A Performance Analysis of a Modified Leaky Bucket Mechanism

Zhonghui Liang

A Thesis
in
The Department
of
Electrical and Computer Engineering

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

August 2001

0-612-64059-0

ABSTRACT

A Performance Analysis of a Modified Leaky Bucket Mechanism

Zhonghui Liang

ATM is universally accepted as the transfer mode of choice for B-ISDN, because it is the only standard technology that allows the integration of different services with diversified characteristics and requirements. Congestion control becomes extremely important in ATM networks for achieving a desired level of performance, QoS (Quality of Service). Due to the high link speed, preventive congestion control is more effective and widely used than reactive control. It is usually implemented in two ways: Connection Admission Control (CAC) and Usage Paramenter Control (UPC). Leaky bucket (LB) is a typically used UPC mechanism. A traditional LB can either police a bursty source's Peak Cell Rate (PCR) or Average Cell Rate (ACR) but not both of them simultaneously, meanwhile, it introduces cell delays by buffering cells before they are transmitted into the network.

In this thesis, we propose a modified Leaky Bucket mechanism, in which a single LB can police both the PCR and ACR simultaneously. The first characteristic of the modified LB is that the token generator is designed to be dynamic which alternates between On and Off states, tokens will be generated during the token generator's On period at a speed equalling to the policed bursty source's PCR.

The second characteristic of this modified LB is that no token pool or cell buffer is utilized. Therefore, cells generated by a bursty source can be transmitted into the network at its PCR only when the token generator is simultaneously in On state. Cells can either be transmitted or discarded, no cell delay occurs. As a result the source's PCR cannot be exceed the token generator's PCR. By properly designing the statistical distributions of the dynamic token generator's On and Off periods, the modified LB can also police a bursty source's ACR. Furthermore, the modified LB doesn't introduce any cell delay so it is more appropriate for real-time traffics than the traditional LB.

Since most bursty sources' On and Off periods can be modeled as geometrically distributed, the dynamic token generator's On and Off periods are also designed to be geometrically distributed. We study three protocols that may be used to control the operation of the modified LB. These protocols are called forced, unforced and compensation protocol. We derive the Cell Loss Probability for each of these protocols, and compare the performance of the modified LB against the traditional LB, in terms of the performance measure of cell delay and cell loss probability.

Acknowledgments

I would like to express my deepest appreciation to my thesis supervisors, Professor M. Mehmet Ali, whose support, invaluable guidance, constructive criticism and encouragement has been of considerable help throughout the course of this work. He introduced me to the various aspects in the performance analysis of communications networks. I also would like to thank Professors Jeremiah. F. Hayes and E. Plotkin for stimulating my interest and teaching me in stochastics processes and queueing theory.

I also would like to thank my family for their sacrifices, love and act of support. This thesis would have never been completed without their encouragement and full support.

Thanks also to all my friends in Concordia and my colleagues in Nortel for their constant moral support and for sharing with me a challenging experience.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

B-ISDN:	Broadband Integrated Service Digital Network
ATM:	Asynchronous Transfer Mode
ATDM:	Asynchronous Time-Division Multiplexing
STM:	Synchronous Transfer Mode
STDM:	Synchronous Time-Division Multiplexing
VPI:	Virtual Path Identification
VCI:	Virtual Channel Identification
PT:	Payload Type
CLP:	Cell Loss Probability
HEC:	Head Error Control
GFC:	Generic Flow Control
SONET:	Synchronous Optical Network
SDH:	Synchronous Digital Hierarchy
QoS:	Quality of Service
VOIP:	Voice Over IP
IP:	Internet Protocol
PCR:	Peak Cell Rate
ACR:	Average Cell Rate
SCR:	Sustain Cell Rate

MBS:	Maximum Burst Size
UPC:	Usage Parameter Control
CAC:	Connection Admission Control
LB:	Leaky Bucket
FCFS:	First Come First Served
PGF:	Probability Generating Function

List of Symbols

α_s: Probability of the source will be On in the next time slot given that it is On in the current slot

β_s: Probability of the source will be Off in the next time slot given that it is Off in the current slot

k_s: Number of time slots that the source remains active during a burst

l_s: Number of time slots that the source remains passive during each visit to the Off state

α_t: Probability of the token generator will be On in the next time slot given that it is On in the current slot

β_t: Probability of the token generator will be Off in the next time slot given that it is Off in the current slot

k_t: Number of time slots that the token generator remains active during each visit to its On state

l_t: Number of time slots that the token generator remains passive during each visit to its Off state

i_k: Number of cells generated by the end of slot k

a_k: State of the bursty source during slot *k* (it assumes the value 0 if the source is Off and 1 if the source is On)

b_k: Number of cells generated during slot *k*

f_k: Nnumber of cells generated by an active source during slot *k*

c: $$c = \begin{cases} 1 & \text{if the source remains active in the next slot} \\ & \text{given that it is active in the present slot} \\ 0 & \text{otherwise} \end{cases}$$

d: $$\begin{cases} 1 & \text{if the source becomes active in the next} \\ & \text{given that it is inactive in the present} \\ 0 & \text{otherwise} \end{cases}$$

$Q_k(z, y)$: Transient joint PGF of the number of cells generated by the end of slot k and the state of the bursty source during slot k

$P_k(z)$: Transient marginal PGF of the number of cells generated by a bursty source by the end of *k*th slot

$A_k(y)$: Transient marginal PGF of the state of the source at the end of *k*th slot

$\ddot{N}_{AB}$: Number of cells generated by the source during the interval A to B

$\overline{\ddot{N}_{AB}}$: Average number of $\ddot{N}_{AB}$

$P_{l_t}(z)$: Transient marginal PGF of the number of cells generated by the end of

l_t slot during the token generator's Off period

$A_{l_t}(y)$: Transient marginal PGF of the source's state at the end of l_t'th slot during the token generator's Off period

$P_{k_t}(z)$: Transient marginal PGF of the number of cells generated by the end of k_t slot during the token generator's tagged period

$A_{k_t}(y)$: Transient marginal PGF of the source's state at the end of k_t'th slot during the token generator's tagged period

N_{BG}: Number of cells successfully transmitted into the network during the source's tagged period

$\ddot{N}_{BG}$: Number of cells lost during the source's tagged period

$Pr(k_s > k_t)$:Probability of the source's tagged period is longer than the token generator's tagged period

$\overline{N_{on}}$: Average number of cells transmitted into the network during a random cycle

$\overline{N_{off}}$: Average number of cells lost during a random cycle

P_{loss} : Cell loss probability during a random cycle

$Pr(\mathrm{On})$: Probability of the source is On when the token generator's tagged period ends

$Pr(\mathrm{Off})$: Probability of the source is Off when the token generator's tagged period ends

ACR: Output ACR from the modified Leaky Bucket during a random cycle

$\bar{R}$: Average duration of a random cycle in number of slots

$\bar{i}$: Average number of tokens wasted in the first random cycle

γ: Source utilization

L: Average number of cells generated during an On period

ρ: Leaky bucket load. Here:

B: Token pool size

M: Cell buffer size

ξ: $= -\frac{\rho(1-\rho)}{L(1-\gamma)(\rho-\gamma}$

CHAPTER 1:

Introduction

Today's telecommunication technologies are developing at an ever fast speed. Optical networks, wireless, internet, and ecommerce, etc., all these products are utilizing new technologies and allow communication networks to support many new services, most of which we could not even imagine ten years ago. These new services not only change the society and the business, but also change the way we work and live. Speed, capacity, and flexibility provided by modern communication networks allow people to communicate to anywhere around the world, in anyway they want. Global Village becomes reality with the modern communication networks.

There are different ways and steps leading to today's highly developed communications. But the entire process can be said to be an evolution to a more direct and natural form of communication.

1.1. Evolution toward Broadband Integrated Service Digital Network (B-ISDN)

1.1.1 Integrated Service Digital Network (ISDN)

ISDN is an integrated services digital communication network that provides various types of services in an integrated fashion. In other words, it can support a wide range of voice and non-voice applications in the same network. ISDN is based on 64-kbps switching technology and is intended to support basic voice facilities and low quality video and data services.

1.1.2 The Concept of B-ISDN

As social and business activities become more diversified, the demand for various multimedia and broadband services increases more rapidly. It has also brought out a diverse set of broadband services such as video conference, high-resolution data and picture transmission, video surveillance, video retrieval services, and broadband videotex.

Among these services, the interactive and distributive services, the circuit-mode and packet-mode services are mixed together, each could develop a separate communication network of its own. However, the construction of an independent service-oriented network imposes a heavy financial burden and can bring about the obstruction of communication information transfer and disorder in the administration of the communication networks. Therefore, it is desired to integrate various networks into one universal network so that all the services can be provided in an integrated manner. The B-ISDN is such an integrated service network which expands the already standardized narrowband ISDN ([1], [2]).

Conceptually, B-ISDN can be described as a digital communication network that utilizes broadband transmission and switching technologies to interconnect concentrated and distributed subscribers and service providers, and to support integrated services with wide-bandwidth distribution that ranges from a few bits per second to several gigabits per second.

1.1.3 The Characteristics of B-ISDN

The primary goal of the B-ISDN is the accommodation of all existing services along with those that will come into being in the communication networks of the future. These may include services of all types and characteristics.

The most notable characteristic of B-ISDN services is that their bandwidth distribution is extremely wide, from 64 kbps voice signal to video signals and high-speed data signals which require hundreds of megabits rate. The service time also exhibits a wide-ranging distribution from a few seconds to hours. In its wide sense, the term *broad* in B-ISDN means that the frequency (bit rate) distribution and service time distribution are both widely spread over the broad band.

Another distinct feature is that continuous signals such as voice and video co-exist with bursty signals such as terminal data. The former requires a constant bit rate while the latter requires a widely varied bit rate. On the other hand, voice and video signals require real-time processing, which is not critical for data services.

Such diversities in B-ISDN services make the switching and transmission a real challenge. Because packet switching is more appropriate for low-rate or bursty signals, whereas circuit switching is more appropriate for continues signals such as voice or video. On the other hand, time-division is more efficient for voice, while space-division is more efficient for high-speed video signals. It is difficult to find a way that enables the switching and transmission of various low-speed and high-speed signals along with the continuous and bursty signals.

1.2. Asynchronous Transfer Mode

The transfer mode defines how information supplied by network users is eventually mapped onto the physical network.

Asynchronous Transfer Mode (ATM) is ITU-T's transfer mode of choice for B-ISDN. In simple terms, ATM is a connection-oriented, packet-switching and multiplexing technique with a particular form that employs ATDM (Asynchronous Time Division Multiplexing), ([3], [6], [7]). Information is transferred through a continues flow of packets of fixed size called ATM cells in a B-ISDN. Consequently, service information is first curtailed to a fixed size, then mapped into the ATM cells, and subsequently goes through the ATDM (Asynchronization Time-Division Multiplexing) procedure with other ATM cells, thus forming the B-ISDN internal transmission signal.

1.2.1 Asynchronous Time-Division Multiplexing

ATDM is a type of statistical multiplexing technique that time-division-multiplexes mutually asynchronous ATM cells from several different channels. It stores each of the incoming low-speed signals inside a buffer, then retrieves and inserts the stored signals one by one into multiplexing slots according to a priority-scheduling principle.

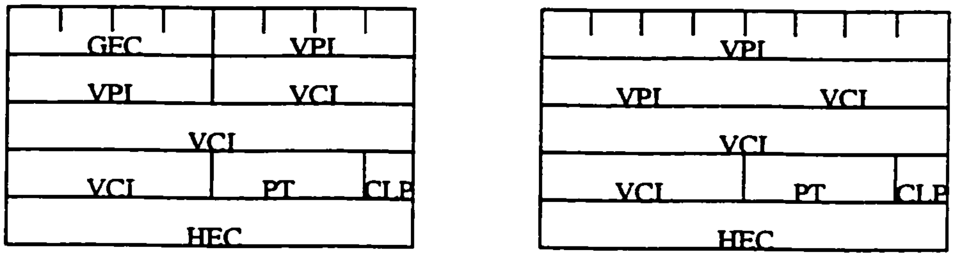
ATM is superior to STM (Synchronous Transfer Mode) since it achieves higher channel utilization factor. STM assigns a dedicated channel to each of the incoming service signals, therefore even when the channel is idle it cannot be used to transmit other information through it. On the other hand, there is no dedicated channel allocation in ATM, an idle channel can be used to transmit any incoming signals, resulting in a higher channel utilization factor.

1.2.2 ATM cells

Some of the design objectives used to define ATM include: integrating voice, audio, video, image and data services; minimizing the switching complexity, and the buffering needed at intermediate nodes mainly to keep up with the high-speed transmission links in the network. These design objectives are met at high transmission speeds by keeping the ATM cell short in length and fixed. The short cell length provides a great deal of flexibility in the way the bandwidth can be used. This flexibility in turn provides the basic framework for B-ISDN to support a wide range of services that emerging applications require. Moreover, the use of short, length-fixed cells fosters the efficient use of the transmission buffer and bandwidth through statistical multiplexing.

An ATM cell consists of 53 bytes, which is divided into 5 bytes of header which contains network control information and 48 bytes of payload space which carries user information. The main function of the cell header is to identify those that are a part of the same virtual channel among the ATM cells existing inside the ATDM information flow. The user data in cell payload is transported transparently by the ATM network and no processing is performed on this field.

The functions of the ATM cell header are Virtual Path Identification (VPI) and Virtual Channel Identification (VCI) in figure 1.1. Other cell header functions include identifying Payload Types (PT), indicating Cell Loss Priority (CLP), and providing Header Error Control (HEC). At the UNI, the cell header additionally provides the Generic Flow Control (GFC) function.



(a) UNI

(b) NNI

Figure 1.1. ATM Cell Header Structure

1.2.3 The Characteristics of ATM

The short cell size of ATM at high transmission rates is expected to offer full bandwidth flexibility and provide the basic framework for guaranteeing the quality of service requirements of applications with a wide range of performance metrics, while allowing statistical multiplexing. The term *statistical multiplexing* refers to the fact that several variable-bit-rate connections can share a link with a capacity less than the sum of their peak bit rate requirements, whereas the term *asynchronous* is used to reflect the fact that the cells of an information unit may appear at irregular intervals over the network links.

On the other hand, although it can and will support connectionless data transfer, ATM is a connection-oriented technique in that end-to-end paths are required to be established prior to the beginning of the information transfer. The packet-oriented nature of ATM is well suited to applications with bursty traffic characteristics. ATM can carry constant-bit-rate traffic equally well. Hence, ATM is an attempt to combine the properties of both the packet- and circuit-switched networks in an integrated network.

The universal acceptance of ATM among every segment of the networking industry comes from the fact that it is a compromise, that allows the integration of different services with different characteristics and requirements. The 53 bytes cell size is a compromise between the conflicting requirements of voice and data applications. A small packet size is preferred for voice since the packets must be delivered with only a short delay. A large packet size is preferable for data since the overheads involved in large packets are smaller. ATM does not handle voice as efficiently as a synchronous network does; it does not handle video as easily as isynchronous transfer; it does not handle data as efficiently as a packet-switching technology with variable-length packets either. But ATM is the only standard technology that handle all of them in an integrated manner.

ATM offers the potential to standardize in one network architecture defining the multiplexing and switching method, with SONET/STM providing the basis for the physical transmission standard for very high-speed rates.

1.2.4. Comparison of ATM with IP

ATM can provide QoS for diversified services, this is a distinguished characteristic from IP (Internet Portocol). Since IP cannot guarantee QoS, it cannot provide satisfying service for some applications, for example voice. There is a lot of research work on VOIP (Voice over IP) recently, however, it makes IP too complicated to keep its advantage of simplicity.

Moreover, the fixed short size of ATM cells makes the high speed switch easier such to provide a good delay performance, this is critical for real time services, including voice.

1.3. Traffic Modeling in ATM networks

ATM networks are expected to support a diverse set of applications with a wide range of characteristics. Unfortunately, for the time being, there are no comprehensive measurements to permit designers to satisfactorily address the characteristics of various types of B-ISDN applications in a realistically accurate manner.

In general, traffic corresponds to one of the three broad categories: voice, video and data. In ATM networks, many sources are visualized as bursty, the cell generation process of bursty sources is a succession of active and silent (idle) periods. Cells are only generated at the peak cell rate (PCR) during the active period. A group of successive cells that are not interrupted by idle period is called burst. The most prominent source model exhibiting this behavior is the popular On-Off model. A set of three important parameters that are used to characterize ATM bursty sources is given below:

- PCR (Peak Cell Rate): the maximum transmission cell rate, i.e. the transmission rate during the active period

- SCR (Sustained Cell Rate): the maximum average transmission cell rate
- MBS (Maximum Burst Size): the maximum number of cells that can be sent at the peak rate.

The On-Off source model shown in figure 1.2 is generally used for modeling VBR traffic in ATM networks. It is used not only for packetized voice models, but also for still video, and data messaging applications. Cells are generated periodically during a burst. In the discrete time case, the number of cells per burst is assumed to have a geometric distribution.

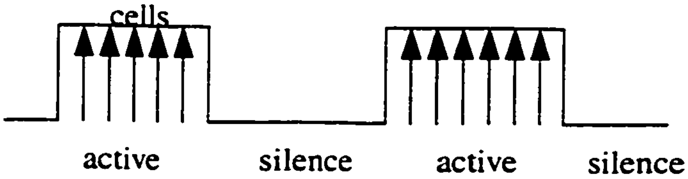


Figure 1.2. On-Off source behavior

1.4. Congestion Control in ATM networks

ATM networks uses the statistically multiplexing to put a number of diverse traffic sources on an ATM link, this introduces considerable flexibility and potential saving of network resources. For instance, a bursty source only generates cells at its peak rate during the active period, therefore it will not be necessary to allocate a continues bandwidth at its peak rate. Statistically multiplexing a large number of such bursty sources can gain bandwidth efficiency. But if the network is not properly designed and controlled in terms of the number of services in different classes it can support, severe network congestion might happen due to the dynamic characteristic of bursty sources, this may cause extensive transient cell loss, cell delay, jitter and deteriorate the performance of the network. Therefore, congestion control is very important in ATM networks in order to provide a desired level of performance, i.e. QoS.

Due to the high link speed, diverse service requirements and various characteristics of ATM traffic, congestion control in ATM networks is extremely difficult and has become one of the active research field in recent years, ([23], [24], [25], [28], [29], [30], [31]). Different congestion control mechanisms fall into two categories: reactive and preventive congestion control.

1.4.1 Reactive Congestion Control

Reactive congestion control mechanism only reacts to the congestion after it has occurred. It uses the feedback information received from the network to adjust the input rate of the sources. But today's very high speed transmission channel increases the ratio of propagation delay to cell transmission time. When congestion happens and the network sends information to the input node, by the time feedback reaches the input node and the control mechanism is triggered, too many cells would have been transmitted into the network, and this will worsen the congestion situation. The reactive congestion control is generally not accepted as effective in ATM networks.

1.4.2 Preventive Congestion Control

Instead, preventive congestion control is more effective and widely used in ATM networks. It does not wait until congestion actually occurs, but rather tries to prevent the network from reaching an unacceptable level of congestion.

It is performed in two ways: Connection Admission Control (CAC) and Usage Parameter Control (UPC). CAC determines whether to accept or reject a new connection at the time of call setup; the decision is based on the traffic characteristics and the network load. However, admission control alone is not sufficient to avoid network congestions, because the user may violate the parameters negotiated at the call setup phase either deliberately or unintentionally. So, after a connection is accepted, Usage Parameter Control is required to monitor and police the traffic to ensure that its parameters conform to what has been negotiated, and the QOS of other connections in the network is not affected by one violating connection.

1.4.3 Leaky-Bucket

Leaky-Bucket (LB) algorithm is a typically used UPC mechanism. Though variations exist, the basic operation of LB is simple and shown in figure1.3.

There are two basic parameters: the token generation rate *R* and the token pool size *B*, the buffer size *M* is another parameter for buffered LB. Arriving cells first enter the cell buffer, if there are tokens in the token pool, cells leave the LB

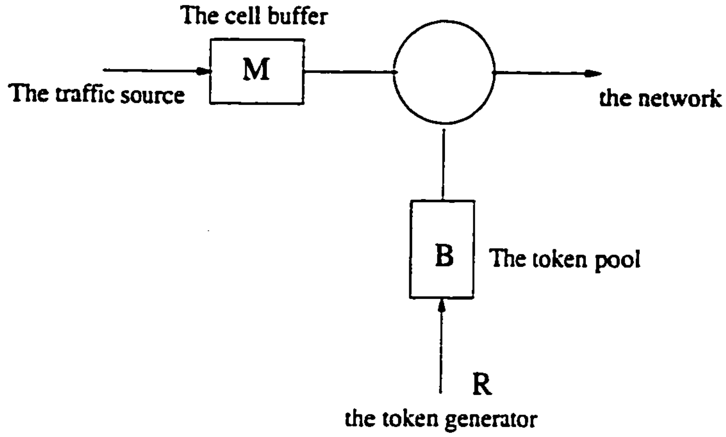


Figure 1.3: The Conventional LB Mechanism

and access to the network on a FCFS (First Come First Served) basis. The number of tokens in the token pool is decremented for each cell leaving the LB. If there is no token in the token pool, cells have to queue in the cell buffer, if the queue is full, new coming cells are discarded.

The proper choice of the parameters *R* and (*B+M*) depends on what the Leaky Bucket is designed to control and how tight a control is required. For example, if it is the source's Peak Cell Rate (PCR) that is to be controlled, then the token generating rate *R* should be close to the source's PCR, and (*B+M*) can be chosen a lower value to obtain a certain CLP. On the other hand, if it is the source's Average Cell Rate (ACR) that is to be controlled, then the token rate *R* should be close to the source's ACR, and a high value of (*B+M*) is needed to obtain the same CLP. Furthermore, the choice of a larger value of (*B+M*) will correspondingly result in a higher cell delay. This may be a major issue for some services, for example the real-time traffic, they have very low tolerance to the cell delay, so buffering them will become inappropriate.

Obviously, a single traditional LB cannot be used to control a bursty source's ACR and PCR simultaneously, while it also introduce different degrees of cell delay. Some recent researches on LB have been conducted either on a more complicated control meachnism, or utilizing multiple LB to police the source's different parameters ([5], [9], [10], [12], [14],[15], [17], [26], [27], [32]. [33]).

1.5 Research Objective

As we have stated, a traditional LB can either police a bursty source's PCR or ACR while allowing for a certain degree of burstiness (B+M), but it can not police both the PCR and SCR simultaneously. On the other hand, bursty sources may be very dynamic and have very different characteristics, for instance, an On-Off source may transmit cells at the peak rate during very short active periods, and generate no cells during long idle periods, thus the sources' utilization (i.e., the ratio of ACR to PCR) is really low.

If we set the token generation rate *R* according to the source's PCR, the network resources will be dramatically wasted; but if *R* is set equal to ACR, it will introduce a very high Cell Loss Probability (CLP). The token pool and cell buffer can decrease the CLP while allowing for a certain degree of burstiness, but it also introduce cell delays and is inappropriate for delay-sensitive sources.

In recent years, significant amount of research has been conducted to explore new schemes for LB mechanisms ([8],[11],[13],[15],[16],[18],[19],[20],

[21], [22], [27]). Most of the research still uses the basic LB structure while using multiple LB or modified protocols to police different characteristics of different sources.

In this thesis, we propose a modified Leaky-Bucket mechanism, which can police the source's PCR and SCR simultaneously without introducing any cell delays. The main properties of the modified LB mechanism are:

1). Utilization of a dynamic token generator which alternates between active and passive periods. Tokens are generated at the same rate as the source's PCR when the token generator is active, otherwise, no tokens are generated.

2). Elimination of the token pool and the cell buffer. When the token generator is active, cells generated by the active bursty source can be transmitted at the same speed as they are generated, i.e. the source's PCR, therefore it is not necessary to use a token pool and a cell buffer. As a result, no cell delay is introduced, and cells can be transmitted only when both the source and the token generator are active.

3). Introduction of a Random Cycle: the time axis is divided into random cycles. The bursty source is able to transmit cells during each random cycle. Different operating protocols allow the source to transmit different number of bursts during each random cycle.

The thesis is organized as follows: Chapter 2 describes the modified leaky-bucket mechanism, introduces the system model, and derives a preliminary mathematical result which will be needed to analyze the performance of the modified leaky-bucket mechanism. In chapter 3, we will introduce our first protocol which is called forced protocol and perform its analysis based on the mathematical results obtained in chapter 2. Chapter 4 will be devoted to the description and analysis of another protocol which is called unforced protocol. In chapter 5, we propose and analyze compensation forced protocol, which is designed to compensate a shortcoming in the forced protocol. Chapter 6 will compare the modified leaky-bucket mechanism with a conventional leaky-bucket, and give the conclusion and the future research work.

Chapter 2

Modified Leaky-Bucket Mechanism and The Derivation of Preliminary Mathematical Results

2.1 Introduction

Leaky-bucket is a widely used UPC mechanism. A traditional LB can police either a bursty source's PCR or ACR, but not both at the same time, and it also introduces cell delays. In this chapter we propose a modified LB mechanism to police a bursty source's PCR and ACR simultaneously without introducing any cell delays, which derives from its two distinguishing mechanisms: the token generator is dynamic alternating between On and Off states; neither cell buffer nor token pool is utilized. We will also describe that by properly designed probability distributions of the token generator's On and Off periods, the modified LB is able to provide not only the cell delay but also the Cell Loss performance.

