1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Acquisitions and Bibliographic Services

Direction des Acquisitions et des Services Bibliographiques

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Leaky Bucket Congestion Control in ATM Networks

by

Xiaoping Wu, B.Eng.

Beijing University of Posts and Telecommunications, China

**A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of
Master of Engineering**

**Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada K1S 5B6
March 20, 1993
© Xiaoping Wu, 1993**

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Vous l'avez vu dans votre bibliothèque

Vous l'avez vu dans votre bibliothèque

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-84126-5

Canada

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

"Leaky Bucket Congestion Control in ATM Networks"

submitted by Xiaoping Wu, B.Eng.
in partial fulfillment of the requirements for
the degree of Master of Engineering

Thesis Supervisor

Chair, Department of Systems and
Computer Engineering

Ottawa-Carleton Institute for Electrical Engineering.
Department of Systems and Computer Engineering
Faculty of Engineering, Carleton University
April 30, 1993

Abstract

In order to regulate the traffic flow and achieve the best use of ATM network resources, congestion control is implemented to prevent the network from performance degradation due to congestion. In this thesis, a leaky bucket approach is used for congestion control in ATM networks. The performance of the leaky bucket scheme is analyzed considering batch data input traffic and the mean, variance and autocovariance of the leaky bucket output process are obtained. A two-minisource model is proposed to characterize the leaky bucket output process and evaluate the network performance. The two-minisource model is also used to study cases with more general traffic arrivals. Extensive simulation studies of various leaky bucket control schemes are performed, including adaptive leaky bucket control, combined leaky bucket and marking scheme, and combined leaky bucket and priority mechanism. Network throughput and cell loss probabilities are evaluated and compared for these leaky bucket schemes.

Acknowledgments

I would like to thank my thesis supervisors, Prof. H. W. Lee and Prof. A. R. Kaye. Their continued guidance, encouragement and numerous suggestions help me to finish this thesis. Their wide knowledge, hard working attitude and generosity stimulated me to learn more and do more.

I would like to thank Prof. I. Lambadaris for reviewing this thesis and giving valuable comments. I would also like to thank the staff members and the graduate students in the Department of Systems and Computer Engineering, Carleton University, for their support.

In addition, I am grateful to the Telecommunication Research Institute of Ontario (TRIO), the Natural Sciences and Engineering Research Council (NSERC) and Carleton University for their generous financial support.

Finally, I would like to thank my husband, Yu-Dong Yao, for his continuous encouragement and support.

Contents

Abstract	iii
Acknowledgments	iv
List of Figures	viii
List of Tables	xiv
List of Acronyms	xv
List of Symbols	xvi
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Evolution of ISDN Towards Broadband ISDN	1
1.1.2 Asynchronous Transfer Mode	3
1.1.2.1 Basic Principles of ATM	3
1.1.2.2 ATM Architecture	4
1.2 Thesis Objectives	6
1.3 Thesis Organization	7
Chapter 2 Traffic Modelling and Congestion Control in ATM Networks	8
2.1 Traffic Modelling of Voice, Data and Video	8
2.1.1 Input Traffic Models for Data Sources	9
2.1.2 Input Traffic Models for Voice Sources	9
2.1.3 Input Traffic Models for Video Sources	9

2.2	Congestion Control Problem	10
2.3	Conventional Congestion Control versus Congestion Control for ATM	11
2.4	Call-Level Congestion Control	11
2.4.1	Admission Control	11
2.5	Cell-Level Congestion Control	12
2.5.1	Traffic Policing	12
2.5.2	Traffic Smoothing	13
2.5.3	Cell Prioritizing	13
2.6	Observations	14
Chapter 3	Performance Analysis of Leaky Bucket and a Two-Minisource Model Approach	15
3.1	Introduction	15
3.2	Performance Analysis of Leaky Bucket	17
3.2.1	Network and Leaky Bucket Model	17
3.2.2	Output Process of the Leaky Bucket	20
3.2.2.1	Cell Arrival Probabilities	24
3.2.2.2	Steady-State Probabilities for Finite User Buffer	27
3.2.2.3	Steady-State Probabilities for Infinite User Buffer	29
3.2.2.4	State Transition Probabilities and Cell Departure Probabilities	31
3.2.2.5	Mean, Variance and Autocovariance of Number of Cell Departures during a Slot	32

3.3	Two-Minisource Model Approach	41
3.3.1	Characterization	41
3.3.2	Network Performance Analysis Using Two-Minisource Model Approach	45
3.3.3	Accuracy of the Two-Minisource Model Approach	48
3.3.3.1	Bursty Batch Data Traffic	48
3.3.3.2	Other Traffic Types	49
3.4	Conclusions	55
Chapter 4	Modified Leaky Bucket Control	66
4.1	Simulation Model and Assumptions	66
4.2	Leaky Bucket Control	69
4.3	Adaptive Leaky Bucket Control	75
4.4	Combined Leaky Bucket and Marking Scheme	76
4.5	Combined Leaky Bucket and Priority Mechanism	79
4.6	Summary of Leaky Bucket Control	82
Chapter 5	Conclusions	88
5.1	Conclusions	88
5.2	Suggestions for Further Investigations	90
References		92

List of Figures

Figure 1.1	Layered model of ATM	4
Figure 1.2	ATM cell structure	5
Figure 3.1	Simplified ATM network model	17
Figure 3.2	Reference model for analysis and simulation	18
Figure 3.3	Queue model for leaky bucket control	19
Figure 3.4	Cells generated from source (before leaky bucket) (burst data traffic: network load = 95% (8 sources), mean burst length = 10 cells)	21
Figure 3.5	Cells delivered to network (after leaky bucket) (burst data traffic: network load = 95% (8 sources), mean burst length = 10 cells, $M = 10$, token interarrival time = 8 simulation time units, user queue length = ∞)	22
Figure 3.6	State transition diagram for leaky bucket output process (a) with two state variables (b) with one state variable	28
Figure 3.7	Normalized autocovariance, comparison of one and two exponential functions (network load = 67% (10 sources), mean burst length = 13.33 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = ∞)	35
Figure 3.8	Normalized autocovariance, comparison of one and two exponential functions (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$, token interarrival time = 10 simulation time units, user queue length = ∞)	36

Figure 3.9	Normalized autocovariance, comparison of one and two exponential functions (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	37
Figure 3.10	Probability density function of the number of departing cells per slot T (network load = 67% (10 sources), mean burst length = 13.33 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = ∞)	38
Figure 3.11	Probability density function of the number of departing cells per slot T (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$, token interarrival time = 10 simulation time units, user queue length = ∞)	39
Figure 3.12	Probability density function of the number of departing cells per slot T (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	40
Figure 3.13	Single minisource model	41
Figure 3.14	State transition diagram for a single-activity-level source model	41
Figure 3.15	Two-minisource model (a) Less bursty traffic level (b) High bursty traffic level	42
Figure 3.16	State transition diagram for an aggregate source model . . .	43
Figure 3.17	Network cell loss probability, comparison of analytical and simulation results (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	50

Figure 3.18	Network cell loss probability, comparison of analytical and simulation results (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$, token interarrival time = 10 simulation time units, user queue length = ∞)	51
Figure 3.19	Network cell loss probability, comparison of analytical and simulation results (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$, token interarrival time = 1 simulation time units, user queue length = ∞)	52
Figure 3.20	Network cell loss probability (no leaky bucket control), comparison of analytical and simulation results (network load = 67% (10 sources), mean burst length = 6.67 cells, user queue length = ∞)	53
Figure 3.21	Normalized autocovariance (MAP traffic, Case 1), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = ∞)	56
Figure 3.22	Network cell loss probability (MAP traffic, Case 1), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = ∞)	57

Figure 3.23	Normalized autocovariance (MAP traffic, Case 2), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	58
Figure 3.24	Network cell loss probability (MAP traffic, Case 2), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	59
Figure 3.25	Normalized autocovariance (MAP traffic, Case 3), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	60
Figure 3.26	Network cell loss probability (MAP traffic, Case 3), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)	61
Figure 3.27	Summary of the work in Chapter 3, Step 1	62
Figure 3.28	Summary of the work in Chapter 3, Step 2	63
Figure 3.29	Summary of the work in Chapter 3, Step 3	64
Figure 4.1	Simulation model for modified leaky bucket control schemes	67
Figure 4.2	Birth-death model for the number of active voice sources . .	68

Figure 4.3	Leaky bucket control, effect of token interarrival time on network throughput (8 sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$)	70
Figure 4.4	Leaky bucket control, effect of mean burst length on network throughput (8 sources, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units)	71
Figure 4.5	Leaky bucket control, effect of mean burst length on user end-to-end delay (8 sources, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units)	72
Figure 4.6	Leaky bucket control, effect of token size on network cell loss probabilities (network load = 80% (10 sources), mean burst length = 8 cells, user queue length = ∞ , token generation interval = 10 simulation time units)	73
Figure 4.7	Leaky bucket control, effect of number of tokens on network cell loss probabilities (network load = 80% (10 sources), mean burst length = 16 cells, user queue length = ∞ , token generation interval = 10 simulation time units)	74
Figure 4.8	Adaptive leaky bucket control, network throughput (8 data sources, mean burst length = 10 cells, user queue length = ∞ , network buffer size = 25 cells, initial token size = 50, token generation interval = 8 simulation time units)	77
Figure 4.9	Adaptive leaky bucket control, network cell loss probabilities (network load = 95% (10 sources), mean burst length = 9.5 cells, user queue length = ∞ , initial token size = 50, token generation interval = 10 simulation time units)	78

Figure 4.10	Combined leaky bucket and marking scheme, network throughput (8 data sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units, small network buffer)	80
Figure 4.11	Combined leaky bucket and marking scheme, network throughput (8 data sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units, large network buffer)	81
Figure 4.12	Leaky bucket with marking and priority mechanism, network data cell loss probabilities (8 data sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units)	83
Figure 4.13	Adaptive leaky bucket control with priority mechanism, network data cell loss probabilities (8 data sources, mean burst length = 10 cells, user queue length = ∞ , initial token size = 50, token generation interval = 8 simulation time units)	84
Figure 4.14	Adaptive leaky bucket control with priority mechanism, network voice cell loss probabilities (8 data sources, mean burst length = 10 cells, user queue length = ∞ , initial token size = 50, token generation interval = 8 simulation time units)	85
Figure 4.15	Summary of modified leaky bucket control schemes	86
Figure 5.1	Summary of the work in this thesis	89

List of Tables

Table 1.1	Packet network characteristics: present versus future . . .	2
Table 3.2	Simulation cases of MAP traffic	54

List of Acronyms

ATM	asynchronous transfer mode
B-ISDN	broadband integrated services digital network
CCITT	International Consultative Committee for Telephone and Telegraph
GFC	generic flow control (ATM cell header format)
HDTV	high definition television
HEC	header error check (ATM cell header format)
IPP	interrupted Poisson process
ISDN	integrated services digital network
MAP	Markovian arrival process
MMPP	Markov modulated Poisson process
MUX	multiplex
N-ISDN	narrowband integrated services digital network
NN	network node
NNI	network-node interface
PMD	physical medium dependent
PR	priority (ATM cell header format)
PT	payload type (ATM cell header format)
UNI	user-network interface
VCI	virtual circuit identifier (ATM cell header format)
VPI	virtual path identifier (ATM cell header format)

List of Symbols

a_i	probability of i cells arriving during a slot
$\tilde{a}_i$	probability of more than i cells arriving during a slot
A	number of cells transmitted in ON state (one-minisource model)
$A(z)$	generating function of cell arrival process to the user queue
A_h	number of cells transmitted in ON state (high rate, two-minisource model)
A_l	number of cells transmitted in ON state (low rate, two-minisource model)
b_i	probability of i batches arriving during a slot
c_a	autocovariance for two-minisource model
C_1	curve fitting parameter
C_2	curve fitting parameter
$d_{i,j}$	number of cells departing when the state of leaky bucket moves from state i to state j during a slot
c_1	curve fitting parameter
c_2	curve fitting parameter
g_i	probability of batch size equal to i
$g_{i,j}$	probability of the process being absorbed and then restarting in state j given that it has been in transient state i
K	user queue size
K_n	network buffer size
$\bar{m}$	mean number of cell departures during a slot
M	token pool size
N	data rate levels (one-minisource model)
N_1	low data rate levels (two-minisource model)
N_2	high data rate levels (two-minisource model)

