce some notation. Let Θ_N denote an N class system to be solved and λ_n, R_n respectively class n arrival rate and loss rate. Let Ψ_g a two class system in which the high superclass corresponds to the aggregation of the g highest classes of Θ_n. Let also $\lambda_{g,h}$ ($\lambda_{g,l}$) and $R_{g,h}$ ($R_{g,l}$) be respectively the high (low) priority superclass arrival rate and loss rate (Figure 8) The Multiple Class Buffer priority algorithm can be state as:

STEP 1 - For $g = 1$ to $N - 1$

Solve Ψ_g

STEP 2- $R_N = R_{N-1,l}$ and $R_1 = R_{1,h}$

STEP 3 - For $n = 2$ to $N-1$ compute:

$$R_n = \frac{1}{\lambda_n}\left[\lambda_{n,h} R_{n,h} - \sum_{k=1}^{n-1} \lambda_k R_k\right]$$

We illustrate the MCBP algorithm for a five class system below:

STEP 1

	HIGH (h)	LOW (l)
Ψ_1	1	2,3,4,5
Ψ_2	1,2	3,4,5
Ψ_3	1,2,3	4,5
Ψ_4	1,2,3,4	5

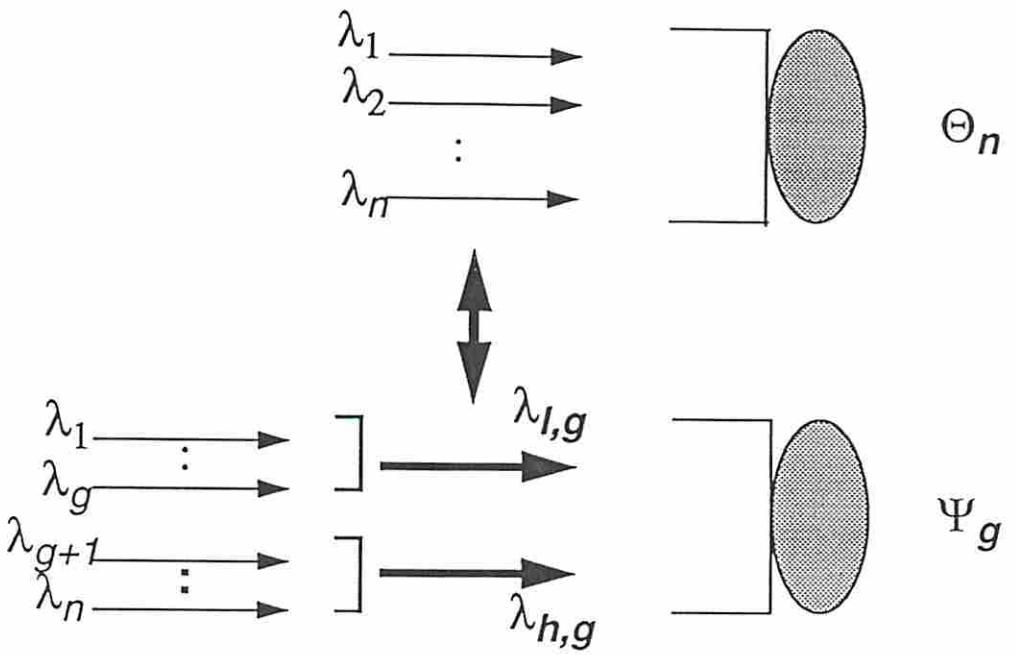

Figure 8: The MCBP algorithm

STEP 2

$$R_5 = R_{4,l}$$

$$R_1 = R_{1,h}$$

STEP 3

$$R_2 = 1/\lambda_2 \, [\lambda_{2,h} \, R_{2,h} - \lambda_1 \, R_1]$$

$$R_3 = 1/\lambda 3 \, [\lambda_{3,h} \, R_{3,h} - \lambda_1 \, R_1 - \lambda_2 \, R_2]$$

$$R_4 = 1/\lambda_4 \, [\lambda_{4,h} \, R_{4,h} - \lambda_1 \, R_1 - \lambda_2 \, R_2 - \lambda_3 \, R_3]$$

Note that the first step of the MCBP algorithm corresponds to *N-1* iterations where each iteration is the solution of a two class system. Step 3 computes the *N-2* remaining loss rates by substitution of known values. Thus, the complexity of the MCBP algorithm is *N*-1 times the complexity of the solution

of the two class system Ψ_g. The MCBP is exact and the accuracy of its results depends only on the accuracy of the two class system.

3.5) Solving a Queue with Two Priority Levels

To solve a queue with multiple levels of buffer priority, we need to solve a queue with two priority levels. Our approach to analyze a queue with two priority levels is [15], [53], [55]: i) to solve the aggregate system, i.e., an equivalent system disregarding the priority mechanism, ii) to compute the loss rate of the low priority class and iii) to derive the high priority loss rate by using the Conservation Law.

Initially, we consider an equivalent aggregated system, i.e., a system with the same arrival process, but with no priority mechanism. We compute the steady state queue length distribution of the embedded Markov chain at departure times [$\Pi T = \Pi$ and $\Pi \bar{e} = 1$] and then compute the overall loss rate:

$$R = \frac{E[L]}{E[A]}$$

where $E[L]$ is the expected number of classes and $E[A]$ is the expected number of arrivals.

$$E[L] = \sum_{k=1}^{K-1} \sum_{y=K-k+1}^{\infty} \sum_{j=1}^{\overline{M}} \sum_{i=1}^{\overline{M}} (y-K-k) \times a_{y,ij} \times \pi(k,i) +$$

$$\sum_{y=K}^{\infty} \sum_{i=1}^{\overline{M}} \sum_{j=1}^{\overline{M}} (y-K+1) \times b_{y,ij} \times \pi(0,i)$$

$$E[A] = \sum_{k=1}^{K-1} \sum_{y=0}^{\infty} \sum_{j=1}^{\overline{M}} \sum_{i=1}^{\overline{M}} y \times a_{y,ij} \times \pi(k, i) +$$

$$\sum_{y=0}^{\infty} \sum_{i=1}^{\overline{M}} \sum_{j=1}^{\overline{M}} y \times b_{y,ij} \times \pi(0, i)$$

To derive the low priority loss rate, R_2, we focus on a tagged cell and compute its survival probability, R_s ($R_2 = 1 - R_s$). The survival probability is the probability that during its waiting time, the tagged cell will not be pushed out by a higher priority cell. We assume that the tagged cell arrives immediately after the beginning of a service period and that the dropping policy is Last in First Drop. To compute the survival probability, we first condition on the position that the tagged cell joins the queue:

$$R_s = \sum_{k=1}^{K} \sum_{i=1}^{M} S(k, i) \times \pi(k-1, i)$$

where: $S(k,i)$ = P (survive / joined the queue at position k and Markov modulated process was in state i)

The tagged cell survives the first service period, if the maximum number of high priority arrivals is equal to $K - k$. It survives the second service period, if and only if, the maximum number of high priority arrivals is equal to $K - k + 1$. In general, it survives up to the n^{th} service period if, and only if, the cumulative number of high priority arrivals is at most equal to $K - k - (n-1)$ [53], [55]. Thus, for a continuous time system:

$$S(k, i) = \sum_{z=0}^{K-1} \beta(z, k, i) \qquad 1 \le k \le K$$

$$\beta_w(z, k, i) \begin{cases} \alpha_w(z, k, i) \otimes \beta_w(z, k, i) & 0 \le z \le K - k + (w - 1) \\ 0 & \textit{otherwise} \end{cases}$$

$$\alpha_w(z, k, i) = \sum_{j=1}^{M} \hat{a}_{z, ij} \qquad w = k$$

$$\alpha_w(z, k, i) = \sum_{j=1}^{M} \sum_{l=1}^{M} \hat{a}_{z, ij} \times \hat{x}_n \qquad w \neq k$$

where $\hat{X} = \hat{A}\hat{X}$, $\hat{X}\bar{e} = 1$ and $A = \sum_{n=0}^{M} \hat{A}_n$. $\hat{A}_n$ is the probability matix of having n high priority arrivals

Finally, we compute the high priority loss rate as:

$$R_1 = 1/\lambda_1 [\lambda R - \lambda_2 R_2]$$

3.6) Numerical Examples

We have claimed that a multi-priority scheme enhances network flexibility in coping with traffic scenarios which have diverse loss requirements. In the first example, we consider four traffic classes with the parameters defined in Table 4 (the transition rates are the same for all MMPP in this section). Class A is the highest priority and class D is the lowest. Figure 9 shows how these four classes are multiplexed when there are 1, 2, 3 or 4 priority levels and a buffer size of 10. Initially, there is no priority, i.e. all classes receive the same performance. When we move to two priority classes, we split off the most

demanding traffic class (A). If three priority levels are available, we use (A), (B) and (C, D). Finally, for four priority levels each traffic class can be assigned to a different level. We can, thus, verify the advantage of an additional priority level by the difference in the loss rate in the corresponding traffic class. The flexibility in accommodating different loss requirements may reduce buffer requirements.

A common description of the burstiness of a traffic stream is the ratio between the peak and the mean rate [65]. In [65], it was shown that the variation of the burstiness of a low priority class does not affect the loss rate of a high priority class. We are interested in investigating the impact of the highest priority class burstiness in a multiple priority queue. We consider four classes described by two state MMPP with mean arrival rate of 0.125, burstiness of 1.5 and transition rates as described in table 4. In Figure 10, we vary the high priority burstiness by fixing the mean arrival rate and the transition probabilities and by varying the ratio between the arrival rate in each state. We notice that at high values of the high priority burstiness (and consequently of the overall loss rate), the differences among the loss rates decrease. However, a multiple class mechanism is still useful for coping with diverse loss requirements.

	transition rate	arrival rate
state 1	10^{-4}	0.175
state 2	10^{-2}	0.2625

Table 4: MMPP parameters for traffic scenario 1

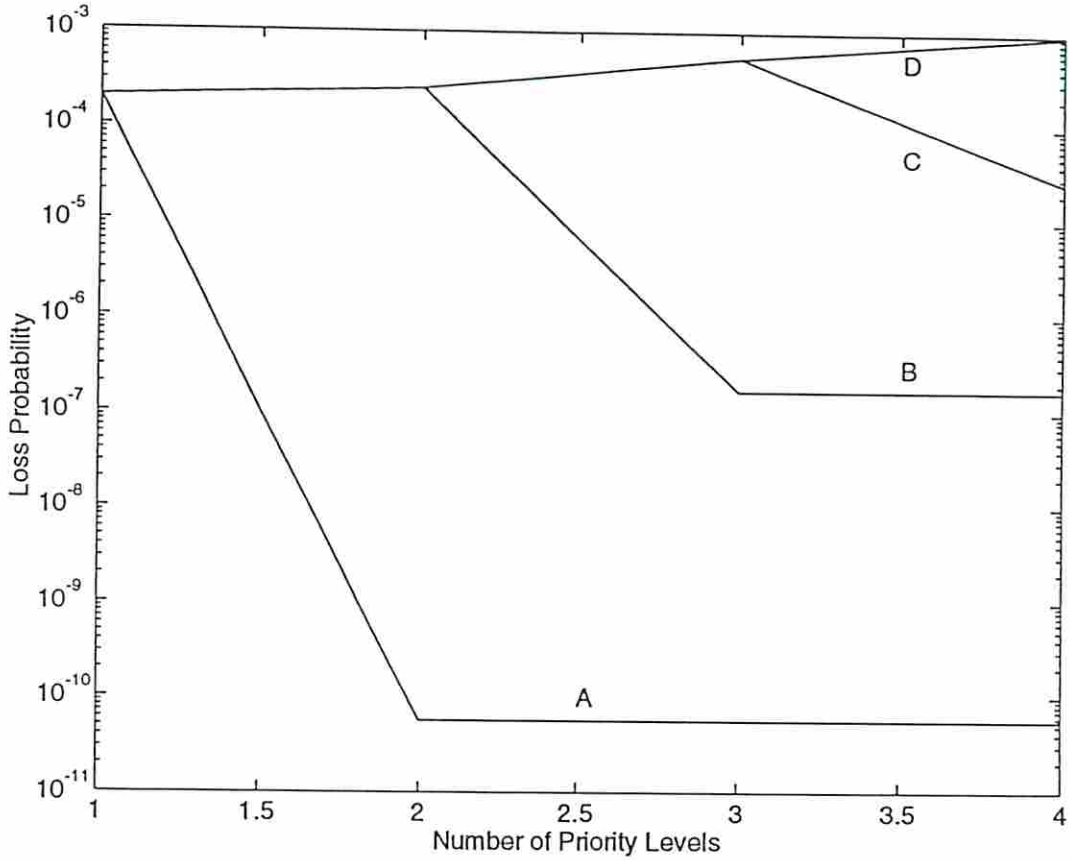

Figure 9: Loss rate as a function of the number of priority levels

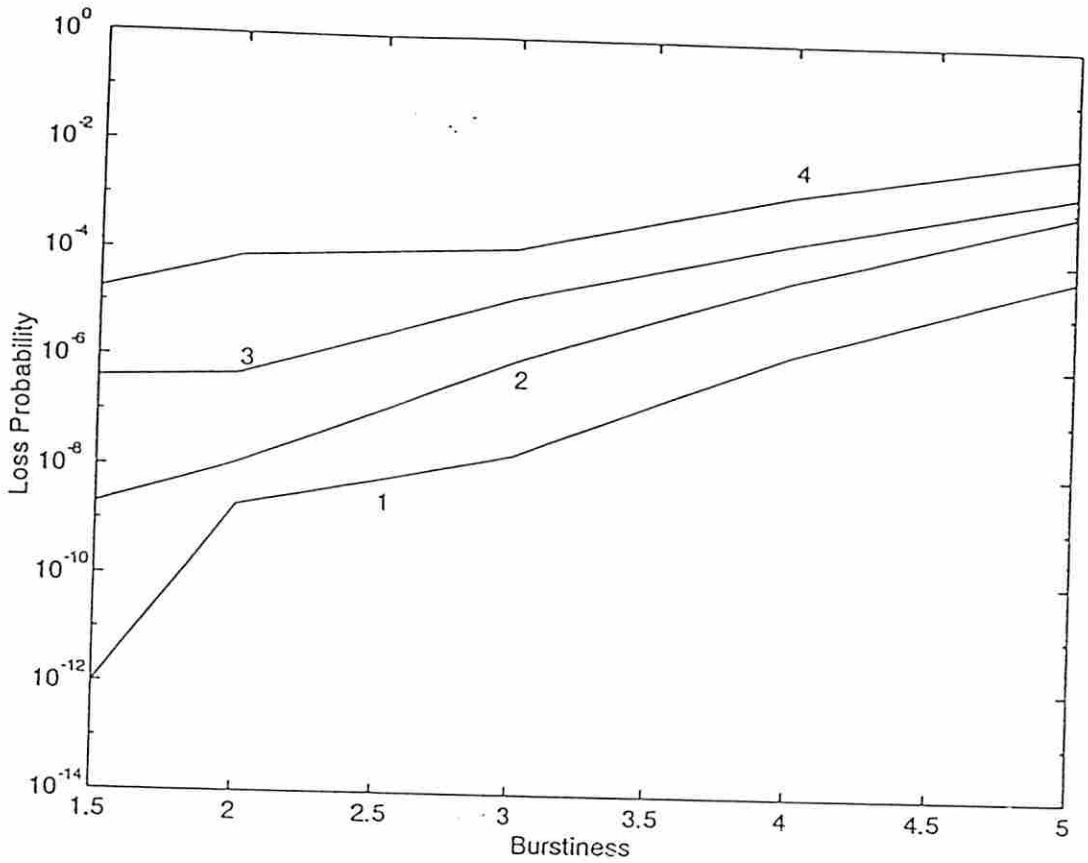

Figure 10: Loss rate as a function of the high priority burstiness

Chapter 4
A Queueing Network Framework for B-ISDN Networks

In B-ISDN networks, both the external traffic and the internal flows are correlated. Developing models for queuing networks with non-renewal flows is a challenge that we need to address in order to better understand B-ISDN networks. In this chapter, we introduce a queueing network framework for the performance evaluation of B-ISDN networks. We first explain the concept of product form queueing networks which admit exact and computationally simple solutions (section 4.1). We, then, specify the essential queueing operators for non product form networks (section 4.2). Finally, in section 4.3, we introduce a framework for queueing networks with Markov modulated flows.