This chapter is composed of two parts. The first part will introduce the traffic model of a bursty source and describe the design of the modified LB mechanism for achieving the objective of policing both PCR and ACR simultaneously without any cell delay. In the second part, we will focus on the derivation of a mathematical result, which will enable us to determine the performance of the modified LB throughout the remaining chapters.

2.2. The Modified Leaky-Bucket

Before we proceed to the description of the modified leaky-bucket, let us describe a bursty source's traffic model first, because it is the basis of the design of the modified LB.

2.2.1 The Traffic Model of A Bursty Source

As shown in figure 2.1, a bursty source alternates between active (On) and passive (Off) states. We assume that the transition from a passive to an active state occurs with a probability $(1-\beta_s)$, while the probability of a transition from an active to a passive state is $(1-\alpha_s)$. Accordingly, the length of the On and Off periods are geometrically distributed in number of slots with means $\frac{1}{1-\alpha_s}$ and $\frac{1}{1-\beta_s}$, respectively.

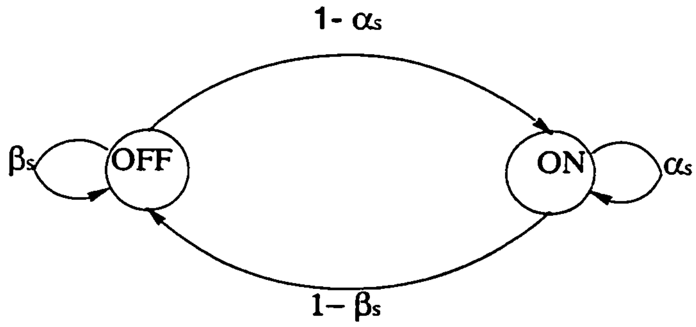


Figure 2.1 The Model of a Bursty Source

we define:

α_s = Pr(the source will be active in the next time slot given that it is active in the present slot)

β_s = Pr(the source will be passive in the next time slot given that it is idle in the present slot)

k_s = the number of time slots that the source remains in active state during a burst

l_s = the number of time slots that the source remains in passive state during each visit to the Off state

then we have the following probability distributions:

$$Pr(k_s) = (1-\alpha_s)\alpha_s^{k_s-1} \tag{2-1}$$

$$Pr(l_s) = (1-\beta_s)\beta_s^{l_s-1} \tag{2-2}$$

2.2.2 The Modified Leaky-Bucket Mechanism

2.2.2.1 Description

Motivated by the bursty source's dynamic cell generation process, we design a dynamic token generator in the modified leaky-bucket mechanism. This token generator alternates also between two states: active and passive. When the token generator is active, it will generate tokens at the same speed as the source's PCR. When the token generator is passive, it will not generate any tokens. Meanwhile we assume that there is neither token pool nor cell buffer in this modified leaky-bucket.

The strategy behind this modified LB mechanism is that for confirming sources, when the source is active, we would want the token generator generate tokens as fast as the source's cell generation rate such that cells can be transmitted into the network at its PCR without any delay; On the other hand, when the source is passive, we would want the token generator stop generating tokens since the tokens virtually represent the network resource, therefore no network resources will be wasted. So ideally, all cells generated during the source's active period will immediately obtain tokens from the token generator and be transmitted into the network at its PCR, so we will not need the token pool and the cell buffer, therefore no cell delay occurs.

Obviously, the source's PCR is policed under this strategy. By properly designing the probability distribution of the token generator's On and Off periods, it is also possible to police the source's ACR. Both parameters are policed while no cell delay occurs.

2.2.2.2 The model of the modified leaky bucket

A bursty source has been modeled in section 2.2.1 as having geometrically distributed On and Off periods. Based on this model, the dynamic token generator is designed to have geometrically distributed On and Off periods too, with parameters α_t and β_t respectively. Thus we similarly define:

$\alpha_t = Pr$(the token generator will be active in the next slot given that it is active in the present slot)

β_t = Pr(the token generator will be passive in the next slot given that it is passive in the present slot)

k_t = the number of time slots that the token generator remains in active state during each burst

l_t = the number of time slots that the token generator remains in passive state during each Off period

then we have the following probability distributions:

$$Pr(k_t) = (1-\alpha_t)\alpha_t^{k_t - 1} \tag{2-3}$$

$$Pr(l_t) = (1-\beta_t)\beta_t^{l_t - 1} \tag{2-4}$$

Ideally, to reach the objective of policing a bursty source's PCR and ACR simultaneously, the token generator's parameters α_t and β_t should be chosen to be equal to the source's parameters α_s and β_s, but randomness of these variables prohibit this. In our analysis, we will present how a set of α_t and β_t can be reached to police a source's PCR and ACR simultaneously.

2.3 Derivation of the Preliminary Mathematical Results

The number of cells generated by a bursty source during a given time interval will be needed for the mathematical analysis in the following chapters when we consider different protocols. In this section we will derive the mathematical result of the PGF of the number of cells generated by a bursty source during a time interval. Before we proceed, we need to introduce the cell arrival process of the modified LB system first.

2.3.1. The Cell Arrival Process of the Modified Leaky-Bucket System

We may model the cell arrival process to the modified LB as a special case of an ATM multiplexer system, let us first review an ATM multiplexer system in the following figure 2.2.

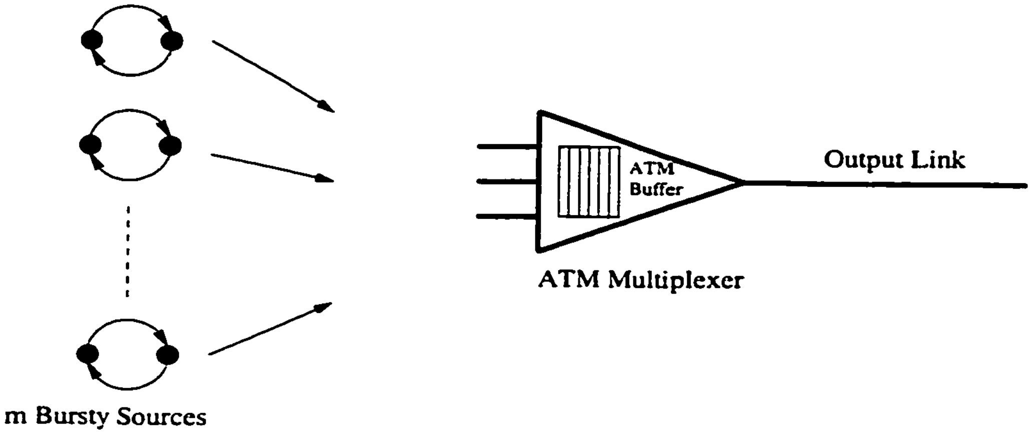


Figure 2.2 An ATM multiplexer fed with *m* i.i.d. bursty sources

This ATM multiplexer system is fed by *m* independently identically distributed (i.i.d.) bursty sources. Cells generated by these sources are first stored in the cell buffer, then they are retrieved and transmitted on the output link by a priority scheme. Such an ATM multiplexer system has been studied in [4].

In the modified Leaky Bucket (LB), we assume that the time axis is also slotted, and during a time slot an active bursty source generates only one cell.

We notice that this is like having an ATM multiplexer which is fed by a single input source, if the output link is turned off, cells generated by this bursty source will all be stored in the cell buffer. If the cell buffer has an unlimited size, then the queue distribution of the cell buffer will represent the distribution of cells generated by the bursty source.

Thus, the cell arrival process to the modified LB can be considered as a special case of the ATM multiplexer system ___ a single source, infinite-sized cell buffer and zero server multiplexer system, the distribution of the queue length of the cell buffer will correspond to the distribution of the number of cells generated by this bursty source. However, the analysis on an ATM multiplexer in [4] is not directly applicable to this special case, fortunately, a parallel analysis applies here and will be presented below.

2.3.2. Derivation of the Preliminary Mathematical Results

Let's define the following notations:

i_k = the number of cells generated by the end of slot *k*

a_k = the state of the bursty source during slot *k* (it assumes the value 0 if the source is Off and 1 if the source is On)

b_k = the number of cells generated during slot *k*

f_k = the number of cells generated by an active source during slot *k*

We also define the following Bernoulli random variables:

$$c = \begin{cases} 1 & \text{if the source remains active in the next slot given that it is active in the present slot} \\ 0 & \text{otherwise} \end{cases}$$

$$d = \begin{cases} 1 & \text{if the source becomes active in the next slot given that it is inactive in the present slot} \\ 0 & \text{otherwise} \end{cases}$$

2.3.2.1 Derivation of the joint Probability Generating Function (PGF) of the number of cells generated by a bursty source and the state of the source

Referring to the bursty source's traffic model in figure 2.1, we can get the PGFs of the Bernoulli random variables c and d defined above as:

$$C(z) = 1 - \alpha_s + \alpha_s \cdot z \tag{2-5}$$

$$D(z) = \beta_s + (1 - \beta_s) \cdot z \tag{2-6}$$

f_k is assumed to be geometrically distributed with the PGF $f(z)$, here

$$f(z) = E[z^{f_k}] \tag{2-7}$$

$$b_k = \begin{cases} f_k & \text{if } a_k = 1 \\ 0 & a_k = 0 \end{cases} \tag{2-8}$$

It is obvious that we can relate a_k and i_k of two consecutive slots by the following equations:

$$a_{k+1} = \begin{cases} c & \text{if } a_k = 1 \\ d & a_k = 0 \end{cases} \tag{2-9}$$

$$i_{k+1} = i_k + b_{k+1} \tag{2-10}$$

Now let $Q_k(z, y)$ denote the transient joint PGF of the number of cells generated by the end of slot k and the state of the bursty source during slot k, then:

$$Q_k(z, y) = E[z^{i_k} y^{a_k}] = \sum_{i=0}^{\infty} \sum_{j=0}^{1} z^i y^j P_k(i, j) \tag{2-11}$$

here $P_k(i, j) = P_r(i_k = \text{i}, a_k = \text{j})$ and j has value 0 when the source is passive and 1 when the source is active. Then we have:

$$Q_{k+1}(z, y) = E[z^{i_k + b_{k+1}} y^{a_{k+1}}]$$

Using equations (2-9) and (2-10) and averaging over f_k, c and d yields:

$$Q_{k+1}(z, y) = [D(y \cdot f(z))] E\left\{ z^{i_k} \left[\frac{C(y \cdot f(z))}{D(y \cdot f(z))} \right]^{a_k} \right\}$$

$$= [D(y \cdot f(z))] \cdot Q_k(z, Y) \tag{2-12}$$

here $Y = \dfrac{C(y \cdot f(z))}{D(y \cdot f(z))} = \dfrac{1 - \alpha_s + \alpha_s y \cdot f(z)}{\beta_s + (1 - \beta_s) y \cdot f(z)}$

So $Q_{k+1}(z, y) = [\beta_s + (1 - \beta_s) y \cdot f(z)] \cdot Q_k(z, Y)$ (2-13)

Before we continue to derive $Q_k(z, y)$, let's make the following definitions:

$$U(0) = y \qquad U(1) = 1 - \alpha_s + \alpha_s y \cdot f(z) \qquad U(k+1) = X(1) U(k)|_{y = Y}$$

$$X(0) = 1 \qquad X(1) = \beta_s + (1 - \beta_s) y \cdot f(z) \qquad X(k+1) = X(1) X(k)|_{y = Y}$$

$$\Phi(k) = \frac{U(k)}{X(k)} \qquad \Phi(0) = \frac{U(0)}{X(0)} = y$$

then it can be demonstrated that:

$$X(k+1) = X(1) \cdot X(k)|_{y=Y} \tag{2-14}$$

$$\Phi(k+1) = \Phi(k)|_{y=Y} \tag{2-15}$$

So equation (2-13) can be expressed as:

$$Q_{k+1}(z,y) = X(1) \cdot Q_k(z,Y) \tag{2-16}$$

In the following we will use recursion method to obtain an expression for $Q_k(z,y)$. From (2-16):

i) For $k = 0$***,***

$$Q_1(z,y) = X(1)Q_0(z,Y)$$

since $Y = \frac{1-\alpha_s+\alpha_s y \cdot f(z)}{\beta_s+(1-\beta_s)y \cdot f(z)} = \frac{U(1)}{X(1)} = \Phi(1)$

so: $Q_1(z,y) = X(1)Q_0(z,\Phi(1))$ (2-17)

ii) For $k = 1$***,***

$$Q_2(z,y) = X(1) \cdot Q_1(z,Y) = X(1) \cdot X(1)|_{y=Y}Q_0(z,\Phi(1))|_{y=Y}$$

from equation (2-14) and (2-15), we can express it as:

$$Q_2(z,y) = X(2) \cdot Q_0(z,\Phi(2)) \tag{2-18}$$

iii) For $k = 3$***,*** following from above,

$$Q_3(z,y) = X(1)Q_2(z,Y) = X(3)\cdot Q_0(z,\Phi(3)) \tag{2-19}$$

From the equations (2-17), (2-18) and (2-19), the functional equation (2-11) may be expressed in the following form:

$$Q_k(z,y) = X(k)\cdot Q_0(z,\Phi(k)) \tag{2-20}$$

Before we proceed further, we need to determine an explicit form for $U(k)$, $X(k)$. It can be shown that $U(k)$ and $X(k)$ satisfy the following recurrence relationships [4]:

$$U(k) = [\beta_s + \alpha_s\cdot f(z)]\cdot U(k-1) + (1-\alpha_s-\beta_s)\cdot f(z)\cdot U(k-2) \tag{2-21}$$

$$X(k) = [\beta_s + \alpha_s\cdot f(z)]\cdot X(k-1) + (1-\alpha_s-\beta_s)\cdot f(z)\cdot X(k-2) \tag{2-22}$$

The above are homogeneous difference equations and have the following characteristic equation:

$$\lambda^2 - (\beta_s + \alpha_s\cdot f(z))\cdot\lambda - (1-\alpha_s-\beta_s)\cdot f(z) = 0 \tag{2-23}$$

the roots of the above equation are given by:

$$\lambda_{1,2} = \frac{\beta_s + \alpha_s f(z) \mp \sqrt{(\beta_s + \alpha_s f(z))^2 + 4(1-\alpha_s-\beta_s)f(z)}}{2} \tag{2-24}$$

then equations (2-21) and (2-22) can be shown to have the following solution [12]:

$$U(k) = D_1\lambda_1^k + D_2\lambda_2^k \tag{2-25}$$

$$X(k) = C_1\lambda_1^k + C_2\lambda_2^k \tag{2-26}$$

here C_1, C_2, D_1, D_2 are constants which may be determined from the initial conditions:

$$X(0) = C_1 + C_2 = 1$$

$$X(1) = C_1\lambda_1 + C_2\lambda_2 = \beta_s + (1-\beta_s)\cdot yf(z)$$

$$U(0) = D_1 + D_2 = y$$

$$U(1) = D_1\lambda_1 + D_2\lambda_2 = (1-\alpha_s) + \alpha_s\cdot yf(z)$$

thus we can solve for $C_{1,2}$, $D_{1,2}$ and obtain the following results:

$$C_{1,2} = \frac{1}{2} \mp \frac{2(y - y\beta_s - \alpha_s)\cdot f(z) + (\beta_s + \alpha_s f(z))}{2\sqrt{(\beta_s + \alpha_s f(z))^2 + 4(1-\alpha_s-\beta_s)f(z)}} \tag{2-27}$$

$$D_{1,2} = \frac{y}{2} \mp \frac{2(1-\alpha_s+\alpha_s yf(z)) - (\beta_s+\alpha_s f(z))\cdot y}{2\sqrt{(\beta_s + \alpha_s f(z))^2 + 4(1-\alpha_s-\beta_s)f(z)}} \tag{2-28}$$

2.3.2.2 The marginal Probability Generating Functions of the number of cells generated by a bursty source and the state of the source

Let's define:

$P_k(z)$ =the transient marginal PGF of the number of cells generated by a bursty source by the end of *k*'th slot

$A_k(y)$ =the transient marginal PGF of the state of the source at the end of *k*'th slot.

From $Q_k(z, y)$ which is shown in equation (2-20), we can get:

$$P_k(z) = Q_k(z, y)\big|_{y=1} = [X(k)Q_0(z, \Phi(k))]\big|_{y=1} = \bar{X}(k) \cdot Q_0(z, \bar{\Phi}(k)) \qquad (2\text{-}29)$$

$$A_k(y) = Q_k(z, y)\big|_{z=1} = [X(k)Q_0(z, \Phi(k))]\big|_{z=1} \qquad (2\text{-}30)$$

where $\bar{X}(k) = X(k)\big|_{y=1} = \bar{C}_1\lambda_1^k + \bar{C}_2\lambda_2^k$

$$\bar{\Phi}(k) = \Phi(k)\big|_{y=1} = \frac{\bar{D}_1\lambda_1^k + \bar{D}_2\lambda_2^k}{\bar{C}_1\lambda_1^k + \bar{C}_2\lambda_2^k}$$

$$\bar{C}_{1,2} = C_{1,2}\big|_{y=1} = \frac{1}{2} \mp \frac{2(1-\alpha_s-\beta_s)f(z) + (\beta_s + \alpha_s f(z))}{2\sqrt{(\beta_s + \alpha_s f(z))^2 + 4(1-\alpha_s-\beta_s)f(z)}} \qquad (2\text{-}31)$$

$$\bar{D}_{1,2} = D_{1,2}\big|_{y=1} = \frac{1}{2} \mp \frac{2(1-\alpha_s+\alpha_s f(z)) - (\beta_s + \alpha_s f(z))}{2\sqrt{(\beta_s + \alpha_s f(z))^2 + 4(1-\alpha_s-\beta_s)f(z)}} \qquad (2\text{-}32)$$

$Q_0(z, \Phi(k))$ in equations (2-29) and (2-30) is the initial joint PGF which is determined by the initial conditions of the number of cells generated by a bursty source and the state of the source for a given time interval.

Equations (2-29) and (2-30) are the mathematical results that will be used in following chapters to analyze the performance of the modified leaky-bucket mechanism.

2.3.2.3. Application in the modified LB model

Since in the modified LB model, we assume that an active bursty source only generates one cell during a time slot, so equation (2-7) becomes:

$$f(z) = E[z^{f_k}] = z \tag{2-33}$$

In the following chapters, we will need equations (2-29), (2-30) and (2-33) to perform the analysis of the modified LB model under different protocols.

Chapter 3

Forced Protocol

3.1 Introduction

In this chapter we propose a forced protocol and present the mathematical analysis of the performance of the modified LB operating under this forced protocol.

The operation of the modified LB is based on a random cycle definition. During each random cycle, the forced protocol will allow a bursty source to transmit cells into the network while limiting the number of cells transmitted to the source's single burst.

The performance will be determined using the preliminary mathematical results derived in chapter 2. Since the modified leaky bucket does not introduce any cell delay, the main performance measure here is Cell Loss Probability (CLP). We note that the time axis may be divided into identically distributed random cycles, as a result all we need is to determine the CLP during such a random cycle.

This chapter is organized as follows: first, we introduce the definition of a random cycle, and then we describe the forced protocol based on this cycle definition. Secondly, we will use the mathematical results derived in chapter 2 to determine the average CLP during a random cycle. In this chapter, we study the performance of the modified LB with the constant token generator's parameters, (Appendix A gives the same analysis for the variant token generator's parameters). Lastly, we will give the numerical results for the forced protocol.

3.2 The definition of a random cycle

As has been described in chapter 2, both the source and the token generator alternate between burst and silent intervals, independent of each other. Thus the burst and silent intervals of the source and the token generator are not in synchronism with each other. We assume that the time axis may be divided into random cycles.

A random cycle is defined with the following properties:

1). A random cycle begins with both the source and the token generator in their Off states.

2). During each cycle, a source will always be able to transmit cells into the network. The number of slots that a source is allowed to transmit cells into the network will not exceed a single On period of the source or the token generator, these On periods will be referred to as the tagged On periods.

A source will be able to transmit cells into the network only when both the source and the token generator are simultaneously in their tagged On periods. A token generator is only allowed to have one tagged period during each cycle.

3). If during the first On period of the token generator, the source is also On for any period of time, then these will be the tagged On periods of the source and the token generator respectively. If on the other hand, during the initial On period of the token generator, the source has never become On, then as soon as the source becomes On, the token generator will be forced immediately to the On state even if it is not On. Again, these On periods will become the tagged periods of the source and the token generator respectively.

4). The cycle will end as soon as the tagged On period of the source terminates.

During the source's tagged On period, it will be able to transmit cells into the network as long as the token generator is also in its tagged On period. The cells generated following the termination of the tagged On period of the token generator will be lost.

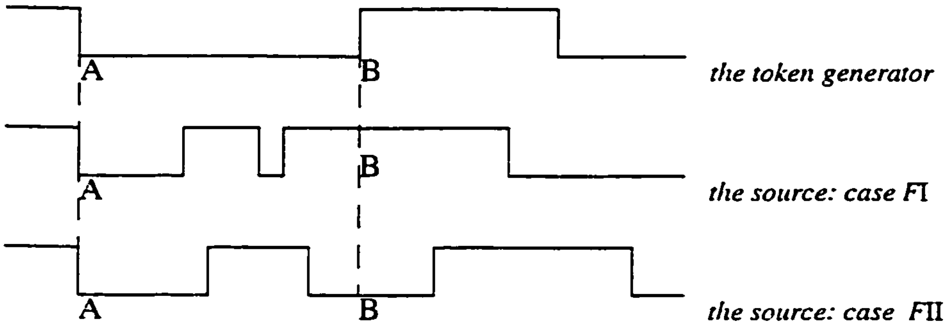
A new random cycle begins as soon as the previous one ends. If at this point the token generator is in its On state, it is forced to the Off state.

We note that random cycles are identically distributed, therefore it is sufficient to analyze the performance of the protocol during one such cycle.

In fact, among the four properties listed above, the first one defines when a random cycle starts, the last one defines when a random cycle terminates which also distinguishes the forced protocol from the unforced protocol as will be described in chapter 4. The second and third properties define how the forced protocol operates under different situations to enable the source transmit cells in each cycle. Next, we will describe the forced protocol in these situations.

3.3 Description of the Forced Protocol:

As we have defined, a random cycle starts when both the token generator and the source are in their Off states. As may be seen in the following figure 3.1, the random cycle starts at time A. During the token generator's initial Off period, i.e. the interval from A to B, the source might alternate between On and Off states.



Legend: A,B: the moment when the token generator changes it's state

Figure 3.1: The source's different state when the token generator's initial Off period terminates

During the random cycle, when the token generator makes its first transition from Off to On at time B, the bursty source might either be in On or Off states, this introduces two possibilities as shown as cases FI and FII in the figure. Here case

FI and FII respectively represent that the source is On and Off at time B in the forced protocol.

Since the forced protocol will operate differently in these two cases, we will discuss them separately in the following.

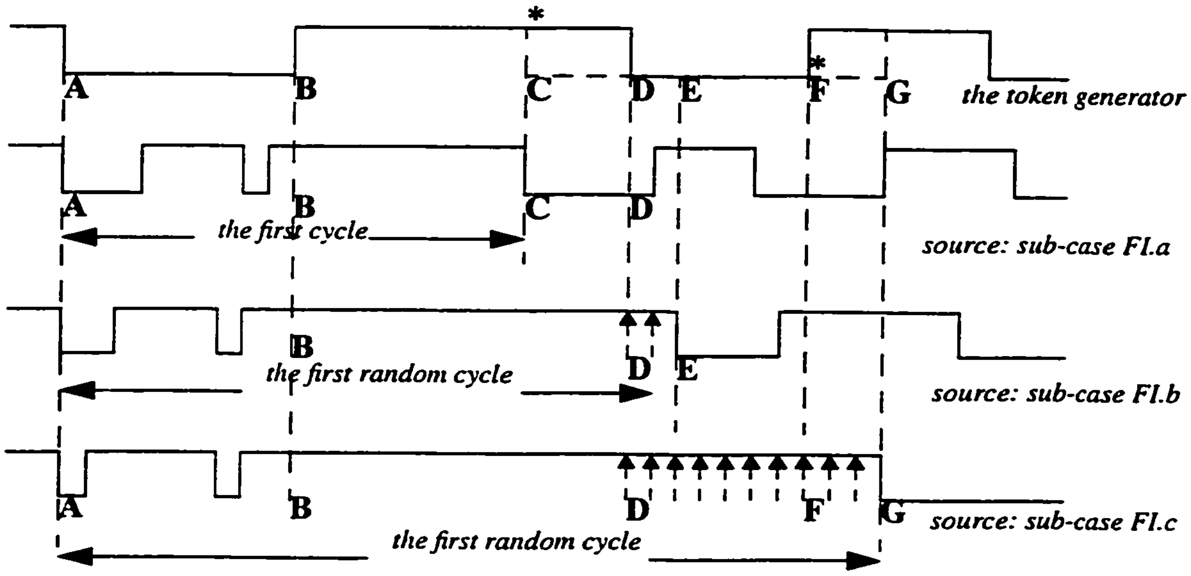
3.3.1. Case FI: the source is in On state when the token generator's initial Off period terminates at time B

From figure 3.1, we can see that when the token generator first makes the transition from Off to On at time B, the source is On, from this point on, both the source and the token generator will be in their tagged periods, so the source can start to transmit cells into the network.