N_a	mean number of active calls
N_s	number of traffic sources
N_v	number of voice sources
p_i^t	probability of leaky bucket in state i at time t
$p_{i,j}$	transition probability from state i to j during a slot
P	state transition probability matrix
$P^t(m, i)$	probability of m tokens in the token pool and i cells in the user queue
$P(z)$	generating function of steady-state probability
P_{loss}	cell loss probability
$P_{lossuser}$	measure of cell loss at user end
$q_{i,j}$	probability of the process going to transient state j given that it has been in transient state i
r	token generating rate
r_i	normalized autocovariance of number of cells departing during a slot (sequence index i)
r_{2m}	normalized autocovariance for two-minisource model
R_i	autocovariance of the number of cells departing during a slot (sequence index i)
$\bar{t}_s$	mean silence length
$\bar{t}_v$	mean talkspurt length
t_{vc}	interarrival time of speech cells
T	token interarrival time
v	variance of the number of cells departing during a slot
α_1	call arrival rate
β_1	call departure rate
$1/\beta$	mean burst length
γ	variance ratio
λ	batch arrival rate

λ_d	batch arrival rate for one data source
λ_i	traffic arrival rate in state i (for MAP)
μ	network service rate
π_i	steady-state probability in state i
ρ_d	data traffic load
ρ_v	voice traffic load

Chapter 1

Introduction

1.1 Background

1.1.1 Evolution of ISDN Towards Broadband ISDN

The integrated services digital network (ISDN) was expected to provide a single and efficient pipe and standard for voice and most non-voice communications needs. The services supported by the original ISDN concept are limited to voice and non-voice applications based on the 64 kb/s transmission rate. This is now being referred to as narrowband ISDN (N-ISDN). The data rates associated with N-ISDN are inadequate for many applications of interest. With the limitation and inflexibility of the N-ISDN, broadband ISDN (B-ISDN) issues for higher rate services have recently become prominent [1, 2]. For broadband services, the problem is not merely an increase of the operation speed, but also the compatibility of the switching technology, the workability of the local loop arrangement, and the efficiency of bandwidth utilization. Although some basic principles of N-ISDN are retained in B-ISDN, the technology and architecture are quite different. For example, the interconnection and transmission problems of video and high speed data are different from those of voice.

It is clear that the data networks of the 1980's are inadequate to handle the applications and capabilities required by 1990's. Today's packet switched data networks have a number of problems: high cost, low speed, switching delays, and relatively high

error rates. Besides, the switches require too much intelligence, the protocols are too heavyweight, and too much processing is done in the network.

Tomorrow's broadband networks require new architectures to handle the changing requirements. The move towards gigabit per second transmission requires dramatic changes in the structure and design. Table 1.1 lists some contrasts between present and future packet network characteristics [3].

Table 1.1 Packet network characteristics: present versus future

	Today	Broadband
Packets/s	Thousands	Millions
Bandwidth	64 kb/s	150 - 620 Mb/s
Bandwidth allocation	Fixed	Dynamic
Services	Voice, data	Integrated voice, data and image
Switch delay	50 - 100 ms	10 ms
Propagation delay	Insignificant	Dominant
Protocols	Heavyweight	Lightweight
Bottleneck	Link bandwidth	Switch bandwidth

In the evolution from the current telecommunication networks towards B-ISDN, some important directions and guidelines have recently been made. The directions taken by B-ISDN are influenced by a number of parameters, including the emergence of a large number of teleservices with different, often as yet unknown requirements. The most significant future teleservices are HDTV (high definition television), video conference, high speed data transfer, videophony, home education and video on demand.

Other parameters influencing the directions taken by B-ISDN are the fast evolution of the semiconductor and optical technology and the evolution in system concepts. The transfer requirements are to handle both narrowband and broadband rates, to handle both

continuous and bursty traffic, to satisfy delay and/or loss sensitive quality requirements, and to meet unforeseen demands.

Neither circuit-mode nor packet-mode can meet all of these requirements. A new switching and multiplexing scheme known as asynchronous transfer mode (ATM) has been adopted as the target solution toward B-ISDN.

1.1.2 Asynchronous Transfer Mode

1.1.2.1 Basic Principles of ATM

ATM [4, 5] is considered by the International Consultative Committee for Telephone and Telegraph (CCITT) as the preferred transfer mode for B-ISDN. Both the need for a flexible network and the progress in technology and system concepts led to the definition of the ATM principle. ATM will provide the means to transport, at broadband rates, the traffic generated by a wide range of multimedia services, which include data, voice, video, and image.

ATM is suitable for a multimedia traffic environment because it offers great flexibility and efficiency in the use of available resources. All available resources in the network are shared by all services, so that statistical sharing of the resources can be obtained. It is achieved through the adoption of fixed length packets, called cells and a simplified transport protocol based on cell switching, and the use of buffering and statistically multiplexing bursty traffic. Since the necessary number of cells is allocated on demand, no bandwidth is consumed unless information is actually being transported. The key techniques of ATM include fixed-sized cell, statistical multiplexing, and lightweight routing and traffic control protocols [2].

1.1.2.2 ATM Architecture

The ATM concept can be described using a layered model [2] as shown in Figure 1.1. The physical medium dependent (PMD) layer deals with the physical cell transmission and reception. The ATM layer describes the multiplexing and switching functions. The ATM adaptation layer provides conversion and restoration between the source information bit stream and the ATM cell stream. The high-layer service layer supports other services such as network terminal signalling and information source coding.

Figure 1.1 Layered model of ATM

An ATM cell consists of a 5-octet header and a 48-octet information field. The header formats which will be used at user-network interface (UNI) and network-node interface (NNI) are shown in Figure 1.2. For UNI, the header contains a 4-bit generic flow control (GFC) field, a 24-bit label field containing virtual path identifier (VPI) and virtual circuit identifier (VCI) subfields (8 bits for the VPI and 16 bits for the VCI), a

2-bit payload type (PT) field, a 1-bit reserved field, a 1-bit priority (PR) field, and an 8-bit header error check (HEC) field. For NNI, the header does not contain a GFC field, and the extra 4 bits are used for a VPI field.

Figure 1.2 ATM cell structure

1.2 Thesis Objectives

ATM networks support services with different traffic statistics and service requirements. In order to regulate the traffic flow, achieve the best use of network resources, and guarantee good service quality to the users, congestion control is implemented to prevent the network from throughput/delay performance degradation due to congestion. In this thesis, congestion control in ATM networks is investigated using a so called leaky bucket approach. The main objectives of the thesis are as follows. For each objective, the highlights of research results are summarized.

- ***Performance analysis of leaky bucket congestion control:*** Analyze the leaky bucket performance considering batch traffic with a Poisson arrival process and a geometrically distributed batch length. The mean, variance and autocovariance of the leaky bucket output process will be derived. A performance analysis of leaky bucket was previously given in [6] assuming a Poisson cell arrival process. However, a simple Poisson cell arrival process may not accurately describe the bursty traffic found in ATM networks since batch arrivals (a large number of cells) occur frequently. It is therefore important to look into the batch arrival process and the batch length distribution.

- ***ATM network performance analysis using a two-minisource model approach:*** Evaluate network performance in terms of cell loss probabilities using two-minisource model. For general traffic arrivals, it is difficult to analyze the output process of the leaky bucket. A two-minisource model is used to approximate the output process and computer simulation is used to validate the approximation.

- ***Extensive simulation of modified leaky bucket control schemes:*** Propose and evaluate the performance of modified leaky bucket control schemes including adaptive leaky bucket control, combined leaky bucket and marking scheme, and combined leaky bucket and priority mechanism. The performance of the basic leaky bucket control

scheme can be improved by incorporating with other control mechanisms such as marking and prioritizing. The performance is evaluated through extensive simulation with emphasis on network throughput and cell loss probabilities.

1.3 Thesis Organization

This thesis consists of five chapters. After this introductory chapter dealing with background issues of ATM networks, in Chapter 2, the congestion control techniques available for ATM networks are discussed. Congestion control mechanisms at different levels including call level and cell level are reviewed. Since traffic modelling is a crucial step in developing congestion control schemes, it is also discussed in Chapter 2 together with congestion control.

The thesis approaches the congestion control problem at the cell level. In Chapter 3, a leaky bucket scheme is used for congestion control. The leaky bucket is first analyzed in terms of the output process mean, variance and autocovariance, and then the output process is characterized using a two-minisource model. The performance of ATM networks with leaky bucket control is analyzed and the analytical results are also validated through computer simulation. A software package SIMSCRIPT is used for simulation throughout this thesis.

More extensive simulation results for the basic leaky bucket control scheme and various modified leaky bucket control schemes are presented in Chapter 4. The modified schemes include adaptive leaky bucket control, combined leaky bucket and marking scheme, and combined leaky bucket and priority mechanism.

Finally in Chapter 5, conclusions are drawn and recommendations are given for further research.

Chapter 2

Traffic Modelling and Congestion Control in ATM Networks

The conventional congestion control mechanisms for packet-switched network are not appropriate in an ATM environment [7] because of its unique traffic characteristics, high speed and service requirements. A layered and distributed congestion-control framework has been proposed to apply to the design of congestion control in ATM networks, which is partitioned into three control domains, network-level congestion control, call-level congestion control, and cell-level congestion control [2, 8].

The objective of this chapter is to review and discuss the congestion control framework for ATM and the emphasis will be cell-level control schemes. The traffic modelling issue is first reviewed in Section 2.1. Section 2.2 describes the congestion control problems in ATM-based networks. Section 2.3 compares conventional congestion control with congestion control for ATM networks. In Section 2.4, call-level congestion control is discussed. Section 2.5 concentrates on cell-level congestion, including traffic policing, traffic smoothing and cell prioritizing. A summary is given in Section 2.6.

2.1 Traffic Modelling of Voice, Data and Video

Modelling of the cell arrival process is an essential part in ATM network design and performance evaluation. There are extensive investigations on this issue [7, 9, 10]. Three

major categories of traffic models have been considered, i.e., input traffic models for data sources, input traffic models for voice sources and input traffic models for video sources.

2.1.1 Input Traffic Models for Data Sources

Generation of data from a single data source is frequently characterized by a Poisson arrival process (continuous time case) or by a geometric interarrival process (discrete time case). For interactive data transmission, a single cell may be generated at a time. For a bulk data transmission, such as a file transfer, a large number of cells may be generated at a time (batch arrivals). Therefore, the traffic needs to be modelled considering batch arrival process and a batch length distribution.

2.1.2 Input Traffic Models for Voice Sources

Modelling of the arrival process of voice traffic is more complex since the arrival traffic is correlated. The arrival process of new voice calls and the distribution of their durations can be characterized by a Poisson process and by an exponential distribution, respectively. Within a call, talkspurts and silent periods alternate. During talkspurts, voice cells are generated periodically, during silent periods, no cells are generated. The correlated generation of voice cells within a call can be modelled by an interrupted Poisson process (IPP) [9, 11]. A common approach for modelling aggregate arrivals from a number of voice sources is to use a two-state Markov modulated Poisson process (MMPP) [10, 12].

2.1.3 Input Traffic Models for Video Sources

Video sources also generate correlated cell arrivals but with very different statistics. Video services have very large bandwidth requirements so that variable-bit-rate compression algorithms have been proposed to reduce the bandwidth consumption in sharing network resources. Two correlated Markov process models (discrete time and continuous

time) have been shown to fit the experimental data well [13, 14]. Note that two types of correlations should be considered in video traffic modelling. One is short-term correlation corresponding to uniform activity levels and the other is long-term correlation corresponding to sudden scene changes with different activity levels [15].