4.1) Markovian Networks

A queueing network is a set of interconnected queues. Although the interest in queueing networks dates from the beginning of this century [65], it was only in the late 50's and early 60's that the first tractable analyses appeared. The successful application of queueing networks to the modelling of computer and communication systems has brought maturity to the field and has provided a source of inspiration for new challenging results.

The taxonomy of queuing networks basically takes into consideration: i) the characteristics of customers requirements (service demand and routing pattern), ii) the interaction with the outside world (open, closed and mixed), and iii) the existence of a product-form solution.

Customers characteristics - In a network, customers are grouped according to the path that they follow, and to the service requirements in the different nodes along their path. Networks are classified either as single class, if all customers have the same routing pattern and same service requirement at each network node or as multiple class, otherwise.

Interaction with outside world - a queueing network can be classified as open, closed or mixed. In an open network, outside arrivals and departures may occur at any node of the network. The population of a closed network is fixed and consequently, it does not accept outside arrivals/departures. A mixed network contains open and closed chains. The choice between an open or a closed model depends essentially on the type of service being considered. For instance, packet switching networks with datagrama service are generally modelled as open networks [67], whereas closed networks are used to analyze window flow controlled virtual circuit services at the transport level (the window size of a connection equals the population size) [68].

Product form - a queueing network admits a solution of the product form type, if the probability mass function of the distribution of customers among the network nodes can be computed by the product of the probability mass function for the number of customers in each node. More precisely, the solution of the network can be expressed as:

$$\pi(\boldsymbol{n}) = G \prod_{i=1}^{N} f_i\left(n_i\right)$$

where $\boldsymbol{n} = \left(n_1, \ldots, n_N\right)$, n_i is the total number of customers at node i and

$\pi(\boldsymbol{n})$ is the probability of having $\boldsymbol{n}$ customers distributed along the network, and G is a normalizing constant.

The concept of product-form solution was established by Jackson in 1957 [69]. He showed that a queueing network with Poisson inputs and exponential services admits solution of the product-form (actually, for closed networks, the product-form is achieved by the computation of a normalization constant). Jackson's results made use of the important theorem that states that the output process of a queue with exponential service time distribution is a Poisson process (Burke's theorem) [70]. The characterization of the type of network that admits a product-form solution is very important in queueing network theory. Baskett, Chandy, Muntz and Palacio [71] came out with the so-called BCMP conditions for a network to have a product-form solution. These conditions are:

i) For each different type of customer (chain), there is a routing probability P_{ij} that the customer leaves station i and goes to station j. This type of routing is called Markovian routing;

ii) The service disciplines and the service time distribution at each node are either: a) FCFS, if every customer class has the same exponential service time distribution, b) Processor sharing and arbitrary phase type (ex. hyperexponential) service time distribution, c) Infinite number of servers and arbitrary phase type service time distribution, d) Last-Come-First-Served-Preemptive-Resume and arbitrary phase type service time distribution;

iii) The service centers may have queue length dependent service rate;

iv) A customer after completion of service in a station may change its class.

Two general principles can be used for the verification of the existence of a product-form solution. In a quasi-reversible queue, the past departure, the current state and the future arrivals are mutually independent [72]. A local bal-

ance equation is one that equates the rate of flow out of the network state due to the departure of a customer from node *i* to the rate of flow into that network state due to the arrival of a customer to node *i*. In a queueing network, if all the queues are quasi-reversible or if the local balance equations are satisfied, then the queueing network admits a product-form solution.

The use of product-form models for the evaluation of computer network performance is only possible due to Kleinrock's independence assumption [72] which says that the transmission times of a packet at each channel (server) that it visits are independent and exponentially distributed. Although this is not correct, since the packet length does not change as it passes from channel to channel through the network, simulation experiments demonstrated that this assumption is quite good when several packet streams are merged onto a transmission line in a moderately to heavily loaded network. However, this assumption is known to fail in a tandem queue situation [73].

The restriction imposed by the product-form conditions led to the development of several approximation techniques for the analysis of real situations found in communication systems. Examples are [74]: i) general service time at a FCFS queue, ii) priority scheduling, iii) blocking, v) simultaneous resource possession.These approximation techniques in general seek an exact solution of an approximate model.

In summary, most advances in queueing network theory have followed two different directions: i) computational - to allow faster computation of the normalization constant for product-form networks; or ii) approximate analysis of non-product form systems, as noted above. However, there has been little work on non-Poisson arrival processes [76]-[78].

4.2) Queueing Networks With Non-Poisson Flows

The internal flows in a queueing network is Poisson, if and only if, the service time in each node is exponentially distributed and the external flow is Poisson. A more realistic approach is to consider the flows in the network as renewal processes which are fully characterized by the arrival rate and the variance of the number of arrivals. This approach was used in the Queueing Network Analyzer, a software package, developed at Bells Lab [78].

To describe the flows in a queueing network, we need to specify three flow transformations: *output, joining* and *splitting* [77], [78]. The *output* operator (transformation) defines the output process of a queue as a function of the input process and of the characteristics of the server. An internal node receives the flow from other nodes in the network and possibly receives external arrivals, too. The *joining* operator describes the process resulting from the merging of several streams. When a packet departs from a node, a decision is made to choose which node the packet will visit next. The *splitting* operator defines the characteristics of the flow for a pair nodes.

The exact solution for product form networks is obtained by solving each individual queue in isolation. In non-product networks, there are dependencies among the nodes which prevent us from obtaining the exact system solution. However, an approximate solution can be achieved by using the parametric decomposition approximation [79]. In a parametric decomposition, a queue is solved in isolation only after the input process is fully characterized. The dependencies among the queues are approximately captured by the input process parameters. The parametric decomposition with renewal approximations were used in the Queueing Network Analyzer.

Kroner et al. [80] considered a queueing network in which each connection at the transport level was represented as an on-off source. The long and

the short term fluctuations in the links were computed respectively by a fluid flow approximation and by an M/D/1-S model. They showed how to compute the end-to-end delay and the distribution of the transfer delay at each queue in a tandem network. Reising analyzed two queues in tandem [81]. The input of the first queue is a two-state Markov Modulated Bernoulli Process and the interfering traffic of both queues is a Bernoulli process. He derived an approximation model for the output process of the first queue as a two-state MMBP. Grienenfield [82] studied two queues in tandem by using a pertubation method to compute approximations for the mean and the variance of the end-to-end delay too. He considered that both the input and the interfering traffic were wide sense stationary processes He used a procedure for matching intermediate results in a Weibuill distribution to compute the jitter delay which is associated with the input traffic (pertubation). In the next section, we introduce a framework which models the flows in the network as Markov modulated processes.

4.3) Queueing Networks with Markov Modulated Flows

In B-ISDN networks, both external and internal flows are correlated and MMPP has been widely used for modelling of the B-ISDN external traffic (multimedia sources). By also modelling the internal flows as a Markov modulated process, we obtain an uniform representation of the flows in B-ISDN networks. With this approach in mind, we developed a procedure for representing the output process of a queue as a Markov Modulated process. It is important to mention that, rather than representing each individual connection which uses

a specific link, we represent the aggregated flow in the network links. Thus, our performance values are per link and not per individual connection that uses a link. For some performance measures, these two values are the same. For instance, the mean delay in networks with FCFS service. For other performance measures, the per link and the per connection values may differ (e.g., the mean number of consecutive losses). In these cases, the per link value can always be interpreted as an approximate estimation of the per connection value. In fact, the bulk of research in the queueing network usually reports the per link values and not the per connection ones.

The fixed size cell of ATM networks calls for a discrete time representation of B-ISDN networks. Thus, we choose to represent B-ISDN flows as a Discrete time Batch Markovian Arrival process. We analyze open networks with feed-forward topology. In other words, we do not allow cycles in the network. Extending the present work to generally connected topologies calls for iterative solutions. This is due to the fact that specification of the input flow of a node in a cycle depends on the input flow of each node of the cycle. In iterative type of solutions, an initial guess for each input flow is given and at each step, the input flows are updated in order to reach the solution of the network [83]. Service is furnished in a FCFS fashion and service time is constant. Each queue has a complete sharing finite capacity buffer space. For now, we do not consider any buffer priority mechanism. To solve a network, we employ the parametric decomposition approximation. In the next three subsections, we respectively define the output, the spliting and joining operators. In subsection 4.3.4, we illustrate the framework through numerical examples.

4.3.1) The Output Operator

Saito [83] studied the output process of an N/G/1 queue and particularly of the MMPP/D/1 queue. He analyzed the interdeparture time process and explicitly detailed the expressions for the mean, variance and covariance at lag 1. The burstiness of a process (C_p (z)) was defined as the Z-transform of the covariance of interdeparture time (in [84] C_p (z) was used to analyze integrated traffic). By comparing the C_p (z) curves for the input and for the output processes, Saito concluded that covariances are likely to be preserved. He also pointed out that the reduction of the coefficient of variation is larger in heavily loaded systems than it is in lightly loaded ones.

In [85] the output process of an ATM switch was modelled as two different memoryless processes: i) a Bernoulli process and ii) an Interrupted Poisson Process with the mean duration of the active state equal to the mean busy period. In both cases, the model failed to produce accurate results.

Takine et al. [86] studied the output process of a D-BMAP/D/1/K queue. They derived not only the expression for the m^{th}-factorial moment of the interdeparture time process, but also the distribution of the idle and the busy periods.

The aforementioned studies emphasize that the output process of a D-BMAP/D/1/K is correlated. At each time slot, we have at most one departure from the queue. The Markov Modulated Bernoulli Process, a correlated process with single arrivals, is a good candidate for modelling the output process of a D-BMAP/D/1/K queue. Our approach to represent the output process of a D-MAP/D/1/K queue is to define an MMBP process some of whose statistics match the numerical values of the true output process.

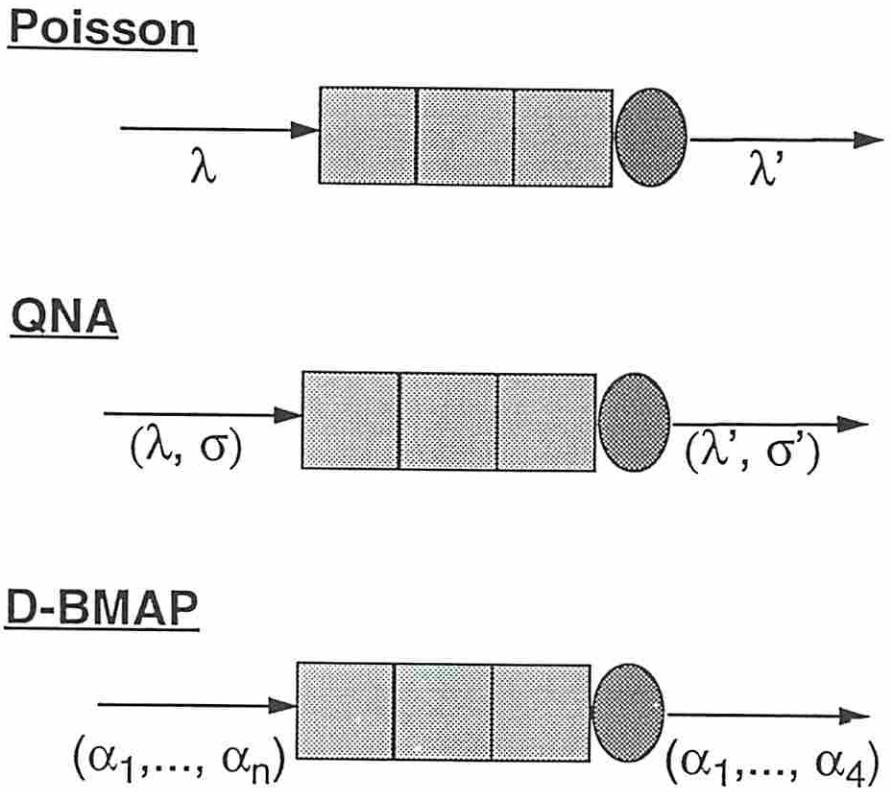

Figure 11: Output process representation in three different queueing network framework

Before showing how to match the statistics of the output process with the statistics of a two-state MMBP, we need to characterize the output process itself. Having exactly one departure at each time instant of a busy period suggests that we can represent the output process as a D-MAP in which the matrices D'_1 and D'_0 correspond respectively to busy and idle periods. In order to capture the behavior of busy/idle periods, we need to associate each state of the D-MAP with the phase of the arrival process and with the number of enqueued cells at the end of each time slot [45]. If we have a gated server (i.e., if a cell finds the server empty at its arrival slot, it can only be transmitted at the next slot) then, the output process is given by [45]:

$$D'_0 = \begin{bmatrix} D_0 & D_1 & \dots & D_{k-1} & \sum_{n=k}^{\infty} D_n \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$D'_1 = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ D_0 & D_1 & D_2 & \dots & D_{k-1} & \sum_{n=k}^{\infty} D_n \\ 0 & D_0 & D_1 & \dots & D_{k-2} & \sum_{n=k-1}^{\infty} D_n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & D_0 & \sum_{n=1}^{\infty} D_n \end{bmatrix}$$

On the other hand, if we have a cut-through type of service (i.e. the cell can be transmitted in the same slot in which it arrives) the output process is specified by:

$$D'_0 = \begin{bmatrix} D_0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$D'_1 = \begin{bmatrix} D_1 & D_2 & D_3 & \cdots & D_K & \sum_{n=K+1}^{\infty} D_n \\ D_0 & D_1 & D_2 & \cdots & D_{K-1} & \sum_{n=K}^{\infty} D_n \\ 0 & D_0 & D_1 & \cdots & D_{K-2} & \sum_{n=K-1}^{\infty} D_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & D_0 & \sum_{n=1}^{\infty} D_n \end{bmatrix}$$

In order to keep the computational complexity low, we choose to represent the output process as a two state MMBP. In a two state MMBP, we need to specify four parameters: the arrival and the transition probabilities in each state. An MMBP with a different number of states could be used but the number of parameters grows quadratically with the number of states. To capture exactly the correlation pattern of a process, we need to take into account all the possible time intervals (lags) from 1 to infinity. We choose to include in our procedure the long term (infinite) index of dispersion for counts which gives us the correlation when times goes to infinity. The long term index of dispersion is the ratio between the mean arrival rate and the variance of the number of arrivals. To avoid unnecessary computational growth, we choose to match the correlation at lag 1 and 2. Our procedure is [35]:

$$output_{mean} = MMBP_{mean}$$

$$output_{variance} = MMBP_{variance}$$

$$output_{covariance\ lag=1} = MMBP_{covariance\ lag=1}$$

$$output_{covariance\ lag=2} = MMBP_{covariance\ lag=2}$$

In [11], it was demonstrated that the mean arrival rate, the variance of number of arrivals and the covariance at lag k are given by:

$$\lambda = \pi\left(\sum_{k=1}^{\infty} kD_k\right)\bar{e}$$

$$var = \pi\left(\sum_{k=1}^{\infty} k^2 D_k\right)\bar{e} - \lambda^2$$

$$cov(x_1, x_k) = \pi\left(\sum_{n=1}^{\infty} nD_n\right)D^{k-2}\left(\sum_{n=1}^{\infty} nD_n\right)\bar{e} - \lambda^2$$

where $\bar{e}$ is the unit column vector and π is the steady state probability of the underlying Markov chain, i.e;

$$\pi D = \pi \qquad \pi\bar{e} = 1$$

To validate the matching procedure, we consider two queues in tandem. The input to the first queue is a two-state D-BMAP as defined in [86]. The input to the second queue is composed by the output process from the first queue and an interfering process. This interfering process is introduced in order to avoid the "non-queueing" phenomenon in tandem queues with constant ser-

vice time. To access the accuracy of the matching procedure, we compare the mean delay and the loss probability at the second queue when the output process is substituted by a two-state MMBP with the results produced by a simulation experiment (figure 12). We report the percentage error defined as ($|X_{sim} - X_{match}| / X_{sim}$) x 100 where X_{sim} and X_{match} are respectively the results produced by the simulation experiment and by the matching procedure. Both input to the first queue and interfering processes are two-state D-BMAP with the same transition probability in each state (a).