Depending on whether the tagged period of the source or the token generator terminates first, we identify three sub-cases FI.a, FI.b and FI.c as shown in the following figure 3.2:

Sub-case FI.a: The tagged period of the source terminates before the token generator's tagged period

From figure 3.2, the source's tagged period terminates at time C while the tagged period of the token generator terminates at time D. According to the definition of the random cycle, the cells generated during the interval A to B will be lost, and those generated during the interval B to C will be transmitted into the network. A new random cycle will begin at time C.



Legend:

A,B ...G: the moment when the soruce or the token generator's state changes

*: the moment when the token generator's state is forced to change

Figure 3.2: Three possibilities of the termination time of the source's tagged period in case FI

Sub-case FI.b: The tagged period of the source terminates after the token generator's tagged period but before the next token generator's On period.

As shown in figure 3.2, the source's tagged period terminates at time E after the tagged period of the token generator which terminates at time D. Cells generated in the interval A to B and D to E will be lost, and cells generated in the interval B to D will be allowed to be transmitted into the network. A new random cycle will begin at time E.

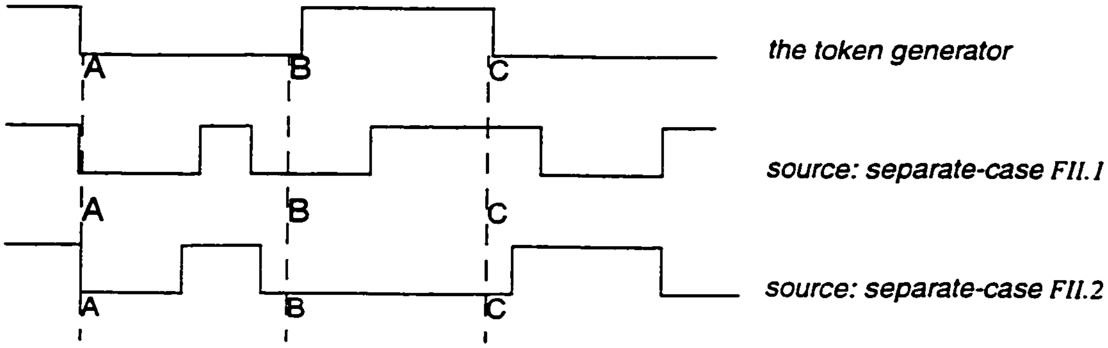
Sub-case FI.c: The tagged period of the source terminates during an On period following the token generator's tagged period.

Again from Figure 3.2, cells generated during the interval A to B and D to G will be lost and those generated during the interval B to D will be transmitted into the network. The new random cycle starts at time G.

Next, we give a similar description of case FII.

3.3.2. Case FII: the source is in Off state when the token generator's initial Off period terminates at time B

Unlike case FI, here we have to further divide case FII into two more separate cases depending on when the source's next burst starts after the token generator makes its first transition from Off to On state at time B. These two separate cases are shown below and to be referred as FII.1 and FII.2.



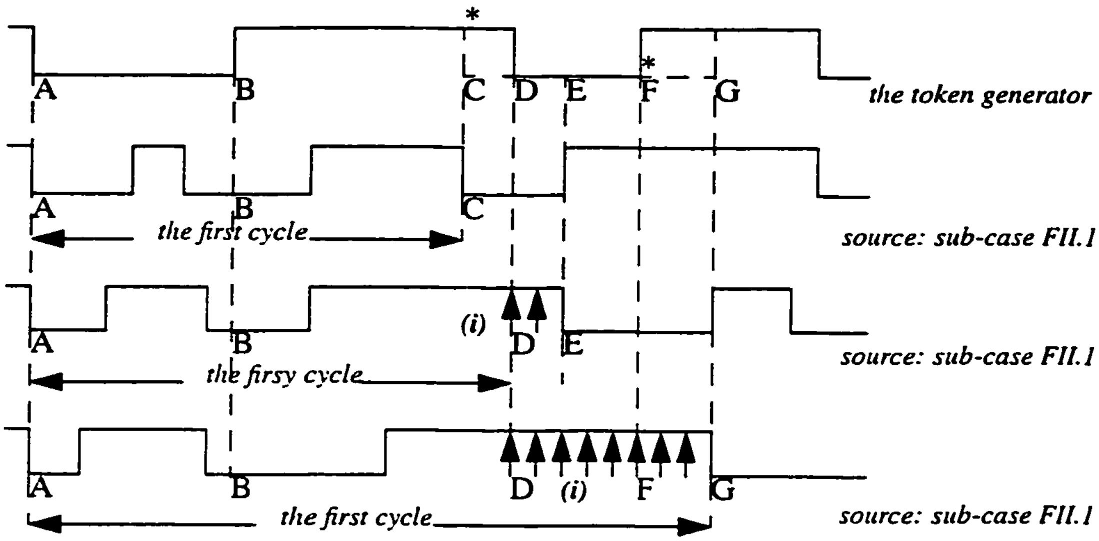
Legend:

A, B...E: the moment when the source or the token generator changes its state

Figure 3.3 The two separate-cases when the source starts its burst after the token generator first makes the transition from Off to On

3.3.2.1. Separate-case FII.1: the source starts its next burst during the token generator's initial On period

As may be seen from figure 3.3, the source's next burst starts during the token generator's initial On period (i.e. time B to D), which becomes their tagged periods. As in the case FI, there are three sub-cases depending on the termination time of the source's tagged period. These are shown in the following figure 3.4 as sub-cases FII.1.a, FII.1.b and FII.1.c.



Legend: A,B...G: the moment when the source or the token generator changes it's state

*: the moment when the token generator is forced to change it's state

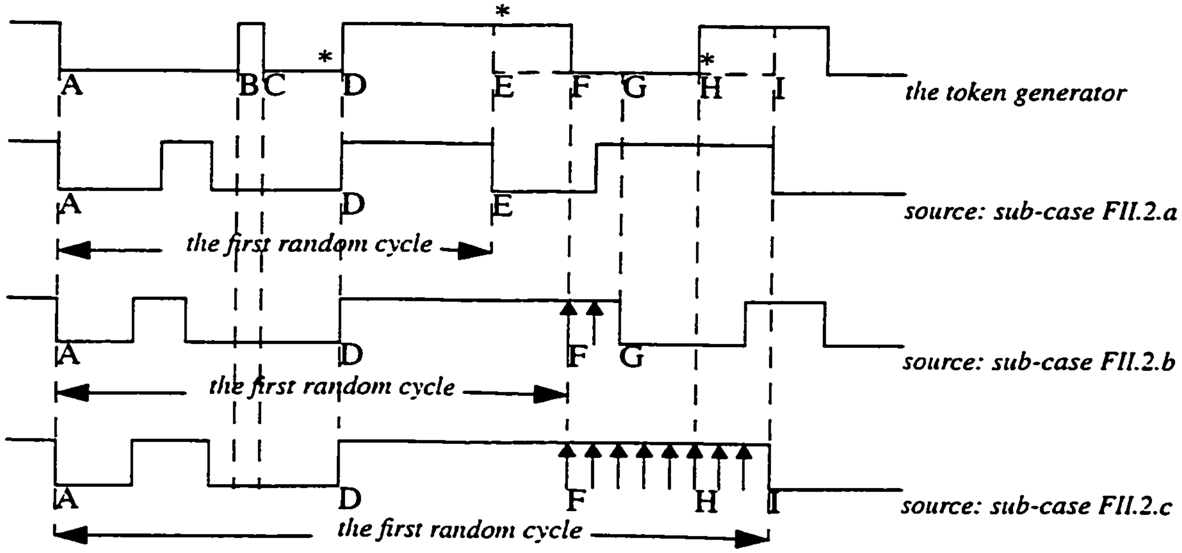
Figure 3.4: Three possibilities of the termination time of the source's tagged period in case FII.1

In each of the sub-cases, the forced protocol operates in the very same way as the corresponding sub-cases in case FI, which we have described in detail in section 3.3.1. The random cycle terminates at times C, E and G with respect to the three sub-cases, as shown in figure 3.4.

3.3.2.2. Separate-case FII.2: the source's next burst starts after the token generator's initial On period has terminated

As shown in the following figure 3.5, the source remains in the Off state during the initial On period of the token generator in the interval B to C. Finally when the source starts its On period at time D, the token generator is in the Off state at time D. In this case the forced protocol will force the token generator to the On state at time D and both the source and the token generator will start this tagged periods.

Depending on the termination time of the source's tagged period, there are three sub-cases as shown below in Figure 3.5. Again, these three case are same as those in case FI.



Legend: A,B...I: the moment when the source or the token generator's state changes

*: the moment when the token generator is forced to change it's state

Figure 3.5 Three possibilities of the termination time of the source's tagged period in case FII.2

Next, we will compare the cases of FI, FII.1 and FII.2. These three cases differ in the relative starting time of their tagged periods. In case FI, the token generator's tagged period begins after the source's tagged period, while in case FII.1 the token generator's tagged period begins before the source's tagged period, and in case FII.2 the tagged periods of the source and the token generator begins simultaneously. Since the source and the token generator's On periods are geometrically distributed and the residual time of the already started tagged periods will have the same geometrical distribution as the original On periods, as a result, these three cases may be treated identically as having geometrically distributed tagged periods starting at the same time. As for the subcases a, b and c as described in each of the three cases, they differ from each other only because of the termination time of the source's tagged period and are completely identical in cases FI, FII.1 and FII.2.

Next we will present the mathematical analysis of the average cell loss probability during such a random cycle using the preliminary result we have derived in chapter 2.

3.4. Analysis of the Average Cell Loss Probability (CLP) during a Random Cycle

As we have defined in section 2.2.2, α_t and β_t are the parameters of the token generator's independent identically distributed On and Off periods respectively. They have been defined as:

α_t = Pr (the token generator will be active in the next slot given that it is active in the present slot)

β_t = *Pr* (the token generator will be passive in the next slot given that it is passive in the present slot)

Obviously, when α_t and β_t are constant, the token generator has geometrically distributed On and Off periods. In this chapter, we will determine the average CLP during a random cycle under this condition.

In chapter 2, we have derived the closed form expressions for the transient PGFs of the number of cells generated by a bursty source and the state of the source at the end of *k*th time slot, which are given in equations (2-29) and (2-30) respectively. Next, we will use these results to analyze the average CLP during a random cycle.

In order to determine the average CLP, we need to know the average number of cells lost and transmitted into the network during a random cycle. The cell losses may occur in the token generator's initial Off period or during the source's tagged period if the source's tagged On period has longer duration than the token generator's tagged period. On the other hand, the cells will be transmitted into the network during the time that both the source and the token generator are simultaneously in their tagged periods. Next, we will determine each of these quantities.

3.4.1. The average number of cells lost during the token generator's initial Off period

Let's first determine the number of cells generated during the token generator's initial Off period, i.e. the time interval A to B as shown in figure 3.1. Let's define:

$\tilde{N}_{AB}$ = the number of cells generated by the source during the interval A to B

Obviously, $\tilde{N}_{AB}$ is the number of cells lost during the interval A to B. To determine the average value of $\tilde{N}_{AB}$, we will use equation (2-29) to derive the PGF of the number of cells generated during this interval, which depends on the initial conditions at time A.

As may be seen in figure 3.1, at time A both the token generator and the source are in their Off states, and the number of cells generated is zero, so we have the following initial probability:

$$Pr(i_0=0,a_0=0) = 1 \tag{3-1}$$

where: i_0 is the number of cells and a_0 is the state of the source, which is assumed to be 0 when the source is Off and to be 1 when the source is On, obviously both variables are 0 at time A.

From equation (3-1), we can get the following initial joint PGF:

$$Q_0(z, \Phi(k)) = z^0 \cdot [\Phi(k)]^0 = 1 \tag{3-2}$$

Define:

$P_{l_t}(z)$ = the transient marginal PGF of the number of cells generated by the end of l_t'th slot during the token generator's initial Off period

Substituting equation (3-2) into equation (2-29), we have:

$$P_{l_t}(z) = \tilde{X}(l_t) = \tilde{C}_1 \lambda_1^{l_t} + \tilde{C}_2 \lambda_2^{l_t} \tag{3-3}$$

Here $\tilde{C}_{1,2}$ and $\lambda_{1,2}$ have been given in chapter 2. Define:

$P(z)$ = the marginal PGF of the number of cells generated during the token generator's initial Off period

Averaging (3-3) with respect to the duration of the token generator's Off period, we get:

$$P(z) = \sum_{l_t = 1}^{\infty} P_{l_t}(z) Pr(l_t) = \sum_{l_t = 1}^{\infty} \left(\tilde{C}_1 \lambda_1^{l_t} + \tilde{C}_2 \lambda_2^{l_t}\right)(1 - \beta_t)\beta_t^{l_t - 1} \tag{3-4}$$

here l_t is the number of time slots of the token generator's initial Off period.

Let $\overline{\tilde{N}_{AB}}$ denote the average number of $\tilde{N}_{AB}$, which is defined as the number of cells generated during the time interval A to B, so:

$$\overline{\tilde{N}_{AB}} = [P(z)]'|_{z=1} = (1 - \beta_t)\left[\left(\frac{C_1\tilde{\lambda}_1}{1 - \lambda_1\beta_t}\right)'\Big|_{z=1} + \left(\frac{\tilde{C}_2\lambda_2}{1 - \lambda_2\beta_t}\right)'\Big|_{z=1}\right] \tag{3-5}$$

Substitution $\tilde{C}_{1,2}$ in (2-31) and λ_{12} in (2-24) will give:

$$\bar{C}_1\Big|_{z=1} = 0 \qquad \bar{C}_2\Big|_{z=1} = 1$$

$$C_1'\Big|_{z=1} = \frac{\beta_s}{2-\alpha_s-\beta_s} + \frac{\alpha_s-1}{(2-\alpha_s-\beta_s)^2}, \qquad C_2'\Big|_{z=1} = -\frac{\beta_s}{2-\alpha_s-\beta_s} + \frac{1-\alpha_s}{(2-\alpha_s-\beta_s)^2}$$

$$\lambda_1\Big|_{z=1} = \beta_s+\alpha_s-1 \ , \qquad \lambda_2\Big|_{z=1} = 1$$

$$\lambda_1'\Big|_{z=1} = \frac{(1-\alpha_s)(\beta_s+\alpha_s-1)}{2-\alpha_s-\beta_s} \ , \qquad \lambda_2'\Big|_{z=1} = \frac{1-\beta_s}{2-\alpha_s-\beta_s}$$

Substituting them into equation (3-5), we have:

$$\overline{N_{AB}} = \frac{(1-\beta_t)\left(\beta_s+\frac{\alpha_s-1}{2-\alpha_s-\beta_s}\right)(\beta_s+\alpha_s-1)}{[1-(\beta_s+\alpha_s-1)\beta_t](2-\alpha_s-\beta_s)} + \frac{\frac{1-\alpha_s}{2-\alpha_s-\beta_s}-2\beta_s+1}{2-\alpha_s-\beta_s} + \frac{\beta_t}{1-\beta_t}\cdot\frac{1-\beta_s}{2-\alpha_s-\beta_s} \tag{3-6}$$

3.4.2. The average number of cells lost and transmitted during the source's tagged period

As explained before, we will be able to treat the three cases as FI, FII.1 and FII.2 together. We note that cell losses will occur during a source's tagged period only if the source's tagged period has a longer duration than the token generator's tagged period.

Let us define:

k_s = the number of time slots in the source's tagged period

k_t = the number of time slots in the token generator's tagged period

From figure 3.2, the source's tagged period begins at time B and ends at time G at the latest. So we let the interval B to G denote the source's tagged period and define:

N_{BG} = the number of cells successfully transmitted into the network during the source's tagged period

$\tilde{N}_{BG}$ = the number of cells lost during the source's tagged period

From figure 3.2, it is obvious that:

$$\tilde{N}_{BG} = \begin{cases} k_s - k_t & \text{when } k_s > k_t \\ 0 & \phantom{\text{when }} k_s \le k_t \end{cases} \qquad (3\text{-}13)$$

Since the distributions of k_s and k_t are independent of each other, so averaging over the values of k_s and k_t, we can obtain the average value of $\tilde{N}_{BG}$ when $k_s > k_t$ as:

$$\overline{\tilde{N}_{BG,k_s > k_t}} = \sum_{k_t = 1}^{\infty} \sum_{k_s = k_t}^{\infty} (k_s - k_t)(1 - \alpha_s)\alpha_s^{k_s - 1}(1 - \alpha_t)\alpha_t^{k_t - 1} = \frac{(1 - \alpha_t)\alpha_s}{(1 - \alpha_s)(1 - \alpha_s\alpha_t)} \qquad (3\text{-}14)$$

Define:

$Pr(k_s > k_t)$ = Pr(the source's tagged period is longer than the token generator's tagged period)

According to the conditional probability, we can obtain:

$$Pr(k_s > k_t) = \sum_{k_t = 1}^{\infty} \sum_{k_s = k_t}^{\infty} Pr(k_s)Pr(k_t) = \frac{1-\alpha_t}{1-\alpha_t\alpha_s} \tag{3-15}$$

$$Pr(k_s \le k_t) = 1 - Pr(k_s > k_t) = \frac{\alpha_t - \alpha_t\alpha_s}{1-\alpha_t\alpha_s} \tag{3-16}$$

Then we can derive the average value of $\tilde{N}_{BG}$ as:

$$\overline{\tilde{N}_{BG}} = 0 \cdot Pr(k_s \le k_t) + \tilde{N}_{BG,k_s > k_t} \cdot Pr(k_s > k_t) = \frac{(1-\alpha_t)^2\alpha_s}{(1-\alpha_s)(1-\alpha_s\alpha_t)^2} \tag{3-17}$$

The above equation gives the average number of cells lost during the source's tagged period. Next, we determine the average number of cells transmitted into the network during the source's tagged period. Clearly,

$$\overline{N_{BG}} = \overline{k_s} - \overline{\tilde{N}_{BG}} \tag{3-18}$$

here $\overline{k_s}$ is the average duration of the source's one tagged period, which equals to the average number of cells generated by the source during its tagged period. Since the source's burst has a geometrical distribution, so we have:

$$\overline{k_s} = \frac{1}{1-\alpha_s} \tag{3-19}$$

$$\overline{N_{BG}} = \overline{k_s} - \overline{\tilde{N}_{BG}} = \frac{(1-\alpha_s\alpha_t)^2 - ((1-\alpha_t)^2 \cdot \alpha_s)}{(1-\alpha_s)(1-\alpha_s\alpha_t)^2} \tag{3-20}$$

The above gives the average number of cells transmitted into the network during the source's tagged period.

3.4.3 Cell Loss Probability during a random cycle

Define:

$\overline{N_{on}}$ = the average number of cells transmitted into the network during a random cycle

$\overline{N_{off}}$ = the average number of cells lost during a random cycle

P_{loss} = the cell loss probability during a random cycle

$\overline{N_{on}}$ equals to the average number of cells transmitted into the network during the source's tagged period, so:

$$\overline{N_{on}} = \overline{N_{BG}} \tag{3-21}$$

$\overline{N_{off}}$ equals to the average number of cells lost during the source's tagged period plus the average number of cells lost during the token generator's initial Off period, so:

$$\overline{N_{off}} = \overline{\bar{N}_{AB}} + \overline{\bar{N}_{BG}} \tag{3-22}$$

Combining equations (3-6), (3-17) and (3-20) into the above equations, we get:

$$P_{loss} = \frac{\overline{N_{off}}}{\overline{N_{off}} + \overline{N_{on}}} \qquad (3\text{-}23)$$

In this chapter, we have assumed that the token generator and the source's parameters α_t, β_t, α_s and β_s are constant. In Appendix A, we will propose to use a variant token generator's On parameter α_t and perform a similar analysis, for the reason of the mathematical complexity, we put it in Appendix A.

3.5. Numerical Results:

From equation (3-23), we can derive the numerical results for the forced protocol as shown in figure 3.6.

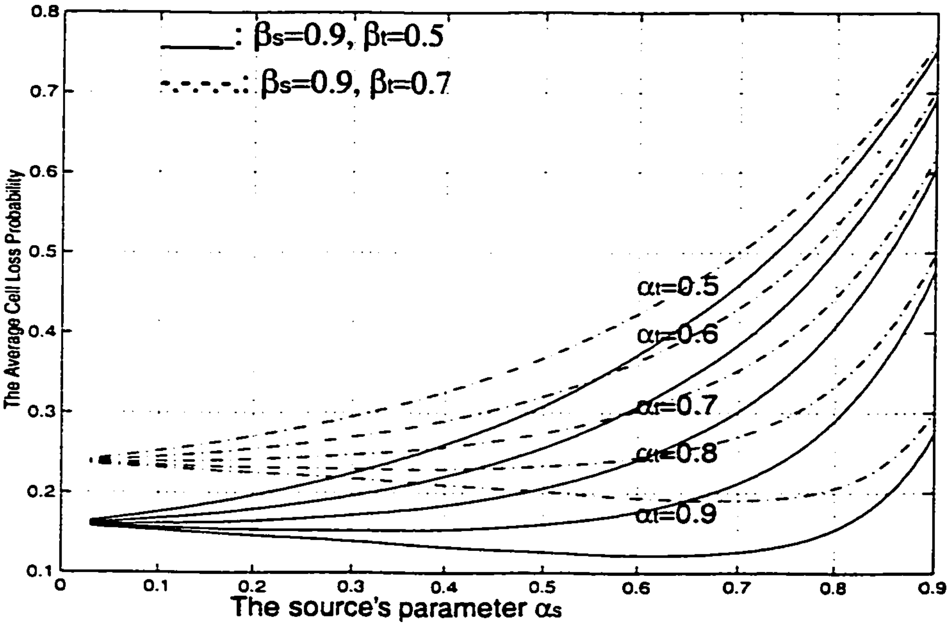


Figure 3.6 The forced protocol with the token generator's different parameters

We can see that for certain given values of the source's parameters α_s and β_s, the token generator's higher On parameter α_t and lower Off parameter β_t will give a lower cell loss probability, this is due to the fact the token generator would have a longer tagged period and a shorter Off period, leading to more cells being transmitted during the source's tagged period as well as less cells lost during the source's tagged period and the token generator's initial Off period. Consequently, CLP will be lower.

The figure also shows that a higher value of the source's parameter α_s will lead to a higher cell loss probability, because the source would have longer tagged period and burst, which would lead to more cells being lost during the source's tagged period and the token generator's initial Off period, thus to cause a higher CLP. Meanwhile, we can see that at lower α_s values, CLPs are every close with different α_t values, because the source's shorter tagged periods have the approximate possibilities to be completely transmitted during the token generator's longer or shorter tagged period.

3.7. Advantages and disadvantages of the forced protocol:

From the description of the forced protocol in section 3.2, we can see that the forced protocol has the following advantages:

(1). The source is able to transmit cells into the network during each random cycle, even when the token generator's initial On period is too short to let the source start its burst.

(2). Once the source starts to transmit cells into the network, no tokens will be wasted during the random cycle, because the token generator will be forced to change to Off state once the source's tagged period terminates. This limits the number of cells transmitted into the network during each random cycle to the source's single tagged period. This is an advantage while facing a violating source which may have more frequent bursts than being claimed at the connection setup phase.

Obviously, this advantage of the forced protocol can prevent the QoS of other connections from being affected by one source's violating behavior.

(3). If the bursty source violates the negotiated parameters and has a very long tagged period, the forced protocol forces the token generator to stay in its Off state after its tagged period terminates till the source's tagged period terminates. Thus violating cells generated after the token generator's tagged period would not be transmitted into the network.

The above second advantage comes from the property 4 of the random cycle's definition in section 3.2, which states that a random cycle ends as soon as the tagged On period of the source terminates. Obviously, this advantage has a tradeoff here, because it limits the number of cells being transmitted into the network during a random cycle to the source's SINGLE tagged period. In this case, if the statistical characteristics of a bursty source give frequent bursts but with short durations, such a limitation will be too strict for policing this type of source's Average Cell Rate (ACR). Next, we will propose *unforced protocol* to compensate for this shortcoming of the forced protocol, and perform the same CLP analysis.

Chapter 4

Unforced Protocol

4.1 Introduction

In order to overcome the forced protocol's shortcoming of policing a bursty source's ACR too strictly, we will propose *unforced protocol* in this chapter. While keeping the forced protocol's advantage of always giving the bursty source a chance to transmit cells into the network during each random cycle, the unforced protocol also allows the source's multiple bursts to be transmitted during each random cycle, thus the bursty source's ACR can be policed more fairly.

To achieve this objective of the unforced protocol, the definition of a random cycle needs to be modified, which will be discussed in the first part of this chapter. Then we will describe how the unforced protocol operates during a random cycle and analyze the cell loss probability. As in Chapter 3, we only analyze CLP with the constant token generator's parameters in this chapter. (In Appendix B, we will perform a similar analysis of the unforced protocol under the variant token generator's parameter $\alpha_t(i)$). At the end of this chapter, we will present the numerical results to compare the forced and unforced protocols.

4.2 The definition of a random cycle under the unforced protocol

In the forced protocol, a random cycle is defined with 4 properties as described in section 3.2, in which property 1 defines how a random cycle starts, property 2 and 3 determines how the protocol operates to allow a source to transmit cells in each cycle, property 4 defines how a random cycle terminates to limit the number of cells transmitted into a network during each random cycle to the source's single burst.