2.2 Congestion Control Problem

Most traffic sources in ATM networks are bursty. A bursty source generates a large amount of traffic (cells) at some high peak rate for a short period of "active" time and generates little or no traffic for some "idle" time. If bandwidth were allocated based on peak rate (deterministic multiplexing), network resources would be wasted when the source is idle. Since an ATM network supports a large number of such bursty traffic sources, statistical multiplexing is more efficient, allowing more traffic sources to share the bandwidth. However, severe network congestion may occur if a large number of traffic sources become active at the same time.

Another key issue is the effect of the high speed channels. Cell processing must be performed at speeds comparable to the high switching speeds. This can only be done by simplifying the schemes enough to avoid excessive processing time. Another problem caused by the high speed channel is the propagation delay. Since the speed of ATM is very high, any action the sources take may be too late to resolve the buffering and switching congestion. High speed channels have changed the network situation dramatically, so that some of the congestion control schemes developed for existing networks are no longer be applicable.

2.3 Conventional Congestion Control versus Congestion Control for ATM

The congestion control schemes used for existing networks usually fall into the class of reactive control which reacts to the congestion after it happens and tries to bring the degree of network congestion to an acceptable level. For example, in the conventional packet switching networks, the applied congestion control schemes are window-based mechanisms typically relying on end-to-end exchange of control messages in order to regulate traffic flow, slow down or stop the source to generate packets. The control messages are used as feedback by the source node to regulate its traffic. However, these reactive control schemes are not appropriate in ATM environments.

Congestion control in ATM networks is based on the philosophy of preventing congestion rather than reacting it, that is, preventive control. Admission control and traffic policing/smoothing are examples of preventive control. In ATM networks, congestion control can be first exercised at a call level. At the call level, it is the decision whether to accept an incoming call into the network in order to avoid long term network congestion, which is called admission control. Once the decision is made to accept the call, the network must allocate the necessary resources to accommodate that call and maintain its quality of service.

2.4 Call-Level Congestion Control

2.4.1 Admission Control

When a new call requests connection, the network should first decide whether it can admit the call or not [16]. The factors that affect the admission of a call include the availability of network resources (network bandwidth), the traffic characteristics of the new call and of existing calls sharing the same resources, and the quality of service

requirements (such as loss probability and delay) of the new and existing calls. If the network does not have enough bandwidth available along the path connecting this call to support the quality of service requirements of the new call and the existing calls, the service request should be rejected; otherwise, the request can be accepted. Meanwhile, some bandwidth needs to be allocated to the accepted new call. Admission control only operates once for each call so that the processing load is relatively low.

The performance of admission control depends upon two factors, precise characterization of input traffic and measurement of bandwidth used by the admitted users (existing calls) statistically sharing the resource.

Accurate characterization of input traffic is not easy owing to the time-varying and burst nature of the input traffic in high-speed networks. The task of traffic modelling discussed in Section 2.1 deals with the characterization issue. A simple parameter set to describe the traffic may consist of the average bit rate, the peak bit rate, and the mean burst length. Another critical problem encountered in traffic characterization is the uncertainty of parameter values caused by incomplete measurements or inaccurate models. Statistical theory or fuzzy set theory may solve this problem to some extent [17, 18].

2.5 Cell-Level Congestion Control

2.5.1 Traffic Policing

Once a connection has been set up, the traffic source needs to be monitored to ensure that it does not violate the traffic pattern declared and authorized in the call admission phase. Traffic policing may be carried out on individual connections, on sets of connections or on all traffic at a network access point.

The so called leaky bucket scheme [19, 20] uses tokens to enforce the authorized source traffic pattern. Tokens are generated at a fixed rate determined by the bandwidth

granted to the call. Tokens may accrue in the token pool up to a certain maximum number which is used to constrain the burstiness of input traffic. An ATM cell must capture a token before it is transmitted; otherwise it must wait in the queue until a token is generated. A cell is discarded if it encounters a full buffer on arrival.

2.5.2 Traffic Smoothing

While traffic policing emphasizes bandwidth enforcement, traffic smoothing emphasizes traffic shaping to reduce burstiness and improve network throughput. The basic premise of traffic smoothing is the use of buffering to achieve a certain transmission bits per smoothing interval [21, 22].

With traffic smoothing, minimized delay variance is obtained compared to non-smoothed traffic. In order to reduce the required buffer sizes in performing traffic smoothing, smaller smoothing intervals are anticipated. It is noted that smoothing the traffic entering the network is achieved at the cost of increased delay and occasional cell losses to maintain overall performance throughout the network.

2.5.3 Cell Prioritizing

In cell-level congestion control, we can use priorities between and within service classes to provide multiple grades of service with ATM. Having determined priority levels for various services, we must handle prioritized cells in an appropriate manner during cell discarding and scheduling. Priority in discarding determines which cells are dropped when buffer overflow occurs [23]. Scheduling priority determines the order of cell transmissions [24]. Priority schemes exploit the fact that there may be more and less significant cells in voice and video coding.

A fixed, or head-of-line, priority scheme is a simple priority scheduling scheme to serve multiple classes with various delay requirements. The traffic is classified into a number of priorities. The input buffer is divided into a number of queues, and an arriving

cell is placed in its corresponding queue. Cells in a higher priority queue are always served first.

In priority discarding schemes, discarding of cells based on the priorities of the voice or video coding scheme is used to prevent congestion. Normally, cells are accepted into the input buffer until the buffer becomes full. With multiple priorities, higher priority cells can push out lower priority cells when buffers are full.

2.6 Observations

Admission control is performed only once per call. It may be combined with a cell-level control scheme. Both traffic policing and traffic smoothing schemes enforce allocated bandwidth and both schemes reduce burstiness. Traffic smoothing schemes are better at ensuring smoothness compared to traffic policing. In traffic policing, for instance, a leaky bucket may release all its tokens at once during a burst. However, with a spacer [25], the smoothness of other traffic smoothing schemes can be achieved with a leaky bucket. Priority schemes are primarily used in an environment with multiple grades of service which can be implemented in combination with the traffic policing or traffic smoothing scheme.

The rest of the thesis concentrates on the leaky bucket approach in performing congestion control.

Chapter 3

Performance Analysis of Leaky Bucket and a Two-Minisource Model Approach

3.1 Introduction

In this chapter, the leaky bucket performance is investigated, including characterizing its output process and deriving cell loss probabilities in the network. Although there have been some studies on the leaky bucket control [6, 26], the performance analysis of leaky bucket is very limited, in particular, its output process and the performance of the network with leaky bucket control have not been investigated. This chapter presents analyses on the leaky bucket output process in terms of mean, variance and autocovariance, and the network performance including the distribution of the network queues and the cell loss probabilities. Considering a batch arrival process (data traffic) with geometrically distributed batch (or burst) size and exponentially distributed batch interarrival time, the mean, variance and autocovariance of the output process of the leaky bucket and the network throughput are explicitly derived. Previous work on leaky bucket control assumes a simple Poisson cell arrival process. In order to model the bursty traffic more accurately, a batch arrival process is used here. For more general traffic types, an approximated output process of the leaky bucket is given in this chapter using a two-minisource model approximation. With this approximation, the output process of leaky bucket is considered as a combination of two types of minisource [15]. From this model, the network buffer queue distribution and the cell loss probabilities are obtained.

When a user establishes a connection, its cell arrival traffic characteristics are declared to the access point for congestion control purpose by leaky bucket. The parameters of the leaky bucket are determined at the time of call establishment, such as the allowed maximum degree of traffic burstiness as well as the mean and variance of the traffic entering the network. However, the characteristics of the traffic after leaky bucket is not the same as the original traffic entering the access point. In order to build a mathematical model to characterize a real cell arrival stream to the network, some statistics such as the mean and variance of the number of arriving cells to the network and the corresponding autocovariance function are needed. Methods for collecting and processing data in order to produce the required statistics will be presented in this chapter.

A cell stream in ATM networks is bursty and correlated, even after the leaky bucket. It is important to choose a proper statistical model and parameters to characterize the leaky bucket output process. In this chapter, the cell burst interarrival time and the number of cells entering the network during a fixed interval are possible candidate parameters that should be measured. However, it is easier to measure the number of arriving cells than to measure the cell burst interarrival time.

This chapter is organized as follows. Section 3.2 first describes the network and leaky bucket model. Assuming batch data traffic (a Poisson arrival process and a geometrically distributed batch length), the behavior of leaky bucket is then analyzed in terms of output process mean, variance and autocovariance. Section 3.3 proposes to use a two-minisource model to approximate the leaky bucket output process according to the mean, variance and autocovariance obtained in Section 3.2. The two-minisource model approach is used to analyze the network performance, such as cell loss probabilities and the network queue distribution. A comparison of simulation results and the theoretical results using the two-minisource model is made to show the accuracy of the two-minisource model. A general Markovian arrival process [26] is also studied in this section showing the applicability

of the two-minisource model. Conclusions are given in Section 3.4.

3.2 Performance Analysis of Leaky Bucket

3.2.1 Network and Leaky Bucket Model

Figure 3.1 depicts a simplified model of an ATM network. N_s traffic sources are multiplexed (MUX) on high speed link to reach a network node (NN).

Figure 3.1 Simplified ATM network model

To understand the relation between the traffic characteristics, the bandwidth assignment policy, and network performance, the reference model of a network shown in Figure 3.2 is used in this thesis. N_s statistically independent traffic sources, each specified by traffic parameters, are multiplexed on a link of bandwidth 150 Mb/s through a buffer of capacity K_n cells. Note that one ATM link is considered in the study by assuming one network queue. In the interface between the data source and the network, an access control mechanism called "leaky bucket" is used (Figure 3.3).

Figure 3.2 Reference model for analysis and simulation

Figure 3.3 Queue model for leaky bucket control

The leaky bucket controls the traffic flow by means of tokens and the basic operation can be described by a queueing model shown in Figure 3.3. Input cells first enter a user queue. If the queue is full, the cell is discarded. In order to enter the network, a cell must obtain a token from a token pool (pool size M); If there is no token in the token pool, a cell waits in the queue until a new token is generated. Tokens are generated at a fixed rate that corresponds to the specified average rate of the connection. If the number of tokens in the token pool reaches a pre-defined maximum number, say M , newly generated tokens are discarded. This maximum number is determined by the allowed burstiness of transmissions and different maximum token values can cause different degrees of burstiness to enter the network. Therefore, the leaky bucket method enforces the average input rate and at the same time it allows a certain degree of burstiness while limiting the peak bandwidth.

Figures 3.4 and 3.5 are obtained from simulation results. Figure 3.4 shows the cell generating rate of a particular bursty data source and Figure 3.5 shows the output of the leaky bucket for the same data source. Comparing these two figures, it is seen that the leaky bucket limits the peak consumption and smooths the incoming bursty traffic. Note that the original leaky bucket method [27] does not include the input buffer (user queue). The input buffer is proposed later as a result of the trade-off between the cell loss probability and the cell waiting time [28].

The leaky bucket control is analyzed in two steps. First, in Section 3.2.2, the source is combined with the leaky bucket to derive the output process of Figure 3.3. Second, in Section 3.3, a two-minisource model approach is used to analyze the network performance.

3.2.2 Output Process of the Leaky Bucket

For a leaky bucket with a Poisson cell arrival process and an input queue, an exact analysis is given in [6]. In this section, the leaky bucket (with an input queue) performance for *batch traffic* with a Poisson arrival process and a geometrically distributed batch length is analyzed. More general input traffic cases (Markovian arrival processes) will be studied in later sections.