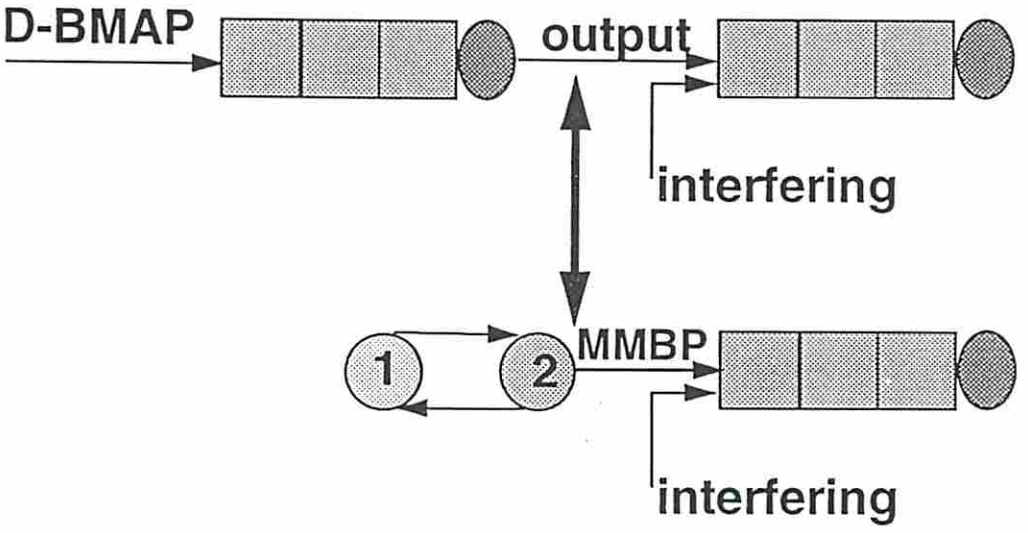

Figure 12: Scheme for the validation of the output procedure

The batch size is Poisson distributed with mean $(1 + c)\rho$ (state 1) and $(1 - c)\rho$ (state 2) where ρ is the overall traffic intensity and c is a parameter. It was demonstrated in [86] that the square coefficient of variation (C_v^2) and the correlation coefficient of the number of arrivals at lag n ($C_c(n)$) are respectively given by:

$$C_v^2 = \rho^{-1} + c^2$$

$$C_c(n) = \frac{c^2\rho}{1 + c^2\rho} \times (2\alpha - 1)^n$$

The data shown is this chapter corresponds to a server with gated service and buffer size 100. Time is normalized to one slot which has the same duration of a service time. In order to validate the accuracy of the computational procedure over a wide range of delay values, we vary the input parameters in a way such that we obtain the desired value at the second queue. Table 5 presents some results from our experiment. Errors are under 6% [35].

To evaluate the impact of the input process mean arrival rate, variance and correlation, we keep constant two of three input parameters: ρ, α and c, and vary the third one. the parameters of the interfering process are set in a way to avoid the "non-queueing" phenomenon in tandem networks with constant service time. Tables 6 and 7 display the delay as a function of c respectively for positively ($\alpha = 0.9$) and negatively ($\alpha = 0.1$) correlated streams. We observe that for positive correlated streams, the matching procedure provides slightly more precise results for higher values of the coefficient of variation than it does for lower ones. This trend did not emerge for negatively correlated streams. Tables 8 and 9 show the delay as a function of a for $c = 0.9$ and $c = 0.1$, respectively. We notice that the correlation coefficient does not affect the accuracy of the procedure. Table 10 shows the delay at the second queue as a function of the arrival rate. We note that the matching procedure provides more accurate results for highly loaded systems than it does for moderately and lightly loaded ones.

Regarding the loss estimation, Table 11 presents the results of an experiment similar to one displayed in Table 5. We note that the matching procedure is more precise for the estimation of higher values of the loss rate than is for lower values. We can also conclude that the accuracy of the procedure increases as ρ increases. In Table 12, we fix ρ at 0.75 and α at 0.9 and vary c. We observe that results are more accurate for higher values of the coefficient

of variation. In Table 13, we fix ρ at 0.75 and *c* at 0.9. The correlation coefficient did not affect the results.

To make sure that the interfering traffic parameters do not impact the obtained results, we vary ρ (Table 14), *c* (Table 15) and α (Table 16). From Tables 14 to 16, we can draw the conclusion that the interfering parameters exerted no influence in our findings. Finally, we investigate whether the buffer size affect our findings. Table 17 displays the results for buffer size ranging from 50 to 200. The buffer size did not interfere with the accuracy of the matching procedure. Overall, the percentage error for the delay estimation and for the loss rate estimation were respectively under 7% and 10%.

Our results are consonant with results found in [88]. Park et al. developed a procedure for modelling the output process of a D-BMAP/D/1/k queue as a two state MMBP by matching: i) the mean arrival rate, ii) the variance of the number of arrivals, iii) the interarrival correlation (lag1) and iv) the correlation of the number of arrivals (lag 1).

Input (ρ, *c*, α)	Interfering (ρ, *c*, α)	Analytical	Simula	conf interval	error
(0.8, 0.3, 0.9)	(0.1, 0.1, 0.9)	6.11832	5.772	0.02	6.0
(0.8, 0.7, 0.9)	(0.1, 0.1, 0.9)	12.3299	11.676	0.07	5.6
(0.9, 0.7, 0.9)	(0.05, 0.1, 0.9)	21.787	20.809	0.03	4.7
(0.65, 0.9, 0.9)	(0.3, 0.5, 0.9)	30.6381	29.375	0.03	4.3
(0.75, 0.9, 0.7)	(0.24, 0.9, 0.9)	44.808	43.251	0.008	3.6
(0.9, 0.3, 0.9)	(0.1, 0.1, 0.9)	52.6774	51.044	0.06	3.2
(0.85, 0.9, 0.9)	(0.2, 0.5, 0.9)	73.8761	72.004	0.5	2.6
(0.9, 0.9, 0.9)	(0.2, 0.1, 0.9)	87.494	85.947	0.11	1.8
(0.9, 0.9, 0.9)	(0.3, 0.1, 0.9)	95.7508	94.522	0.18	1.3

Table 5: Delay at the second queue

input (ρ, c)	interf ρ	Analytical	Simulation	conf interval	error
(0.4, 0.1)	0.4	3.182	2.988	0.004	6.5
(0.4, 0.3)	0.4	3.539	3.329	0.07	6.3
(0.4, 0.5)	0.4	4.644	4.377	0.03	6.1
(0.4, 0.7)	0.4	5.078	4.799	0.03	5.8
(0.4, 0.9)	0.4	5.694	5.397	0.20	5.5
(0.5, 0.1)	0.27	3.185	2.985	0.05	6.7
(0.5, 0.3)	0.27	3.467	3.259	0.06	6.4
(0.5, 0.7)	0.27	4.362	4.092	0.42	6.6
(0.5, 0.9)	0.27	5.917	5.593	0.01	5.8
(0.6, 0.1)	0.175	3.318	3.116	0.005	6.5
(0.6, 0.3)	0.175	3.565	3.338	0.10	6.8
(0.6, 0.7)	0.175	5.029	4.744	0.26	6.0
(0.6, 0.9)	0.175	6.336	5.994	0.008	5.7
(0.7, 0.1)	0.1	3.524	3.297	0.09	6.9
(0.7, 0.3)	0.1	3.701	3.469	0.17	6.7
(0.7, 0.7)	0.1	5.219	4.924	0.06	6.0
(0.7, 0.9)	0.1	6.219	5.906	0.53	5.2
(0.8, 0.1)	0.05	4.015	3.759	0.009	6.8
(0.8, 0.3)	0.05	4.097	3.843	0.12	6.6
(0.8, 0.5)	0.05	4.419	4.157	0.09	6.3
(0.8, 0.7)	0.05	5.753	5.438	0.008	5.8
(0.8, 0.9)	0.05	7.571	7.163	0.06	5.7
(0.9, 0.1)	0.03	4.630	4.364	0.17	6.1
(0.9, 0.3)	0.03	5.798	5.475	0.22	5.9
(0.9, 0.5)	0.03	8.413	7.967	0.05	5.6
(0.9, 0.7)	0.03	11.648	11.072	0.08	5.2
(0.9, 0.9)	0.03	13.945	13.281	0.04	5.0

Table 6: Delay as a function of c for input $\alpha = 0.9$ and interfering ($\alpha = 0.9$, $c = 0.1$)

input (ρ, c)	interfering (ρ, c, α)	Analytical	Simulation	conf interval	error
(0.4, 0.1)	0.4	3.154	2.959	0.12	6.5
(0.4, 0.3)	0.4	3.163	2.962	0.008	6.8
(0.4, 0.5)	0.4	3.163	2.967	0.05	6.6
(0.4, 0.7)	0.4	3.187	2.981	0.04	6.9
(0.4, 0.9)	0.4	3.193	2.987	0.007	6.9
(0.5, 0.1)	0.27	3.145	2.954	0.02	6.7
(0.5, 0.3)	0.27	3.148	2.958	0.008	6.4
(0.5, 0.7)	0.27	3.159	2.959	0.03	6.7
(0.5, 0.9)	0.27	3.193	2.993	0.07	6.7
(0.6, 0.1)	0.175	3.280	3.071	0.004	6.8
(0.6, 0.3)	0.175	3.291	3.099	0.005	6.2
(0.6, 0.7)	0.175	3.313	3.111	0.18	6.4
(0.6, 0.9)	0.175	3.317	3.127	0.16	6.0
(0.7, 0.1)	0.1	3.467	3.249	0.04	6.7
(0.7, 0.3)	0.1	3.485	3.286	0.12	6.0
(0.7, 0.7)	0.1	3.509	3.292	0.009	6.6
(0.7, 0.9)	0.1	3.525	3.301	0.07	6.6
(0.8, 0.1)	0.05	3.937	3.697	0.14	6.3
(0.8, 0.3)	0.05	3.945	3.718	0.14	6.1
(0.8, 0.5)	0.05	3.973	3.745	0.03	6.1
(0.8, 0.7)	0.05	4.034	3.795	0.05	6.3
(0.8, 0.9)	0.05	4.778	4.508	0.07	6.0
(0.9, 0.1)	0.03	4.519	4.251	0.13	6.3
(0.9, 0.3)	0.03	4.539	4.274	0.07	6.2
(0.9, 0.5)	0.03	4.542	4.281	0.21	6.1
(0.9, 0.7)	0.03	4.548	4.283	0.04	6.2
(0.9, 0.9)	0.03	4.556	4.286	0.16	6.3

Table 7: Delay as a function of c for input $\alpha = 0.1$ and interfering ($\alpha = 0.9$, $c = 0.1$)

input (ρ, α)	interfering ρ	Analytical	Simulation	conf interval	error
(0.4, 0.1)	0.4	3.185	2.983	0.04	6.7
(0.4, 0.3)	0.4	3.291	3.076	0.009	7.0
(0.4, 0.7)	0.4	3.951	3.699	0.03	6.8
(0.4, 0.9)	0.4	5.694	5.397	0.02	5.5
(0.5, 0.1)	0.27	3.195	2.998	0.007	7.0
(0.5, 0.3)	0.27	3.309	3.101	0.04	6.7
(0.5, 0.7)	0.27	4.156	3.895	0.18	6.7
(0.5, 0.9)	0.27	5.917	5.593	0.05	5.8
(0.6, 0.1)	0.175	3.341	3.122	0.008	7.0
(0.6, 0.3)	0.175	3.411	3.191	0.008	6.9
(0.6, 0.7)	0.175	3.981	3.738	0.11	6.5
(0.6, 0.9)	0.175	6.336	5.994	0.20	5.7
(0.7, 0.1)	0.1	3.536	3.307	0.06	6.9
(0.7, 0.3)	0.1	3.597	3.371	0.07	6.7
(0.7, 0.7)	0.1	4.082	3.822	0.04	6.8
(0.8, 0.1)	0.05	4.034	3.777	0.03	6.8
(0.8, 0.3)	0.05	4.061	3.806	0.006	6.7
(0.8, 0.7)	0.05	4.441	4.178	0.17	6.3
(0.8, 0.9)	0.05	7.571	7.163	0.19	5.7
(0.9, 0.1)	0.03	4.511	4.251	0.18	6.1
(0.9, 0.9)	0.03	14.025	13.281	0.23	5.6

Table 8: Delay as a function of α for input $c = 0.9$ and interfering ($\alpha = 0.9$, $c = 0.1$)

input (ρ, α)	interfering ρ	Analytical	Simulation	conf interval	error
(0.4, 0.1)	0.4	3.182	2.988	0.008	6.5
(0.4, 0.3)	0.4	3.196	2.993	0.04	6.8
(0.4, 0.7)	0.4	3.202	2.993	0.03	7.0
(0.4, 0.9)	0.4	3.207	2.997	0.008	7.0
(0.5, 0.1)	0.27	3.185	2.985	0.03	6.7
(0.5, 0.3)	0.27	3.188	2.982	0.005	6.9
(0.5, 0.7)	0.27	3.192	2.991	0.06	6.7
(0.5, 0.9)	0.27	3.194	2.993	0.07	6.7
(0.6, 0.1)	0.175	3.311	3.106	0.09	6.6
(0.6, 0.3)	0.175	3.309	3.105	0.02	6.6
(0.6, 0.7)	0.175	3.311	3.106	0.009	6.6
(0.6, 0.9)	0.175	3.318	3.116	0.03	6.5
(0.7, 0.1)	0.1	3.522	3.301	0.05	6.7
(0.7, 0.3)	0.1	3.522	3.304	0.11	6.6
(0.7, 0.7)	0.1	3.524	3.301	0.03	6.7
(0.7, 0.9)	0.1	3.532	3.306	0.15	6.8
(0.8, 0.1)	0.05	4.034	3.795	0.009	6.3
(0.8, 0.3)	0.05	4.058	3.796	0.03	6.9
(0.8, 0.7)	0.05	4.061	3.795	0.21	7.0
(0.8, 0.9)	0.05	4.015	3.759	0.09	6.8
(0.9, 0.1)	0.03	4.506	4.246	0.18	6.1
(0.9, 0.9)	0.03	4.630	4.364	0.13	6.1