In the unforced protocol, we aim at eliminating the forced protocol's shortcoming of only being able to transmit one burst while also keeping its advantage of transmitting cells in each cycle. Therefore, in the random cycle's definition, only the property 4 in the random cycle's definition will be modified for the unforced protocol, the other three properties will remain same as in section 3.2.

According to the fourth property, a random cycle ends when the source's tagged period ends. If at this moment the token generator is On, it is forced to the Off state and a new cycle begins. We modify this property as:

A random cycle ends at the later finishing time of the source or the token generator's tagged period.

We note that from property 2 of the random cycle's definition a source is allowed to transmit cells into the network only during a single token generator's tagged period. On the other hand, if during this token generator's tagged period, the source has several On periods, all of the cells generated will be transmitted into the network with the above modified property. Each of these source's On periods will be referred to as the source's tagged periods.

When the token generator's tagged period terminates, if the source is in Off state, then the random cycle ends and a new one begins immediately; if the source is in On state, this On period becomes the last tagged period of the source, then the cells generated during the residual duration of this tagged period will be lost, a new cycle begins when this last tagged period of the source terminates.

With this modification, the source is able to transmit multiple tagged periods into the network during a random cycle, as we can see in the following figure 4.1.

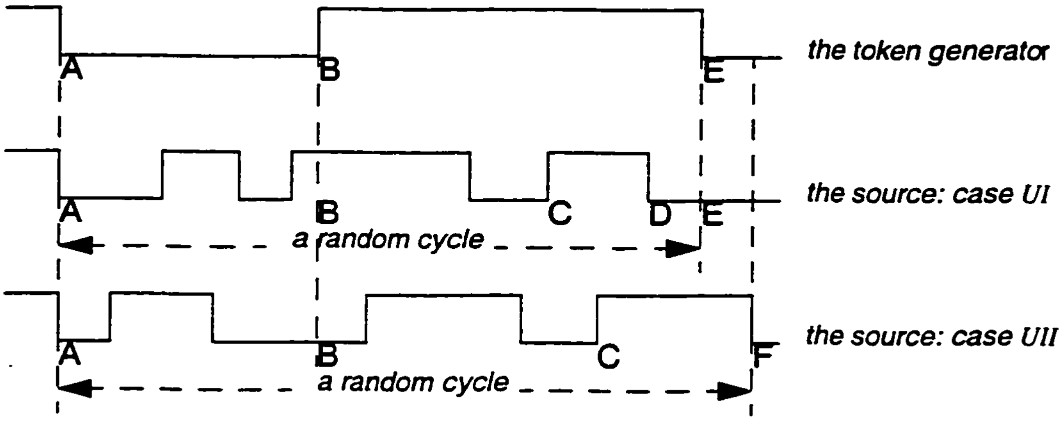


Figure 4.1: the source's multiple bursts being transmitted

As in the forced protocol, there exist two cases depending on the source's state when the token generator's initial Off period terminates at time B, which are denoted by cases UI and UII in figure 4.1 to present the unforced protocol.

The last tagged period of the source is the last On period of the source that has started during the token generator's tagged period. The source's last tagged period may terminate before or after the token generator's tagged period.

Since the source is able to transmit multiple bursts during a random cycle under the unforced protocol, it would be able to reach its ACR. As may be seen in chapter 2, the modified leaky bucket will allow the source to reach its PCR by generating tokens at the source's PCR during the token generator's On period, so the unforced protocol will be more fair and effective to police a bursty source's parameters, in terms of both PCR and ACR.

Next, we will describe the operation of the unforced protocol.

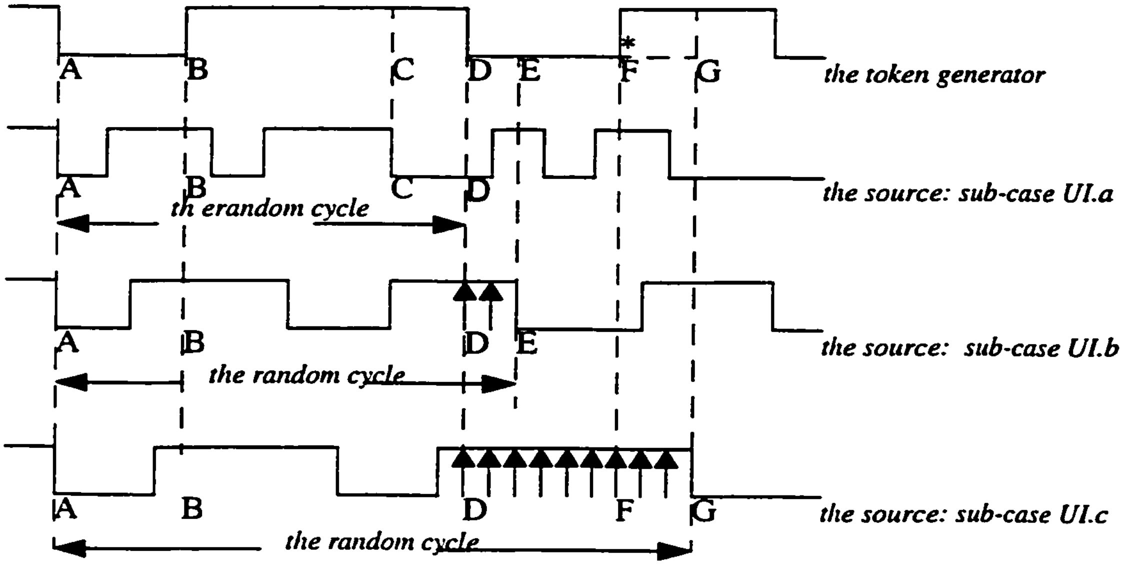
4.3 Descriptions of the unforced protocol

As we have described in chapter 3, the forced protocol operates differently for the two cases FI and FII. Same in the unforced protocol, the two cases UI and UII shown in Figure 4.2 also have different operations, so we describe them separately below.

4.3.1 Case UI: the source is On when the token generator first makes the transition from Off to On state at time B

As may be seen from figure 4.1, in case UI, the source is active and starts to transmit cells into the network when the token generator makes the first transition from Off to On at time B, and these On periods become the source and the token generator's tagged periods respectively.

During the token generator's tagged period, the source might have multiple tagged periods. Depending on the termination time of the source's last tagged period, there exist three possibilities as presented by sub-cases UI.a, UI.b and UI.c in the following figure 4.2.



Legend:

A,B ...G: the moment when the soruce or the token generator's state changes

*: the moment when the token generator is forced to change its state

Figure 4.2: Three possibilities of the termination time of the source's last tagged period in case UI

Similar to the respective sub-cases in the forced protocol, the random cycle also terminates differently in these sub-cases, as we explain below:

Sub-Case UI.a: The source's last tagged period terminates before the token generator's tagged period

From figure 4.2, we can see that no cells will be lost during the source's tagged periods. The random cycle ends when the token generator's tagged period terminates at time D. The cells generated during the interval A to B will be lost and those generated during the interval B to D will be transmitted into the network.

Sub-case UI.b: The source's last tagged period terminates after the token generator's tagged period but before the token generator's next On period

Cells generated in the intervals A to B, D to E will be lost and cells generated in the interval B to D will be allowed to be transmitted into the network. A new random cycle starts at time E.

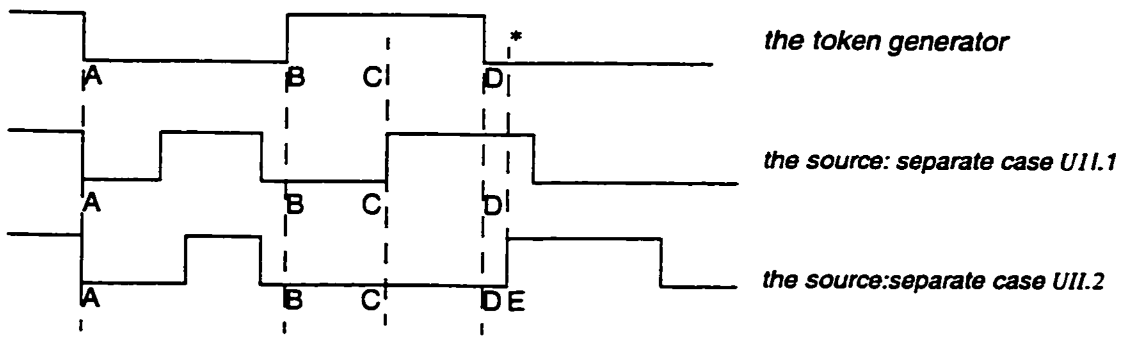
Sub-case UI.c: The source's last tagged period terminates during the token generator's On period following the token generator's tagged period.

From the property 2 of a random cycle's definition, which is same for both forced and unforced protocols, the source is only allowed to transmit cells into the network during ONE of the token generator's tagged period in each random cycle. Therefore, the token generator will be forced to stay with Off state during the residual duration of the source's last tagged period. Cells generated during the intervals A to B, D to G will be lost and those generated during the interval B to D will be transmitted into the network. The new random cycle begins at time G.

4.3.2 Case UII: the source is in the Off state when the token generator first makes the transition from Off to On state at time B

Similar to section 3.3.2, we have 2 separate situations regarding to when the source's burst starts after the token generator's initial Off period terminates at time

B, which are represented by separate cases UII.1 and UII.2 respectively in figure 4.3, we briefly explain the difference of the two separate cases in the following:



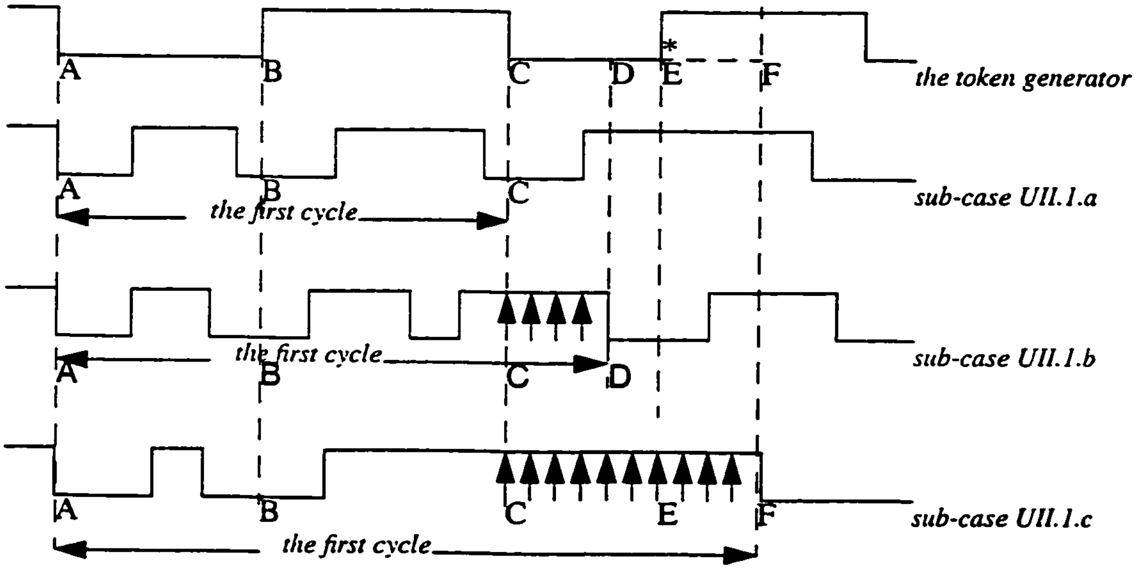
*
A
B
C
D
the token generator
A
B
C
D
the source: separate case UII.1
A
B
C
D E
the source:separate case UII.2

Legend:

A, ...E: the moment when the token generator or the source changes its state

Figure 4.3: the two separate cases in case UII

4.3.2.1 Separate case UII.1: the source starts its burst during the token generator's initial On period



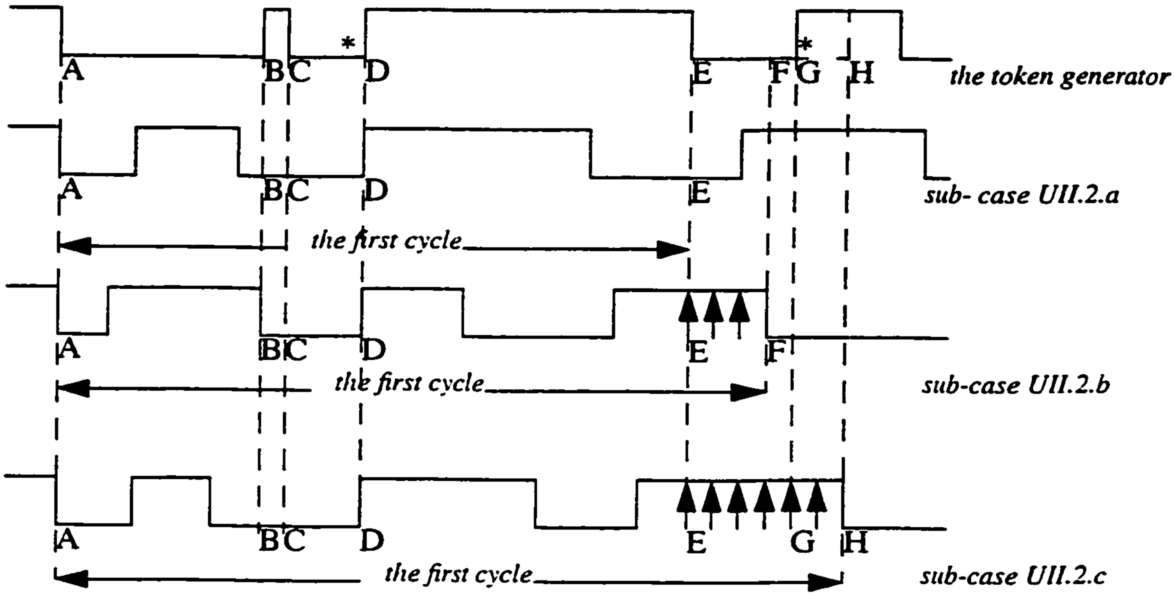
Legend:

A,B ...G: the moment when the source or the token generator's state changes

Figure4.4:Three possibilities of the termination time of the source's last tagged period in case UII..

As may be seen from the above Figure 4.4, the source's next burst starts during the token generator's initial On period (B to D), same as in case UI, depending on the termination time of the source's last tagged period, we have three sub cases UII.1.a, UII.1.b and UII.1.c. The random cycle ends at time C, D and F respectively.

4.3.2.2 Separate case UII.2: the source starts its burst after the token generator's initial On period



Legend:

A,B ...H: the moment when the soruce or the token generator's state changes

*: the moment when the token generator is forced to change it's state

Figure4.5:Three possibilities of the termination time of the source's last tagged period in case UII.2

Same as in case FII.2 in the forced protocol the source should be able to transmit cells during each random cycle, so in this case, once the source starts its next burst at time D, the token generator will be forced to change to the On state simultaneously, both the source and the token generator starts their tagged periods at this moment.

Similarly, depending on the termination time of the source's last tagged period, there are three sub-cases as UII.2.a, UII.2.b and UII.2.c in the above figure. The random cycle terminates at time E, F and H respectively in these sub-cases.

Next, before we perform the mathematical analysis, we need to describe the analysis methodology for the unforced protocol and compare cases UI, UII.1 and UII.2 accordingly.

In the unforced protocol, the source's multiple tagged periods will successfully transmit cells into the network during the token generator's tagged period, so we cannot use the same method to determine the average number of cells being transmitted during each random cycle as we used in equation (3-18), which is based on the condition that only one tagged period of the source would be transmitted into the network. Instead, here we have to use the preliminary PGF results that we have derived in chapter 2 to analyze the average number of cells being successfully transmitted and lost.

The PGF method depends on the initial conditions when the token generator's tagged period starts, this is reasonable because the token generator's tagged period can host different number of the source's tagged periods, which may be affected by the source's state when the token generator's tagged period starts.

Obviously, different number of the source's tagged periods during the token generator's tagged period will cause different number of cells being successfully transmitted into the network.

Based on this analysis method, next we need to compare the initial conditions of cases UI, UII.1 and UII.2 when the token generator's tagged period begins.

From figure 4.2 and 4.5, we can see that in cases UI and UII.2 the source is in its On state when the token generator's tagged period begins at time B and D respectively, this means that case UI and UII.2 have the same initial conditions when the token generator's tagged period begins. Therefore, we still can treat these two cases together in the analysis of the unforced protocol. But for the case UII.1 as shown in Figure 4.4, the source is in the Off state when the token generator's tagged period begins at time B, so the initial conditions will be different from cases UI and UII.2, we have to treat this case separately in the following analysis.

4.4 Analysis of the average Cell Loss Probability

As in the forced protocol, in order to determine the average CLP, we need to determine the average number of cells lost and transmitted during a random cycle.

4.4.1 The average number of cells lost during the token generator's initial Off period

Because the token generator's behavior during its initial Off period from A to B has not changed under the unforced protocol, so the average number of cells lost during the token generator's initial Off period is same as in the forced protocol, which has been given in equation (3-6) for all cases.

4.4.2 The average number of cells lost and successfully transmitted during the source's tagged periods and the token generator's tagged period

Next, we will separately determine the number of cells being successfully transmitted and lost for case UI and UII.1, and case UII.2 will be treated together with case UI.

4.4.2.1 Case UI (and UII.2)

As can be seen from Figure 4.2, the source's multiple tagged periods terminate at time G at the latest in case UI, so we let the time interval B to G denote the duration of the source's multiple tagged periods and define:

N_{BG} = the number of cells successfully transmitted into the network during the source's tagged periods

$\tilde{N}_{BG}$ = the number of cells lost during the source's tagged periods

From figure 4.2, when the token generator's On period starts at time B, the initial conditions are that the source is On and the initial number of cells is zero, which can be described as a_0=1 and i_0=0. Thus we have the following initial probability:

$$Pr(i_0{=}0, a_0{=}1) = 1 \tag{4-1}$$

This leads to the following initial joint PGF:

$$Q_0(z, \Phi(k)) = z^0[\Phi(k)]^1 = \Phi(k) \tag{4-2}$$

Define:

$P_{k_t}(z)$ = the transient marginal PGF of the number of cells generated by the end of k_t'th slot during the token generator's tagged period

Substituting equation (4-2) into equation (2-29), we have:

$$P_{k_t}(z) = \tilde{X}(k_t) \bullet \tilde{\Phi}(k_t) = \tilde{D}_1\lambda_1^{k_t} + \tilde{D}_2\lambda_2^{k_t} \tag{4-3}$$

here $\tilde{D}_{1,2}$ and $\lambda_{1,2}$ have been given in equations (2-32) and (2-24).

Define:

$P_1(z)$ = the marginal PGF of the total number of cells generated during the token generator's tagged period in case UI

From equation (4-3) and the definition of the conditional probability, we have:

$$P_1(z) = \sum_{k_t=1}^{\infty} P_{k_t}(z)Pr(k_t) = \sum_{k_t=1}^{\infty} \left(\tilde{D}_1\lambda_1^{k_t} + \tilde{D}_2\lambda_2^{k_t}\right)(1-\alpha_t)\alpha_t^{k_t-1}$$

$$= \left(\frac{\bar{D}_1\lambda_1}{1-\lambda_1\alpha_t} + \frac{\bar{D}_2\lambda_2}{1-\lambda_2\alpha_t}\right)(1-\alpha_t) \tag{4-4}$$

Let $\overline{N_{BH}}$ denote the average value of N_{BH} which is defined as the number of cells being successfully transmitted during the token generator's tagged period, so:

$$\overline{N_{BG}} = P_1(z)'\Big|_{z=1} = (1-\alpha_t)\left[\left(\frac{\bar{D}_1\lambda_1}{1-\lambda_1\alpha_t}\right)'\Bigg|_{z=1} + \left(\frac{\bar{D}_2\lambda_2}{1-\lambda_2\alpha_t}\right)'\Bigg|_{z=1}\right] \tag{4-5}$$

here $\bar{D}_1\Big|_{z=1} = 0$ $\qquad\qquad$ $\bar{D}_2\Big|_{z=1} = 0$

$$D_1'\Big|_{z=1} = \frac{1-\beta_s}{(2-\alpha_s-\beta_s)^2} - \frac{\alpha_s}{2-\alpha_s-\beta_s} \qquad D_2'\Big|_{z=1} = -\frac{1-\beta_s}{(2-\alpha_s-\beta_s)^2} + \frac{\alpha_s}{2-\alpha_s-\beta_s}$$

Combine $\lambda_{1,2}$ and $\lambda'_{1,2}$ in section 3.4.1 into (4-5), we get:

$$\overline{N_{BG}} = \frac{(1-\alpha_t)(\beta_s+\alpha_s-1)\left(\frac{1-\beta_s}{2-\alpha_s-\beta_s}-\alpha_s\right)}{[1-(\beta_s+\alpha_s-1)\alpha_t][2-\alpha_s-\beta_s]} + \frac{1+\alpha_s-\beta_s-\frac{1-\beta_s}{2-\alpha_s-\beta_s}}{2-\alpha_s-\beta_s} + \frac{\alpha_t}{1-\alpha_t}\cdot\frac{1-\beta_s}{2-\alpha_s-\beta_s} \tag{4-6}$$

Next we will determine $\bar{N}_{BH}$, the number of cells lost during the source's multiple tagged periods. Cell losses will only occur only the source's last tagged period terminates after the token generator's tagged period, so first we need to determine the source's state at this moment. Let us define:

$A_{k_t}(y)$ = the transient marginal PGF of the source's state at the end of k_t'th slot

during the token generator's tagged period

$A_1(y)$ = the marginal PGF of the source's state at the end of the token generator's tagged period in case UI

Substituting the initial probability from equation (4-2) into the preliminary PGF result in equation (2-30), we have:

$$A_{k_t}(y) = X(k_t)\Big|_{z=1} \cdot \Phi(k_t)\Big|_{z=1} = \left(D_1\lambda_1^{k_t} + D_2\lambda_2^{k_t}\right)\Big|_{z=1} \tag{4-7}$$

$$A_1(y) = \sum_{k_t=1}^{\infty} A_{k_t}(y) \cdot Pr(k_t) = \sum_{k_t=1}^{\infty} A_{k_t}^1(y) \cdot (1-\alpha_t)\alpha_t^{k_t-1} \tag{4-8}$$

Substituting $D_{1,2}$, $\lambda_{1,2}$ into the above, we derive the following:

$$A_1(y) = \frac{1}{2-\alpha_s-\beta_s}\left[(1-\beta_s) + \frac{(1-\alpha_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{1-\alpha_t(\beta_s+\alpha_s-1)} + y\left((1-\alpha_s) - \frac{(1-\alpha_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{1-\alpha_t(\beta_s+\alpha_s-1)}\right)\right] \tag{4-9}$$

Let us define:

Pr(On)=Pr(the source is On when the token generator's tagged period ends)

Pr(Off)=Pr(the source is Off when the token generator's tagged period ends)

From equation (4-9) we can get the closed form expressions for the above two probabilities as:

$$Pr(\text{On}) = \frac{(1-\alpha_s)}{(2-\alpha_s-\beta_s)} - \frac{(1-\alpha_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{(1-\alpha_t(\beta_s+\alpha_s-1))(2-\alpha_s-\beta_s)} \tag{4-10}$$

$$Pr(\mathrm{Off}) = \frac{(1-\beta_s)}{(2-\alpha_s-\beta_s)} + \frac{(1-\alpha_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{(1-\alpha_t(\beta_s+\alpha_s-1))(2-\alpha_s-\beta_s)} \qquad (4\text{-}11)$$

As we have stated, cells are lost during the source's tagged periods only if the source is On when the token generator's tagged period terminates, this is the case for UI.b and UI.c in figure 4.2. If the source is Off when the token generator's tagged periods terminates, no cell losses occur during the source's tagged periods, as we show in UI.a in figure 4.2. Therefore, we can derive the average number of cell losses during the source's tagged periods as:

$$\overline{N_{BG}} = 0 \cdot Pr(\mathrm{Off}) + \overline{N_{BG,k_s > k_t}} Pr(\mathrm{On}) \qquad (4\text{-}12)$$

Here k_s and k_t are the number of time slots of the source's and the token generator's tagged periods respectively, $\overline{N_{BH,k_s > k_t}}$ is the average number of cell losses when the source's last tagged period terminates after the token generator's tagged period, since the token generator's tagged period is geometrically distributed which has the memoryless property, we can have:

$$\overline{N_{BG,k_s > k_t}} = \sum_{k_t = 1}^{\infty} \sum_{k_s = k_t}^{\infty} (k_s - k_t)(1-\alpha_s)\alpha_s^{k_s - 1}(1-\alpha_t)\alpha_t^{k_t - 1} = \frac{(1-\alpha_t)\alpha_s}{(1-\alpha_s)(1-\alpha_s\alpha_t)} \qquad (4\text{-}13)$$

So the average number of cells lost during the source's tagged periods in case UI will be:

$$\overline{N_{BG}} = \frac{(1-\alpha_t)\alpha_s}{(1-\alpha_s)(1-\alpha_s\alpha_t)} \cdot \left(\frac{(1-\alpha_s)}{(2-\alpha_s-\beta_s)} - \frac{(1-\alpha_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{(1-\alpha_t(\beta_s+\alpha_s-1))(2-\alpha_s-\beta_s)} \right) \qquad (4\text{-}14)$$

4.4.2.2 Case UII.1

From Figure 4.4, the source's multiple tagged periods terminates at time H at the latest in case UII.1, so we let the time interval B to G denote the duration of the source's multiple tagged periods and define:

N_{BH} = the number of cells successfully transmitted into the network during the source's tagged periods

$\tilde{N}_{BH}$ = the number of cells lost during the source's tagged periods

When the token generator's On period starts at time B, the initial conditions are that the source is Off and the initial number of cells is 0, which can be described as a_0=0 and i_0=0, so we have the initial probability as:

$$Pr(i_0=0, a_0=0) = 1 \tag{4-15}$$

Then the initial joint PGF when the token generator's On period starts in case UII.1 becomes:

$$Q(z, \Phi(k)) = z^0[\Phi(k)]^0 = 1 \tag{4-16}$$

Define:

$P_{k_t}(z)$ = the transient marginal PGF of the number of cells generated by the end of k_t'th slot during the token generator's tagged period

$P_2(z)$ = the marginal PGF of the number of cells generated during the token generator's tagged period in case UII.1

As in equation (4-4), we obtain:

$$P_{k_t}(z) = [X(k_t)Q(z, \Phi(k_t))]\Big|_{y=1} = \bar{X}(k_t) = \bar{C}_1\lambda_1^{k_t} + \bar{C}_2\lambda_2^{k_t} \tag{4-17}$$

$$P_2(z) = \sum_{k_t=1}^{\infty} P_{k_t}(z)Pr(k_t) = \sum_{k_t=1}^{\infty}\left(\bar{C}_1\lambda_1^{k_t} + \bar{C}_2\lambda_2^{k_t}\right)(1-\alpha_t)\alpha_t^{k_t-1}$$

$$= \left(\frac{\bar{C}_1\lambda_1}{1-\lambda_1\alpha_t} + \frac{\bar{C}_2\lambda_2}{1-\lambda_2\alpha_t}\right)(1-\alpha_t) \tag{4-18}$$

Same as in case UI, we can determine the average number of cells successfully transmitted into the network from the above equation as follow:

$$\overline{N_{BH}} = P_2(z)'\Big|_{z=1} = (1-\alpha_t)\left[\left(\frac{\bar{C}_1\lambda_1}{1-\lambda_1\alpha_t}\right)'\Bigg|_{z=1} + \left(\frac{\bar{C}_2\lambda_2}{1-\lambda_2\alpha_t}\right)'\Bigg|_{z=1}\right] \tag{4-19}$$

Substituting $\bar{C}_{1,2}$ and $\lambda_{1,2}$ into it, we get:

$$\overline{N_{BH}} = = \frac{(1-\alpha_t)\left(\beta_s + \frac{\alpha_s - 1}{2-\alpha_s-\beta_s}\right)(\beta_s+\alpha_s-1)}{[1-\alpha_t(\beta_s+\alpha_s-1)](2-\alpha_s-\beta_s)} + \frac{\frac{1-\alpha_s}{2-\alpha_s-\beta_s} - 2\beta_s + 1}{2-\alpha_s-\beta_s} + \frac{1-\alpha_t}{\alpha_t}\cdot\frac{1-\beta_s}{2-\alpha_s-\beta_s} \tag{4-20}$$

Next, to determine the average number of cells lost during the source's tagged periods, we also need to determine the source's state when the token generator's tagged periods terminates. Define:

$A_{k_t}(y)$ = the transient marginal PGF of the source's state at the end of k_t'th slot during the token generator's tagged period in case UII.1

$A_2(y)$ = the marginal PGF of the source's state at the end of the token generator's tagged period in case UII.1

Substituting sub-case UII.1's initial PGF in equation (4-16) into equation (2-30), we can have:

$$A_{k_t}(y) = X(k_t)\Big|_{z=1} = \left(C_1\lambda_1^{k_t} + C_2\lambda_2^{k_t}\right)\Big|_{z=1} \tag{4-22}$$

$$A_2(y) = \sum_{k_t=1}^{\infty} A_{k_t}(y)\cdot Pr(k_t) = \sum_{k_t=1}^{\infty} A_{k_t}(y)\cdot(1-\alpha_t)\alpha_t^{k_t-1} \tag{4-23}$$

here $C_1\Big|_{z=1} = \dfrac{(1-\beta_s)(1-y)}{2-\alpha_s-\beta_s}$, $\qquad C_2\Big|_{z=1} = \dfrac{(1-\alpha_s)+y(1-\beta_s)}{2-\alpha_s-\beta_s}$

$\lambda_1\Big|_{z=1} = \beta_s+\alpha_s-1$, $\qquad \lambda_2\Big|_{z=1} = 1$.

Substituting them into the above, we obtain:

$$A_2(y) = \frac{1}{2-\alpha_s-\beta_s}\left[(1-\beta_s)-\frac{(1-\beta_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{1-\alpha_t(\beta_s+\alpha_s-1)}+y\left((1-\alpha_s)+\frac{(1-\beta_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{1-\alpha_t(\beta_s+\alpha_s-1)}\right)\right] \tag{4-24}$$

Define:

Pr(On)=Pr (the source is On when the token generator's tagged period ends)

Pr(Off)=Pr (the source is Off when the token generator's tagged period ends)

From equation (4-24) we can get the closed form expressions of the above two probabilities as:

$$Pr(\mathrm{On}) = \left(\frac{(1-\alpha_s)}{(2-\alpha_s-\beta_s)} + \frac{(1-\beta_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{(1-\alpha_t(\beta_s+\alpha_s-1))(2-\alpha_s-\beta_s)} \right) \tag{4-25}$$

$$Pr(\mathrm{Off}) = \left(\frac{(1-\beta_s)}{(2-\alpha_s-\beta_s)} - \frac{(1-\beta_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{(1-\alpha_t(\beta_s+\alpha_s-1))(2-\alpha_s-\beta_s)} \right) \tag{4-26}$$

For the same reason, cell losses occur only if the source is On when the token generator's tagged period terminates, so the average number of cells lost during the source's multiple tagged periods will be:

$$\overline{\bar{N}_{BH}} = 0 \cdot Pr(\mathrm{Off}) + \overline{\bar{N}_{BH.k_s > k_t}}(Pr)(\mathrm{On}) \tag{4-27}$$

Here $\overline{\bar{N}_{BH.k_s > k_t}}$ is the average number of cells lost if the source is On when the token generator's tagged period terminates. Due to the fact that the source and the token generator's geometrically distributed tagged periods have memoryless property, this is same as in case UI which has been derived in equation (4-13).

Combining equations (4-25), (4-26) and (4-13), we obtain the average number of cells lost during the source's multiple tagged periods in case UII.2 as:

$$\overline{\bar{N}_{BH}} = \frac{(1-\alpha_t)\alpha_s}{(1-\alpha_s)(1-\alpha_s\alpha_t)} \cdot \left(\frac{(1-\alpha_s)}{(2-\alpha_s-\beta_s)} + \frac{(1-\alpha_s)(1-\alpha_t)(\beta_s+\alpha_s-1)}{(1-\alpha_t(\beta_s+\alpha_s-1))(2-\alpha_s-\beta_s)}\right) \tag{4-28}$$

Since cases UI, UII.1 and UII.2 have different number of cell losses and successfully transmitted cells during the source's tagged periods, they will have different cell loss probabilities during a random cycle. Next we need to determine the probabilities for each of these cases.

4.4.3. Probabilities of cases UI, UII.1 and UII.2

The differentiation between cases UI and UII is that the source is On for case UI and Off for case UII when the token generator's initial Off period terminates. So we need to determine the probabilities of the source's different state at time B when the token generator's initial Off period terminates, seeing figure 4.1.

Define:

$A_{l_t}(y)$ = the transient marginal PGF of the source's state at the end of l_t'th slot during the token generator's initial Off period

Similarly, this transient marginal PGF will be determined by the initial conditions when the token generator's initial Off period begins, which have been given in equations (3-1) and (3-2) in chapter 3.

Substituting equation (3-2) into the PGF result in equation(2-30), we have:

$$A_{l_t}(y) = Q_{l_t}(z,y)\Big|_{z=1} = X(l_t)\Big|_{z=1} = \left(C_1\lambda_1^{l_t} + C_2\lambda_2^{l_t}\right)\Big|_{z=1} \tag{4-29}$$

Substituting $c_{1,2}$ and $\lambda_{1,2}$ into it, we have:

$$A_{l_t}(y) = \frac{(1-\beta_s)(1-y)}{2-\alpha_s-\beta_s}(\alpha_s+\beta_s-1)^{l_t} + \frac{(1-\alpha_s)+y(1-\beta_s)}{2-\alpha_s-\beta_s} \tag{4-30}$$

Define:

$A_{off}(y)$ = the marginal PGF of the state of the source at the end of the token generator's initial Off period

Averaging with respect to the duration of the token generator's Off period, we get:

$$A_{off}(y) = \sum_{l_t=1}^{\infty} A_{l_t}(y)(1-\beta_t)\beta_t^{l_t-1} \tag{4-31}$$

$$= \left[\frac{1-\alpha_s}{2-\alpha_s-\beta_s} + \frac{(1-\beta_t)(1-\beta_s)(\beta_s+\alpha_s-1)}{(2-\alpha_s-\beta_s)(1-\beta_t(\beta_s+\alpha_s-1))}\right] + y\left[\frac{1-\beta_s}{2-\alpha_s-\beta_s} - \frac{(1-\beta_t)(1-\beta_s)(\beta_s+\alpha_s-1)}{(2-\alpha_s-\beta_s)(1-\beta_t(\beta_s+\alpha_s-1))}\right]$$

Define:

$Pr(\mathrm{UI}) = Pr$(case UI occurs)

$Pr(\mathrm{UII}) = Pr$(case UII occurs)

Obviously, the source is On when case UI occurs and Off when case UII occurs, then the marginal PGF $A_{off}(y)$ can be expressed as:

$$A_{off}(y) = Pr(\mathrm{UII}) + y \cdot Pr(\mathrm{UI}) \tag{4-32}$$

Compare equations (4-31) and (4-32), we obtain:

$$Pr(\mathrm{UI}) = \frac{1-\beta_s}{2-\alpha_s-\beta_s} - \frac{(1-\beta_t)(1-\beta_s)(\beta_s+\alpha_s-1)}{(2-\alpha_s-\beta_s)(1-\beta_t(\beta_s+\alpha_s-1))} \tag{4-33}$$

$$Pr(\mathrm{UII}) = \frac{1-\alpha_s}{2-\alpha_s-\beta_s} + \frac{(1-\beta_t)(1-\beta_s)(\beta_s+\alpha_s-1)}{(2-\alpha_s-\beta_s)(1-\beta_t(\beta_s+\alpha_s-1))} \tag{4-34}$$

Because the further separate cases UII.1 and UII.2 in case UII have different numbers of cells lost and transmitted during the source's tagged periods, which result in different CLPs, next we need to determine the probabilities of case UII.1 and UII.2. Define:

$Pr(\mathrm{UII.1/UII}) = Pr$(separate case UII.1 occurs given that case UII occurs)

$Pr(\mathrm{UII.2/UII}) = Pr$(separate case UII.2 occurs given that case UII occurs)

From figure 4.3, we can see that the difference between these two separate cases is that the token generator's initial On period is longer than the residual duration of the source's Off period after time B for case UII.1 and shorter for case UII.2, so we can obtain these two probabilities as:

$$Pr(\mathrm{UII.1/UII}) = Pr(k_t > l_s) = \sum_{l_s=1}^{\infty} \sum_{k_t=l_s+1}^{\infty} Pr(k_t)Pr(l_s) = \frac{\alpha_t(1-\beta_s)}{1-\alpha_t\beta_s} \tag{4-35}$$

$$Pr(\mathrm{UII.2/UII}) = 1 - Pr(\mathrm{UII.1/UII}) = 1 - \frac{\alpha_t(1-\beta_s)}{1-\alpha_t\beta_s} = \frac{1-\alpha_t}{1-\alpha_t\beta_s} \tag{4-36}$$

Next, we will combine these cases and determine the average CLP during a random cycle.

4.4.4 The average Cell Loss Probability during a random cycle

Let us define:

$\overline{N_{on}}$ = the average number of cells successfully transmitted during a random cycle

$\overline{N_{off}}$ = the average number of cells lost during a random cycle

As we have stated, the number of cells lost and transmitted during the source's tagged periods for cases UI and UII.2 is identical but different for case UII.1, and the number of cells lost during the token generator's initial Off period is same for all cases, so we can define the following CLPs and determine them as:

$P_{loss}(1)$ = the cell loss probability during a random cycle in cases UI and UII.2

$P_{loss}(2)$ = the cell loss probability during a random cycle in case UII.1

$$P_{loss}(1) = \frac{\overline{N_{off}}}{\overline{N_{off}}+\overline{N_{on}}} = \frac{\overline{\tilde{N}_{AB}}+\overline{\tilde{N}_{BG}}}{\overline{\tilde{N}_{AB}}+\overline{\tilde{N}_{BG}}+\overline{N_{BG}}} \qquad (4\text{-}37)$$

here $\overline{\tilde{N}_{AB}}$, $\overline{\tilde{N}_{BG}}$ and $\overline{N_{BG}}$ have been determined in equations (3-6), (4-14) and (4-6) respectively for cases UI and UII.2.

$$P_{loss}(2) = \frac{\overline{N_{off}}}{\overline{N_{off}}+\overline{N_{on}}} = \frac{\overline{\tilde{N}_{AB}}+\overline{\tilde{N}_{BH}}}{\overline{\tilde{N}_{AB}}+\overline{\tilde{N}_{BH}}+\overline{N_{BH}}} \qquad (4\text{-}38)$$

here $\overline{\tilde{N}_{AB}}$, $\overline{\tilde{N}_{BH}}$ and $\overline{N_{BH}}$ have been determined in equations (3-6), (4-28) and (4-

20) respectively for case UII.1.

Define:

P_{loss} = the cell loss probability during a random cycle

Combine equations (4-33), (4-34), (4-35), (4-36), (4-37) and (4-38) together, we have:

$$P_{loss} = P_{loss}(1)[Pr(\mathrm{UI}) + Pr(\mathrm{UII.2/UII})Pr(\mathrm{UII})] + P_{loss}(2)Pr(\mathrm{UII.1/UII})Pr(\mathrm{UII}) \qquad (4\text{-}39)$$

As we have stated, the unforced protocol is designed to police a bursty source's ACR more fairly, so next we will determine the output ACR for both unforced and forced protocols, then in the mathematical results we will compare both the CLP and the output ACR for these two protocols.

4.4.5. The output ACR from the modified LB

Here we will determine the output ACR during one random cycle since we have assumed that the time axis is divided into identically distributed random cycles.

4.4.5.1 The output ACR for the unforced protocol

Let us define:

ACR = the output ACR from the modified Leaky Bucket during a random cycle

$\bar{R}$ = the average duration of a random cycle in number of slots

Obviously the output ACR can be expressed as:

$$ACR = \frac{\overline{N_{on}}}{\bar{R}} \tag{4-40}$$

here $\overline{N_{on}}$ is the average number of cells transmitted into the network during a random cycle which equals to the average number of cells transmitted during the source's tagged periods and has been derived in section 4.4.4. It has the same value for cases UI and UII.2 but a different value for case UII.2.

From Figures 4.2, 4.4 and 4.5, it is obvious that the average duration of a random cycle $\bar{R}$ is different for cases UI, UII.1 and UII.2, we let $\overline{R_{UI}}$, $\overline{R_{UII;1}}$ and $\overline{R_{UII;2}}$ denote them respectively. From these figures we can obtain:

$$\overline{R_{UI}} = \bar{l}_t + \bar{k}_t + \overline{\bar{N}_{BG}}$$

Where $\bar{l}_t$ and $\bar{k}_t$ are the average duration of the token generator's Off and On period respectively, $\overline{\bar{N}_{BG}}$ is the average residual duration of the source's last tagged period which has been derived in equation (4-14). So $\overline{R_{UI}}$ becomes:

$$\overline{R_{UI}} = \frac{1}{1-\beta_t} + \frac{1}{1-\alpha_t} + \overline{\bar{N}_{BG}} \tag{4-41}$$

Similarly, from figures 4.4. and 4.5, we get the following:

$$\overline{R_{UII,1}} = \bar{l}_t + \bar{k}_t + \overline{N_{BH}} = \frac{1}{1-\beta_t} + \frac{1}{1-\alpha_t} + \overline{N_{BH}} \qquad (4\text{-}42)$$

here $\overline{N_{BH}}$ is given by equation (4-28).

$$\overline{R_{UII,2}} = \bar{l}_t + \bar{k}_t + \bar{l}_s + \overline{N_{BG}} = \frac{1}{1-\beta_t} + \frac{1}{1-\alpha_t} + \frac{1}{1-\alpha_s} + \overline{N_{BG}} \qquad (4\text{-}43)$$

here $\overline{N_{BG}}$ is same as in case UI which is given in equation (4-14).

Then from equation (4-40), we obtain the output ACR for the cases UI, UII.1 and UII.2 respectively as:

$$ACR_{UI} = \frac{\overline{N_{BG}}}{\overline{R_{UI}}}$$ here $\overline{N_{BG}}$ is given in equation (4-6) for case UI (4-44)

$$ACR_{UII;1} = \frac{\overline{N_{BH}}}{\overline{R_{UII.1}}}$$ here $\overline{N_{BH}}$ is given in equation (4-20) for case UII.1 (4-45)

$$ACR_{UII;2} = \frac{\overline{N_{BG}}}{\overline{R_{UII.2}}}$$ here $\overline{N_{BG}}$ is given in equation (4-6) for case UII.2 (4-46)

Substituting the probabilities of these cases as being given in the section 4.4.4 into the above, we can get the output ACR from the modified LB for the unforced protocol as:

$$ACR = ACR_{UI}Pr(\mathrm{UI}) + ACR_{UII;1}Pr(\mathrm{UII.1/UII})Pr(\mathrm{UII}) + ACR_{UII;2}Pr(\mathrm{UII.2/UII})Pr(\mathrm{UII}) \qquad (4\text{-}47)$$

4.4.5.2 The output ACR for the forced protocol

Using the same methodology as in the unforced protocol, we can get the average duration of a random cycle for cases FI, FII.1 and FII.2 from Figures 3.2, 3.4 and 3.5 in the following:

$$\overline{R_{FI}} = \bar{l}_t + \bar{k}_t + \overline{\bar{N}_{BG}} = \frac{1}{1-\beta_t} + \frac{1}{1-\alpha_t} + \overline{\bar{N}_{BG}} \tag{4-48}$$

$$\overline{R_{FII;1}} = \bar{l}_t + \bar{k}_t + \overline{\bar{N}_{BG}} = \frac{1}{1-\beta_t} + \frac{1}{1-\alpha_t} + \overline{\bar{N}_{BG}} \tag{4-49}$$

$$\overline{R_{FII;2}} = \bar{l}_t + \bar{k}_t + \bar{l}_s + \overline{\bar{N}_{BG}} = \frac{1}{1-\beta_t} + \frac{1}{1-\alpha_t} + \frac{1}{1-\alpha_s} + \overline{\bar{N}_{BG}} \tag{4-50}$$

here $\overline{\bar{N}_{BG}}$ is the average residual duration of the source's tagged period after the token generator's tagged period, which is same for cases FI, FII.1 and FII.2 as given in equation (3-17).

Then the output ACR for the cases FI, FII.1 and FII.2 can be obtained as:

$$ACR_{FI} = \frac{\overline{N_{BG}}}{\overline{R_{FI}}} \tag{4-51}$$

$$ACR_{FII;1} = \frac{\overline{N_{BG}}}{\overline{R_{FII;1}}} \tag{4-52}$$

$$ACR_{FII;2} = \frac{\overline{N_{BG}}}{\overline{R_{FII;2}}} \tag{4-53}$$

here $\overline{N_{BG}}$ is same for cases FI, FII.1 and FII.2 too and is given in equation (3-20).

Substituting the probabilities of these cases which have the same value as in the unforced protocol, as we have determined in section 4.4.4, we can obtain the output ACR from the modified LB for the forced protocol as:

$$ACR = ACR_{FI}Pr(FI) + ACR_{FII;1}Pr(FII.1/FII)Pr(FII) + ACR_{FII;2}Pr(FII.2/FII)Pr(FII) \qquad (4\text{-}54)$$

Next, we will give the mathematical results for the unforced protocol and compare it with the forced protocol.

4.5 Numerical Results

In this section, we present numerical results for the average CLP and the output ACR.

4.5.1 The average CLP during a random cycle

Equation (4-39) gives the average cell loss probability during a random cycle under the unforced protocol. Together with equation (3-23), we compare the average CLP for the unforced and forced protocols in Figure 4.6.

We can see that the unforced protocol always gives a lower CLP than the forced protocol, because it allows the source's multiple tagged periods to be successfully transmitted into the network during a random cycle. We also notice that when α_s has lower values, the unforced protocol gives different CLPs for different

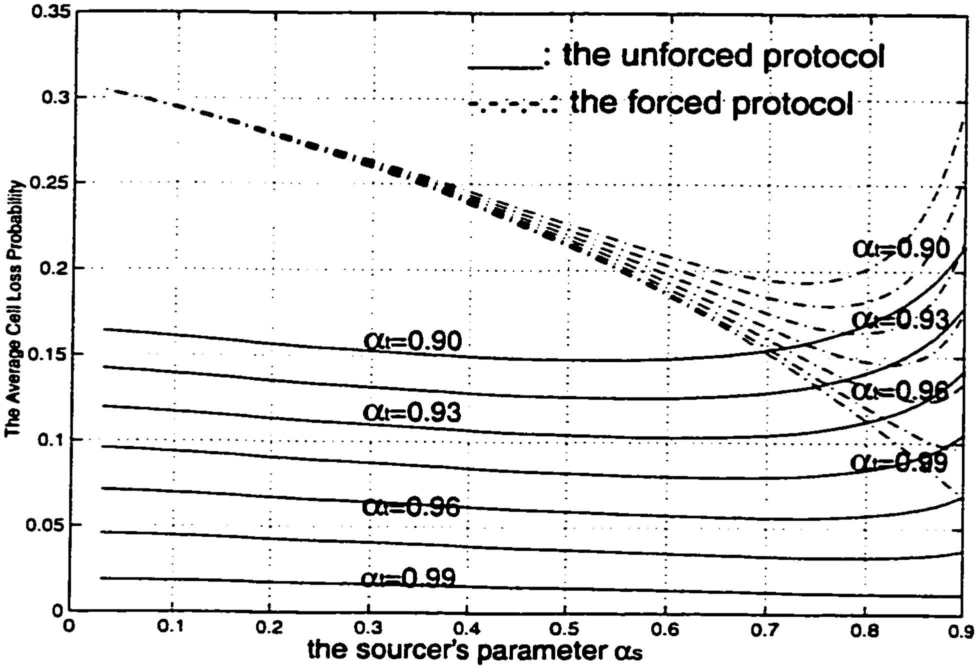


Figure 4.6: The Comparison of the forced and unforced protocol when βs=0.7 and βt=0.4

values of the token generator's On parameter α_t, but the forced protocol gives closed CLPs. This is because the different durations of the token generator's tagged On period can host different number of the source's tagged periods in the unforced protocol, so the number of cells transmitted into the network would be different. On the other hand, the forced protocol only allows cells to be transmitted into the network during the source's single tagged period no matter what the duration of the token generator's tagged period is.

Another phenomena in the figure is that the CLP will reach a minimum value as we increase the source's On parameter α_s, after this point the CLP will increase again, this is more obvious for the forced protocol. Because when α_s is increased from a lower value, the number of cells transmitted will increase much faster than the number of cells lost. As α_s reaches a certain value, the source's tagged period becomes too long so the number of cells lost during the residual duration of the source's tagged period after the token generator's tagged period will increase fast, leading the CLP starts to increase. (This can also be observed from the following Table 4.1).

Table 4.1 shows more details of the comparison of the forced and unforced protocol. It is composed of 5 groups of data, in each group α_t, β_t and β_s are same but α_s is different. We make the following observations:

1). When β_t is increased while keeping everything else constant, the number of cells lost is increased for both protocols, this is because the token generator's Off period is longer so cell losses during this period will also increase. For example this may be seen from the comparison of the CLPs in rows 1 and 6.

2). When β_s is decreased with everything else being kept constant, the number of cell losses increase for the forced protocol, because the source has a shorter Off period so the number of cells lost during the token generator's Off period is increased, the number of cells transmitted is not affected because the source's tagged period is same. For the unforced protocol, on the other hand, the number of cells transmitted is increased by the decrease of β_s, because the source's shorter Off periods enable more tagged periods to be hosted during the token generator's tagged period so more cells are transmitted. This may be seen from the comparison of CLP results in rows 1 and 11.