As shown in Figure 3.3, the maximum token pool size is M and the token generating process is deterministic with rate r . Batches arrive and enter a user queue with size K according to a Poisson process with rate λ . The burst length is geometrically distributed with mean $1/\beta$. Consider a slotted time axis where a slot is the interarrival time between tokens, T . A new token is generated at each slot boundary, i.e., just after the beginning of a new slot. If the token pool is full, the newly generated token is discarded; otherwise, it joins the token pool. Let $P^t(m, i)$ be the probability of having m tokens in the token pool and i cells in the user buffer queue at time t , ($t=0, T, 2T \dots$), just before the token

Figure 3.4 Cells generated from source (before leaky bucket)
(burst data traffic: network load = 95% (8 sources), mean burst length = 10 cells)

Figure 3.5 Cells delivered to network (after leaky bucket)
 (burst data traffic: network load = 95% (8 sources), mean burst length = 10 cells,
 $M = 10$, token interarrival time = 8 simulation time units, user queue length = ∞)

generation instance. The operation of the leaky bucket implies that the token pool and the user queue are not simultaneously non-empty. That is to say, there are cells waiting in the user queue if and only if no token is in the token pool, and the number of tokens can not be beyond M . Therefore, we have

$$P^t(m, i) = 0, \quad m > M, \quad \text{or } 1 \leq m \leq M \text{ and } i \geq 1. \quad (3.1)$$

The notations listed below are used in the following derivations.

T = token interarrival time.

a_i = the probability of i cells arriving during a slot, T .

π_i = the steady-state probability in state i .

$1/\beta$ = mean batch size.

b_i = the probability of i batches arriving during a slot, T .

p_{ij} = the transition probability from state i to state j during a slot, T .

d_{ij} = the number of cells departing when the state of leaky bucket moves from state i to state j during a slot, T .

M = maximum token queue size.

K = maximum user queue size.

r_i = normalized autocovariance of number of cells departing during a slot, T , (sequence index i).

v = variance of the number of cells departing during a slot, T .

λ = the batch arrival rate.

r = deterministic token generating rate.

R_i = autocovariance of the number of cells departing during a slot, T , (sequence index i).

For simplicity, the following notations are used: $p_0^t = P^t(M, 0)$, $p_1^t = P^t(M-1, 0)$, ..., $p_M^t = P^t(0, 0)$, ..., $p_{M+i}^t = P^t(0, i)$, and $\pi_i = \lim_{t \rightarrow \infty} p_i^t$ ($i \geq 0$) (the corresponding steady-state probability).

The probability of i batches arriving during a slot T is

$$b_i = \exp(-\lambda T)(\lambda T)^i / i!, \quad (i \geq 0). \quad (3.2)$$

The probability that the batch size equals to i is

$$g_i = (1 - \beta)\beta^{i-1}, \quad i > 1. \quad (3.3)$$

Notice that, since the batch arrival is being considered, the probability of i cells arriving during a slot is different from the probability of i batch arrivals.

3.2.2.1 Cell Arrival Probabilities

Lemma 1: The probability vector of i cells arriving during a slot T is

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_i \\ \vdots \end{bmatrix} = Q \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \end{bmatrix} \quad (3.4)$$

where $Q = [Q_0 \quad Q_1 \quad Q_2 \quad \cdots \quad Q_i \quad \cdots]$. Q_i is

$$Q_0 = q,$$

$$Q_1 = Q_0 * g,$$

$$\begin{aligned}
Q_2 &= Q_1 * g, \\
&\vdots \\
Q_i &= Q_{i-1} * g,
\end{aligned} \tag{3.5}$$

where $*$ stands for a convolution operation, $g = [1 \ 0 \ 0 \ \dots \ 0]'$ with size of $M + K$, g is the probability vector of batch size,

$$\begin{aligned}
g' &= [g_0 \ g_1 \ g_2 \ g_3 \ \dots] \\
&= [0 \ \alpha \ \alpha\beta \ \alpha\beta^2 \ \dots] ,
\end{aligned} \tag{3.6}$$

where $\alpha = 1 - \beta$ and b_i is the probability of i batches arriving during a slot, T , as given in Equation (3.2).

Proof.

The probability of i cells arriving during a slot is

$$\begin{aligned}
a_i &= \text{prob}(1 \text{ batch}) \text{prob}(\text{batch size} = i) \\
&+ \text{prob}(2 \text{ batches}) \sum_{\text{batch size } j=1}^{i-1} \text{prob}(j) \text{prob}(i-j) \\
&+ \text{prob}(3 \text{ batches}) \sum_{\text{batch size } j=1}^{i-1} \text{prob}(j) \sum_{k=1}^{i-j-1} \text{prob}(k) \text{prob}(i-j-k) \\
&+ \dots
\end{aligned} \tag{3.7}$$

The above expression can be simplified by using a matrix expression,

$$\begin{aligned}
 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_i \\ \vdots \\ a_{M+K} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 0 & \alpha & 0 & 0 & \cdots \\ 0 & \alpha\beta & \alpha^2 & 0 & \cdots \\ 0 & \alpha\beta^2 & 2\alpha^2\beta & \alpha^3 & \cdots \\ 0 & \alpha\beta^3 & 3\alpha^2\beta^2 & 3\alpha^3\beta & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix} \\
 &= [Q_0 \quad Q_1 \quad Q_2 \quad Q_3 \quad \cdots] \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix} \\
 &= Q \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix} \tag{3.8}
 \end{aligned}$$

which gives

$$a_{k+1} = \alpha \sum_{l=0}^k \binom{k}{l} \alpha^l \beta^{k-l} b_{l+1} \tag{3.9}$$

for $k = 0, 1, 2, \dots, K + M - 1$.

Q.E.D.

Figure 3.6 shows the state transition diagram of a leaky bucket operation, where (m, i) is the state corresponding to m tokens in the token pool and i cells in the user queue, $a_i = \text{prob}(i \text{ cells arriving during } T)$, and $\tilde{a}_i = \text{prob}(\geq i \text{ cells arriving during } T) = \sum_{k=i}^{\infty} a_k$.

Since $i \cdot m$ always equals to 0 ($i \cdot m = 0$), for simplicity, one state variable i is used to represent the state instead of two variables (m, i). Therefore, state $(M, 0)$ is simplified as state (0), $(M-1, 0)$ as (1), $(M-2, 0)$ as (2), ..., $(0, 0)$ as (M) , $(0, 1)$ as $(M+1)$, ..., $(0, K)$ as $(M+K)$. Figure 3.6 (b) gives the simplified state transition diagram. Using this diagram, the steady-state probability, π_i can be derived.

3.2.2.2 Steady-State Probabilities for Finite User Buffer

Lemma 2: When the user queue size is finite, K , the steady-state equations are

$$\pi_i = \begin{cases} \pi_0 a_i + \sum_{j=0}^i \pi_{j+1} a_{i-j}, & 0 \leq i \leq M+K-1 \\ \pi_0 \tilde{a}_{M+K} + \sum_{j=1}^i \pi_j a_{i-j}, & i = M+K \\ 0, & i > M+K \end{cases} \quad (3.10)$$

Proof:

For state i , the input traffic load should be equal to output traffic load, which gives

$$\begin{aligned} & (a_0 + a_2 + a_3 + \dots + a_i + \dots + \tilde{a}_{M+K}) \pi_i \\ &= a_0 \pi_{i+1} + a_2 \pi_{i-1} + a_3 \pi_{i-2} + \dots + a_{i-1} \pi_1 + a_i \pi_0 \end{aligned} \quad (3.11)$$

add $a_1 \pi_i$ to both sides, we get

$$\pi_i = \pi_0 a_i + \sum_{j=0}^i \pi_{j+1} a_{i-j}, \quad 0 \leq i \leq M+K-1 \quad (3.12)$$

Q.E.D.

Using the condition that $\sum_{i=0}^{M+K} \pi_i = 1$ (normalization), it is easy to get the solution for Equation (3.12). First, an arbitrary value $\pi_0 > 0$ is assumed. Then a recursion equation $\pi_i = \left(\pi_{i-1} - \pi_0 a_{i-1} - \sum_{j=0}^{i-2} \pi_{j+1} a_{i-1-j} \right) / a_0$ is used to find all π_i , $1 \leq i \leq M+K$. Finally, π_i is obtained by using the above normalization condition.

Figure 3.6 State transition diagram for leaky bucket output process (a) with two state variables (b) with one state variable

With the values (a_i, π_i) obtained above, the mean number of cells entering the network per slot (i.e. the network traffic load) can be obtained as

$$\begin{aligned} \text{Network Traffic Load} = & \pi_0 \left(\sum_{i=0}^{K+M} i a_i + (M+K) \tilde{a}_{M+K} \right) \\ & + \sum_{j=1}^{K+M} \pi_j \left(\sum_{i=1}^{K+M-j+1} i a_i + (M+K-j+1) \tilde{a}_{M-K-j+1} \right) \end{aligned} \quad (3.13)$$

where $\tilde{a}_i = 1 - \sum_{j=0}^i a_j$. The measure of cell loss at the leaky bucket is given by

$$P_{\text{loss user}} = 1 - \frac{\text{Network Traffic Load} \times \beta}{\lambda T} \quad (3.14)$$

3.2.2.3 Steady-State Probabilities for Infinite User Buffer

The above analyses assume finite buffer size at the leaky bucket. For the case of infinite user buffer size, the steady-state equation is the same as Equation (3.17) except that it holds for all i ,

$$\pi_i = \pi_0 a_i + \sum_{j=0}^i \pi_{j+1} a_{i-j}, \quad i \geq 0. \quad (3.15)$$

For infinite user buffer size, the same approach can not be used to determine π_0 . Here, the generating function $P(z)$ of Equation (3.15) is derived first and then the normalization condition $P(z)|_{z=1} = 1$ is used to obtain π_0 .

Let $P(z) = \sum_{i=0}^{\infty} \pi_i z^i$ and apply it to Equation (3.15).

$$P(z) = \sum_{i=0}^{\infty} \pi_0 a_i z^i + \sum_{i=0}^{\infty} \sum_{j=0}^i \pi_{j+1} a_{i-j} z^i$$

$$\begin{aligned}
&= \sum_{i=0}^{\infty} \pi_0 a_i z^i + \sum_{i=0}^{\infty} \sum_{j=0}^i \pi_{j+1} a_{i-j} z^i \\
&= \pi_0 A(z) + \sum_{i=0}^{\infty} \sum_{j=0}^i \pi_{j+1} z^{j+1} / z,
\end{aligned} \tag{3.16}$$

where $A(z)$ is the generating function of the cell arriving process to the user queue. Since

$$\sum_{i=0}^{\infty} \sum_{j=0}^i (\cdot) = \sum_{j=0}^{\infty} \sum_{i=j}^{\infty} (\cdot),$$

$$\begin{aligned}
P(z) &= \pi_0 + \sum_{j=0}^{\infty} \pi_{j+1} z^{j+1} \sum_{i=j}^{\infty} a_{i-j} z^{i-j} / z \\
&= \pi_0 A(z) + \sum_{j=0}^{\infty} \pi_{j+1} z^{j+1} \sum_{k=0}^{\infty} a_k z^k / z \\
&= \pi_0 A(z) + P(z) A(z) / z - \pi_0 A(z) z,
\end{aligned} \tag{3.17}$$

where $k=i-j$. The generating function of π_i is obtained as

$$P(z) = \frac{\pi_0 A(z)(z-1)}{z - A(z)}. \tag{3.18}$$

Using the normalization condition $P(z)|_{z=1} = 1$ and L'Hospital rule,

$$\begin{aligned}
P(z)|_{z=1} &= \frac{(\pi_0 A(z)(z-1))'}{(z - A(z))'} \\
&= \frac{\pi_0 A(z)}{1 - A'(z)} = 1
\end{aligned} \tag{3.19}$$

and therefore $\pi_0 = \frac{1-A'(1)}{A(1)} = 1 - A'(1)$, where $A'(1)$ is the mean of the number of cell arrivals during a slot T and $A(1) = 1$.