Table 9: Delay as a function of α for input $c = 0.1$ and interfering ($\alpha = 0.9$, $c = 0.1$)

input ρ	analytical	simulation	conf interval	error
0.4	3.259	3.049	0.009	6.9
0.5	4.436	4.169	0.06	6.4
0.6	7.359	6.943	0.13	6.0
0.7	17.955	17.116	0.18	4.9
0.8	54.726	53.029	0.03	3.2
0.9	87.494	85.947	0.15	1.8

Table 10: Delay as a function of ρ for input (c = 0.9, α =0.9) and interfering (ρ = 0.2, c = 0.1, α = 0.9)

input (ρ, c)	Analytical	simulation	conf interval	error
(0.8, 0.9)	1.3962e-1	1.3047e-1	2.35e-3	4.2
(0.75, 0.9)	2.98797e-2	2.8322e-2	6.38e-4	5.5
(0.7, 0.9)	2.5675e-3	2.4245e-3	5.18e-5	5.2
(0.675, 0.9)	6.01925e-4	5.6839e-4	4.93e-6	5.9
(0.65, 0.9)	8.4937e-05	8.0281e-05	1.82e-07	6.8
(0.75, 0.1)	5.2704e-06	4.8397e-06	2.94e-08	8.9
(0.7, 0.47)	3.0113e-08	2.7779-e-07	5.10e-09	9.4

Table 11: Loss rate at the second queue for input α = 0.9 and interfering (ρ = 0.2, c = 0.1, α = 0.9)

input c	Analytical	Simulation	conf interval	error
0.1	1.7783e-06	1.6496e-06	7.45e-08	7.8
0.3	1.1964e-05	1.1266e-05	4.38e-07	6.2
0.5	3.0732e-4	2.9241e-4	2.49e-06	5.1
0.7	2.1992e-3	2.1086e-3	6.73e-05	4.3
0.9	2.9031e-2	2.8322e-2	8.74e-04	2.5

Table 12: Loss rate as a function of c for input ($\alpha = 0.9$ $\rho = 0.75$) and interfering ($\rho = 0.2$, $c = 0.1$, $\alpha = 0.9$)

input α	Analytical	Simulation	conf interval	error
0.3	4.2764e-05	4.0535e-05	7.62e-07	5.5
0.7	8.5021e-4	8.08944e-4	2.53e-06	5.1

Table 13: Loss rate as a function of α for input ($\rho = 0.75$, $c = 0.9$) and interfering ($\rho = 0.2$, $c = 0.1$, $\alpha = 0.9$)

interf. ρ	delay anal	delay simu	error	loss rate anlytical	loss rate simu.	error
0.2	53.013	51.369	3.2	8.5709e-2	8.3864e-2	2.2
0.3	91.355	89.739	1.8	4.2126e-1	4.1463e-1	1.6
0.4	97.565	96.218	1.4	6.2832e-1	6.2087e-1	1.2
0.5	99.144	98.066	1.1	7.5129e-1	7.4239e-1	1.2
0.6	99.543	98.851	0.7	8.2811e-1	8.1991e-1	1.0
0.7	99.842	99.255	0.3	8.8049e-1	8.7177e-1	1.0

Table 14: Accuracy of the results as a function of the interfering process ρ for input ($c = 0.1$, $\alpha = 0.9$) and interfering ($\rho = 0.2$, $c = 0.1$, $\alpha = 0.9$)

interf c	delay analyt	delay simula	error	loss rate anlytical	loss rate simula	error
0.3	11.317	10.697	5.8	2.4742e-4	2.3564e-4	5.0
0.5	11.474	10.907	5.9	2.8768e-4	2.7346e-4	5.2
0.7	11.778	11.154	5.6	3.5054e-4	3.3417e-4	4.9
0.9	12.037	11.399	5.6	4.5878e-4	4.3735e-4	4.9

Table 15: Accuracy of the results as a function of the interfering process c for input ($\rho = 0.8$, $c = 0.7$, $\alpha = 0.9$) and interfering ($\rho = 0.1$, $\alpha = 0.9$)

interf α	delay analyt	delay simula	error	loss rate anlytical	loss rate simula	error
0.1	53.091	51.296	3.5	8.78646-2	8.3284e-2	5.5
0.3	52.928	51.287	3.2	8.7931e-2	8.3425e-2	5.4
0.7	53.258	51.408	3.6	8.8812e-2	8.4182e-2	5.5

Table 16: Accuracy of the results as a function of the interfering process α for input ($\rho = 0.8$, $c = 0.7$, $\alpha = 0.9$) and interfering ($\rho = 0.2$, $c = 0.1$)

buffer size	delay analytical	delay simula	error	loss rate analytical	loss rate simula	error
50	17.982	17.158	6.1	1.8722e-3	1.7985e-3	4.1
100	17.982	17.158	6.1	1.8722e-3	1.7985e-3	4.1
150	17.982	17.158	6.1	1.8722e-3	1.7985e-3	4.1
200	17.982	17.158	6.1	1.8722e-3	1.7985e-3	4.1

Table 17: Accuracy of the results as a function of the buffer size buffer for input ($\rho = 0.7$, $c = 0.9$, $\alpha = 0.9$) and interfering ($\rho = 0.2$, $c = 0.1$, $\alpha = 0.9$)

4.3.2) Splitting

We assume that the routing decisions are memoryless. In other words, every cell that departs from queue *i* goes to queue *j* with a certain fixed probability. When characterizing the flow between two nodes, we represent the output process of the first queue as an MMBP process, and then model the flow that goes to the second queue as an MMBP with parameters:

$$(p_{ij} \times p_1, p_{ij} \times p_2, \alpha_1, \alpha_2)$$

where p_{ij} is the probability that a cell leaves node *i* and goes to node *j*, p_1 (p_2) and α_1 (α_2) are respectively the arrival and transition probability in state 1 (2).

Stavrakakis [87]advocates that there is a destination correlation between two consecutive cells that departure from a queue given the likelihood that these two cells belong to the same connection. He showed that significant errors can be made, if we neglect the destination correlation. He substantiates his claim by defining an MMBP whose states takes into account the destination correlation and compares the actual traffic that goes to a fixed destination to the traffic when memoryless routing is used. However, he fails to address the mixing that exists when several connections share the same buffer space. In our investigation, we did not find any significant impact of this destination correlation on our results.

4.3.3) Joining

The superposition of two D-BMAP processes with m_1, m_2 states and n_1, n_2 maximum batch size is also a D-BMAP with $m_1 \times m_2$ states and $n_1 + n_2$ maximum batch size. The matrix D_k whose elements $(d_{ij})_k$ which give the probabil-

ity of going from state i to state j and having a batch arrival of size k is computed as:

$$D_k = \sum_{q=0}^{n_1} D_q^{(1)} \otimes D_{k-q}^{(2)}$$

For instance, the superposition of an MMBP and a D-BMAP with maximum batch size of 2 is given by:

$$D_0 = D_0^{(1)} \otimes D_0^{(2)}$$

$$D_1 = D_0^{(1)} \otimes D_1^{(2)} + D_1^{(1)} \otimes D_0^{(2)}$$

$$D_2 = D_0^{(1)} \otimes D_2^{(2)} + D_1^{(1)} \otimes D_1^{(2)} + D_2^{(1)} \otimes D_0^{(2)}$$

$$D_3 = D_0^{(1)} \otimes D_3^{(2)} + D_1^{(1)} \otimes D_2^{(2)} + D_2^{(1)} \otimes D_1^{(2)} + D_3^{(1)} \otimes D_0^{(2)}$$

where $A \otimes B$ denotes the Krockener product of matrix A by matrix B.

The growth of the aggregated process number of states restricts the use of the present framework to the analysis of networks whose nodes have low connectivity. This restriction could be attenuated, if procedures for reducing the number of states of a Markov modulated process were developed. An alternative solution would be to develop techniques to solve a queue with Markov Modulated process which do not depend on the number of states.

We have now defined all the elementary queueing network operations. In the section 4.4, we illustrate how these operations can be used to predict the performance in ATM feed-forward networks, and in section 4.6, we give some numerical examples.

4.4) The Computation of End-to-End Performance

To compute the end-to-end delay in an ATM virtual path, we make use of the parametric decomposition approximation, i.e., the queues are analyzed in isolation after their input process are fully characterized. In this approach, the dependencies among the queues are approximated by the flow parameters. We concentrate on ATM networks whose topology can be described as an acyclic directed graph. Otherwise, if we consider generally connected networks, we would have to define iterative procedures for determining the input flow of nodes in a cycle. We assume that there are two distinct sets of nodes: sets *E* and *I*. The elements of set *E* receive only input (external) traffic to the network (i.e., elements of set *E* are network's entry points). The elements of set *I* are nodes whose input is composed of the output process of other nodes and possibly input traffic to the network (i.e. nodes belonging to set *I* are network internal nodes which can also receive external traffic). We define S_k as the set of nodes whose input traffic can be determined only at iteration *k* of the computational procedure. In other words, nodes belonging to S_k have at least one input link whose flow parameters can only be computed at step *k-1*. We compute the occupancy distribution of all nodes of S_k at step *k*, and we denote a link whose traffic parameters have been determined as a marked link. The computational procedure can be summarized as [88]:

1 - $k = 1$ and $S_1 = E$;

2 - While $S_k \neq \varnothing$ do:

2.1 - Characterize the input process of every S_k node by performing a joining operation of all input links to each node. For S_1 nodes, the input processes are given by the input process to the network;

2.2 - Compute the steady state queue length distribution of every S_k node. Compute the mean delay seen by an arriving cell at an S_k node;

2.3 - Characterize the output process of every S_k node by matching the statistics of the S_k nodes' output process with the statistics of a two-state MMBP;

2.4 - Characterize the process of every outgoing link of S_k nodes by performing a spliting operation on the S_k nodes' output process;

2.5 - Mark all outgoing links of every S_k node;

2.6 - $k = k + 1$.

By assuming a feed-forward topology, we guarantee that the computational procedure terminates.

4.5) Numerical Examples

In this section, we illustrate our queueing network framework for the estimation of end-to-end performance in ATM networks. In the first example (Table 18), we show the end-to-end delay for a five nodes tandem network (Figure 13). We observe that there is an increase of less than 1% in the percentage error when compared to the results found in the study with an isolated queue. Figure 14 displays the end-to-end delay as a function of ρ for a five node network with interfering process parameters (ρ=0.075,c=0.1,α=0.9) and input process parameters (c = 0.9, α = 0.9)

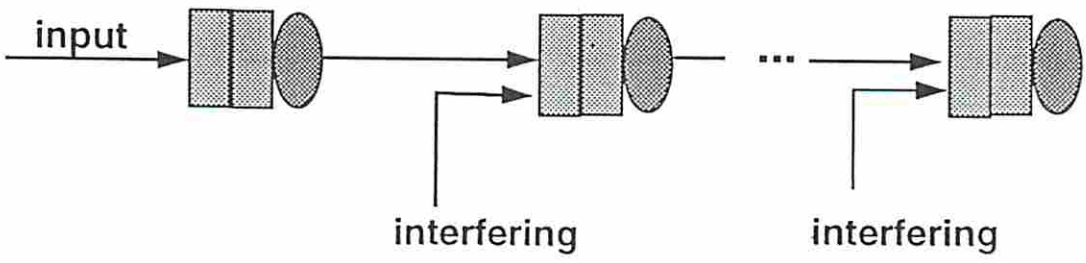

Figure 13: Tandem network

input (ρ, c, α)	interfering (ρ, c, α)	analyt	simula	conf interval	error
(0.65, 0.9, 0.9)	(0.65, 0.9, 0.9)	51.791	48.906	0.52	5.9
(0.7, 0.9, 0.9)	(0.7, 0.9, 0.9)	98.84	94.313	0.08	4.8
(0.75, 0.9, 0.9)	(0.075, 0.1, 0.9)	163.396	156.214	1.33	4.6
(0.8, 0.9, 0.9)	(0.075, 0.1, 0.9)	231.947	223.456	0.49	3.8
(0.825,0.9, 0.9)	(0.075, 0.1, 0.9)	240.065	256.631	0.05	3.5
(0.85, 0.9, 0.9)	(0.075, 0.1, 0.9)	297.728	289.337	1.89	2.9
(0.9, 0.9, 0.9)	(0.075, 0.1, 0.9)	359.814	350.014	0.72	2.8
(0.8, 0.9, 0.9)	(0.4, 0.1, 0.9)	367.73	407.877	0.14	2.2

Table 18: End-to-end delay for a five nodes tandem network

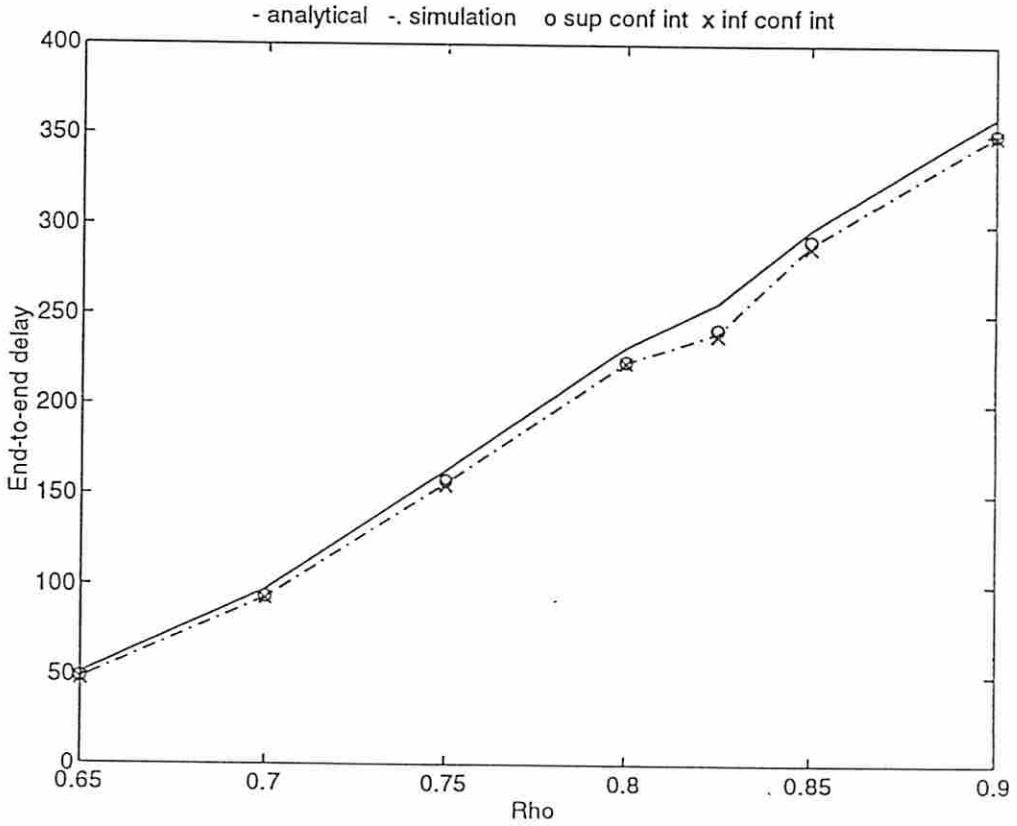

Figure 14: End-to-end delay as a function of ρ

We show two examples of feed-forward topology in Figures 14 and 15 [88]. The loads at nodes A, B and C are ($\rho = 0.75$, $c = 0.9$, $\alpha = 0.9$), ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$) ($\rho = 0.75$, $c = 0.1$, $\alpha = 0.9$) respectively. Table 19 presents the routing probabilities for the network of figure 14. Table 20 and Table 21 respectively display the delay at each node and the end-to-end delays. For Figure 15 network, the arrival process parameters are ($\rho = 0.75$, $c = 0.9$, $\alpha = 0.9$) for nodes G, H, and J. The routing probabilities are given in Table 22. Table 23 and Table 24 respectively show the delay at each node and the end-to-end delays.