Table 4.1: The Comparison of the forced and unforced protocol

Row	The source			The token generator			Forced Protocol			Unforced Protocol		
#	α_s	β_s	$\frac{1}{1-\alpha_s}$	α_t	β_t	$\frac{1}{1-\alpha_t}$	$\overline{N_{on}}$	$\overline{N_{off}}$	CLP	$\overline{N_{on}}$	$\overline{N_{off}}$	CLP
1	0.9	0.7	10	0.9	0.4	10	7.5	3.25	0.29	7.5	2.0	0.20
2	0.8	0.7	5	0.9	0.4	10	4.5	1.1.4	0.20	6.0	1.2	0.17
3	0.7	0.7	3.3	0.9	0.4	10	3.2	0.77	0.19	5.0	0.91	0.15
4	0.6	0.7	2.5	0.9	0.4	10	2.4	0.64	0.21	4.3	0.75	0.15
5	0.5	0.7	2.0	0.9	0.4	10	2.0	0.15	0.22	3.8	0.66	0.15
6	0.9	0.7	10	0.9	**0.7**	10	7.5	4.2	0.36	7.5	3.0	0.29
7	0.8	0.7	5.0	0.9	**0.7**	10	4.5	2.0	0.31	6.0	2.1	0.26
8	0.7	0.7	3.3	0.9	**0.7**	10	3.2	1.6	0.33	5.0	1.7	0.25
9	0.6	0.7	2.5	0.9	**0.7**	10	2.4	1.3	0.36	4.3	1.4	0.25
10	0.5	0.7	2.0	0.9	**0.7**	10	2.0	1.2	0.38	3.8	1.3	0.25
11	0.9	**0.5**	10	0.9	0.4	10	7.5	3.48	0.32	8.23	1.83	0.18
12	0.8	**0.5**	5.0	0.9	0.4	10	4.5	1.46	0.25	7.11	1.36	0.16
13	0.7	**0.5**	3.3	0.9	0.4	10	3.2	1.07	0.25	6.25	1.14	0.15
14	0.6	**0.5**	2.5	0.9	0.4	10	2.4	0.94	0.28	5.6	1.01	0.15
15	0.5	**0.5**	2.0	0.9	0.4	10	2.0	0.87	0.31	5.0	0.92	0.15
16	0.9	0.7	10	**0.95**	0.4	20	8.9	1.73	0.16	14.6	1.5	0.09
17	0.8	0.7	5.0	**0.95**	0.4	20	4.8	0.80	0.14	11.9	1.0	0.07
18	0.7	0.7	3.3	**0.95**	0.4	20	3.3	0.65	0.16	10.0	0.77	0.07
19	0.6	0.7	2.5	**0.95**	0.4	20	2.5	0.59	0.19	8.6	0.67	0.07
20	0.5	0.7	2,0	**0.95**	0.4	20	2.0	0.55	0.22	7.5	0.60	0.07
21	0.9	0.7	10	**0.99**	0.4	100	9.9	0.73	0.07	75	0.87	0.01
22	0.8	0.7	5.0	**0.99**	0.4	100	5.0	0.63	0.11	60	0.70	0.01
23	0.7	0.7	3.3	**0.99**	0.4	100	3.3	0.60	0.15	50	0.63	0.01
24	0.6	0.7	2.5	**0.99**	0.4	100	2.5	0.57	0.19	43	0.60	0.01
25	0.5	0.7	2.0	**0.99**	0.4	100	2.0	0.54	0.21	37	0.56	0.01

3). When α_t is increased, the number of cells transmitted increases and the number of cell losses decreases for both protocols, because the token generator's longer tagged period allows more cells being transmitted and less cells being lost.

We also notice that the forced protocol limits the number of cells being successfully transmitted during a random cycle to a burst of the source even when α_t is very high. For the unforced protocol, it can transmit multiple tagged periods of the source, so the number of cells transmitted increases dramatically as α_t is increased to give a longer token generator's tagged period. From the results in row 21 in the table, it can be seen that 75 cells can be transmitted with the unforced protocol and 9.9 cells can be transmitted with the forced protocol, while the average durations of the source and the token generator's tagged periods are 10 and 100 time slots respectively. This will result in different output ACR as we will present in the next section.

4). When α_s is decreased, the number of cells transmitted in the forced protocol is decreased much more dramatically than in the unforced protocol when α_t has a higher value, as may be seen from the results of row 21. Because a shorter tagged period of the source causes much less cells being transmitted in the forced protocol. But in the unforced protocol, multiple tagged periods can be transmitted, so the decrease in the number of cells transmitted is slower.

4.5.2 The output ACR during a random cycle

Equations (4-47) and (4-54) give the output ACR from the modified LB during a random cycle. To compare the effectiveness of the forced and unforced protocol in terms of policing a source's ACR, first we need to determine the source's ACR.

A geometrically distributed bursty source's ACR can be expressed by:

$$ACR = \frac{\bar{k}_s}{\bar{k}_s + \bar{l}_s} = \frac{\frac{1}{1-\alpha_s}}{\frac{1}{1-\alpha_s} + \frac{1}{1-\beta_s}} = \frac{1-\beta_s}{2-\alpha_s-\beta_s} \qquad (4\text{-}55)$$

Here $\bar{k}_s$ and $\bar{l}_s$ is the average duration of the source's On and Off period respectively.

From equations (4-47), (4-54) and (4-55), we obtain the following figure 4.7.

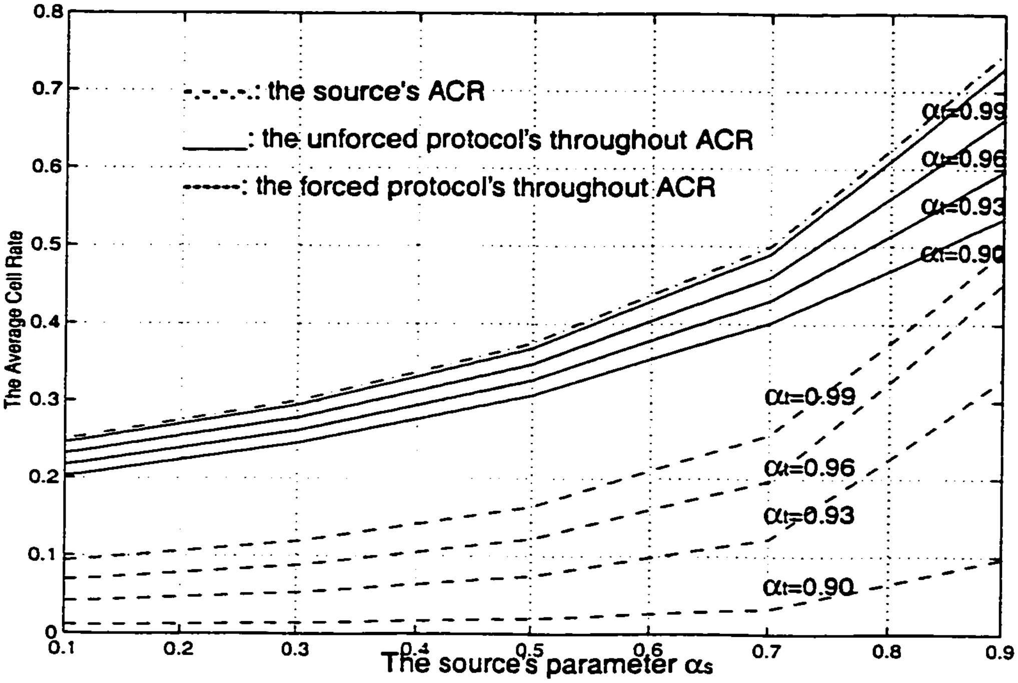


Figure 4.7: The comparison of the throughout ACR for forced and unforced protocols when βs=0.7, βt=0.4

Obviously, the unforced protocol achieves a higher output ACR than the forced protocol, which tracks closely the source's ACR when the token generator's parameter α_t is chosen to have a higher value, for example 0.99. As we can see, the forced protocol has a much lower output ACR for a more bursty source which has a lower ACR, on the other hand, the unforced protocol can provide a very closed throughout ACR for all sources if α_t is selected to have a high value, this indicates the unforced protocol's effectiveness on policing a source's ACR.

As a summary to the above numerical results, we can see that for a given pair of the bursty source's parameters α_s and β_s, a token generator's higher α_t and lower β_t parameter values will give a lower CLP and a higher output ACR for both forced and unforced protocols, while the unforced protocol always performs better than the forced one. When the source is very bursty with a lower ACR and a lower α_s value, the token generator's higher α_t and lower β_t values still can give a very good result of CLP and output ACR under the unforced protocol, while the forced protocol does not perform well with this type of sources. This is because the unforced protocol gives the source the opportunity to transmit multiple bursts into the network while the forced protocol only allow the source's one burst being transmitted during a random cycle.

As a conclusion to the above analysis and observations, the modified LB can police a bursty source's PCR since tokens are generated at the same speed as the source's PCR. To police a source's ACR, the token generator's parameters α_t and β_t have to be properly chosen. Because of the statistical distributions of the source and the token generator's On and Off periods, we cannot simply choose them as identical as the source's α_s and β_s. But eventually a set of α_t and

β_t can be reached depending on the value of α_s and β_s to police a source's ACR. Compared to the forced protocol, the unforced protocol can better realize the modified LB's objective of policing a source's PCR and ACR simultaneously.

4.6 The advantage and disadvantage of the unforced protocol

As we have stated, the unforced protocol can police the ACR more fairly, this is the advantage of the unforced protocol, which achieves the promise made for this protocol.

But there is a trade-off in the unforced protocol. As we can see, during the token generator's tagged period, the source may alternate between On and Off states. When the source is Off, no cells are generated, but the token generator still generates tokens and these tokens are wasted.

Next, we will propose *Compensation Protocol* to take advantage of these wasted tokens. We will also give the comparison of the modified LB to a traditional LB.

Chapter 5

Compensation Protocol and Comparisons

5.1 Introduction

In chapters 3 and 4, we have determined the average cell loss probability during a random cycle under the forced and unforced protocols.

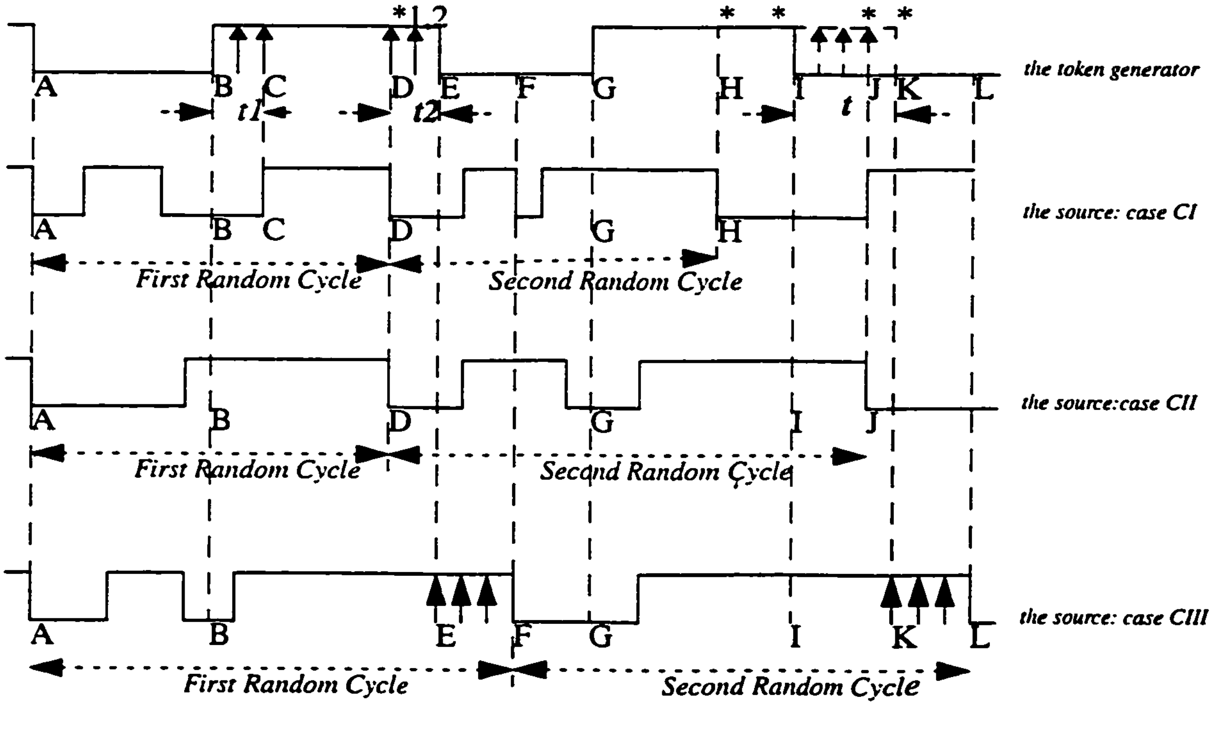
As we have described, during a random cycle, when the token generator first makes a transition from Off to On, the source might be On or Off. If the source is Off at this moment, tokens will be wasted until the source becomes On. In the unforced protocol, during the token generator's tagged period, the source might be Off, so some tokens would be wasted too. These wasted tokens imply that the network resources which have been assigned to this source is not effectively used.

In order to take make use of the network resources assigned to the source, in this chapter we propose *Compensation Protocol* which operates in two random cycles. In the first part of this chapter, we will describe this protocol and perform the CLP analysis. Since the unforced protocol would make the mathematical analysis for the compensation method untractable, here we will only discuss this compensation method under the operation of the forced protocol in each random cycle (which has been described in Chapter 3).

The second part of this chapter is devoted to comparisons. First, we will compare the three protocols in our modified leaky bucket mechanism, i.e. the forced, unforced and compensation protocols, then we will make a comparison with a traditional leaky bucket mechanism, which has been studied by Hluchyj et al. in [9].

5.2 Description of the Compensation Protocol

As we have stated, in this thesis the compensation protocol will obly be discussed under the operation of the forced protocol, so the definition of a random cycle will be same as the one described in the forced protocol in section 3.2. The following figure 5.1 shows the compensation protocol which is designed to operate in two random cycles.



Legend: A, B ...L: the time points that the token generaotr or the source changes state

*: the time points that the token generator is forced to change state

Figure 5.1 The Compensation Protocol in Two Random Cycles

Depending on the source's state when the token generator's tagged period starts and terminates in the first ransom cycle, there exist three scenarios which cause waste of tokens, they are shown as cases CI, CII and CIII in Figure 5.1. Below we describe the compensation protocol in respect to the two random cycles:

(i). During the first random cycle: when the token generator first makes a transition from Off to On at time B, the source's state might be Off (in case CI and CIII) or On (in case CII), thus tokens generated during the interval B to C are wasted in cases CI and CIII, which is presented by *t1*. When the source's tagged period terminates at time D in cases CI and CII during the token generator's tagged period, the forced protocol will operate to force the token generator to change to the Off state at time D and next random cycle starts. But the token generator would remain On until time E according to its statistical distribution, as a result some more tokens would be wasted, which is presented by *t2*. So we have:

In case CI: (*t1*+*t2*) tokens are wasted before and after the source's tagged period in the first random cycle.

In case CII: *t2* tokens are wasted after the source's tagged period in the first random cycle.

In case CIII: *t1* tokens are wasted before the source's tagged period in the first random cycle.

(ii). During the second random cycle: in order to take advantage of these wasted tokens, we will implement a compensation operation in the second random cycle which is described as below:

(ii.1). If the source's tagged period terminates during the token generator's tagged period, i.e. at time H as shown in case CI, then the token generator will be forced to change to Off state at H and next random cycle starts immediately. This is the same operation as used in the first random cycle.

(ii.2). If the source's tagged period is longer than the token generator's tagged period as shown in cases CII and CIII, then the token generator's tagged period is forced to extend an extra $t=t1+t2$ time slots. i.e. from time I to K, then:

if: the source's tagged period terminates during these extended time slots as shown in case CII at time J, then the token generator is forced to become Off too at time J and thenext random cycle starts immediately;

if: the source's tagged period terminates after these extended time slots as shown in case CIII at time L, then the second random cycle terminates at time L and the next random cycle starts immediately. From time K to L the token generator is not allowed to become On again.

From the above description, we can see that the compensation protocol differs from the forced protocol only in the second random cycle, in which it extends the token generator's tagged period for a certain time slots, which equals to the number of tokens wasted in the first random cycle. Next, we will determine the average

CLP of this compensation protocol during two random cycles.

5.3. Analysis of the Cell Loss Probability

Referring to figure 5.1, we let the time interval A to F denote the first random cycle, and the interval F to L denote the second random cycle. So we define:

$\tilde{N}_{AB}$ = the number of cells lost during the token generator's initial Off period in the first random cycle

N_{BF} = the number of cells successfully transmitted into the network during the source's tagged period in the first random cycle

$\tilde{N}_{BF}$ = the number of cells lost during the source's tagged period in the first random cycle

$\tilde{N}_{FG}$ = the number of cells lost during the token generator's initial Off period in the second random cycle

N_{GL} = the number of cells successfully transmitted into the network during the source's tagged period in the second random cycle

$\tilde{N}_{GL}$ = the number of cells lost during the source's tagged period in the second random cycle

$\bar{t_1}$ = the average number of tokens wasted *before* the source's tagged period starts during the token generator's tagged period in the first random cycle

$\bar{t_2}$ = the average number of tokens wasted *after* the source's tagged period terminates in the first random cycle

$\bar{t}$ = the average number of tokens wasted in the first random cycle

j_s = the number of time slots of the source's Off period

j_t = the number of time slots of the token generator's Off period

Same as the analysis in section 3.2.3, we can obtain:

$$\overline{\tilde{N}_{BF,k_s>k_t}} = \sum_{k_t=1}^{\infty}\sum_{k_s=k_t}^{\infty}(k_s-k_t)(1-\alpha_s)\alpha_s^{k_s-1}(1-\alpha_t)\alpha_t^{k_t-1} = \frac{(1-\alpha_t)\alpha_s}{(1-\alpha_s)(1-\alpha_s\alpha_t)} \qquad \text{(5-1)}$$

$$Pr(k_s>k_t) = \sum_{k_t=1}^{\infty}\sum_{k_s=k_t}^{\infty}Pr(k_s)Pr(k_t) = \frac{1-\alpha_t}{1-\alpha_t\alpha_s} \qquad \text{(5-2)}$$

$$\overline{\tilde{N}_{BF}} = 0\cdot Pr(k_s\le k_t)+\overline{\tilde{N}_{BF,k_s>k_t}}Pr(k_s>k_t) = \frac{(1-\alpha_t)^2\alpha_s}{(1-\alpha_s)(1-\alpha_s\alpha_t)^2} \qquad \text{(5-3)}$$

$$\overline{N_{BF}} = \bar{k}_s-\overline{\tilde{N}_{BF}} = \frac{1}{1-\alpha_s}-\overline{\tilde{N}_{BF}} = \frac{(1-\alpha_s\alpha_t)^2-((1-\alpha_t)^2\cdot\alpha_s)}{(1-\alpha_s)(1-\alpha_s\alpha_t)^2} \qquad \text{(5-4)}$$

From figure 5.1, we can obtain the average number of tokens wasted during the first random cycle as:

$$\bar{t}_1 = \sum_{j_t=1}^{\infty}\sum_{j_s=j_t}^{\infty}(j_s-j_t)(1-\beta_t)\alpha_t^{j_t-1}(1-\beta_s)\alpha_s^{j_s-1} = \frac{(1-\beta_s)\beta_t}{(1-\beta_t)(1-\beta_t\beta_s)} \qquad \text{(5-5)}$$

$$\bar{t}_2 = \sum_{k_s=1}^{\infty}\sum_{k_t=k_s}^{\infty}(k_t-k_s)(1-\alpha_t)\alpha_t^{k_t-1}(1-\alpha_s)\alpha_s^{k_s-1} = \frac{(1-\alpha_s)\alpha_t}{(1-\alpha_t)(1-\alpha_t\alpha_s)} \qquad \text{(5-6)}$$

$$\bar{i} = \bar{i}_1 + \bar{i}_2 = \frac{(1-\beta_s)\beta_t}{(1-\beta_t)(1-\beta_t\beta_s)} + \frac{(1-\alpha_s)\alpha_t}{(1-\alpha_t)(1-\alpha_t\alpha_s)} \tag{5-7}$$

From the operation of the compensation protocol in the second random cycle, we can obtain the average value of $\tilde{N}_{GL}$ as:

$$\overline{\tilde{N}_{GL,k_s>k_t+\bar{i}}} = \sum_{k_t=1}^{\infty}\sum_{k_s=k_t+\bar{i}}^{\infty}(k_s-k_t-\bar{i})(1-\alpha_s){\alpha_s}^{k_s-1}(1-\alpha_t){\alpha_t}^{k_t-1}$$

$$= \frac{(1-\alpha_t)\alpha_s^{\bar{i}}}{(1-\alpha_s\alpha_t)}\left(\frac{\alpha_s}{1-\alpha_s}-\bar{i}\right) \tag{5-8}$$

$$Pr(k_s>k_t+\bar{i}) = \sum_{k_t=1}^{\infty}\sum_{k_s=k_t+\bar{i}}^{\infty}Pr(k_s)Pr(k_t) = \frac{(1-\alpha_t)\alpha_s^{\bar{i}}}{1-\alpha_t\alpha_s} \tag{5-9}$$

$$\overline{\tilde{N}_{GL}} = 0\cdot Pr(k_s\le k_t+\bar{i}) + \overline{\tilde{N}_{BG,k_s>k_t+\bar{i}}}(Pr(k_s>k_t+\bar{i}))$$

$$= \frac{(1-\alpha_t)^2\alpha_s^{2\bar{i}}}{(1-\alpha_s\alpha_t)^2}\left(\frac{\alpha_s}{1-\alpha_s}-\bar{i}\right) \tag{5-10}$$

$$\overline{N_{GL}} = \bar{k}_s - \overline{\tilde{N}_{GL}} = \frac{1}{1-\alpha_s} - \frac{(1-\alpha_t)^2\alpha_s^{2\bar{i}}}{(1-\alpha_s\alpha_t)^2}\left(\frac{\alpha_s}{1-\alpha_s}-\bar{i}\right) \tag{5-11}$$

Define:

$\overline{N_{on,1}}$ = the average number of cells successfully transmitted during the first random cycle

$\overline{N_{off,1}}$ = the average number of cells lost during the first random cycle

$\overline{N_{on,2}}$ = the average number of cells successfully transmitted during the second random cycle

$\overline{N_{off,2}}$ = the average number of cells lost during the second random cycle

From figure 5.1, we have:

$$\overline{N_{on,1}} = \overline{N_{BF}} \tag{5-12}$$

$$\overline{N_{off,1}} = \overline{\bar{N}_{AB}} + \overline{\bar{N}_{BF}} \tag{5-13}$$

$$\overline{N_{on,2}} = \overline{N_{GL}} \tag{5-14}$$

$$\overline{N_{off,2}} = \overline{\bar{N}_{FG}} + \overline{\bar{N}_{GL}} \tag{5-15}$$

Here $\overline{\bar{N}_{AB}}$ and $\overline{\bar{N}_{FG}}$ are the average number of cells lost during the token generator's initial Off period in the first and second random cycle, which are same as being given in equation (3-6). Let us define:

P_{loss} = the average cell loss probability during two random cycles for the compensation protocol

Combine equations (3-6), (5-3), (5-4), (5-10), and (5-11) into above, we have:

$$P_{loss} = \frac{\overline{N_{off,1}} + \overline{N_{off,2}}}{\overline{N_{on,1}} + \overline{N_{on,2}} + \overline{N_{off,1}} + \overline{N_{off,2}}} \tag{5-16}$$

5.4 Numerical Results

From equation (5-16) and the results obtained in previous chapters, we can obtain the following figure 5.2 as a comparison of the three protocols in the modified leaky bucket mechanism.

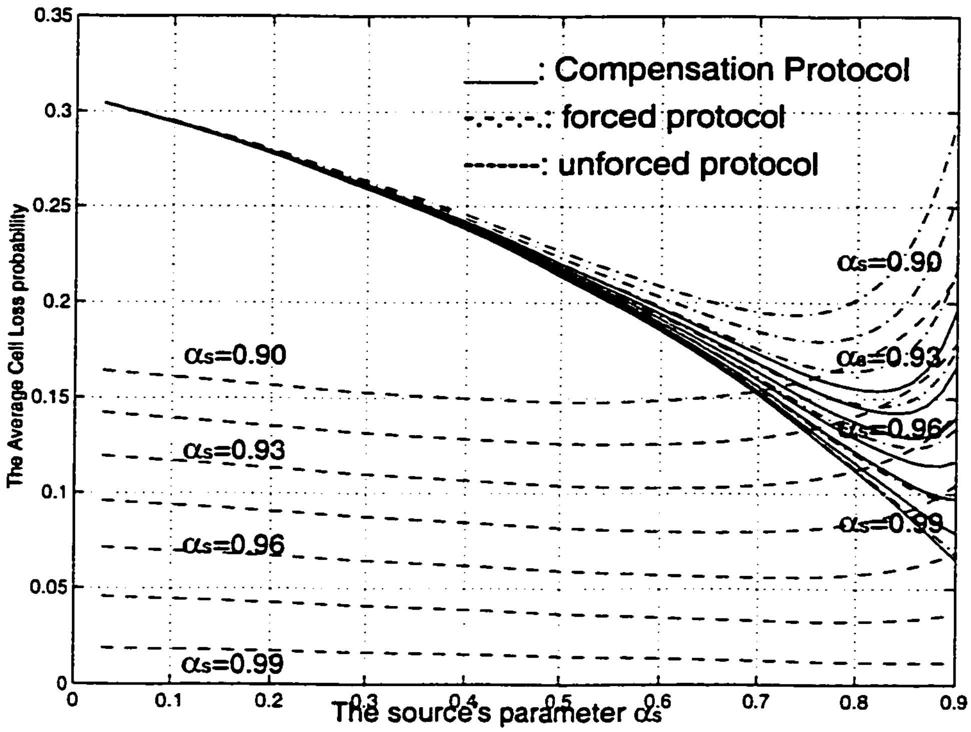


Figure 5.2 The Comparison of the three protocols when $\beta_s=0.7$ and $\beta_t=0.4$

Obviously, the compensation protocol has a lower cell loss probability than the forced protocol, especially with a higher α_s, because a higher α_s gives the source opportunity to take better use of the wasted tokens in the second random cycle.