Finally, it is obtained that

$$\pi_0 = 1 - \frac{\lambda T}{\beta}. \tag{3.20}$$

3.2.2.4 State Transition Probabilities and Cell Departure Probabilities

So far, the steady-state probabilities of the transition diagram (Figure 3.6 (b)) have been obtained. In order to evaluate the output process of the leaky bucket (mean, variance and autocovariance), it is necessary to find the state transition probabilities and cell departure (from the leaky bucket) probabilities.

The transition probabilities from state i to state j , $p_{i,j}$, can be obtained using the following procedures.

Let's first consider the boundary cases,

$$\begin{aligned} \text{when } i=0, j \leq M+K-1: & \quad p_{0,j} = a_j; \\ \text{when } i=0, j=M+K: & \quad p_{0,j} = \tilde{a}_{M+K}; \\ \text{when } i \neq 0, j=M+K: & \quad p_{i,j} = \tilde{a}_{M+K-i+1}. \end{aligned}$$

For the non-boundary cases,

$$\begin{aligned} \text{when } i>0, j \leq M+K \text{ and } j > i: & \quad p_{i,j} = a_{i-j+1}; \\ \text{when } i>0, j=i-1: & \quad p_{i,j} = a_0; \\ \text{when } i \leq 0 \text{ or } i > M+K, j < 0 \text{ or } j > M+K: & \quad p_{i,j} = 0. \end{aligned}$$

It has been assumed that $P^t(m,i)$ is the probability of having m tokens in the token pool and i cells in the user queue at time t , just before a token arrival. The following considers the number of cells departing during T (token interarrival time) in the $i \rightarrow j$ transition.

For $i \geq M+1$:

$$\text{there is only 1 departure,} \quad d_{i,j} = 1;$$

For $i \leq M$:

$$\text{if } i-1 \leq j \leq M, \text{ i.e., there are } j-i+1 \text{ departures,} \quad d_{i,j} = j-i+1;$$

$$\text{if } j \geq M+1, \text{ i.e., there are } M-i+1 \text{ departures,} \quad d_{i,j} = M-i+1.$$

The number of departures during T in the $i \rightarrow j$ transition is therefore

$$d_{i,j} = \begin{cases} \min(M, j) - i + 1, & 0 < i \leq M \\ \min(M, j) - i, & i = 0 \\ 1 & i \geq M + 1 \end{cases} \quad (3.21)$$

3.2.2.5 Mean, Variance and Autocovariance of Number of Cell Departures during a Slot

Based on above equations, the mean ($\bar{m}$) and the variance (v) of the number of cell departures during a slot time T can be obtained as

$$\bar{m} = \sum_{i=0}^{M+K} \pi_i \sum_{j=0}^{M+K} d_{i,j} p_{i,j} \quad (3.22)$$

and

$$\begin{aligned} v &= \overline{m^2} - \bar{m}^2 \\ &= \sum_{i=0}^{M+K} \pi_i \sum_{j=0}^{M+K} d_{i,j}^2 p_{i,j} - \bar{m}^2 \end{aligned} \quad (3.23)$$

By denoting $i(t)$ as the state at the end of time slot $t - 1$ and just before the beginning of time slot t , and similarly, $j(t)$ as the state at the end of time slot t and just before the beginning of time slot $t + 1$, we can compute the autocovariance of the number of cell departures during two slots having a lag of τ time units as follows:

$$R_\tau = E[d_{i(t),j(t)} d_{i(t+\tau),j(t+\tau)}] - (\bar{m})^2 \quad (3.24)$$

For the sake of notation, we set $i(t) = i$, $j(t) = j$, $i(t + \tau) = k$ and $j(t + \tau) = l$. We have $d_{i(t),j(t)} = d_{i,j}$ and $d_{i(t+\tau),j(t+\tau)} = d_{k,l}$. Equation (3.24) is derived as follows.

In the case of R_1 , $k = j$,

$$\begin{aligned} R_1 &= E[d_{i,j} d_{j,l}] - (m)^2 \\ &= \sum_{i=0}^{M+K} \sum_{j=0}^{M+K} \sum_{l=0}^{M+K} d_{i,j} d_{j,l} p_{i,j} p_{j,l} \pi_i - (m)^2 \end{aligned} \quad (3.25)$$

For R_2 , we have

$$\begin{aligned} R_2 &= E[d_{i,j} d_{k,l}] - (m)^2 \\ &= \sum_{i=0}^{M+K} \sum_{j=0}^{M+K} \sum_{k=0}^{M+K} \sum_{l=0}^{M+K} d_{i,j} d_{k,l} p_{i,j} p_{j,k} p_{k,l} \pi_i - (m)^2 \end{aligned} \quad (3.26)$$

For R_3 , we have

$$\begin{aligned} R_3 &= E[d_{i,j} d_{k,l}] - (m)^2 \\ &= \sum_{i=0}^{M+K} \sum_{j=0}^{M+K} \sum_{k=0}^{M+K} \sum_{l=0}^{M+K} d_{i,j} d_{k,l} p_{i,j} p_{j,k}^{(2)} p_{k,l} \pi_i - (m)^2 \end{aligned} \quad (3.27)$$

Similarly, we obtain

$$R_n = \sum_{i=0}^{M+K} \sum_{j=0}^{M+K} \sum_{k=0}^{M+K} \sum_{l=0}^{M+K} d_{i,j} d_{k,l} p_{i,j} p_{j,k}^{(n-1)} p_{k,l} \pi_i - (m)^2 \quad (3.28)$$

By denoting $d_{in}(j) = \sum_{i=0}^{M+K} d_{i,j} p_{i,j} \pi_i$ and $d_{out}(i) = \sum_{j=0}^{M+K} d_{i,j} p_{i,j}$, we have in general the following shorthand notation,

$$R_n = \sum_{j=0}^{M+K} \sum_{k=0}^{M+K} d_{in}(j) p_{j,k}^{(n-1)} d_{out}(k) - (m)^2 \quad (3.29)$$

In above equations, $p_{j,k}^{(t)}$ is the jk^{th} entry of the multiple step transition probability matrix $P^{(t)}$. P is the state transition probability matrix and $P^{(t)} = P \times P^{(t-1)}$ [29].

Finally, the normalized autocovariance is

$$r_t = R_t/v. \quad (3.30)$$

The normalized autocovariance is plotted in Figures 3.7, 3.8 and 3.9 for different leaky bucket and network parameters. The vertical axis shows the normalized autocovariance while the horizontal axis shows the time index for covariance sequences (in time unit T). A simulation time unit equals to one cell transmission time at network node. There are three curves in each figure and one of them is plotted using the exact analytical results obtained in this section. The other two curves are the result of curve fitting using exponential functions, one with a single exponential function (in the form of $C_1 \exp(-e_1 t)$, where C_1 and e_1 are the curve fitting parameters) and the other with two exponential functions ($C_1 \exp(-e_1 t) + C_2 \exp(-e_2 t)$, where C_2 and e_2 are also the curve fitting parameters). A minimum mean squared error criteria is used in the curve fitting process. It can be seen that the normalized autocovariance of the leaky bucket output process is very well approximated by a sum of two exponential functions. This is the motivation for adopting a two-minisource approximation in the next section. It is thus seen that the autocovariance is used to justify the use of a two-minisource model. The probability density function of the number of departing cells per slot T is also obtained through simulation. The probability density function curves with different leaky bucket and network parameters are presented in Figures 3.10, 3.11 and 3.12. A unimodal and bell-shaped density, truncated to zero, is observed. Since multiple minisources correspond to a bell-shaped probability density function [13], this favors the adoption of the two-minisource model instead of the single minisource model.

**Figure 3.7 Normalized autocovariance,
comparison of one and two exponential functions
(network load = 67% (10 sources), mean burst length = 13.33 cells, $M = 3$,
token interarrival time = 10 simulation time units, user queue length = ∞)**

Figure 3.8 Normalized autocovariance,
 comparison of one and two exponential functions
 (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$,
 token interarrival time = 10 simulation time units, user queue length = ∞)

Figure 3.9 Normalized autocovariance,
 comparison of one and two exponential functions
 (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 3$,
 token interarrival time = 10 simulation time units, user queue length = 20)

Figure 3.10 Probability density function of the number of departing cells per slot T (network load = 67% (10 sources), mean burst length = 13.33 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = ∞)

Figure 3.11 Probability density function of the number of departing cells per slot T (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$, token interarrival time = 10 simulation time units, user queue length = ∞)

Figure 3.12 Probability density function of the number of departing cells per slot T
(network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 3$,
token interarrival time = 10 simulation time units, user queue length = 20)

Figure 3.13 Single minisource model

Figure 3.14 State transition diagram for a single-activity-level source model

3.3 Two-Minisource Model Approach

3.3.1 Characterization

In this section, a two-minisource model is used to approximate the output process of the leaky bucket and hence the input traffic to the ATM network. For a single minisource model, there are two states, "ON" and "OFF" (Figure 3.13). In the OFF state, no cells are transmitted while in the ON state $[13]$ A cells/slot are transmitted. There are $N + 1$ data rate levels and the corresponding state transition diagram is shown in Figure 3.14.

The two-minisource model contains two different minisources. Figure 3.15 (a) represents less bursty traffic and Figure 3.15 (b) represents highly bursty traffic. In the less bursty traffic case, there are $N_1 + 1$ data rate levels and in the high bursty traffic

case, there are $N_2 + 1$ data rate levels. A two-minisource model was first proposed as a model of variable-bit-rate video sources [15] and represents the video source as one which changes among two different fixed-rate levels, a high rate A_h and a low rate A_l , via integer combinations up to a maximum of $N_1 + 1$ low rate levels and $N_2 + 1$ high rate levels. The corresponding state transition diagram for the aggregate source is shown in Figure 3.16. In this thesis, this two-minisource model is applied to characterize the output process of the leaky bucket.

Figure 3.15 Two-minisource model

(a) Less bursty traffic level (b) High bursty traffic level

The parameters (a , b , c and d) in Figure 3.15 are related to the fraction of the average time spent in the low activity level and the average time spent in the high level. Following [15], the characteristics of the two-minisource model are found as follows:

Figure 3.16 State transition diagram for an aggregate source model

1. mean:

$$\bar{\lambda} = N_1 p_a A_l + N_2 p_b A_h. \quad (3.31)$$

$$\text{where } p_a = \frac{a}{a+b}, \quad p_b = \frac{c}{c+d};$$

2. variance:

$$v = A_l^2 p_a (1 - p_a) N_1 + A_h^2 p_b (1 - p_b) N_2. \quad (3.32)$$

3. variance ratio:

$$\gamma = \frac{A_h^2 p_b (1 - p_b) N_2}{A_l^2 p_a (1 - p_a) N_1}, \quad (3.33)$$

4. autocovariance:

$$c_a = A_l^2 p_a (1 - p_a) N_1 \exp(-(a+b)t) + A_h^2 p_b (1 - p_b) N_2 \exp(-(c+d)t), \quad (3.34)$$

5. normalized autocovariance:

$$\begin{aligned} r_{2m} &= \frac{c}{v} \\ &= \frac{A_l^2 p_a (1 - p_a) N_1 \exp(-(a+b)t) + A_h^2 p_b (1 - p_b) N_2 \exp(-(c+d)t)}{A_l^2 p_a (1 - p_a) N_1 + A_h^2 p_b (1 - p_b) N_2} \\ &= \frac{\exp(-xt) + \gamma \exp(-yt)}{1 + \gamma}, \end{aligned} \quad (3.35)$$

$$\text{where } x = a + b, \quad y = c + d.$$

Non-linear optimization can be used to find the values of x , y and γ which match the normalized autocovariance obtained from the exact analysis (see Section 3.2.2). For a given traffic condition (N_1, N_2 and A_l), Equation (3.31) to Equation (3.35) are combined with γ , x , and y to determine a , b , c , d and A_h of the two-minisource model.

It is important to mention that the two-minisource model can be used to evaluate the network performance even under some general input traffic conditions, which will be discussed in later sections.