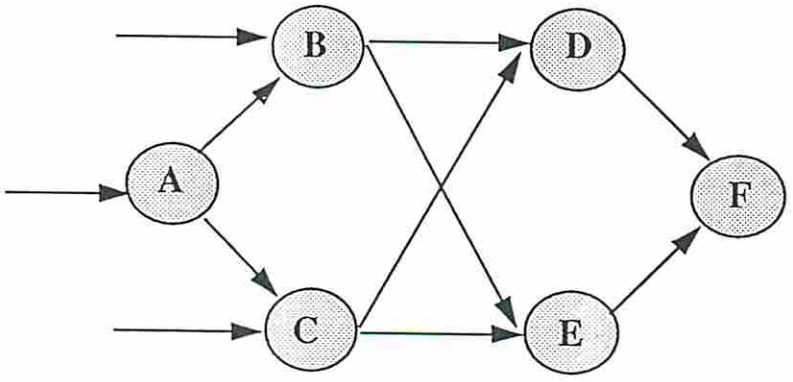

Figure 15: First feed-forward network

	B	C	D	E
A	0.7	0.3		
B			0.4	0.6
C			0.6	0.4

Table 19: Routing probabilities for figure 14 network

	estimated	simulation	error
A	11.73	11.01 ± 0.16	6.5
B	69.38	66.65 ± 0.65	4.1
C	26.81	25.49 ± 0.43	5.2
D	22.68	21.42 ± 0.11	5.9
E	25.91	24.56 ± 0.10	5.5

Table 20: Delay per node for figure 14 network

	estimated	simulation	error
ABD	103.79	99.08	4.7
ABE	107.08	102.22	4.9
ACD	61.22	57.92	5.7
ACE	64.45	61.06	5.5

Table 21: End-to-end delays for Figure 14 network

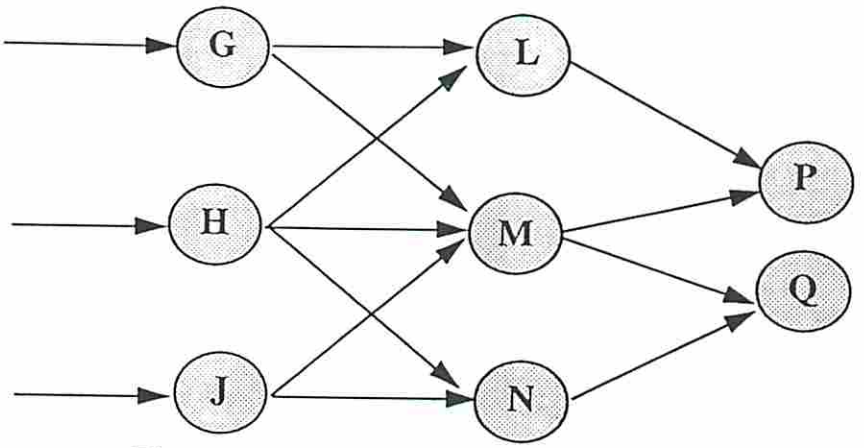

Figure 16: Second Feed-forward network

	L	M	N	P	Q
G	0.9	0.1			
H	0.1	0.8	0.1		
J		0.3	0.7		
M				0.5	0.5

Table 22: Routing probabilities Figure 15 network

	estimated	simulation	error
G	11.64	11.04 ± 0.08	5.4
H	11.64	11.00 ± 0.07	5.8
J	11.64	11.05 ± 0.06	5.3
L	2.62	2.46 ± 0.02	6.5
M	12.97	12.34 ± 0.12	5.1
N	2.52	2.36 ± 0.01	6.8
P	48.07	46.08 ± 0.78	4.2
Q	47.89	45.83 ± 0.61	4.5

Table 23: Delay per node Figure 15 network

	estimated	simulation	error
GLP	62.33	59.57	4.6
GMP	72.68	69.45	4.7
GMQ	72.5	69.21	4.8
HLP	62.33	59.59	4.6
HMP	72.68	69.42	4.7
HMQ	72.5	69.17	4.8
HNQ	62.05	59.44	4.4
JMP	72.68	69.47	4.6
JMQ	72.5	69.22	4.7
JNQ	62.05	59.24	4.7

Table 24: End-to-end delays for Figure 15 network

We study the ATM network shown in Figure 16 with the traffic sessions specified in Table 25 (where R is the channel data rate) [89]. In this example, all links are assumed to have a data rate $R = 22.5$ Mbps and all traffic sessions are assumed to support a similar traffic mix of (roughly) 40%, 40% and 20% from video, voice and data sources respectively. The corresponding (open) queueing network for this example is given by Figure 17. More specifically, for analytical results, we approximate each traffic session as a two-state D-BMAP using the model described in [32]. We first compare the analytical results with that of simulation where the traffic sources are exactly represented and routing of cells is done according to the session (whereas the analytical model uses random traffic splitting). Note that at node *AB* in Figure 17, we assume that cells from different sessions are served alternately. The average delay and loss rate for each node are presented in Table 26 and Table 27 respectively. In this simulation, we also measure average delay and loss rate on a per ses-

sion basis which is also shown in the tables. We see that the delay results from our model are in very good agreement with the simulation. The loss figures also shown reasonable agreement but not quite as good as the delay results.

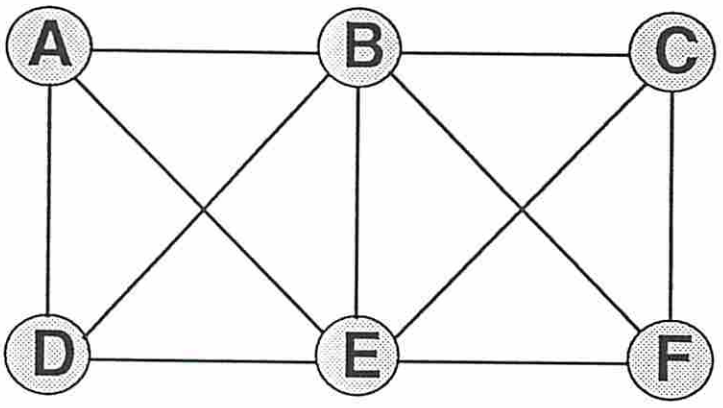

Figure 17: Communication network

traffic session	load	source	destination	routing
γ_1	$0.4335R$	*A*	*C*	via *B*
γ_2	$0.4335R$	*A*	*E*	via *B*
γ_3	$0.8671R$	*D*	*F*	via *E*
γ_4	$0.4335R$	*B*	*C*	---
γ_5	$0.4335R$	*B*	*E*	---

Table 25: Traffic Sessions of Figure 16 communication network

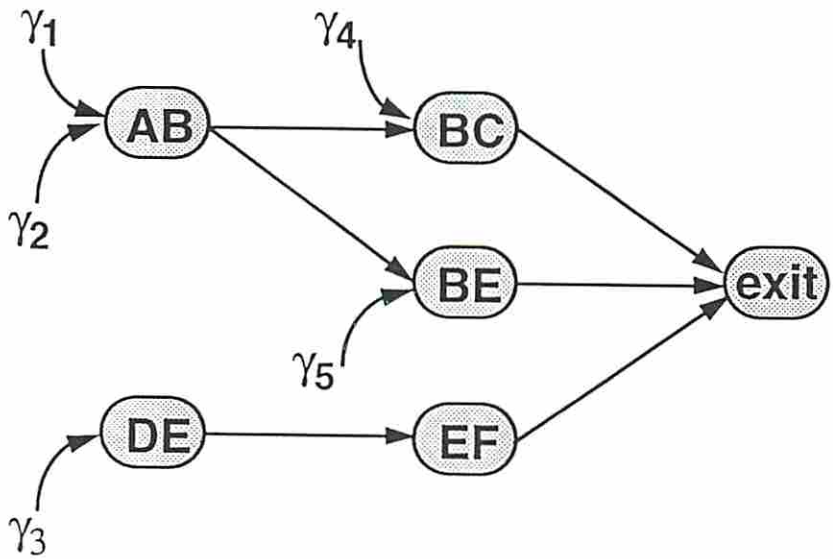

Figure 18: Queueing network of Figure 16 connections

	analytical	simulation	error
Link AB Session γ_1 Session γ_2	24.0	25.3 (±0.7) 25.3 (±0.7) 25.4 (±0.7)	5.1
Link DE Session γ_3	24.0	25.2 (±0.6) 25.2 (±0.6)	4.5
Link BC Session γ_1 Session γ_4	20.0	21.3 (±0.8) 21.2 (±0.8) 21.6 (±0.8)	6.3
Link BE Session γ_2 Session γ_5	20.0	21.4 (±0.5) 21.2 (±0.5) 21.9 (±0.5)	6.7

Table 26: Delay at each link of Figure 17 network

	analytical	simulation	error
Link AB Session γ_1 Session γ_2	$7.1\text{x}10^{-3}$	$8.1(\ 0.7)\text{x}10^{-3}$ $8.0(\ 0.6)\text{x}10^{-3}$ $8.3(\ 0.6)\text{x}10^{-3}$	13.0
Link DE Session γ_3	$7.1\text{x}10^{-3}$	$8.2(\ 0.5)\text{x}10^{-3}$ $8.2(\ 0.5)\text{x}10^{-3}$	14.4
Link BC Session γ_1 Session γ_4	$1.5\text{x}10^{-3}$	$1.3(\ 0.8)\text{x}10^{-3}$ 0.0 $2.5(\ 0.2)\text{x}10^{-3}$	19.6
Link BE Session γ_2 Session γ_5	$1.5\text{x}10^{-3}$	$1.3(\ 0.5)\text{x}10^{-3}$ 0.0 $2.2(\ 0.2)\text{x}10^{-3}$	17.8

Table 27: Loss rate at each link of Figure 17 network

End-to-end performance results on a per session basis are given in Table 28 and Table 29. In the analytical results, the average delay of a session is calculated as the sum of the average delay of all queues on the path of the session and the loss rate is calculated as:

$$1 - \prod_{k \in \xi} (1 - p_k)$$

where ξ is the set of queues alone the path of the session and P_k is the loss probability (over all sessions for the analytical results) at queue k We find surprisingly good agreement except in the loss rates of sessions γ_1 and γ_5. We believe this to be due to the fact that, as seen in the simulation, session γ_1 and γ_2 traffic suffer no loss (since it is given priority over new traffic) here but the analytical model lumps the loss of session γ_4 with γ_1 at link *BE*.

session	analytical	simulation	error
γ_1	44.0	46.3	5.2
γ_2	44.0	47.3	6.8
γ_3	25.0	26.1	4.2
γ_4	20.0	21.6	7.5
γ_5	20.0	21.2	5.5

Table 28: End-to-end delay per session of Figure 16

session	analytical	simulation	error
γ_1	$8.5\text{x}10^{-3}$	$8.0\text{x}10^{-3}$	7.3
γ_2	$8.5\text{x}10^{-3}$	$8.3\text{x}10^{-3}$	2.7
γ_3	$7.1\text{x}10^{-3}$	$8.2\text{x}10^{-3}$	14.3
γ_4	$1.5\text{x}10^{-3}$	$2.5\text{x}10^{-3}$	39.2
γ_5	$1.5\text{x}10^{-3}$	$2.2\text{x}10^{-3}$	31.4

Table 29: End-to-end loss rate per session of Figure 16

We study a higher capacity network with link rates of R = 155.52 Mbps (standard OC-3 data rate). The traffic mix is shown in Table 30. Table 31 and Table 32 respectively show the delay and the loss rate. We see better agreement between the analytical model and the simulation from which we can conclude that the errors introduced by simplification necessary to solve the queueing network tend to reduce as the network is scaled up.

traffic session	load	source	destination	routing
γ_1	0.5R	A	C	via B
γ_2	0.3R	A	E	via B
γ_3	0.6R	D	F	via E
γ_4	0.35R	B	C	---
γ_5	0.55R	B	E	---

Table 30: Traffic Sessions of Figure 16 communication network with OC-3 rate

session	analytical	simulation	error
AB	7.13	6.72(0.001)	5.3
DE	4.84	4.51(0.003)	6.1
BC	6.18	5.77(0.016)	6.1
BE	7.57	7.12(0.003)	5.3

Table 31: End-to-end delay per session of Figure 16

session	analytical	simulation	error
AB	4.8×10^{-5}	$4.4(\ 0.2)\times10^{-5}$	7.8
DE	2.0×10^{-5}	$1.9(\ 0.04)\times10^{-5}$	8.8
BC	1.9×10^{-5}	$1.6(\ 0.3)\times10^{-5}$	8.6
BE	4.3×10^{-5}	$4.0(\ 0.3)\times10^{-5}$	8.1

Table 32: End-to-end delay per session of Figure 16

Chapter 5
A Framework for Queueing Networks with Prioritized Flows

In the ATM standard for Broadband Integrated Service Digital Network, selective discard was adopted as a mechanism for copying with diverse quality of service loss requirements. The advantage of selective discard is well accepted. However, researchers have studied selective discard only in the single node context. Understanding selective discard in a network wide context is still a challenge. Developing tools for the computation of end-to-end loss performance is of paramount importance for overcoming this challenge. In this chapter, we extend the chapter 4 queueing network framework to incorporate prioritized flows. Although the generalization to multiple classes is straightforward, we present results for only two levels of priority. We assume that the flows can be modelled as D-BMAP$^{[H,L]}$ processes. A D-BMAP$^{[c_1, \cdots, c_N]}$process which was introduced in section 2.4.2. A D-BMAP$^{[H,L]}$ is a D-BMAP process in which each cell of a batch is classified as either high or low priority.

In a D-BMAP$^{[H, L]}$, the probability of a cell having a certain priority (priority probability) is independent of other cells and is a function of the state of the underlying Markov chain. The i^{th} element of $\vec{p}_{high}$ ($\vec{p}_{low}$), $p_{high}(i)$ [$p_{high}(i)$] gives the high (low) priority probability when the process is in state i. We define $p_{high} = \vec{p}_{high} \cdot \vec{\pi}$ ($p_{low} = \vec{p}_{low} \cdot \vec{\pi}$) to be the unconditional high (low) priority probability.

A D-BMAP $^{[H,L]}$ is fully specified by the matrices D_n of the corresponding D-BMAP process plus the vector p_{high} (p_{low}). Sections 5.1, 5.2 and 5.3 respectively define the output, the splitting and the joining operator for queueing networks with prioritized Markov modulated flows.

5.1) The Output Process

If we observe a work conserving D-BMAP$^{[H,L]}$/D/1/K queue and disregard the priority classification, we notice that the statistics of the output process is the same of the output process of a D-BMAP/D/1/K. Therefore, we compute the parameters for the output process in two steps. In the first step, we model the output process as a two state MMBP without taking into account the priority classification. In the second step, we compute the probability of a cell having a certain priority (figure 18).