As expected, the unforced protocol still gives a lower CLP than the compensation protocol because of its ability to transmit multiple tagged periods of the source during the token generator's one tagged period.

5.5 Comparison of the modified leaky Bucket with the traditional leaky bucket mechanism

In [9], Hluchyj et al. have performed analysis on the traditional Leaky Bucket algorithm for bursty sources using the fluid-flow traffic model. Closed-form expressions were derived that relate the LB parameters (token generation rate *R* and bucket size *B*) and the bursty source characteristics to the cell loss probability and the queueing delay caused by the LB.

As we have described in section 2.2.2, the modified leaky bucket is designed to police a bursty source's PCR and ACR simultaneously, this is the first advantage over a traditional LB because a traditional LB can either police a source's PCR or ACR, but not both at the same time. Secondly, since no cell buffer and token pool are designated in the modified LB mechanism, thus no cell delay occurs. A traditional leaky bucket needs either token pool or cell buffer, or both to achieve its performance, as a result cells are delayed before being transmitted into the network. Therefore, the modified LB performs better than the traditional LB in terms of delay and it is more appropriate for real-time traffic sources.

Next, we will compare performance measurement of the cell loss probability between the modified LB and a traditional LB mechanism.

5.5.1. Hluchyj's mathematical results

Under the assumption that a bursty source's On and Off periods are exponentially distributed, Hluchyj et al. performed the analysis, in which γ denotes the source utilization, L denotes the average number of cells during an On period, and ρ denotes the leaky bucket load. Here:

$$\gamma = \frac{\text{average cell rate}}{\text{peak cell rate}} \tag{5-17}$$

$$\rho = \frac{\text{average cell rate}}{\text{average token rate}} \tag{5-18}$$

Based on these notations, the following result was obtained in [9]:

$$P_{loss} = \frac{(\rho-\gamma)(1-\rho)e^{\xi(B+M)}}{\rho(1-\gamma)\left(1-\frac{\rho-\gamma}{1-\gamma}e^{\xi(B+M)}\right)} \tag{5-19}$$

here B and M are the token pool size and the cell buffer size respectively and ξ is $\xi = -\frac{\rho(1-\rho)}{L(1-\gamma)(\rho-\gamma)}$ in which L is the average number of cells generated during a bursty source's On period.

From equation (5-19), we obtain the following figure 5.3, which shows:

1). a lower leaky-bucket load ρ will give a lower CLP under the same γ and $(B+M)$, because a lower ρ means the token rate R is close to the source's PCR, so more tokens are generated to give a lower CLP.

2). a lower source's utilization γ will introduce a higher CLP under the same ρ, especially when ρ has a higher value. This is as to the expectation because a lower γ means a higher burstiness, a higher ρ means the token generation rate *R* is closer to the source's ACR, obviously this will give a higher CLP.

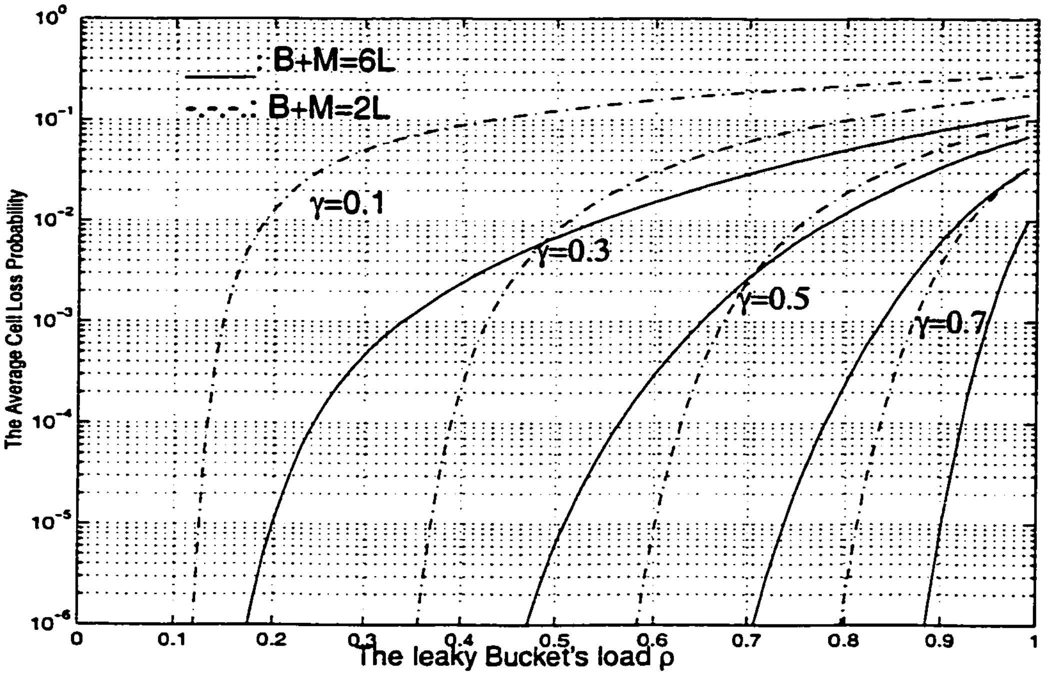


Figure 5.3 The traditional LB with different token pool and cell buffer

3) The traditional Leaky Bucket achieves a better performance through buffering more cells and tokens, i.e. the utilization of a higher $(B+M)$ value will give a lower CLP under the same γ and ρ, but this will introduce a higher cell delay. There is a trade-off between the cell loss probability and the cell delay.

5.5.2 Comparison of the modified LB with the traditional LB

As we have stated, the modified LB is more suitable for real-time sources since it does not introduce cell delays. Meanwhile it is also more appropriate for very bursty sources, which transmit cells at a high PCR during a very short active period. According to Hluchyj's notations, these sources have a lower source utilization parameter γ.

On the other hand, as figure 5.3 shows, a traditional leaky-bucket introduces a higher CLP when γ has a lower value, this means the traditional LB doesn't perform very well with very bursty sources. So in the following we will compare our modified LB with the traditional LB mechanism especially for very bursty sources which have lower γ values. We will use γ=0.01 in the following comparison.

Since our analysis is based on the discrete-time model, we need to transfer the parameters of our modified leaky bucket into the parameters γ, ρ and L as in Hluchyj's model shown below:

$$\gamma = \frac{\text{average cell rate}}{\text{peak cell rate}} = \frac{\text{average On duration}}{\text{average On +average Off duration}} = \frac{1-\beta s}{2-\alpha_s-\beta_s} \tag{5-20}$$

$$\rho = \frac{\text{average cell rate}}{\text{average token rate}} = \frac{1-\beta_t}{2-\alpha_t-\beta_t} \tag{5-21}$$

$$L = \frac{1}{1-\alpha_s} \tag{5-22}$$

Then we can convert the parameters in our discrete-time model into the following expressions:

$$\alpha_s = 1-\frac{1}{L} \tag{5-23}$$

$$\beta_s = 1-\frac{\gamma}{L-L\cdot\gamma} \tag{5-24}$$

$$\alpha_t = 2-\beta_t-\frac{\rho\cdot(1-\beta_t)}{\gamma} \tag{5-25}$$

Substituting them into equation (4-39), then we can obtain the following figure

5.4 from equations (4-39) and (5-19) to compare the unforced protocol of the modified LB with the traditional LB.

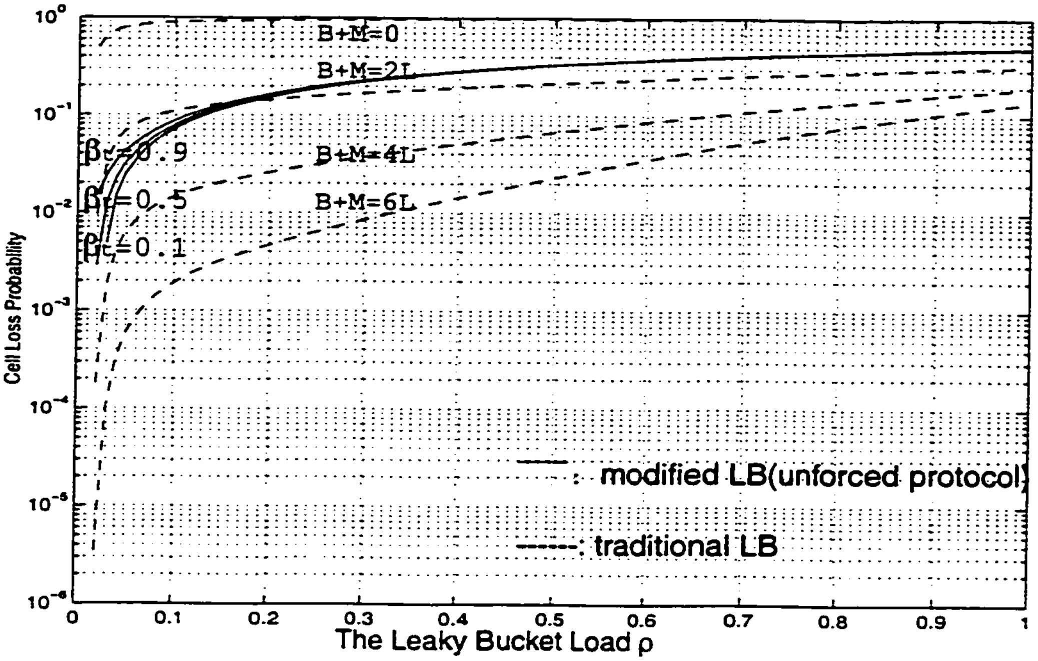


Figure 5.4 The Comparison of two Leaky Bucket when $\gamma = 0.01$

As expected the modified LB is more suitable for real-time sources since this type of traffic is not suitable for buffering. From Figure 5.4, the traditional LB achieves a better performance through buffering. When there is no buffering or even for small-size buffering, the modified LB performs better than a traditional LB.

Chapter 6

Conclusions and Further Research

6.1 Conclusions

In this thesis, we have proposed a new dynamic Leaky-Bucket mechanism in order to police a bursty source's PCR and ACR simultaneously with a single LB, while also eliminating cell delays caused by the traditional LB.

Motivated by a bursty source's geometrically distributed On and Off periods, we have designed a two-state token generator with geometrically distributed On and Off periods. As tokens are generated at the source's PCR during the token generator's On period (so to police the source's PCR), cells generated by the source will immediately obtain tokens and be transmitted into the network. Therefore we will not need a token pool or a cell buffer, as a result no cell delay is introduced in this mechanism. We have also shown that the modified LB can police the source's ACR at the same time by properly choosing the token generator's On and Off periods based on a source's statistical characteristics.

We have derived a closed-form expression for the average cell loss probability in this modified LB system. Numerical results have shown that this dynamic LB mechanism is more appropriate for real-time traffic and very bursty sources which have very low source utilizations than the traditional LB, in terms of the cell delay and cell loss probability.

6.2 Further Research

This work could be further developed in the following directions:

- Since no token pool exists in the modified leaky bucket, the cell loss probability is pretty high for some sources. This could be improved by storing and reusing tokens which are wasted when the token generator is active but the source is inactive, then analyze the average CLP for a longer time interval.

- Adjust the two-state token generator's statistic parameters to obtain a more effective policing function for a certain types of bursty sources. This can be achieved by adjusting the token generator's statistical behavior based on the source's behavior.

- Perform a simulation to measure the cell loss probability and cell delay while a token pool and a cell buffer have been added to the modified LB.

Appendix A

Analysis of the average CLP while the token generator's parameter α_t is variant

In this appendix, we will analyze the CLP under the forced protocol with a variant token generator's parameter α_t, this analysis is parallel to that of chapter 3. First, we describe why and how to vary α_t.

A.1. The motivation to vary α_t

As we know, the bursty source and the dynamic token generator's On-Off distributions are independent of each other, so long bursts of the source may coincide with the token generator's long On periods. As a result, a large number of cells generated by these long bursts would be transmitted into the network. The admission of long bursts places a heavy burden on the resources of the network. Therefore, we would like to reduce the possibility of transmitting the source's long bursts into the network.

We will assume that during a source's tagged period, cells generated during the first *K* time slots are *conforming cells*, and cells generated after these *K* slots are *violating cells*. Obviously, if the token generator's On parameter α_t is constant, it will have the same possibility to stay at On state before and after these *K* time slots, so the source's conforming and violating cells will have the same possibility to be admitted into the network, as shown in following Figure A.1.

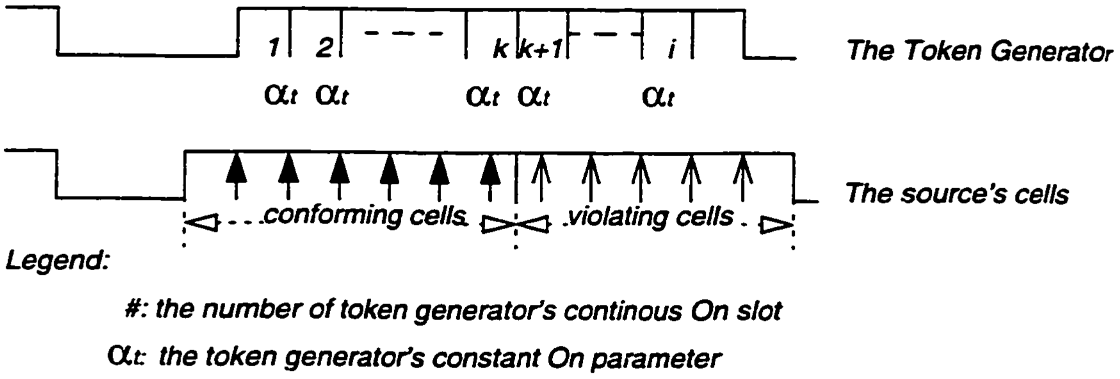

1
2
k
k+1
i
αt αt αt αt αt
The Token Generator
The source's cells
conforming cells
violating cells
Legend:
#: the number of token generator's continous On slot
αt: the token generator's constant On parameter

Figure A.1: The constant token generator's parameter α_t

Since transmitting tagged periods longer than K time slots are not desired, this may be achieved by reducing the token generator's On parameter α_t after slot K. In the next section, we will show how to vary α_t to achieve this objective.

A.2 How to vary α_t

As we have stated, the cells generated during the initial K slots of a source's tagged period is referred as conforming cells and the ones after these K slots as violating cells. The objective here is to keep the probability of transmitting conforming cells at a constant level, while reducing it for violating cells. So the token generator's On parameter α_t should be kept constant for these K time slots for transmitting the conforming cells with the same probability. After these K slots, α_t should be decreased dramatically to give violating cells less possibility to be transmitted. Based on this strategy, we vary α_t as follows.

First let's define:

$\alpha_t(i)$ = the probability that the token generator will stay at On state during the ith time slot given that it is active during the previous slot

In this thesis, we vary $\alpha_t(i)$ as:

$$\alpha_t(i) = \begin{cases} \alpha_t & i \le K \\ \alpha_t^{i-K+1} & i > K \end{cases} \tag{A-1}$$

In the above equation, α_t and K are two constants which should be determined at the service negotiation phase. The value of K represents the number of time slots during which the token generator's parameter $\alpha_t(i)$ has a constant value α_t. The variant $\alpha_t(i)$ is shown in following figure A.2.

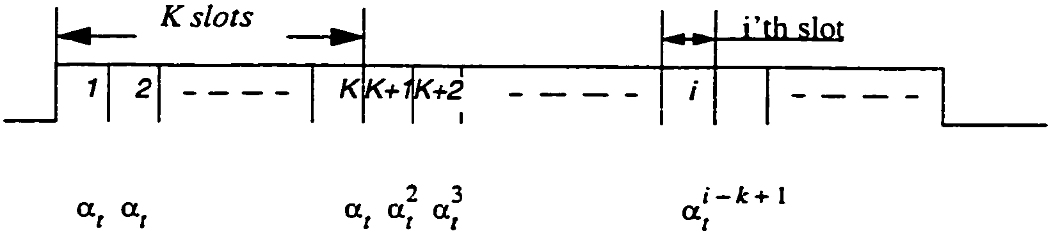


Figure A.2: the token generator's parameter $\alpha_t(i)$

According to equation (A-1), $\alpha_t(i)$ is a constant within the first K time slots during the token generator's On period to give the source a fair chance to transmit its conforming cells into the network. After these K slots, $\alpha_t(i)$ will be geometrically decreased as α_t is a constant less than 1, the token generator will have a less chance to stay at On state during the next time slot, thus to give the violating cells less possibilities to be transmitted into the network. The longer the source's burst is, the more violating cells may be generated, the less possible for these violating cells being transmitted.

Define:

$Pr(k_t) = Pr$ (the number of time slots of the token generator's On period is k_t)

From equation (A-1), it is easy to get the probability distribution of the token generator's On period in the following equation:

$$Pr(k_t) = \begin{cases} (1-\alpha_t)\alpha_t^{k_t - 1} & k_t \le K \\ (1-\alpha_t^2)\alpha_t^K & k_t = K+1 \\ \alpha_t^{\sum_{i=2}^{k_t - K} i} \left(1-\alpha_t^{k_t - K + 1}\right) \cdot \alpha_t^K & k_t \ge K+2 \end{cases} \tag{A-2}$$

Next, we will use the preliminary PGF results obtained in chapter 2 to analyze the average CLP during a random cycle with such a variant $\alpha_t(i)$, which is in parallel with the analysis of the constant parameters in section 3.4.

A.3 Analysis of the CLP during Random Cycle

As has been described in section 3.4.3, we can analyze all cases together in the forced protocol because the source and the token generator's geometrically distributed On periods have memoryless property, which enables us to consider the token generator's and the source's tagged periods start simultaneously, this is shown in figure 3.6.

But in this section, the variant parameter $\alpha_t(i)$ makes the token generator's On period no longer geometrically distributed, as may be observed in equation (A-2). The probability distribution of the token generator's tagged period doesn't have memoryless property any more.

Therefore, as figure 3.2 shows, case FII.1 cannot be treated together with case FI, because the source's tagged period starts at time C after the token generator's tagged period has started at time B, from time C on, the residual time distribution of the token generator's tagged period is different from its original distribution. So we cannot consider that the token generator and the source's tagged period start at the same time when we consider case FII.1.

On the other hand, since the source's tagged period is still geometrically distributed with the memoryless property, so we can still consider that the token generator and the source's tagged period start at the same time in case FI. and FII.2, we still can analyze them together in this section.

In the following, we will present the analysis on case FI (case FII.2) and case FII.1 separately.

A.3.1 The average umber of cells lost during the token generator's initial Off period

Since the parameter of the token generator's Off period is not changed, so the number of cells lost during the token generator's initial Off period (from A to B) is same as been given in equation (3-6), this is identical for all cases.

A.3.2 The average CLP during a random cycle

As been explained in chapter 3, we need to analyze case FI(FII.2) and case FII.1 separately.

A.3.2.1 Case FI (and case FII.2)

Refer to figure 3.2, we define:

N_{BG} = the number of cells transmitted during the source's tagged period

$\tilde{N}_{BG}$ = the number of cells lost during the source's tagged period

As in section 3.4.3, we can obtain the average number of them as:

$$\overline{\tilde{N}_{BG,(k_s > k_t)}} = \sum_{k_t = 1}^{\infty} \sum_{k_s = k_t}^{\infty} (k_s - k_t) \cdot Pr(k_s) \cdot Pr(k_t)$$

$$= \sum_{k_t = 1}^{K} \sum_{k_s = k_t}^{\infty} (k_s - k_t)(1-\alpha_s)\alpha_s^{k_s - 1}(1-\alpha_t)\alpha_t^{k_t - 1} + \sum_{k_s = k+1}^{\infty} (k_s - K + 1)(1-\alpha_s)\alpha_s^{k_s - 1}\left(1 - \alpha_t^{2}\right)\alpha_t^{K}$$

$$+ \sum_{k_t = K+2}^{\infty} \sum_{k_s = k_t}^{\infty} (k_s - k_t)(1-\alpha_s)\alpha_s^{k_s - 1}\left(1 - \alpha_t^{k_t - K + 1}\right)\alpha_t^{K}\alpha_t^{\sum_{i=2}^{k_t - K} i}$$

$$= \frac{(1-\alpha_t)\alpha_s\left[1 - (\alpha_s\alpha_t)^{K}\right]}{(1-\alpha_s)(1-\alpha_s\alpha_t)} + \frac{\left(1-\alpha_t^{2}\right)\alpha_s(\alpha_s\alpha_t)^{K}}{(1-\alpha_s)} + \frac{\alpha_t^{K-1}\alpha_s^{K}}{(1-\alpha_s)} \sum_{k_t - k = 2}^{\infty} \left(1 - \alpha_t^{k_t - K + 1}\right)\alpha_t^{\frac{(k_t - K)^2 + k_t - K}{2}} \alpha_s^{k_t - k} \quad \text{(A-4)}$$

$$Pr(k_s > k_t) = \sum_{k_t = 1}^{\infty} \sum_{k_s = k_t}^{\infty} Pr(k_s)Pr(k_t)$$

$$= \frac{(1-\alpha_t)\cdot\left(1 - (\alpha_s\alpha_t)^{K}\right)}{1 - \alpha_s\alpha_t} + (\alpha_s\alpha_t)^{K}\left(1 - \alpha_t^{2}\right) + \alpha_t^{K-2} \sum_{k_t = K+2}^{\infty} \alpha_s^{k_t - 1}\left(1 - \alpha_t^{k_t - K + 1}\right)\alpha_t^{\frac{(k_t - K)^2}{2}} \quad \text{(A-5)}$$

$$\overline{\bar{N}_{BG}} = \overline{\bar{N}_{BG, k_s > k_t}} Pr(k_s > k_t) \quad \text{(A-6)}$$

$$\overline{N_{BG}} = \overline{k_s} - \overline{\bar{N}_{BG}} = \frac{1}{1 - \alpha_s} - \overline{\bar{N}_{BG}} \quad \text{(A-7)}$$

Define:

$\overline{N_{on}}$ = the average number of cells successfully transmitted during the random cycle

$\overline{N_{off}}$ = the average number of cells lost during the random cycle

$P_{loss}(1)$ = the average CLP during a random cycle in case FI (and case FII.2)

Combining equations (3-6), (A-6), and (A-7), we obtain:

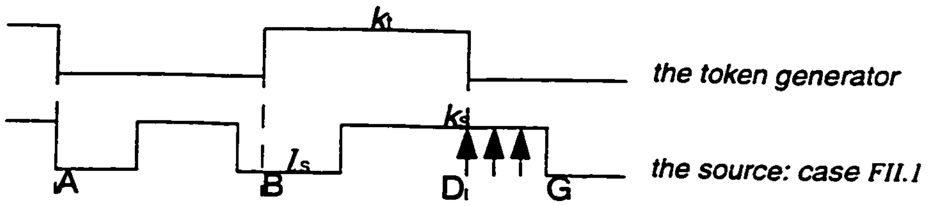
$$\overline{N_{on}} = \overline{N_{BG}} \tag{A-8}$$

$$\overline{N_{off}} = \overline{\tilde{N}_{AB}} + \overline{\tilde{N}_{BG}} \tag{A-9}$$

$$P_{loss}(1) = \frac{\overline{N_{off}}}{\overline{N_{on}} + \overline{N_{off}}} \tag{A-10}$$

A.3.2.2 Case FII.1

Let us review case FII.1 in the following figure A.3:



Legend:

ks, kt : the number of time slots of the sourc's burst and the token generator's On period

ls: the number of time slots of the source's Off period

Figure A.3: Case FII.1

Since the probability distributions of k_s, l_s and k_t are independent of one another, we can get the average number of $\tilde{N}_{BG}$ as the following:

$$\overline{\tilde{N}_{BG, k_s > k_t - l_s}} = \sum_{l_s = 1}^{\infty} \sum_{k_t = l_s}^{\infty} \sum_{k_s = k_t - l_s}^{\infty} [k_s - (k_t - l_s)] Pr(k_s) Pr(k_t) Pr(l_s) \tag{A-11}$$

$$\overline{\tilde{N}_{BG}} = \overline{\tilde{N}_{BG, k_s > k_t - l_s}} Pr(k_s > k_t - l_s) \tag{A-12}$$

Here $Pr(k_t)$ is given in equation (A-2). The constant K in $Pr(k_t)$ is defined to represent the number of time slots during which the token generator has a constant probability α_t to stay in On state during the next time slot given it is On in the current slot. The value of K should be determined at the connection set-up phase. Substituting equation (A-2) into (A-11), we will have two equations depending on the different value of K:

i). For $K = 1$, we have:

$$\overline{\bar{N}_{BG.k_s > k_t - l_s}} = \frac{(1-\beta_s)\alpha_s}{(1-\alpha_s)(\alpha_s-\beta_s)}\left[\left(1-\alpha_t^2\right)\alpha_t(\alpha_s-\beta_s) + \sum_{k_t=3}^{\infty}\alpha_t\alpha_s^{\sum_{i=2}^{k_t-1} i}\left(1-\alpha_t^{k_t}\right)\left(\alpha_s^{k_t-1}-\beta_s^{k_t-1}\right)\right]$$

$$= \frac{(1-\beta_s)\alpha_s\alpha_t(1-\alpha_t^2)}{(1-\alpha_s)} + \frac{(1-\beta_s)\alpha_s}{(1-\alpha_s)(\alpha_s-\beta_s)}\sum_{j=2}^{\infty}\alpha^{\frac{j^2+j}{2}}(1-\alpha_t^{j+1})(\alpha_s^j-\beta_s^j) \qquad \text{(A-13)}$$

here $j = k_t - l_s$

ii). For $K \geq 2$, we have:

$$\overline{\bar{N}_{BG.k_s > k_t - l_s}} = \frac{(1-\beta_s)\alpha_s}{(1-\alpha_s)(\alpha_s-\beta_s)}\left[\sum_{k_t=2}^{K}(1-\alpha_t)\alpha_t^{k_t-1}\left(\alpha_s^{k_t-1}-\beta_s^{k_t-1}\right)+\left(1-\alpha_t^2\right)\alpha_t^K\left(\alpha_s^K-\beta_s^K\right)\right]$$

$$+ \frac{(1-\beta_s)\alpha_s}{(1-\alpha_s)(\alpha_s-\beta_s)}\cdot\left[\sum_{k_t=K+2}^{\infty}\left(1-\alpha_t^{k_t-K+1}\right)\alpha_t^{K+\sum_{i=2}^{k_t-K} i}\left(\alpha_s^{k_t-1}-\beta_s^{k_t-1}\right)\right]$$

$$= \frac{(1-\beta_s)\alpha_s}{(1-\alpha_s)(\alpha_s-\beta_s)}\left[(1-\alpha_t)\cdot\frac{\alpha_s\alpha_t-(\alpha_s\alpha_t)^K}{1-\alpha_s\alpha_t}-(1-\alpha_t)\frac{\beta_s\alpha_t-(\beta_s\alpha_t)^K}{1-\beta_s\alpha_t}+\left(1-\alpha_t^2\right)\left((\alpha_s\alpha_t)^K-(\alpha_{tt}\beta_s)^K\right)\right]$$

$$+ \frac{(1-\beta_s)\alpha_s}{(1-\alpha_s)(\alpha_s-\beta_s)}\cdot\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m+m^2}{2}}\left(\alpha_s^{k+m-1}-\beta_s^{k+m-1}\right)\cdot\alpha_t^{k-1} \qquad \text{(A-14)}$$

Here $m = k_s - K$.