3.3.2 Network Performance Analysis Using Two-Minisource Model Approach

Using the two-minisource model to approximate the output process of the leaky bucket or the input traffic process to the network, this subsection analyzes the network performance, including the network buffer queue length distribution and cell loss probabilities.

Let μ be the network service rate and $q(t)$ the instantaneous network queue size. Considering the forward transition equations for the joint probability of the source state (input to the network, i.e., output process of the leaky bucket) and network queue size in a steady state, it is shown in [15] that

$$\begin{aligned} \frac{dF_{i,j}(x)}{dx} &= \frac{(N_1 - i + 1)a}{\lambda_{i,j} - \mu} F_{i-1,j}(x) \\ &\quad - \frac{ib + jd + (N_1 - i)a + (N_2 - j)c}{\lambda_{i,j} - \mu} F_{i,j}(x) \\ &\quad + \frac{(i + 1)b}{\lambda_{i,j} - \mu} F_{i+1,j}(x) \\ &\quad + \frac{(N_2 - j + 1)c}{\lambda_{i,j} - \mu} F_{i,j-1}(x) \\ &\quad + \frac{(j + 1)d}{\lambda_{i,j} - \mu} F_{i,j+1}(x) \end{aligned} \quad (3.36)$$

where

$$F_{i,j}(x) = \text{prob}(\text{source is in state } (i, j), \text{ network queue size } \leq x) \quad (3.37)$$

and $\lambda_{i,j} = iA_l + jA_h$. Equation (3.36) can be written as

$$D\dot{F} = HF \quad (3.38)$$

in which D and H are appropriate matrices and F is the vector formed from $F_{i,j}$.

The solution for Equation (3.38) is given by

$$F(x) = F(\infty) + \sum_z a_z \phi^{(z)} e^{zx} \quad (3.39)$$

where z are eigenvalues of $D^{-1}H$ in the left half complex plane and $\phi^{(z)}$ are the corresponding eigenvectors

$$zD\phi^{(z)} = H\phi^{(z)}. \quad (3.40)$$

Note that for the solution of Equation (3.38) to be a probability distribution function, only the left half plane eigenvalues can appear.

To obtain the complete queue distribution, the eigenvalues and eigenvectors of $D^{-1}H$, and the coefficients a_z need to be evaluated. Following the procedure in [15], the eigenvalues and eigenvectors can be obtained. First, the eigenvector for a given eigenvalue is determined. Let $\Phi^{(z)}(u, v)$ denote the generating function of $\phi^{(z)}$,

$$\Phi^{(z)}(u, v) = \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} \phi_{ij}^{(z)} u^i v^j \quad (3.41)$$

Using Equation (3.41), the following equation for the generating function can be obtained

$$\begin{aligned} & \frac{\partial \Phi(u, v)}{\partial u} [-b + (A_l z + b - a)u + au^2] \\ & + \frac{\partial \Phi(u, v)}{\partial v} [-d + (A_h z + d - c)v + cv^2] \\ & = \partial \Phi(u, v) [\mu z - (N_1 a + N_2 c) + N_1 au + N_2 cv] \end{aligned} \quad (3.42)$$

Equation (3.42) is solved by using a separation of variables. Thus letting

$$\Phi(u, v) = h(u)g(v) \quad (3.43)$$

and substituting in Equation (3.42), it is obtained that

$$h(u) = (u - s_1)^{c_1} (u - s_2)^{c_2} \quad (3.44)$$

$$g(v) = (v - s_3)^{c_3}(v - s_4)^{c_4} \quad (3.45)$$

where

$$\begin{aligned} s_1 &= \left\{ -(A_l z + b - a) + [(A_l z + b - a)^2 + 4ab]^{1/2} \right\} / 2a \\ s_2 &= \left\{ -(A_l z + b - a) - [(A_l z + b - a)^2 + 4ab]^{1/2} \right\} / 2a \\ s_3 &= \left\{ -(A_h z + d - c) + [(A_h z + d - c)^2 + 4cd]^{1/2} \right\} / 2c \\ s_4 &= \left\{ -(A_h z + d - c) - [(A_h z + d - c)^2 + 4cd]^{1/2} \right\} / 2c \end{aligned} \quad (3.46)$$

and

$$\begin{aligned} c_1 &= \frac{\mu z/2 + \Gamma - N_1 a(1 - s_1)}{a(s_1 - s_2)}; & c_2 &= N_1 - c_1 \\ c_3 &= \frac{\mu z/2 - \Gamma - N_2 c(1 - s_3)}{c(s_3 - s_4)}; & c_4 &= N_2 - c_3 \end{aligned} \quad (3.47)$$

The eigenvector generating function $\Phi(u, v)$ can be obtained from Equation (3.43) to Equation (3.45), and the eigenvector components can be obtained by comparing this expression to the expansion of Equation (3.41). Given an eigenvalue z , the complete procedure for finding the eigenvector is as following [15].

Compute $s_1 - s_4$ and c_1, c_3 from the above, with Γ chosen to make both c_1 and c_3 integers. Use Equations (3.43) to (3.45) to obtain Φ and Equation (3.40) to get the eigenvector components.

To obtain the eigenvalues, the procedure is as follows. Solving for Γ in the expressions for c_1 and c_3 above and squaring both sides of the resulting equation, a fourth-order equation is obtained

$$T_0 z^4 + T_1 z^3 + T_2 z^2 + T_3 z + T_4 = 0 \quad (3.48)$$

where the coefficients T_0 , T_1 , T_2 , T_3 , and T_4 are all functions of only the model parameters. The eigenvalues are obtained by solving Equation (3.48) for all possible combinations of c_1 and c_3 .

Once the eigenvalues and eigenvectors are obtained, the coefficients a_z of Equation (3.39) have to be evaluated in order to determine the complete network distribution. It is noted that the queue size is nonzero with probability one if the instantaneous input rate is greater than the queue output rate. This gives a set of linear equations

$$F_{i,j}(0) = 0, \quad \text{if } \lambda_{ij} = (iA_l + jA_h) > \mu. \quad (3.49)$$

The buffer overflow probability, or the cell loss probability for given network queue size x is therefore

$$P_{loss} = \text{prob}(\text{queue size} > x) = 1 - \sum_{n,m} F_{nm}(x). \quad (3.50)$$

Equation (3.50) will be used to show the network performance in terms of cell loss probabilities.

3.3.3 Accuracy of the Two-Minisource Model Approach

This subsection presents some comparisons between the results obtained using the theoretical two-minisource model (Equation (3.50)) and simulation results.

3.3.3.1 Bursty Batch Data Traffic

For bursty batch data traffic, the accuracy of the two-minisource model approach is shown in Figure 3.17 to 3.20. In Figures 3.17, 3.18 and 3.19, network cell loss probabilities are presented with leaky bucket control congestion. The analytical two-minisource

model approach and simulation give very similar results. Figure 3.20 compares theoretical and simulation results for a case without leaky bucket control showing good approximation.

3.3.3.2 Other Traffic Types

So far it has been assumed that the input traffic is bursty batch data with Poisson distributed arrivals and exponentially distributed batch size. In the following, a more generalized input traffic model, the Markovian arrival process (MAP) [26] is used. This is a class of semi-Markov processes that can model a broad range of arrival processes, for example, voice, data, and video arrivals [26]. It will be shown that the two-minisource model is also a good approach for such general input traffic sources.

The MAP is a Markov process with a finite number of transient states, m , and one absorbing state; the epoch of absorption constitutes an arrival of a modelled job arrival process. Upon absorption, the MAP immediately restarts in a transient state, and the cycle repeats.

More formally, using the notation in [30], if the MAP is in transient state i , $i = 1, \dots, m$, it will remain in i for a length of time that is exponentially distributed with parameter λ_i . At the end of this period, the process may move to another transient state or may be absorbed and then instantaneously restart in some transient state. Let q_{ij} be the probability that the process goes to transient state j given that it has been in transient state i . Let g_{ij} be the probability that the process is absorbed and then restarts in state j given that it has been in transient state i . Note that,

$$\sum_{j=1, j \neq i}^m q_{ij} + \sum_{j=1}^m g_{ij} = 1, \quad 1 \leq i \leq m. \quad (3.51)$$

Figure 3.17 Network cell loss probability,
 comparison of analytical and simulation results
 (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 3$,
 token interarrival time = 10 simulation time units, user queue length = 20)

Figure 3.18 Network cell loss probability,
 comparison of analytical and simulation results
 (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$,
 token interarrival time = 10 simulation time units, user queue length = ∞ .)

Figure 3.19 Network cell loss probability,
 comparison of analytical and simulation results
 (network load = 67% (10 sources), mean burst length = 6.67 cells, $M = 20$,
 token interarrival time = 1 simulation time units, user queue length = ∞)

Figure 3.20 Network cell loss probability (no leaky bucket control), comparison of analytical and simulation results (network load = 67% (10 sources), mean burst length = 6.67 cells, user queue length = ∞)

Let the matrix $P(n, t)$ denote the counting function whose ij^{th} entry, $P_{ij}(n, t)$, is the probability that the number of job arrivals over the interval $(0, t]$ is n and the state of the MAP at time 0 is i and at time n is j . The z-transform of $P(n, t)$ with respect to n , $P^*(z, t) = \sum_{n=0}^{\infty} P(n, t)z^n$, is equal to

$$P^*(z, t) = e^{(C + zB)t}, \quad |z| \leq 1, \quad t \geq 0. \quad (3.52)$$

where C is an $m \times m$ matrix whose ij^{th} entry equals $\lambda_i q_{ij}$ for $i \neq j$ and equals $-\lambda_i$ for $i = j$, and D is an $m \times m$ matrix whose ij^{th} entry equals $\lambda_i g_{ij}$

Assume that the cell arrival process is an MAP and is independent of the token arrival process. In the following, it is assumed that $m = 2$ and MAP becomes a 2-branch hyperexponential distributed renewal process with

$$C = \begin{bmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{bmatrix}, \quad B = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \cdot [\delta_1 \quad \delta_2] \quad (3.53)$$

where $\delta_1 + \delta_2 = 1$.

Three MAP cases with different coefficient of variation and skewness [26] as shown in Table 3.2 are to be examined.

Table 3.2 Simulation cases of MAP traffic

Case	Coefficient of variation	Skewness
1	2	10
2	2	50
3	3	50

The coefficient of variation is defined as $\sqrt{E(X^2) - (E(X))^2}/E(X)$ and skewness is defined as $E[(X - E(X))^3]/(E(X^2) - (E(X))^2)^{3/2}$, where X is the number of cell arrivals in a time unit.

In order to use the two-minisource model to analyze the network performance such as cell loss probabilities, The two-minisource model parameters (see Section 3.3.1) is determined using the mean, variance and normalized autocovariance which are obtained through simulation. Note that a theoretical expression of the two-minisource model autocovariance (Equation (3.34)) can be obtained using nonlinear optimization and minimum mean squared error criteria.

For the three MAP traffic cases, simulation results are used to confirm the analytical results using the two-minisource model approach. The normalized autocovariance and network cell loss probability curves are presented in Figure 3.21 to Figure 3.26. The network load, mean burst length and leaky bucket parameters used in the simulation are given in each figure and the analytical results are obtained using Equations (3.35) (normalized autocovariance) and (3.50) (cell loss probability). In all cases, a good match is observed between the analytical and simulation results. It is therefore concluded that the two-minisource model is a good way to characterize the leaky bucket output process and to evaluate ATM network performance under quite general input traffic conditions.

3.4 Conclusions

In this chapter, the performance of congestion control using a leaky bucket scheme is analyzed. As summarized in Figures 3.27, 3.28 and 3.29, three steps were taken in the investigation.

In the first step, the exact leaky bucket model was considered and the input traffic was assumed to be batch data with a Poisson arrival process and a geometrically distributed burst length. The mean, variance and autocovariance of the leaky bucket output process were obtained. In particular, it was observed that the normalized autocovariance can be approximated by aggregated two exponential functions.