Regarding the computation of the probability of a cell having a certain priority, if we had an infinite buffer space this probability would be the same as the priority probability of the input process. However, in a finite buffer queue, we need to take into account the loss rate per class due to buffer overflow. Thus, our procedure is:

$$\Pi_{high} = \frac{p_{high} \times (1 - R_{high})}{p_{high} \times (1 - R_{high}) + p_{low} \times (1 - R_{low})}$$

$$\Pi_{low} = \frac{p_{low} \times (1 - R_{low})}{p_{high} \times (1 - R_{high}) + p_{low} \times (1 - R_{low})}$$

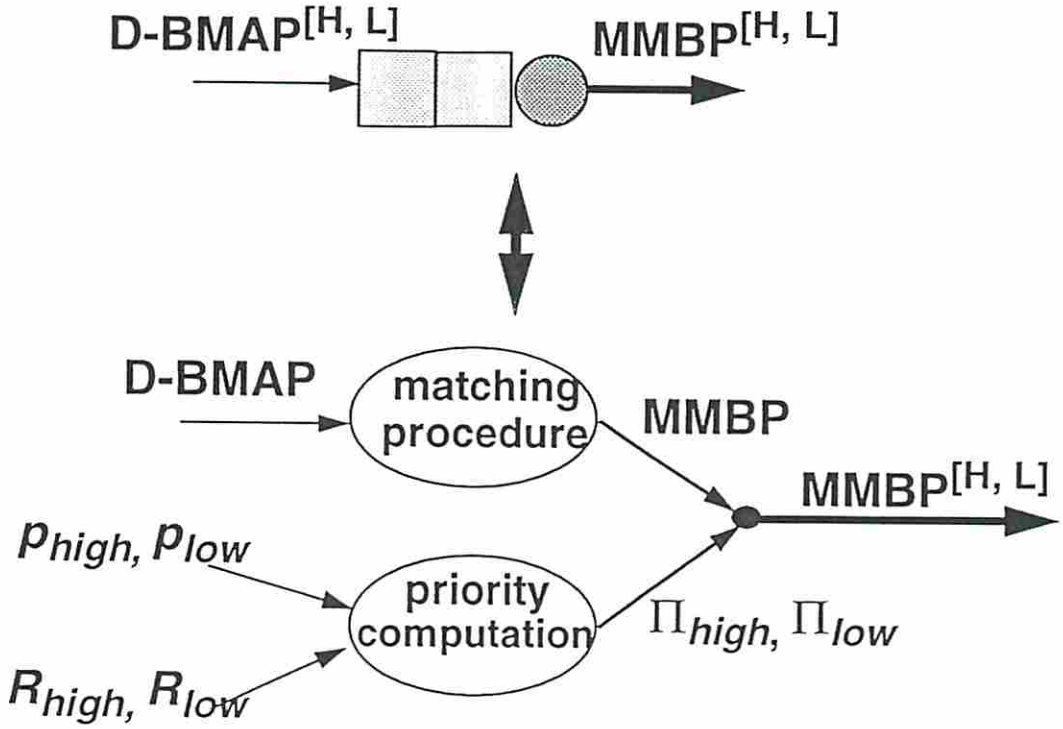

Figure 18: The modeling procedure of a prioritized output process.

where R_{high} (R_{low}) is the high (low) priority loss rate

Π_{high} (Π_{low}) is the output high (low) priority probability,

$p_{high} = \hat{p}_{high} \cdot \hat{\pi}$ ($p_{low} = \hat{p}_{low} \cdot \hat{\pi}$) is the input process high (low) priority probability

To validate the matching procedure, we consider the same two queues in tandem used in section 5.3.1. Both input and interfering process are D-BMAP$^{[H,L]}$, and we use the D-BMAP process as defined in [44] to create the validation examples. We generate examples in a numerical range which can reasonably be verified employing the Monte Carlo simulation, i.e. up to loss of about 10^{-7}

Tables 33 and 34 show respectively the high and low priority loss rate for a wide range of values. Our procedure is more accurate when it estimates loss

rate for the low priority class than it is for the high priority class. The errors of the low priority loss rate estimation are similar to the errors of the aggregated loss rate. We also notice that our procedure is more precise for high values of the loss rate than it is for lower ones. Errors were below 15% for the high priority class and below 10% for the low priority class. Tables 33 and 34 illustrate that our procedure is more accurate for higher values of ρ. Tables 35-36 and 37-38 display respectively the precision of the procedure as a function of c for positively and negatively correlated streams. Our procedure is slightly more accurate for higher values of c than it is for lower ones. The precision increases significantly as p_{high} increases. The estimations were slightly more accurate (1.5) for positively than for negatively correlated streams. The difference in the precision is much more pronounced in the estimation of high priority values than it is in the estimation of the low priority ones. Tables 39-40 and 41-42 show respectively the impact of α in our results for high (c = 0.9) and low (c = 0.1) values of coefficient of variation. The same patterns observed for the impact of c on our results are observed for α, as well.

To make sure that the interfering traffic parameters do not impact the obtained results, we vary ρ (Tables 43 and 44), c (Tables 45 and 46) and α (Tables 47 and 48). As the interfering utilization increases, so does the fraction of the loss rate due to the interfering process. Consequently the overall error estimation at the second queue decreases. From Table 43 to Table 46, we can conclude that the interfering coefficient of variance and correlation coefficient do not impact the accuracy of the matching procedure.

input (ρ, c, p_{high})	interf (ρ, c, p_{high})	Analytical	simulation	conf interval	error
0.85, 0.95, 0.9	0.5, 0.1, 0.7	0.1262	0.1205	4.63e-3	4.7
0.85, 0.9, 0.8	0.5, 0.1, 0.5	2.4049e-2	2.2309e-2	7.72e-4	7.8
0.9, 0.95, 0.8	0.3, 0.1, 0.5	2.2711e-3	2.0912e-3	2.54e-5	8.6
0.85, 0.95, 0.8	0.2, 0.1, 0.5	1.8265e-4	1.6742e-4	2.01e-6	9.1
0.8, 0.9, 0.8	0.3, 0.1, 0.5	4.2467e-5	3.8019e-5	8.29e-7	11.7
0.8, 0.9, 0.8	0.2, 0.1, 0.5	3.9843e-6	3.5197e-6	4.29e-8	13.2
0.75, 0.9, 0.8	0.2, 0.1, 0.5	2.3188e-7	2.0287e-7	3.73e-9	14.3

Table 33: High priority loss rate for input and interfering $\alpha = 0.9$

input (ρ, c, p_{high})	Analytical	simulation	conf interval	error
(0.85, 0.95, 0.8)	0.2167	0.2066	3.82e-3	4.9
(0.75, 0.9, 0.8)	2.0134e-2	1.9048e-2	5.38e-4	5.7
(0.7, 0.9, 0.8)	1.9641e-3	1.8617e-3	1.17e-5	5.5
(0.65, 0.9, 0.8)	1.8444e-4	1.7189e-4	3.85e-6	7.3
(0.625, 0.9, 0.8)	3.2688e-5	3.0211e-5	1.62e-7	8.2
(0.6, 0.9, 0.8)	1.9371e-6	1.7722e-6	5.72e-8	9.3
(0.5875, 0.9, 0.8)	4.9388e-7	4.5062e-7	1.72e-9	9.6

Table 34: Low priority loss rate for input $\alpha = 0.9$ and interfering ($\rho = 0.2$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.5$)

input (c, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	3.5641e-6	3.1569e-6	5.83e-8	12.9
(0.1, 0.7)	8.1167e-4	7.4534e-4	1.16e-6	8.9
(0.1, 0.8)	1.8963e-2	1.7542e-2	2.27e-4	8.1
(0.1, 0.9)	7.3916e-2	6.9996e-2	5.83e-4	5.6
(0.3, 0.6)	7.8701e-6	6.9894e-6	8.23e-8	12.6
(0.3, 0.7)	1.3834e-3	1.2727e-3	3.71e-5	8.7
(0.3, 0.8)	2.2685e-2	2.0985e-2	5.27e-4	8.1
(0.3, 0.9)	7.6788e-2	7.2785e-2	2.82e-4	5.5
(0.5, 0.6)	1.4371e-5	1.2865e-5	1.52e-7	11.7
(0.5, 0.7)	2.5103e-3	2.3179e-3	3.92e-5	8.3
(0.5, 0.8)	2.8961e-2	2.6865e-2	3.81e-4	7.8
(0.5, 0.9)	8.2004e-2	7.7951e-2	2.27e-4	5.2
(0.7, 0.6)	2.5435e-5	2.2791e-5	1.84e-7	11.6
(0.7, 0.7)	4.1678e-3	3.8591e-3	6.49e-5	8.0
(0.7, 0.8)	3.5801e-2	3.3272e-2	2.74e-4	7.6
(0.7, 0.9)	8.8421e-2	8.4291e-2	9.28e-4	4.9
(0.9, 0.6)	4.4318e-5	3.9819e-5	2.83e-7	11.3
(0.9, 0.8)	4.2529e-2	3.9599e-2	2.73e-4	7.4
(0.9, 0.9)	4.4613e-2	9.0842e-2	2.28e-4	4.9

Table 35: High priority loss rate as a function of c, input ($\rho = 0.8$, $\alpha = 0.9$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.7$)

input (c, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	0.6651	0.63826	3.28e-3	4.2
(0.1, 0.7)	0.7987	0.76724	5.38e-3	4.1
(0.1, 0.8)	0.9156	0.88216	6.94e-3	3.8
(0.1, 0.9)	0.9418	0.91262	4.78e-3	3.2
(0.3, 0.6)	0.6644	0.63828	2.18e-3	4.1
(0.3, 0.7)	0.7987	0.76733	9.69e-3	4.1
(0.3, 0.8)	0.9483	0.91274	1.82e-3	3.9
(0.3, 0.9)	0.9411	0.91366	2.93e-3	3.0
(0.5, 0.6)	0.6655	0.63874	3.93e-3	4.2
(0.5, 0.7)	0.7968	0.76613	1.82e-3	4.0
(0.5, 0.8)	0.8734	0.84144	5.73e-3	3.8
(0.5, 0.9)	0.9708	0.94257	8.64e-3	3.0
(0.7, 0.6)	0.6636	0.63815	5.93e-3	4.0
(0.7, 0.7)	0.7945	0.76437	9.58e-3	4.0
(0.7, 0.8)	0.9344	0.90111	3.82.e-3	3.7
(0.7, 0.9)	0.9992	0.97969	8.49e-3	2.0
(0.9, 0.6)	0.6643	0.63885	4.82e-3	4.0
(0.9, 0.8)	0.9451	0.91142	6.82e-3	3.7
(0.9, 0.9)	0.9994	0.97982	1.82e-3	2.0

Table 36: Low priority loss rate as a function of c, input ($\rho = 0.8$, $\alpha = 0.9$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.1$, $p_{high} = 0.7$)

input (c, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	2.6463e-6	2.3377e-6	1.72e-8	13.2
(0.1, 0.7)	7.5036e-4	6.8777e-4	6.94e-6	9.1
(0.1, 0.8)	1.8502e-2	1.7037e-2	3.94e-4	8.6
(0.1, 0.9)	7.3876e-2	6.9629e-2	5.92e-4	6.1
(0.3, 0.6)	3.6455e-6	3.229e-06	5.92e-8	12.9
(0.3, 0.7)	7.6566e-4	7.0309e-4	1.39e-6	8.9
(0.3, 0.8)	1.8572e-2	1.7149e-2	4.97e-4	8.3
(0.3, 0.9)	7.3783e-2	6.9738e-2	5.98e-4	5.8
(0.5, 0.6)	1.4082e-5	1.2528e-5	2.83e-7	12.4
(0.5, 0.7)	2.4586e-3	2.2618e-3	4.83e-5	8.7
(0.5, 0.8)	2.8795e-2	2.6662e-2	5.98e-4	8.0
(0.5, 0.9)	8.2285e-2	7.7848e-2	2.16e-4	5.7
(0.7, 0.6)	2.1634e-5	1.9351e-5	5.98e-7	11.8
(0.7, 0.7)	4.1514e-3	3.8297e-3	4.85e-5	8.4
(0.7, 0.8)	3.5705e-2	3.3091e-2	3.92e-4	7.9
(0.7, 0.9)	8.8682e-2	8.4218e-2	1.92e-4	5.3
(0.9, 0.6)	4.2746e-5	3.8303e-5	5.93e-7	11.6
(0.9, 0.8)	4.2461e-2	3.9462e-2	5.98e-4	7.6
(0.9, 0.9)	9.5412e-2	9.0696e-2	3.98e-4	5.2

Table 37: High priority loss rate as function of c, input ($\rho = 0.8$, $\alpha = 0.1$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.1$, $p_{high} = 0.7$)

input (c, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	0.6657	0.6377	3.94e-3	4.4
(0.1, 0.7)	0.7928	0.7608	1.92e-3	4.2
(0.1, 0.8)	0.8745	0.8409	1.93e-3	4.0
(0.1, 0.9)	0.9103	0.8812	5.93e-3	3.3
(0.3, 0.6)	0.6655	0.6381	4.93e-3	4.3
(0.3, 0.7)	0.7963	0.7642	6.94e-3	4.2
(0.3, 0.8)	0.8976	0.8623	2.93e-3	4.1
(0.3, 0.9)	0.9396	0.9123	5.93e-3	3.0
(0.5, 0.6)	0.6652	0.6384	2.93e-3	4.2
(0.5, 0.7)	0.7911	0.7606	1.93e-3	4.0
(0.5, 0.8)	0.9159	0.8815	6.98e-3	3.9
(0.5, 0.9)	0.9715	0.9423	8.52e-3	3.1
(0.7, 0.6)	0.6641	0.6386	2.84e-3	4.0
(0.7, 0.7)	0.7994	0.7672	9.48e-3	4.2
(0.7, 0.8)	0.8951	0.8623	2.74e-3	3.8
(0.7, 0.9)	0.9951	0.9661	9.48e-3	3.0
(0.9, 0.6)	0.6644	0.6389	2.83e-3	4.0
(0.9, 0.8)	0.9156	0.8829	4.81e-3	3.7
(0.9, 0.9)	0.9978	0.9783	4.95e-3	2.0

Table 38: Low priority loss rate as a function of c, input ($\rho = 0.8$, $\alpha = 0.1$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.7$)

input (α, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	2.6463e-6	2.3377e-6	5.93e-8	13.2
(0.1, 0.7)	7.5036e-4	6.8777e-4	2.73e-6	9.1
(0.1, 0.8)	1.8502e-2	1.7037e-2	1.72e-4	8.6
(0.1, 0.9)	7.3876e-2	6.9629e-2	6.94e-4	6.1
(0.3, 0.6)	4.0338e-6	3.5697e-6	2.73e-8	13.0
(0.3, 0.7)	7.6993e-4	7.0571e-4	7.89e-6	9.1
(0.3, 0.8)	1.7728e-2	1.6339e-2	4.62e-4	8.5
(0.3, 0.9)	7.192e-2	6.7912e-2	1.27e-4	5.9
(0.7, 0.6)	2.5283e-6	2.2374e-6	7.49e-8	13.0
(0.7, 0.7)	7.6483e-4	7.0168e-4	4.83e-6	9.0
(0.7, 0.8)	1.7762e-2	1.6416e-2	2.83e-4	8.2
(0.7, 0.9)	7.2385e-2	6.8482e-2	8.75e-4	5.7
(0.9, 0.6)	3.5641e-6	3.1569e-6	3.62e-8	12.9
(0.9, 0.7)	8.1167e-4	7.4534e-4	6.94e-6	8.9
(0.9, 0.8)	1.8963e-2	1.7542e-2	3.92e-4	8.1
(0.9, 0.9)	7.3915e-2	6.9996e-2	5.84e-4	5.6