Figure 3.21 Normalized autocovariance (MAP traffic, Case 1),
 comparison of two-minisource model and simulation results
 (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$,
 token interarrival time = 10 simulation time units, user queue length = ∞)

Figure 3.22 Network cell loss probability (MAP traffic, Case 1), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = ∞ .)

**Figure 3.23 Normalized autocovariance (MAP traffic, Case 2),
comparison of two-minisource model and simulation results
(network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$,
token interarrival time = 10 simulation time units, user queue length = 20)**

Figure 3.24 Network cell loss probability (MAP traffic, Case 2), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)

Figure 3.25 Normalized autocovariance (MAP traffic, Case 3),
 comparison of two-minisource model and simulation results
 (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$,
 token interarrival time = 10 simulation time units, user queue length = 20)

Figure 3.26 Network cell loss probability (MAP traffic, Case 3), comparison of two-minisource model and simulation results (network load = 70% (10 sources), mean burst length = 7 cells, $M = 3$, token interarrival time = 10 simulation time units, user queue length = 20)

Figure 3.27 Summary of the work in Chapter 3, Step 1

Figure 3.28 Summary of the work in Chapter 3, Step 2

Figure 3.29 Summary of the work in Chapter 3, Step 3

In the second step, the output process of the leaky bucket was approximated using a two-minisource model and the network performance in terms of cell loss probabilities was evaluated. The network queue length distribution was also obtained. Simulations were performed to evaluate network cell loss probabilities showing very similar results compared to the analytical approach.

In the third step, more general traffic cases were also considered using the two-minisource model approach. Three MAP traffic cases were simulated presenting network cell loss probabilities. The simulation results confirmed the analytical results using the two-minisource model approach. This demonstrates that the two-minisource model is an important and valuable way to characterize the leaky bucket output process and to evaluate ATM network performance.

The performance of leaky bucket control is expected to be improved by combining with other control mechanisms such as marking and prioritizing. This is the topic for investigation in next chapter.

Chapter 4

Modified Leaky Bucket Control

The basic leaky bucket scheme operates in a strict "two state" mode: transmission with a token and no-transmission without a token. This can be modified by implementing more transmission options such as transmissions with marking and/or prioritizing. This chapter considers those modified approaches. Due to the complexity of the modified leaky bucket schemes, we perform the studies through simulation.

4.1 Simulation Model and Assumptions

Figure 4.1 shows the simulation model to be used in this chapter. Both data and voice traffic are considered. The basic leaky bucket scheme is modified through a control parameter input. The parameter could be an adaptive control, marking, or prioritizing parameter. The output of the modified leaky bucket is delivered to MUX which is connected to an ATM network through a high speed link.

The link transmission capacity is assumed to be 150 Mb/s, which is equivalent to $150 \times 10^6 / (53 \times 8) \approx 2.83 \times 10^5$ cells/second and one cell payload is equal to 48 octets. It is also assumed that one simulation time unit is required to transmitting one cell at peak bandwidth. Thus 1 simulation time unit is equal to $53 \times 8 / (150 \times 10^6) \approx 2.82 \times 10^{-6}$ sec.

For data sources, Poisson batch arrivals with geometrically distributed batch length are considered. The total data traffic load is assumed to be ρ_d . The batch arrival rate

Figure 4.1 Simulation model for modified leaky bucket control schemes

Figure 4.2 Birth-death model for the number of active voice sources

for a data source is

$$\lambda_d = \frac{\rho_d}{(\text{number of sources}) \times (\text{mean batch length})} \quad \text{batch/unit time} \quad (4.1)$$

For example, if the total data traffic load is 50% of the link capacity, the number of sources is 8, and the mean batch length is 10 cells, the batch arrival rate $\lambda_d = 0.5/80 = 6.25 \times 10^{-3}$ batch/unit time.

In general, the arrival process of new voice calls and the distribution of their durations are characterized by a Poisson process and by an exponential distribution respectively. When a number of voice sources are multiplexed, the aggregated voice-call cell arrivals can be modelled by a Markov process (birth-death) [7] as shown in Figure 4.2. The total number of voice sources is assumed to be N_v , a call arrival rate is α_1 and its departure rate is β_1 . N_a is the mean number of active calls.

The voice traffic load ρ_v is:

$$\begin{aligned} \rho_v &= \frac{t_v}{t_v + t_s} \times C_v \times N_v \times \frac{\beta_1}{\alpha_1 + \beta_1} \\ &= \frac{t_v}{t_v + t_s} \times C_v \times N_a \end{aligned} \quad (4.2)$$

where t_v is the mean talkspurt length, t_s the mean silence length, and C_v is the voice speech rate 64 kb/s.

For simplicity, it is assumed that t_r is equal to t_s . The interarrival time of speech cell is $t_{rc} = 53 \times 8 / (64 \times 10^3) = 6.47 \times 10^{-3} \text{ sec} \approx 2343 \text{ simulation time units}$. For example, if the voice traffic load is 30% of the link capacity, the mean number of active voice sources is $N_a = 2 \times \frac{150 \times 10^3}{64 \times 10^3} \times 0.3 = 1405$.

4.2 Leaky Bucket Control

The basic leaky bucket scheme was analyzed and simulated in Chapter 3. In this section, some additional simulation results (Figure 4.3 to Figure 4.7) are presented.

The effect of token interarrival time on the leaky bucket performance is shown in Figure 4.3 in which the network throughput is normalized to the offered traffic load where network throughput is defined as total number of cells departing the network node over total number of cells generated by the sources per slot. With an increase of the token interarrival time, the network throughput increases due to the fact that more cells successfully reach the destinations without suffering congestion. However, the cell loss probabilities may increase at the user queue if it is not large enough. The network throughput presented in Figure 4.3 is normalized to the offered traffic load to show the relation between traffic load and network congestion behavior. Under low traffic conditions, for example, 20% , the normalized throughput is 100% indicating no network congestion and no network cell loss.

A larger mean burst length represents a higher degree of burstiness, which introduces a higher probability of network congestion. This results in lower network throughput. As shown in Figure 4.4, the effect of the mean burst length becomes significant under high traffic conditions. The mean burst length also affects the end-to-end delay performance, which is shown in Figure 4.5.

Figure 4.3 Leaky bucket control, effect of token interarrival time on network throughput (8 sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$)

Figure 4.4 Leaky bucket control, effect of mean burst length on network throughput (8 sources, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units)

Figure 4.5 Leaky bucket control, effect of mean burst length on user end-to-end delay (8 sources, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units)

Figure 4.6 Leaky bucket control, effect of token size on network cell loss probabilities
(network load = 80% (10 sources), mean burst length = 8 cells, user queue length = ∞ , token generation interval = 10 simulation time units)

Figure 4.7 Leaky bucket control, effect of number
of tokens on network cell loss probabilities

(network load = 80% (10 sources), mean burst length = 16 cells, user
queue length = ∞ , token generation interval = 10 simulation time units)

Figures 4.6 and 4.7 demonstrate the effect of leaky bucket control in reducing cell loss probabilities in the network (It is considered that a cell is lost when network size is larger than network buffer size). It is seen that the cell loss probability is substantially reduced with leaky bucket control, especially when the maximum token pool size is very small. Note that the leaky bucket effect decreases with an increase of token size. Of course, an infinite token pool size corresponds to no leaky bucket control. Figure 4.6 and Figure 4.7 assume the same parameters except the mean burst length. In Figure 4.6, the mean burst length is 8 cells, while in Figure 4.7, the mean burst length is 16 cells. The network cell loss probability is higher in the case of larger mean burst length. However, with a higher degree of leaky bucket control (smaller token size), the burst length effect diminishes. The two figures show similar network cell loss probabilities when the token size is 3 or 0.

4.3 Adaptive Leaky Bucket Control

The maximum token pool size and token generation rate are fixed in the basic leaky bucket scheme. In the adaptive leaky bucket control, the maximum token pool size and token generation rate may vary depending on the network traffic load. If the traffic load is very low, the token pool size and/or token generation rate can be increased. In the case of high traffic load (congestion state), the token pool size and/or token generation rate should be reduced. The network traffic load can be determined by monitoring the network buffer occupancy. Clearly, lower network buffer occupancy corresponds to lower traffic load. The network sends a traffic load status message to end users as a control parameter to the leaky bucket. The end users adjust the leaky bucket (maximum token pool size and token generation rate) accordingly.

The simulation results of adaptive leaky bucket control are given in Figure 4.8 and Figure 4.9. Only the maximum token pool size varies following the traffic load status message (control parameter). Two network buffer occupancy thresholds, 40% and 80%, are chosen. The initial token pool size is assumed to be 50. In each adaptation instant, the token pool size is reduced by half if the network buffer occupancy is larger than 80% and the minimum token pool size is 1. If the network buffer occupancy is smaller than 40%, the token pool size is increased by 1 and the maximum token size is 50. The token pool size keeps the same if the network buffer occupancy is between 40% and 80%. Figure 4.8 shows two curves with different adaptive control intervals (adaptation intervals). It is shown that a smaller adaptation interval is desirable in order to achieve higher network throughput. However, this implies a higher control message processing load. There is thus a trade-off between network performance and processing load. Figure 4.9 compares the adaptive leaky bucket control with basic leaky bucket control and a case without leaky bucket control. Substantial performance improvement in cell loss probability is achieved with adaptive leaky bucket control.

4.4 Combined Leaky Bucket and Marking Scheme

In ATM networks, input traffic changes during a call and it might exceed the allocated bandwidth. In considering bandwidth allocation and congestion control, it is necessary to investigate how violating cells (i.e., incoming cells which exceed the allocated bandwidth) should be handled. In preceding studies, the violating cell issue is not considered. That is to say, all incoming cells are treated as non-violating cells.

In this subsection, two approaches dealing with the violating cells are examined. One approach is that violating cells are discarded locally at the user queue [31]. In other words, a violating cell is discarded before it enters a (leaky bucket) user queue.

Figure 4.8 Adaptive leaky bucket control, network throughput (8 data sources, mean burst length = 10 cells, user queue length = ∞ , network buffer size = 25 cells, initial token size = 50, token generation interval = 8 simulation time units)

2 OF 2

PM-1 3 1/2" x 4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

Figure 4.9 Adaptive leaky bucket control, network cell loss probabilities
 (network load = 95% (10 sources), mean burst length
 = 9.5 cells, user queue length = ∞ , initial token size =
 50, token generation interval = 10 simulation time units)

A different approach is that a violating cell is marked and admitted into the user queue. The marked cell is transmitted later upon obtaining a token. This is the combined leaky bucket and marking scheme [19, 31, 32]. The violating cell is transmitted into the network with marking (a violation tag) in its cell header and is discarded upon arriving at a congested node. If the node becomes congested, the violating cell could also be discarded from a network queue when a normal cell arrives. If there are no congested nodes along the routes, the violating cell is successfully transmitted without being discarded. Clearly, in the combined leaky bucket and marking scheme, congestion control is performed in a leaky bucket through a token generator and in a congested node through discarding violating cells. Marking is performed at the user end before the user queue.

As discussed above, violating cells are those exceeding the allocated bandwidth. In the simulation, a (batch data) interarrival time threshold is set and incoming cells are considered to be violating cells when the interarrival time exceeds the threshold. The simulation results of the combined leaky bucket and marking scheme are given in Figure 4.10 and Figure 4.11 together with the results of discarding instead of marking. Both figures indicate that network throughput is increased when leaky bucket control is combined with the marking method. Compared to Figure 4.10, Figure 4.11 shows higher network throughput since the network buffer size is larger.

4.5 Combined Leaky Bucket and Priority Mechanism

Cells may be discarded at a network node when the node becomes congested. With cell prioritizing, low-priority cells are discarded before high-priority cells. This is particularly useful in ATM networks to provide multiple grades of services to different service classes.