Table 39: High priority loss rate as a function of α input ($\rho = 0.8$, $c = 0.1$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.7$)

input (α, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	0.6658	0.6377	6.94e-3	4.4
(0.1, 0.7)	0.7927	0.7608	4.42e-3	4.2
(0.1, 0.8)	0.8745	0.8409	2.29e-3	4.0
(0.1, 0.9)	0.9103	0.8812	4.29e-3	3.3
(0.3, 0.6)	0.6614	0.6335	1.19e-3	4.4
(0.3, 0.7)	0.7952	0.7624	9.83e-3	4.3
(0.3, 0.8)	0.9428	0.9074	6.93e-3	3.9
(0.3, 0.9)	0.9964	0.9769	8.53e-3	2.0
(0.7, 0.6)	0.6669	0.6395	2.93e-3	4.3
(0.7, 0.7)	0.7967	0.7646	5.93e-3	4.2
(0.7, 0.8)	0.9432	0.9078	9.84e-3	3.9
(0.7, 0.9)	0.9973	0.9778	4.72e-3	2.0
(0.9, 0.6)	0.6651	0.6382	1.91e-3	4.2
(0.9, 0.7)	0.7986	0.7672	3.92e-3	4.1
(0.9, 0.8)	0.9156	0.8821	8.74e-3	3.8
(0.9, 0.9)	0.9418	0.9126	2.93e-3	3.2

Table 40: Low priority loss rate as a function of α input ($\rho = 0.8$, $c = 0.1$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.7$)

input (α, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	4.2516e-5	3.8303e-5	4.92e-7	11.6
(0.1, 0.8)	4.2461e-2	3.9462e-2	2.92e-4	7.6
(0.1, 0.9)	9.5412e-2	9.0696e-2	8.39e-4	5.2
(0.3, 0.6)	5.1247e-6	4.5962e-6	1.62e-8	11.5
(0.3, 0.7)	1.0757e-3	9.8965e-4	4.09e-6	8.7
(0.3, 0.8)	2.0488e-2	1.9041e-2	2.83e-4	7.6
(0.3, 0.9)	7.4783e-2	7.1087e-2	5.84e-4	5.2
(0.7, 0.6)	1.0911e-5	9.803e-6	9.27e-8	11.3
(0.7, 0.7)	2.7705e-3	2.5535e-3	5.92e-5	8.5
(0.7, 0.8)	2.7678e-2	2.5771e-2	1.71e-4	7.4
(0.7, 0.9)	2.9062e-2	7.6623e-2	8.72e-4	5.0
(0.9, 0.6)	4.4318e-5	3.9819e-5	3.93e-7	11.3
(0.9, 0.8)	4.2529e-2	3.9599e-2	2.81e-4	7.4
(0.9, 0.9)	9.5293e-2	9.0842e-2	1.01e-4	4.9

Table 41: High priority loss rate as a function of α input ($\rho = 0.8$, $c = 0.9$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.7$)

input (α, p_{high})	Analytical	Simulation	Conf Interval	Error
(0.1, 0.6)	0.6644	0.6389	6.93e-3	4.0
(0.1, 0.8)	0.9156	0.8829	2.83e-3	3.7
(0.1, 0.9)	0.9988	0.9783	5.97e-3	2.1
(0.3, 0.6)	0.6569	0.6311	4.93e-3	4.1
(0.3, 0.7)	0.7974	0.7682	4.02e-3	3.8
(0.3, 0.8)	0.91784	0.8851	2.93e-3	3.7
(0.3, 0.9)	0.9739	0.9465	1.02e-3	2.9
(0.7, 0.6)	0.6651	0.6389	3.98e-3	4.1
(0.7, 0.7)	0.8101	0.7805	2.54e-3	3.8
(0.7, 0.8)	0.9407	0.9071	1.82e-3	3.7
(0.7, 0.9)	0.9955	0.9731	1.64e-3	2.3
(0.9, 0.6)	0.6643	0.6388	2.73e-3	4.0
(0.9, 0.8)	0.9451	0.9114	4.92e-3	3.7
(0.9, 0.9)	0.9964	0.9798	1.87e-3	1.7

Table 42: Low priority loss rate as a function of α input ($\rho = 0.8$, $c = 0.9$) interfering ($\rho = 0.5$, $c = 0.1$, $\alpha = 0.9$, $p_{high} = 0.7$)

interfering ρ	Analytical	Simulation	Conf Interval	Error
0.6	2.6072e-4	2.3919e-4	4.92e-6	9.0
0.7	6.6228e-3	6.1379e-3	1.72e-5	7.9
0.8	4.4951e-2	4.2567e-2	4.91e-4	5.6
0.9	0.1041	9.9434e-2	2.89e-4	4.7

Table 43: High priority loss rate as a function of the interfering ρ for input ($\rho = 0.8$, $c = 0.9$, $\alpha = 0.1$, $p_{high} = 0.6$) and interfering ($c = 0.1$, $\alpha = 0.1$, $p_{high} = 0.7$)

interfering ρ	Analytical	Simulation	Conf Interval	Error
0.6	0.8318	0.7998	5.93e-3	4.0
0.7	0.9566	0.9324	3.32e-3	2.6
0.8	0.9975	0.9925	9.76e-3	0.5
0.9	0.9999	0.9997	3.64e-3	0.02

Table 44: Low priority loss rate as a function of the interfering ρ for input ($\rho = 0.8$, $c = 0.9$, $\alpha = 0.1$, $p_{high} = 0.6$) and interfering ($c = 0.1$, $\alpha = 0.1$, $p_{high} = 0.7$)

interfering c	Analytical	Simulation	Conf Interval	Error
0.1	4.4438e-5	3.9819e-5	3.94e-7	11.6
0.3	8.4174e-5	7.5901e-5	2.81e-7	10.9
0.5	1.0897e-4	9.8798e-5	9.53e-7	10.3
0.7	1.3107e-4	1.1981e-4	4.73e-6	9.4
0.9	3.7411e-4	3.4322e-4	2.21e-6	9.0

Table 45: Impact of interfering c on the high priority loss rate for input ($\rho = 0.8$, $c = 0.9$,$\alpha = 0.1$,$p_{high} = 0.6$) and interfering ($c = 0.1$,$\alpha = 0.1$, $p_{high} = 0.7$)

interfering c	Analytical	Simulation	Conf Interval	Error
0.3	0.6618	0.6382	4.54e-3	3.7
0.5	0.6621	0.6379	5.38e-3	3.8
0.7	0.6601	0.6371	9.85e-3	3.6
0.9	0.6592	0.6363	6.59e-3	3.6

Table 46: Impact of interfering c on the low priority loss rate for input ($\rho = 0.8$, $c = 0.9$, $\alpha = 0.1$,$p_{high} = 0.6$) and interfering ($c = 0.1$,$\alpha = 0.1$,$p_{high} = 0.7$)

interfering α	Analytical	Simulation	Conf Interval	Error
0.1	2.6787e-5	2.4089e-5	3.87e-7	11.2
0.3	3.8743e-5	3.4747e-5	7.94e-7	11.5
0.7	4.0036e-5	3.6134e-5	4.86e-7	10.8
0.9	4.4079e-5	3.9819e-5	2.85e-7	10.7

Table 47: Impact of interfering α on the high priority loss rate for input ($\rho = 0.8$, $c = 0.9$, $\alpha = 0.1$,$p_{high} = 0.6$) and interfering ($\rho = 0.5$,$c = 0.1$,$p_{high} = 0.7$)

interfering α	Analytical	Simulation	Conf Interval	Error
0.1	0.6478	0.6241	3.93e-3	3.8
0.3	0.6501	0.6269	5.94e-3	3.7
0.7	0.6623	0.6387	7.84e-3	3.7
0.9	0.6631	0.6388	1.52e-3	3.8

Table 48: Impact of interfering α on the low priority loss rate for input ($\rho = 0.8$, $c = 0.9$, $\alpha = 0.1$,$p_{high} = 0.6$) and interfering ($c = 0.1$,$\alpha = 0.1$,$p_{high} = 0.7$)

The priority computational procedure can be extended to a queue with multiclass selective discard mechanism, i.e., a D-BMAP$^{[c_1, ..., c_N]}$/D/1/K queue. In this case, the probability of a cell's priority being equals to j is given by:

$$\Pi_j = \frac{p_j \times (1 - R_j)}{\sum_{k=1}^{n} p_k \times (1 - R_k)}$$

where p_k and R_k are respectively class k priority probability and loss rate.

5.2) Splitting

When characterizing the flow between two nodes, we represent the output process of the first queue as an MMBP$^{[H,L]}$ process, and then we model the flow that goes to the second queue as an MMBP$^{[H,L]}$ with the same p_{high} and p_{low} and parameters ($p_{ij} \times p_1$, $p_{ij} \times p_2$, α_1, α_2) where p_{ij} is the probability that a cell leaves node i and goes to node j

5.3) Joining

For the joining operator, we first compute the D_n matrices of the aggregated process and then we compute the vector $\vec{p}_{high}$. To compute the aggregate process priority probability, we need to take into consideration not only the priority probability of each aggregating process but also their probability of arrivals. Thus, the i^{th} component of process c $\vec{p}_{high}$ which is the result of the aggregation of processes a and b is given by:

$$p_{high}^{(C)}(i_c) = \sum_{j_c=1}^{M_c} \sum_{n_c=1}^{N_a+N_b} \sum_{n_a=\min(0,\, n_c-N_b)}^{\min(n_c,\, N_a)} H_a \times H_b \times \frac{z_a+z_b}{n_a+n_b} \times \left(d_{i_c j_c}\right)_{n_a,\, n_b}$$

$$H_a = \sum_{z_a=0}^{n_a} \binom{z_a}{n_a} \times p_{high}^{z_a}(i_a) \times p_{low}^{n_a-z_a}(i_a)$$

$$H_b = \sum_{z_b=0}^{n_b} \binom{z_b}{n_b} \times p_{high}^{z_b}(i_b) \times p_{low}^{n_b-z_b}(i_b)$$

where:

$n_c = n_a + n_b$

$p_{low}(i_a) = 1 - p_{high}(i_a)$

$p_{low}(i_b) = 1 - p_{high}(i_b)$

i_a and i_b are respectively the states of process A and B which correspond to state i_c of the aggregate process C

$\left(d_{i_c j_c}^{(C)}\right)_{n_a,\, n_b}$ is the element in the i_c^{th} row and the j_c^{th} column of

$D_{n_a}^{(a)} \otimes D_{n_b}^{(b)}$

Chapter 6

Conclusions

6.1) Summary of the Contributions

In this dissertation, we investigated queueing network models for multiple class B-ISDN networks. More precisely, we studied in depth a multiple class selective discard mechanism.

The future B-ISDN network will carry video, voice and data applications with diverse quality of service requirements. In Chapter 3, we showed that the introduction of a multiple class selective discard mechanism at the cell level of the network protocol hierarchy provides flexibility for coping with diverse loss QOS without excessive complexity. To draw this conclusion, we proved a loss rate conservation law and introduced an algorithm for solving a complete sharing queue with multiple levels of buffer priority. Additionally, we solve an MMPP1....MMPPN/D/1/K and compared the impact of adopting different push-out policies in a selective discard mechanism.

Being able to estimate end-to-end performance is of paramount importance not only for controlling traffic, but also for dimensioning the future B-ISDN networks. Along with that, we introduced a framework for queueing networks with Markov Modulated flow in Chapter 4. We demonstrated how to model the output, the splitting and the joining operators. Our analysis was approximate, given the non-product form characteristics of the B-ISDN net-

works. Our performance estimation was reasonably accurate for both delay and loss rate. We also illustrated the queueing network framework in the analysis of tandem and feed-forward networks. In Chapter 5, we extended the framework for prioritized flow. We took into account only buffer priority. We showed the output, the splitting and the joining operators in this type of networks. This queueing network framework can be used for the analysis of traffic control mechanisms such as multiple class selective discard.

6.2) Future Research

The original contributions of this dissertation generate topics for several investigations. Among them are:

• The analysis of multiple class selective discard mechanisms with other types of buffer management - in chapter 3, we concentrated on the complete sharing buffer management policy. It would be interesting to develop solutions for other policies, as well as to examining the extent to which the advantages of a multiple class mechanism are preserved;

• The evaluation of the relationship between service and buffer priority mechanisms - the order which cells are served impact their waiting time, and consequently their likelihood of being lost. Assessing the relationship between service and buffer priority would help the creation of strategies for copying with diverse loss and delay QOS requirements;

• The definition of policies for the guarantee of minimum loss performance requirement - the assignment of a fixed priority level to a cell may lead to situations where the QOS requirements of a low priority class is not satisfied. Increasing the loss probability of high priority classes can change the loss

scenario in a way to satisfy the overall loss requirements. Defining dynamic priority assignment policies targeted at achieving the overall QOS requirements by guaranteeing minimum loss performance for the low priority classes is undoubtedly a research problem worth investigating;

• The evaluation of the importance of destination correlation when splitting a traffic stream - after a customer receives service, a decision has to be made to choose which queue it will visit next. In our analysis, we assume that a cell goes to a certain destination based on a fixed random value. If there is a likelihood that consecutive departing cells go to the same destination, then this likelihood has to be taken into account when modelling the traffic stream. Finding which traffic mix scenario leads to such likelihood is still an opening question in the literature;

• The definition of procedures for reducing the state space of the underlying Markov chain of Markov modulated processes - the exponential state space growth of Markov processes resulting from the joining operation restricts the use of the framework developed in this dissertation to networks in which the nodes have a low degree of connectivity. Therefore, we need to define procedures for reducing the state space of the underlying Markov chain of a Markov modulated process. An alternate approach would be to develop solutions of systems with Markov modulated input which are independent of the state space size;

• The definition of priotized Markov modulated processes with correlation between the arrivals' priorities - to evaluate networks with priotized flow, we create a batch arrival process which cells are independently classified in one of the priority levels. Future efforts should be concentrated on generalizing this process by allowing a correlated criteria for cell classification;

• The conception of queueing network frameworks which allows the rep-

resentation of individual connections - in our queueing network framework, we assume an aggregated traffic model. Although we are able to estimate some performance which the aggregated value and the per connection value are the same, we are not able to estimate other per connection performances such as average burst loss size. Therefore, research is needed to create network models which allow the prediction of any per connection performance

- The investigation of the selective discard benefits in a network wide context - Although it has been shown that selective discard is an efficient mechanism when considering an isolated multiplexer, no studies were carried out attempting to generalize the conclusion to a network wide context. Using the framework developed in Chapter 5 to access the advantages of selective discard in a network context is a task of paramount importance which should be addressed by future research.

References

[1]L. Kleinrock, "ISDN- the path to broadband networks", *Proc. of IEEE*, vol. 79, pp. 112-117, Feb. 1991.

[2]V.O. K. Li, J. F. Chang, K. C. Lee and T. S. Yang, "A survey of research and standards in high speed networks", *Int. J. Digital Analog Commun. Sys.*, vol. 4, pp. 269-309, 1991.

[3]CCITT Rec. I311. recommendation on broadband aspect of ISDN, 1990.

[4]*IEEE J. Select. Areas Commun.*, Special issue in congestion control in high speed packet switched networks, vol. 9, Sep. 1991.

[5]*IEEE Network*, Special issue on congestion control for B-ISDN networks, Sep. 1992.

[6]J. J. Bae and T. Suda, "Survey of traffic control schemes and protocols in ATM networks", *Proc. of IEEE*, pp. 170-189, Feb. 1991.

[7]J. Y. Hui, "Resource allocation for broadband networks", *IEEE J. Select. Areas Commun.*, vol. 6, pp. 1598-1608, Dec. 1988.

[8]H. Saito and K. Shiomoto, "Dynamic call admission control in ATM networks", *IEEE J. Select. Areas Commun.*, vol 9, pp. 982-989, Sep. 1991.

[9]G. Gallassi, G. Riogolio and L. Fratta, "Bandwidth assignement in prioritized ATM Networks", in *Proc. of IEEE GLOBECOM'90*, pp. 852-856.

[10]L. Dittmann, S. B. Jacobsen and K. Moth,"Flow enforcement algorithm for ATM Networks", *IEEE J. Select. Areas Commun.*, vol 9, n 3, April 1991.

[11]E. P. Rathgeb, "Modeling and performance of policing mechanisms for ATM networks:, *IEEE J. Select. Areas Commun.*, vol 9, pp. 325-334, Apr. 1991.

[12]L. P. Clare and I Rubin, "On the design of prioritized multiplexing systems", in *Proc. IEEE ICC'83*, pp E5.3.1-E5.3.5.

[13]L. P. Clare and I. Rubin. "Preemptive buffering disciplines for time-critical sensor communications", in *Proc. IEEE ICC'8*6, pp. 904-909, 1986.

[14]N. L. S. Fonseca and J. A. Silvester, "A Multiple Class Buffer Priority Algorithm for the Design of B-ISDN networks, First International Conference of Computer and Communications and Networks, pp. 38-42, 1992.

[15]N. L. S. Fonseca and J. A. Silvester, "Estimating the loss probability in a multiplexer loaded with multi-priority MMPP streams", in *Proc. IEEE ICC*, pp. 1037-1041,1993.

[16]N. L. S. Fonseca and J. A. Silvester, "A comparison of push-out policies in an ATM multiplexer", in *Proc. of IEEE Pac. Rim Conf. on Commun. Comp. and Signal Proc.*, pp. 338-341, 1993.

[17]G. M. Woodruff and R. Kositpaiboon", "Multimedia traffic management principles for guaranteed ATM network performance", *IEEE J. Select. Areas Commun.*, vol 8, pp. 437-46, Apr. 1990.

[18]R. Nagarajan, J. Kurose and D. Towesley,"Quality-of-service issues in high speed networks", submitted to IEEE ICC'93.

[19]H. Heffes, "A class of data traffic processes - covariance function characterization and related queueing results", *The Bell Sys. Tech. J.*, n 6, Jul/Aug, 1980.

[20]C Yuan and J. A. Silvester, "Queueing analysis of delay constrained voice traffic in a packect switching system", *IEEE J. Select. Areas Commun.*, vol. 7, pp. 729-738, Jun1989.

[21]K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexer for voice and data", *IEEE J. Select. Areas Commun.*, vol. 4, pp. 833-846, Sep. 1986.

[22]J. N. Daigle and J. D. Langford, "Models for analysis of packet voice communications systems", *IEEE J. Select. Areas Commun.*, vol. 4, pp. 847-855, Sep. 1986.

[23]D. Anick, D. Mitra and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources", *The Bell Sys. Tech J.*, pp. 1871-1894, 1982.

[24]A. I. Elwalid, D. Mitra and T. E. Stern, "Statistical multiplexing of Markov modulated sources: Theory and computataional algorithms", in *Proc. of 13th Int. Teletraffic Conf.*, pp. 495-500, 1991.

[25]R. C. F. Tucker, "Accurate method for analysis of a packet-speech multi-

plexer with limited delay", *IEEE Trans. Commun.*, vol 36, pp. 479-483, Apr. 1988.

[26]I. Norros, J. W. Roberts, A. Simoniam and J. T. Virtamo, "The superposition of variable bit rate sources in an ATM multiplexer", *IEEE J. Select. Areas Commun.*, vol. 9, n 3, pp. 378-387, Apr. 1991.

[27]S.Q. Li, "A general technique for discrete queueing analysis of multimedia traffic on ATM", *IEEE Trans. Commun.*, vol. 39, pp. 1115-1132, Jul.1991.

[28]M.F. Neuts, Matrix-geometric solutions in stochastic models: An algorithmic approach, John Hopkins University Press, Baltimore, MD, 1981.

[29]D. M. Lucantoni, "New results on the single server queue with a batch markovian arrival process", *Stochastic Models*, vol.7, no. 1, pp. 1-46, 1991.

[30]H. Heffes and D. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance", *IEEE J. Select. Areas Commun.*, vol. 4, pp. 856-868, Sep. 19.86

[31]A. Baiocchi, N.B. Melazzi, M. Listani, A. Roveri and R. Winkler, "Loss performance analysis of an ATM multiplexer loaded with high speed on-off processes", *IEEE J. Select. Areas Commun.*, vol. 9, pp. 388-393, Apr. 1991.

[32]S. Wang and J. A. Silvester, "A discrete-time performance model for integrated service ATM multiplexers", in *Proc. of IEEE GLOBECOM'93*, pp.757-761, 1993.

[33]R. Narajan, J.F. Kurose and D. Towsley,"Approximation techniques for computing packet loss in finite-buffered voice multiplexer", *IEEE J. Select. Areas Commun.* , vol. 9, pp. 368-377, Apr. 1991.

[34]C. Blondia and O. Casals, "Performance analysis of a Statistical multiplexing of VBR sources", *Proc of IEEE INFOCOM*, pp 828-838, 1992.

[35]N.L.S. Fonseca and J. A. Silvester, "Modelling the output process of an ATM multiplexer with Markov modulated arrivals", in *Proc of IEEE ICC'94*, pp. 721-725.

[36]P. T. Brady, "A statistical analysis of on-off pattern in 16 conversations", *The Bell Sys. tech J.*, pp 73-91, jan 1968.

[37]B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson and J. D. Robins, "Performance models of statistical multiplexing in packet video communications", *IEEE Trans. Commun.*, vol 36, pp. 834-844, July, 1988.

[38]P. Sen, B. Maglaris, N. E. Rikli and D. Anastassiou, "Models for packet switching of variable-bit-rate video sources", *IEEE J. Select. Areas Commun.*, vol. 7, pp. 865-869, Jun. 1989.

[39]P. Pancha and M. El Zarki, "A look at the MPEG video coding standard for variable bit rate video transmission", in *Proc of IEEE INFOCOM'92*, pp. 85-94.

[40]D. Le Gall, "MPEG: a video compress for multimedia applications", *Commun. ACM*, vol. 34, pp. 46-58, Apr. 1991.

[41]R. Grunenfelder, J. P. Cosmas, S. Manthorpe and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance", *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 284-295, Apr. 19.91

[42]B. Melamed and B. SEngupta, "TES Modeling of video traffic", IEICE Trans. on Commun., vol E75-B, no 12, pp. 1292-1300, 1992.

[43]P. Skelly, S. Dixit and M. Schwartz, "A histogram based model for video traffic behavior in an ATM network node with an application to congestion control", in *Proc of IEEE INFOCOM'92.*

[44]Blondia Belgian C. Blondia, "A discrete-time batch Markovian arrival process as B-ISDN traffic model", *Belgian J. of Oper Res., Stat. and Comp. Science*, vol. 32 (3), pp. 3-23, 1992.

[45]S. M. Ross, Introduction to probability models, Academic Press Inc, San Diego, 1989.

[46]R Bellman, Introduction to matrix analysis, McGraw Hill, New York, 1960.

[47]M. F. Neuts and G. Latouche,"The superposition of two PH-renewal processes. In "Semi-Markov Models: Theory and Applications", J. Janssen, ed., London: Plenum Publishers, pp. 131-177, 1986.

[48]F. Kamoun and L. Kleinrock, "Analysis of shared finite storage in a computer network node environment under general traffic conditions", *IEEE Trans. Commun.*, vol. 28, pp. 992-1003, 1980.

[49]P Landsberg and C Zukowski, "A novel buffer sharing method: complete sharing subject to guaranteed queue minimum", in *Proc. of The First Conf. Comp. Commun. Net.*, pp. 43-48, 1992.

[50]L. P. Clare and I Rubin, "Performance boundaries for prioritized multiplexing systems", IEEE trans. on Info. Theory, n0 3, pp. 329-340, May 1987.I

[51]I. Rubin and M. Quaily, "Performance of finite capacity communication and queueing systems under various service and buffer preemptive policies", in *Proc. INFOCOM'88*, pp. 505-514.

[52]G. Hebuterne and A Gravey., "A space priority queueing mechanism for multiplexing ATM channels", in *Proc. ITC Spec. Sem.'89*, paper no. 7.4.

[53]H. Kroner, "Comparative performance study of space priority mechanisms for ATM networks", in *Proc. IEEE INFOCOM'90*, pp.1136-1143.

[54]J. Y. Le Boudec, "An efficient solution method for Markov models of ATM links with loss priorities", *IEEE J. Select. Areas Commun.*, vol. 9, pp. 408-417, Apr. 1991.

[55]A. Y-M Lin and J. A. Silvester, "Priority queueing strategies and buffer allocation protocols for traffic control at an ATM integarted broadband switching system, *IEEE J. Select. Areas Commun.* , vol. 9 , pp. 1524-1536 , Dec. 1991.

[56]S. Sumita and T. Ozawa: Achievability of performance objectives in ATM switching nodes, in *Proc. of the Int. Sem. Perfor. of Dist. Paral. Sys.*, pp. 45-56, Dec. 1988.

[57]S. Q. Li, "Overload control in a finite message storage buffer", *IEEE Trans. Commun.*, vol. COM-37, pp. 1330-1338, Dec. 1988.

[58]N. Yin, S Q. Li and T. E. Stern, "Congestion control for packet voice by selective packet discarding", *IEEE Trans. Commun.*, vol. COM-38, pp. 674-683, May 1990.

[59]D. Peter and V. S. Frost, "Nested threshold cell discarding for ATM overload control: optimization under cell loss constraints", in *Proc of INFOCOM'91*, pp 1403-1412.

[60]M Karol, M. G. Hiluchyj and S. P. Morgan, "Input versus output queueing on a space-division packet switch", *IEEE Trans. Commun.*, vol. COM-35, Dec. 1987.

[61]P. Ferrandiz and A. Lazar, "Modeling and analysis of real-time packet traffic", in *Proc. of Fourth Int. Conf. on Data Commun.* Sys., pp. 306-324, 1990.

[62]H. Schulzrinne and J. F. Kurose, "Distribution of the loss period for some queues in continuous and discrete time", In *Proc. of INFOCOM'91,* pp. 1446-1455.

[63]Y. H. Jeon amd I. Viniotis, "Achievable loss probabilities and buffer allocation policies in ATM nodes with correlated arrivals", in *Proc of ICC'93*, pp. 365-369.

[64]J. Garcia and Olga Casals, "Stochastic models of space priority mechanisms with Markovian arrival processes", *Annals of Operation Research* 35, pp. 271-296, 1992.

[65]J. J. Bae, T. Suda and R. Simha, "Analysis of individual packet loss in a finite buffer queue with heterogeneous Markov modulated arrival processes: a study of traffic burstiness and priority packet discarding", in *Proc INFOCOM'92*, pp. 0219-0230.

[66]A. K. Erlang, "Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges, *Post Office Electical Engineer's Journal*, 10:189-197, 1917.

[67]L. Kleinrock, Communication nets: stochastic message flow and delays, Mc Graw-Hill, New York, 1964.

[68]S. S. Lam and J. Wong, "Queueing nerwok models of packet switching networks, Part 2: networks with population size constraints"N, *Perfor Eval.*, 2, pp 161-180, 1982.

[69]J. R. Jackson, "Networks of waiting lines", *Oper. Res.*, vol. 5, pp. 518-521, 1957.

[70]P. J. Burke, "The output of a queueing system", *Oper. Res.*, vol. 4, pp. 699-704.

[71]F. Baskett, K.M. Chandy, R.R. Muntz and F. Palacios, "Open, closed and mixed networks of queue with different classes of customers", *J. ACM*, pp. 248-260, 1975.

[72]F. P. Kelly, "Reversibility and stochastic networks", Wiley, New York, 1980.

[73]I. Rubin, "Communication networks: message path delays", *IEEE Trans. on Inform. Theory*, vol.20; pp. 738-745, 1974.

[74]E. de Souza e Silva and R.R. Muntz, "Queueing Networks: Solutions and Applications", in Stochastic Analysis of Computer and Communication Systems, H. Takagi editor, North Holland, 1990.

[75]S. C. Agrawal, Metamodeling: A Study of approximations in queueing models, The MIT Press, 1985.

[76]R. L. Disney and D. Konig, "Queueing networks: A survey of its random processes", *SIAM Rev.*, vol. 27, 3, pp. 335-403, 1985.

[77]R. L. Disney and P. C. Kiessler, Traffic processes in queueing networks: A markov renewal approach, The Johns Hopkins University Press, Baltimore, 1987.

[78]W. Whitt, "The Queueing network analyzer", *The Bell Sys. Tech.I* J., vol. 62, pp. 2779-2815, Nov. 1983.

[79]P. J. Kuehn, "Approximative analysis of general queueing networks by decomposition", *IEEE Trans. Commun.*, vol COM-27, 1, pp. 113-126, 1979.

[80]H. Kroner, M. Eberspacher, T. H. Theimer, P. J. Kuhn and U. Briem, "Approximate analysis of the end-to-end delay in ATM networks", in *Proc of IEEE INFOCOM'92*, pp. 879-985.

[81]J.A. Resing, "ATM cell stream through tandem queues", *Mini Symposium on Performance Aspects of ATM Networks*, Leidschendam, 1991.

[82]R. Grunenfelder, "A correlation based end-to-end cell queueing delay characterization in an ATM network", in *Proc. of 13th ITC*, vol on queueing and control in ATM, pp. 59-64, 1991.

[83]K. P. Jun and H. G. Perros, "Approximate analysis of arbritary configurations of queueing networks with blocking and deadlock", in *Proc. First International Workshop on Queueing Networks with Blocking*, Nort Holland, pp. 259-280, 1989.

[84]H. Saito, "The Departure process of an N/G/1 Queue", *Perfor. Eval.*, 11, pp. 241-251, 1990.

[85]H. Saito, M. Kawarasaki and H. Yamak, "An analysis of statistical multi-

plexing in an ATM transport network", *IEEE J. Select. Areas Commun.*, vol 9, n 3, pp 359-367, April 1991.

[86]F. Bonomi, S. Montagna and R. Paglino , "Busy period analysis for an ATM switching element output line", in *Proc. of IEEE INFOCOM*, pp. 544-550, 1992.

[87]T. Takine, T. Suda and T. Hasegawa, "Cell loss and output process analyses of finite buffer discrete time queueing system with correlated arrivals", in Proc. of *IEEE INFOCOM*, pp 1259-1268, 1993.

[88]D. Park, H. G. Perros and H. Yamashita, "Approximate analysis of discrete-time tandem queueing networks with bursty and correlated input traffic and customers", to appear in *Operation Research Letters*.

[89]I. Stavrakakis, "Efficient Modeling of merging and spliting processes in large networking structures:, *IEEE J. Select. Areas Commun.*, vol. 9, no 8, pp. 1336-1347, Oct 1991.

[90]N. L. S. Fonseca and J. A. Silvester, "On the computation of end-to-end delay in feed-forward ATM networks", *Proc of International Telecommunications Symposium'94*, pp. 460-464.

[91]J. A. Silvester, N. L. S. Fonseca and S. S. Wang, "D-BMAP models for the performance evaluation of ATM networks", in *Proc of 2nd IFIP Workshop on Performance and Modeling of ATM networks*, July 1994.