Figure 4.10 Combined leaky bucket and marking scheme, network throughput (8 data sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units, small network buffer)

Figure 4.11 Combined leaky bucket and marking scheme, network throughput (8 data sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units, large network buffer)

Figures 4.12, 4.13 and 4.14 show the effect of the priority mechanism in leaky bucket control assuming that data traffic has high priority over voice traffic with respect to the discarding. Figure 4.12 compares two modified leaky bucket schemes, one with marking method only and another with both marking method and priority mechanism. It shows that the leaky bucket scheme with both marking method and priority mechanism results in a lower data cell loss probability. Figure 4.13 also compares two modified leaky bucket schemes, one with adaptive control only and another with both adaptive control and priority mechanism. The case with both marking method and priority mechanism shows a lower data cell loss probability. However, as shown in Figure 4.14, the voice cell loss probability is increased due to the priority mechanism. Clearly, the performance improvement of high-priority services is achieved at the expense of low-priority services.

4.6 Summary of Leaky Bucket Control

This chapter has presented the simulation results of the basic leaky bucket control scheme and various modified leaky bucket control schemes. For the basic leaky bucket scheme, the effects of token interarrival time, token size and traffic burst length on the network performance are examined in terms of network throughput, user end-to-end delay and network cell loss probabilities.

Three modified leaky bucket control schemes were considered. As summarized in Figure 4.15, an adaptive leaky bucket control scheme was obtained by varying the token pool size of the basic leaky bucket scheme adaptively according to a control parameter. Marking and transmitting violation cells gave a combined leaky bucket and marking scheme. Finally, a combined leaky bucket and priority scheme was based on giving higher priority to data cells over voice cells with respect to discarding when congestion occurs.

Figure 4.12 Leaky bucket with marking and priority mechanism, network data cell loss probabilities
 (8 data sources, mean burst length = 10 cells, user queue length = ∞ , $M = 10$, token generation interval = 8 simulation time units)

Figure 4.13 Adaptive leaky bucket control with priority mechanism, network data cell loss probabilities
 (8 data sources, mean burst length = 10 cells, user queue length = ∞ , initial token size = 50, token generation interval = 8 simulation time units)

Figure 4.14 Adaptive leaky bucket control with priority mechanism, network voice cell loss probabilities
 (8 data sources, mean burst length = 10 cells, user queue length = ∞ , initial token size = 50, token generation interval = 8 simulation time units)

Figure 4.15 Summary of modified leaky bucket control schemes

Generally speaking, modified leaky bucket control schemes outperform the basic leaky bucket control scheme at the expense of extra processing load. Further investigations are required to reach a leaky bucket control architecture with enhanced performance and, at the same time, with a simple leaky bucket operation mechanism.

Chapter 5

Conclusions

5.1 Conclusions

This thesis addressed the congestion control issue in ATM networks and concentrated on leaky bucket congestion control techniques.

Figure 5.1 summarizes the flow of work of this thesis. Through analytical and simulation approaches, the basic leaky bucket control scheme were investigated. Modified leaky bucket schemes were then examined through simulation.

Previous work on leaky bucket control assumed a Poisson cell arrival process. However, a simple Poisson cell arrival process cannot model the bursty traffic found in ATM networks since batch arrivals occur frequently. Due to the bursty traffic nature in ATM networks, batch traffic arrivals need to be considered in evaluating congestion control performance. The leaky bucket performance was analyzed assuming batch traffic with a Poisson arrival process and a geometrically distributed batch length. For more generalized input traffic, a Markovian arrival process model was used which characterizes a wide range of arrival processes. Assuming a batch traffic arrival, the mean, variance and autocovariance of the basic leaky bucket output process was derived. It is shown that the normalized autocovariance can be well approximated by aggregated two exponential functions.

In order to evaluate the performance of ATM networks with leaky bucket control, a two-minisource model was proposed to approximate the leaky bucket output process.

Figure 5.1 Summary of the work in this thesis

Simulations were performed to validate the two-minisource model showing that the approximation is reasonably accurate. The network queue length distribution was also derived. Using the two-minisource model, the network performance in terms of cell loss probabilities was analyzed.

More generalized traffic cases are also examined using the two-minisource model approach. Three MAP traffic cases were simulated and network cell loss probabilities were obtained. The simulation results confirm the analytical results using the two-minisource model. Thus it demonstrates that the two-minisource modelling is an accurate method to characterize the leaky bucket output process and to evaluate ATM network performance.

Extensive simulation was performed to investigate the performance of the basic leaky bucket scheme further. The leaky bucket control and the conventional acknowledgement-based control were compared. The effects of leaky bucket parameters (token interarrival time and token size) and traffic parameters (mean burst length) were observed. Network performance was presented in terms of network throughput, network cell loss probability and user end-to-end delay.

It was demonstrated that the performance (network throughput and cell loss probabilities) of the basic leaky bucket scheme can be improved by incorporating with other congestion control mechanisms. The simulated schemes include adaptive leaky bucket control (varying the token pool size adaptively), combined leaky bucket and marking scheme (marking and transmitting violation cells), and combined leaky bucket and priority mechanism (giving higher priority to data cells over voice cells).

5.2 Suggestions for Further Investigations

To conclude this thesis, a number of congestion control research topics for further

investigation are recommended.

- ***Optimum Token Generation Rate:*** In the basic leaky bucket scheme, tokens are generated at a fixed rate. The leaky bucket performance depends on the token generation rate. It is desirable to find an optimum rate for a given arriving call.

- ***Congestion Period Behavior:*** The network performance in a steady state is analyzed in this thesis. An ATM network operates dynamically between congestion and non-congestion even with congestion control. It is therefore necessary to look into the network performance during the congestion period.

- ***Error Control Effect on Congestion Control:*** Error control is used for recovering cell/packet errors or losses due to transmission media noise, cell misrouting, and/or cell dropping for congestion control. It is considered that the cell loss because of congestion or congestion control may be recovered through error control. The error control effect on congestion control is worthy to be investigated.

References

- [1] R. Handel, "Evolution of ISDN towards broadband ISDN," *IEEE Network Magazine*, pp. 7–13, Jan. 1989.
- [2] V. O. K. Li, J. Chang, K. Lee, and T. Yang, "A survey of research and standards in high-speed networks," *Int. J. of Digital and Analog Commun. Systems*, vol. 4, pp. 269–309, 1991.
- [3] L. Kleinrock, "ISDN - The path to broadband networks," *Proc. of IEEE*, vol. 79, no. 2, pp. 112–117, Feb. 1991.
- [4] S. E. Minzer, "Broadband ISDN and asynchronous transfer mode (ATM)," *IEEE Communications Magazine*, pp. 17–24, Sept. 1989.
- [5] M. DePrycker, *Asynchronous Transfer Mode*. Chichester, U.K.: Ellis Horwood, Simon and Schuster Int'l., 1991.
- [6] M. Sidi, W. Liu, I. Cidon, and I. Gopal, "Congestion Control Through Input Rate Regulation," in *Proc. IEEE GLOBECOM'89*, pp. 1764–1768, 1989.
- [7] J. J. Bae and T. Suda, "Survey of traffic control schemes and protocols in ATM networks," *Proceedings of the IEEE*, vol. 79, no. 2, pp. 170–189, Feb. 1991.
- [8] D. Hong and T. Suda, "Congestion control and prevention in ATM networks," *IEEE Network Magazine*, pp. 10–16, July 1991.
- [9] T. Kamitake and T. Suda, "Evaluation of an admission-control scheme for an ATM network considering fluctuation in cell loss rate," in *Proc. IEEE GLOBECOM'89*, pp. 1774–1780, 1989.

- [10]H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Selec. Areas Commun.*, vol. SAC-4, no. 6, pp. 856–868, Sept. 1986.
- [11]M. Murata, Y. Oie, T. Suda, and H. Miyahara, "Analysis of a discrete-time single-server queue with bursty inputs for traffic control in ATM networks," in *Proc. IEEE GLOBECOM'89*, pp. 1781–1787, 1989.
- [12]K. Q. Liao and L. G. Mason, "A discrete-time single server queue with a two-level modulated input and its applications," in *Proc. IEEE GLOBECOM'89*, pp. 26.1.1–26.1.6, 1989.
- [13]B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," *IEEE Trans. on Commun.*, vol. 36, no. 7, pp. 834–844, July 1988.
- [14]M. Nomura, T. Fujii, and N. Ohta, "Basic characteristics of variable rate video coding in the ATM environment," *IEEE J. Selec. Areas Commun.*, vol. 7, no. 5, pp. 752–760, June 1989.
- [15]P. Sen, B. Maglaris, N. Rikli, and D. Anastassiou, "Basic characteristics of variable rate video coding in the ATM environment," *IEEE J. Selec. Areas Commun.*, vol. 7, no. 5, pp. 865–869, June 1989.
- [16]C. Rasmussen, J. H. Sorensen, K. S. Kvols, and S. B. Jacobsen, "Source-independent call acceptance procedure in ATM networks," *IEEE Trans. on Commun.*, vol. 9, pp. 351–358, 1991.
- [17]J. Holtzman, "Coping with broadband traffic uncertainties: statistical uncertainty, fuzziness, neural networks," in *Proc. IEEE GLOBECOM'90*, pp. 7–11, 1990.
- [18]K. S. Meier-Hellstern and P. E. Wirth, "Packet switch engineering procedures when customer characteristics are uncertain," *IEEE J. Selec. Areas Commun.*, vol. SAC-9,

pp. 142–150, 1991.

- [19]G. Gallassi, G. Rigolio, and L. Fratta, "ATM: Bandwidth assignment and bandwidth enforcement policies," in *Proc. IEEE GLOBECOM'89*, pp. 1788–1793, 1989.
- [20]K. Sohraby and M. Sidi, "On the performance of bursty and correlated sources subject to leaky bucket rate-based access control schemes," in *Proc. IEEE GLOBECOM'89*, pp. 426–434, 1991.
- [21]S. J. Golestani, "Congestion-free transmission of real-time traffic in packet networks," in *Proc. INFOCOM'90*, pp. 527–536, 1990.
- [22]R. S. Dighe, C. J. May, and G. Ramamurthy, "Congestion avoidance strategies in broadband packet networks," in *Proc. INFOCOM'91*, pp. 295–303, 1991.
- [23]H. Kroner, G. Hebuterne, and A. Gravey, "Priority management in ATM switching nodes," *IEEE J. Selec. Areas Commun.*, vol. SAC-9, no. 3, pp. 418–427, Apr. 1991.
- [24]Y. Lim and J. E. Kobza, "Analysis of delay-dependent priority discipline in an Integrated multiclass traffic fast packet switch," *IEEE Trans. Commun.*, vol. COM-38, no. 5, pp. 659–665, May 1990.
- [25]E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Select. Areas Commun.*, vol. SAC-9, no. 3, pp. 325–334, Apr. 1991.
- [26]A. W. Berger, "Performance analysis of a rate control throttle where tokens and jobs queue," in *Proc. IEEE INFOCOM'90*, pp. 30–38, 1990.
- [27]J. S. Turner, "New directions in communications (or which way to the information age?)," *IEEE Commun. Magazine*, vol. 25, pp. 8–15, Oct. 1986.
- [28]I. Cidon and I. S. Gopal, "PARIS: An approach to integrated high-speed private networks," *Int. J. of Digital and Analog Cabled Systems*, vol. 1, pp. 77–86, Apr. 1988.

- [29]L. Kleinrock, *Queueing Systems, Volume 1: Theory*. John Wiley and Sons, 1975.
- [30]D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts, "A single server queue with server vacations and a class non-renewal arrival process," *Advances in Applied Probability*, Sept. 1990.
- [31]M. Hirano and N. Watanabe, "Traffic characteristics and a congestion control scheme for an ATM network," *Int. J. of Digital and Analog Commun. Systems*, vol. 3, pp. 211–217, 1990.
- [32]M. Hirano and N. Watanabe, "Characteristics of a cell multiplexer for bursty ATM traffic," in *Proc. IEEE ICC'89*, pp. 399–403, 1989.

END

15-1 2-93

FIN