In this Appendix we assume that $K \geq 2$.

For the same reason as in the last section, we get:

$$\overline{N_{BG}} = \bar{k}_s - \overline{\bar{N}_{BG}} \qquad \text{(A-15)}$$

Define:

$\overline{N_{on,2}}$ = the average number of cells successfully transmitted during the random cycle

$\overline{N_{off,2}}$ = the average number of cells lost during the random cycle

$P_{loss}(2)$ = the average CLP during a random cycle in case FII.1

Combining equations (3-6), (A-12) and (A-15), we have:

$$\overline{N_{on}} = \overline{N_{BG}} \qquad \text{(A-16)}$$

$$\overline{N_{off}} = \overline{\bar{N}_{AB}} + \overline{\bar{N}_{BG}} \qquad \text{(A-17)}$$

$$P_{loss}(2) = \frac{\overline{N_{off}}}{\overline{N_{on}} + \overline{N_{off}}} \tag{A-18}$$

Next, we will combine case FI (FII.2) and FII.1 together to obtain the final average CLP during a random cycle for a variant token generator's parameter $\alpha_t(i)$.

A.3.2.3 The Average CLP

In section 4.4.3, we have derived the probabilities for cases UI and UII, which were given in equations (4-33), (4-34) respectively. Cases FI and FII have the same probability distributions as UI and UII since the token generator's initial Off period is same for both forced and unforced protocols. FII has two further separate cases FII.1 and FII.2, which have different CLPs in this section, so we need to derive the probabilities of these two separate cases here.

Define:

$Pr(\text{ FII.1}) = Pr$(case FII.1 occurs given that case FII has occurred)

$Pr(\text{ FII.2}) = Pr$(case FII.2 occurs given that case FIl has occurred)

From figure 3.6, we can get:

$$Pr(\text{ FII.1}) = Pr(k_t > l_s) = \sum_{l_s = 1}^{\infty} \sum_{k_t = l_s + 1}^{\infty} Pr(k_t) Pr(l_s) \tag{A-19}$$

$$Pr(\text{ FII.2}) = 1 - Pr(\text{ FII.1}) \tag{A-20}$$

Define:

P_{loss} = the cell loss probability during a random cycle

Then combining equations (4-33), (4-34), (A-10), (A-18), (A-19) and (A-20) together, we get:

$$P_{loss} = P_{loss}(1)[Pr(\text{FI}) + Pr(\text{FII})Pr(\text{FII.2})] + P_{loss}(2)Pr(\text{FII})Pr(\text{FII.1}) \qquad \text{(A-21)}$$

Equation (A-21) gives the average CLP with a variant token generator's parameter $\alpha_t(i)$, and in the following figure A.4 we plot together the cell loss probability of constant and variant token generator's parameter $\alpha_t(i)$.

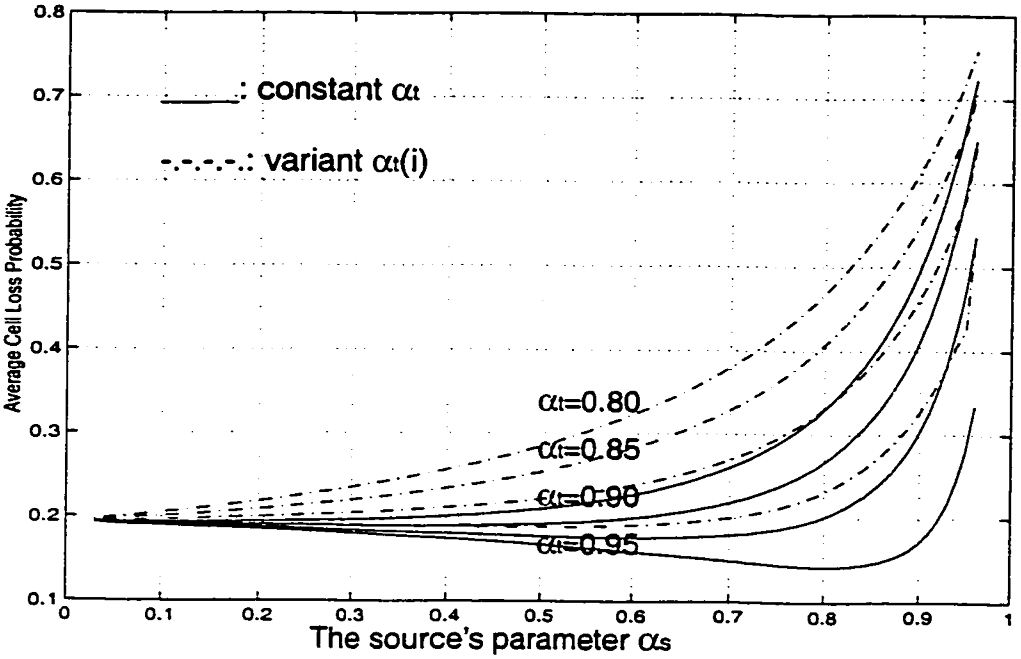


Figure A.4: Comparison of the constant and variant $\alpha_t(i)$ when $\beta_s = 0.95, \beta_t = 0.8$ and $K = 5$

(Note: the infinite summations in equations (A-4), (A-5) and (A-14), we have continued to add terms until the deviation is smaller than 1.0E-5.)

We can see that the variant token generator's On parameter $\alpha_t(i)$ gives a higher cell loss probability than the constant parameter α_t . When the source's On parameter α_s has a low value, the source generates mostly short bursts, the CLPs are very closed for the constant and the variant token generator's On parameter, this is reasonable because the variant parameter $\alpha_t(i)$ is designed for policing the source's longer bursts.

Appendix B

Analysis of the unforced protocol while the token generator's parameter $\alpha_t(i)$ is a variant

As in Appendix A, here we will perform a parallel analysis on the unforced protocol with Chapter 4, which deals with the variant token generator's parameter $\alpha_t(i)$.

We use the same $\alpha_t(i)$ as described in Appendix A to determine the performance of the unforced protocol. The unforced protocol is same as the one we have described in chapter 4, except that the probability distribution of the token generator's On period is no longer geometrically distributed, which has been shown in equation (A-2).

As has been described in chapter 4, we need to analyze case UI and case UII.1 separately here, because these two cases have different initial conditions when the token generator first makes the transition from Off to On state, referring to figure 4.3 and 4.5. Case UII.2 can still be analyzed identically with case UI, because it has the same initial conditions when the token generator is forced to start a new On period at time C.

Next, we will present the analysis of case UI (case UII.2) and case UII.1 separately.

B.1 The number of cells lost generated during the token generator's Off

period

Since the probability distribution of the token generator's Off period has never been changed throughout our analysis, we will have the same result in this section as all previous analysis. The result has been presented in equation (3-6).

B.2 The Average CLP during a Random Cycle in Case UI (and UII.2)

The initial condition and initial joint PGF in this case has been shown in equation (4-1) and (4-2).

B.2.1 The number of cells transmitted during the source's tagged period

Define:

$P_{k_t}(z)$ =the transient marginal PGF of the number of cells generated by the end of k_t'th slot during the token generator's tagged period

From the initial joint PGF shown in equation (4-2) and equation (A-2), the transient marginal PGF will be:

$$P_{k_t}(z) = [X(k_t)(Q_0(z, \Phi(k_t)))]\Big|_{y=1} = \bar{X}(k_t) \cdot \bar{\Phi}(k_t) = \bar{D}_1\lambda_1^{k_t} + \bar{D}_2\lambda_2^{k_t} \qquad \text{(B-1)}$$

Define:

$P(z)$ =the marginal PGF of the number of cells generated during the token generator's tagged period

Averaging equation (B-1) with the duration of the token generators tagged period, we have:

$$P(z) = \sum_{k_t = 1}^{\infty} P_{k_t}(z) \cdot Pr(k_t)$$

$$= \sum_{k_t = 1}^{K} \left(\bar{D}_1 \lambda_1^{k_t} + \bar{D}_2 \lambda_2^{k_t}\right)(1-\alpha_t)\alpha_t^{k_t - 1} + \left(\bar{D}_1 \lambda_1^{K+1} + \bar{D}_2 \lambda_2^{K+1}\right)\left(1-\alpha_t^2\right)\alpha_t^{K} + \sum_{k_t = K+2}^{\infty} \left(\bar{D}_1 \lambda_1^{k_t} + \bar{D}_2 \lambda_2^{k_t}\right)\left(1-\alpha_t^{k_t - K+1}\right)\alpha_t^{K + \sum_{i=2}^{k_t - K} i}$$

$$= \left(\frac{(1-\alpha_t)}{\alpha_t} \cdot \left[\bar{D}_1 \cdot \frac{\alpha_t\lambda_1 - (\alpha_t\lambda_1)^{K+1}}{1-\alpha_t\lambda_1} + \bar{D}_2 \cdot \frac{\alpha_t\lambda_2 - (\alpha_t\lambda_2)^{K+1}}{1-\alpha_t\lambda_2}\right] + \left(\bar{D}_1 \lambda_1^{K+1} + \bar{D}_2 \lambda_2^{K+1}\right)\left(1-\alpha_t^2\right)\alpha_t^{K} + III\right) \qquad \text{(B-2)}$$

$$= I + II + III$$

here I, II and III are respectively the first, second and third terms on the right hand side of the above equation.

Define:

$\overline{N_{BG}}$ = the average number of cells successfully during the source's tagged period

from the PGF equation shown in (A-2), we have:

$$\overline{N_{BG}} = I'|_{z=1} + II'|_{z=1} + III'|_{z=1} \qquad \text{(B-3)}$$

where:

$$I'\big|_{z=1} = \frac{(1-\alpha_t)}{\alpha_t}\cdot\left[\bar{D}_1\cdot\frac{\alpha_t\lambda_1-(\alpha_t\lambda_1)^{K+1}}{1-\alpha_t\lambda_1}+\bar{D}_2\cdot\frac{\alpha_t\lambda_2-(\alpha_t\lambda_2)^{K+1}}{1-\alpha_t\lambda_2}\right]'\Bigg|_{z=1}$$

$$= \left(\frac{(1-\alpha_t)\left(\frac{1-\beta_s}{(2-\alpha_s-\beta_s)^2}-\frac{\alpha_s}{(2-\alpha_s-\beta_s)}\right)(\beta_s+\alpha_d-1)\left(1-\alpha_t^{K}(\beta_s+\alpha_s-1)^{K}\right)}{1-\alpha_t(\beta_s+\alpha_s-1)}+\frac{\alpha_t\left(1-\alpha_t^{K}\right)}{1-\alpha_t}\right)$$

$$+\ \left(1-\alpha_t^{K}\right)\left(\frac{\alpha_s}{(2-\alpha_s-\beta_s)}-\frac{1-\beta_s}{(2-\alpha_s-\beta_s)^2}\right)+\frac{1-\beta_s}{(2-\alpha_s-\beta_s)}\left(1-(k+1)\alpha_t^{K}\right) \qquad \text{(B-4)}$$

$$II'\big|_{z=1} = (D_1'\lambda_1^{K+1}+\bar{D}_1(K+1)\lambda_1^{K}\lambda_1'+\bar{D}_2'\lambda_2^{K+1}+(K+1)D_2\lambda_2^{K}\lambda_2')(1-\alpha_t^{2})\alpha_t^{K}\Big|_{z=1}$$

$$= \frac{\left(1-\alpha_t^{2}\right)\alpha_t^{K}}{2-\alpha_s-\beta_s}\left[\left(\frac{1-\beta_s}{2-\alpha_s-\beta_s}-\alpha_s\right)(\beta_s+\alpha_s-1)^{K+1}+\alpha_s-\frac{1-\beta_s}{2-\alpha_s-\beta_s}+(K+1)(1-\beta_s)\right] \qquad \text{(B-5)}$$

$$III'\big|_{z=1} = \sum_{k_y=K+2}^{\infty}\left(\bar{D}_1\lambda_1^{k_t}+\bar{D}_2\lambda_2^{k_t}\right)'\Big|_{z=1}\left(1-\alpha_t^{k_t-k+1}\right)\alpha_t^{K+\sum_{i=2}^{k_t-K} i}$$

$$= \frac{\alpha_t^{k-1}}{2-\alpha_s-\beta_s}\sum_{m=2}^{\infty}\left[\left(\frac{1-\beta_s}{2-\alpha_s-\beta_s}-\alpha_s\right)(\beta_s+\alpha_s-1)^{K+m}+\alpha_s-\frac{1-\beta_s}{2-\alpha_s-\beta_s}+(K+m)(1-\beta_s)\right]\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m+m^2}{2}} \qquad \text{(B-6)}$$

here, $m = k_t - K$

B.2.2 The probability distribution of the source's state when the token generator's tagged period terminates

In this section, because $\alpha_t(i)$ is a variant, the token generator's On period is not geometrically distributed, it has no memoryless property. So when we analyze the cells lost after the token generator's tagged period during the random cycle, we cannot use the same method as that used in chapter 4, which was presented in equation (4-12).

Instead, we need to use the PGF results derived in chapter 2 to analyze the number of cells lost after the token generator's On period. Because the source's state will determine the initial conditions used in the PGF method, so here we must determine the probability of the source's different state when the token generator's On period ends during the random cycle.

Define:

$A_{k_t}(y)$ = the transient marginal PGF of the state of the source at the end of k_t'th slot during the token generator's tagged period

Substitute the initial conditions when the token generator's On period starts, i.e. equation (4-2) into (2-30), we will have:

$$A_{k_t}(y) = \left(D_1\lambda_1^{k_t} + D_2\lambda_2^{k_t}\right)\Big|_{z=1} \tag{B-7}$$

where

$$\lambda_1\Big|_{z=1} = \beta_s + \alpha_s - 1 \quad , \qquad \lambda_2\Big|_{z=1} = 1$$

$$D_1\Big|_{z=1} = \frac{(1-y)(\alpha_s - 1)}{2-\beta_s-\alpha_s} \quad , \qquad D_2\Big|_{z=1} = \frac{1-\alpha_s+(1-\beta_s)y}{2-\beta_s-\alpha_s}$$

combining them into (B-7), we get:

$$A_{k_t}(y) = \frac{(1-y)(\alpha_s-1)}{2-\beta_s-\alpha_s}(\beta_s+\alpha_s-1)^{k_t}+\frac{1-\alpha_s+(1-\beta_s)y}{2-\beta_s-\alpha_s} \qquad \text{(B-8)}$$

Define:

$A_{on}(y)$ = the marginal PGF of the state of the source at the end of the token generator's tagged period

combining equation (A-2) and (B-8) together, we have:

$$A_{on}(y) = \sum_{a=1}^{\infty} A_{k_t}(y)Pr(k_t)$$

$$= \sum_{a=1}^{\infty}\left[\frac{(1-y)(\alpha_s-1)}{2-\beta_s-\alpha_s}(\beta_s+\alpha_s-1)^{k_t}+\frac{1-\alpha_s+(1-\beta_s)y}{2-\beta_s-\alpha_s}\right]Pr(k_t)$$

$$=\frac{(1-\alpha_t)}{2-\alpha_s-\beta_s}\left[1-\alpha_t^{K}-(1-\alpha_t)(\beta_s+\alpha_s-1)\cdot\frac{1-\left(\alpha_t(\beta_s+\alpha_s-1)^{K}\right)}{1-\alpha_t(\beta_s+\alpha_s-1)}+\left(1-\alpha_t^{2}\right)\alpha_t^{K}\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)\right]$$

$$+\quad\frac{(1-\alpha_t)\alpha_t^{K}}{2-\alpha_s-\beta_s}\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m+m^2}{2}}\left(1-(\beta_s+\alpha_s-1)^{K+m}\right)+$$

$$\frac{y}{2-\alpha_s-\beta_s}\left[(1-\beta_s)\left(1-\alpha_t^{K}\right)+\left((1-\alpha_t)(1-\alpha_s)(\beta_s+\alpha_s-1)\cdot\frac{1-\left(\alpha_t(\beta_s+\alpha_s-1)^{K}\right)}{1-\alpha_t(\beta_s+\alpha_s-1)}\right)+\left(1-\alpha_t^{2}\right)\alpha_t^{K}\left((1-\beta_s)+(1-\alpha_s)(\beta_s+\alpha_s-1)^{K+1}\right)\right]$$

$$+\ \frac{y}{2-\alpha_s-\beta_s}(1-\alpha_s)\alpha_t^{K-1}\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m+m^2}{2}}\left(1-\beta_s-(1-\alpha_s)(\beta_s+\alpha_s-1)^{K+m}\right) \qquad \text{(B-9)}$$

here $m = k_t - K$

Define:

$Pr(Off) = Pr$ (the source is Off when the token generator's tagged period ends)

$Pr(On) = Pr$ (the source is On when the token generator's tagged period ends)

From the definition of the probability generating function, we have:

$$A_{on}(y) = Pr(Off) + y \cdot Pr(On) \tag{B-10}$$

Compare equation (B-9) and (B-10), we can get the two probabilities as the following:

$$Pr(off) = \frac{(1-\alpha_t)}{2-\alpha_s-\beta_s}\left[1-\alpha_t^{K} - (1-\alpha_t)(\beta_s+\alpha_s-1) \cdot \frac{1-\left(\alpha_t(\beta_s+\alpha_s-1)^{K}\right)}{1-\alpha_t(\beta_s+\alpha_s-1)} + \left(1-\alpha_t^{2}\right)\alpha_t^{K}\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)\right]$$

$$+ \frac{(1-\alpha_t)\alpha_t^{K}}{2-\alpha_s-\beta_s}\sum_{m=2}^{\infty}(1-\alpha^{m+1})\alpha_t^{\frac{m+m^2}{2}}\left(1-(\beta_s+\alpha_s-1)^{K+m}\right) \tag{B-11}$$

$$Pr(On) = \frac{1}{2-\alpha_s-\beta_s}\left[(1-\beta_s)\left(1-\alpha_t^{K}\right) + \left((1-\alpha_t)(1-\alpha_s)(\beta_s+\alpha_s-1) \cdot \frac{1-\left(\alpha_t(\beta_s+\alpha_s-1)^{K}\right)}{1-\alpha_t(\beta_s+\alpha_s-1)}\right)\right]$$

$$+ \frac{1}{2-\alpha_s-\beta_s}\left[\left(1-\alpha_t^{2}\right)\alpha_t^{K}\left((1-\beta_s)+(1-\alpha_s)(\beta_s+\alpha_s-1)^{K+1}\right)\right] +$$

$$\frac{1}{2-\alpha_s-\beta_s}(1-\alpha_s)\alpha_t^{K-1}\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m+m^2}{2}}\left(1-\beta_s-(1-\alpha_s)(\beta_s+\alpha_s-1)^{K+m}\right) \tag{B-12}$$

B.2.3 Cell loss probability in case UI

Define:

$I(z)$ = the PGF of the number of cells lost during the source's tagged period

Because the residual time distribution of a geometrical distribution is still geometrical, we can get the following equation from figure 4.2:

$$I(z) = Pr(off) \cdot z^0 + Pr(on) \cdot \left(\frac{(1-\alpha_s)z}{1-\alpha_s z}\right) \tag{B-13}$$

Define:

$\overline{N_{BG}}$ = the average number of cells lost during the source's tagged period

from equation (B-13), we have:

$$\overline{N_{BG}} = I(z)'\big|_{z=1} = Pr(on) \cdot \left(\frac{(1-\alpha_s)z}{1-\alpha_s z}\right)'\Bigg|_{z=1} = Pr(on) \cdot \frac{1}{1-\alpha_s}$$

substitute equation (B-12) into the above equation, we have:

$$\tilde{N}_{BG} = \frac{1}{2-\alpha_s-\beta_s}\left[\frac{(1-\beta_s)\left(1-\alpha_t^{K}\right)}{1-\alpha_s} + \left((1-\alpha_t)(\beta_s+\alpha_s-1) \cdot \frac{1-\left(\alpha_t(\beta_s+\alpha_s-1)^{K}\right)}{1-\alpha_t(\beta_s+\alpha_s-1)}\right) + \left(1-\alpha_t^{2}\right)\alpha_t^{K}\left(\frac{1-\beta_s}{1-\alpha_s} + (\beta_s+\alpha_s-1)^{K+1}\right)\right]$$

$$+ \quad \frac{1}{2-\alpha_s-\beta_s}\alpha^{K-1}\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m+m^2}{2}}\left(1-\beta_s-(1-\alpha_s)(\beta_s+\alpha_s-1)^{K+m}\right) \tag{B-14}$$

Define:

$P_{loss}(1)$ = the average cell loss probability in case UI

Substitute equation (3-6), (B-3) and (B-14) into it, we have:

$$P_{loss}(1) = \frac{\overline{N_{off}}}{\overline{N_{on}} + \overline{N_{off}}} = \frac{\overline{\bar{N}_{AB}} + \overline{\bar{N}_{BG}}}{\overline{\bar{N}_{AB}} + \overline{\bar{N}_{BG}} + \overline{N_{BG}}} \qquad \text{(B-15)}$$

B.3 The Average Cell Loss Probability During a Random Cycle in case UII.1

The initial probability condition and initial joint PGF have been shown in equation (4-15) and (4-16) respectively in section 4.4.2.

B.3.1 The number of cells transmitted during the source's tagged periods

The analysis method for case UII.1 is the same as that for case UI which has been presented in the previous section B.2. Let us define the following notations first:

$P_{k_t}(z)$ =the transient marginal PGF of the number of cells generated by the end of k_t'th slot during the token generator's On period in case UII.1

$P(z)$ =the marginal PGF of the number of cells generated during the token generator's On period in case UII.1

$\overline{N_{BG}}$ = the average number of cells generated during the interval B to G that the token generator is On in case UII.1

So, depending on the initial conditions (equation 4-15) and the initial PGF (equation 4-16) in case UII.1, we will have the following results:

$$P_{k_t}(z) = [X(k_t)(Q_0(z,\Phi(k_t)))]\Big|_{y=1} = \tilde{X}(k_t) = \bar{C}_1\lambda_1^{k_t} + \bar{C}_2\lambda_2^{k_t}$$

$$P(z) = \sum_{k_t=1}^{\infty} P_{k_t}(z)\cdot Pr(k_t) =$$

$$\sum_{k_t=1}^{K}\left(\bar{C}_1\lambda_1^{k_t}+\bar{C}_2\lambda_2^{k_t}\right)(1-\alpha_t)\alpha_t^{\,k_t-1} + \left(\bar{C}_1\lambda_1^{K+1}+\bar{C}_2\lambda_2^{K+1}\right)\left(1-\alpha_t^{\,2}\right)\alpha_t^{\,K} + \sum_{k_t=K+2}^{\infty}\left(\bar{C}_1\lambda_1^{k_t}+\bar{C}_2\lambda_2^{k_t}\right)\left(1-\alpha_t^{\,k_t-K+1}\right)\alpha_t^{\,K+\sum_{i=2}^{k_t-K} i}$$

$$=\frac{(1-\alpha_t)}{\alpha_t}\cdot\left[\bar{C}_1\cdot\frac{\alpha_t\lambda_1-(\alpha_t\lambda_1)^{K+1}}{1-\alpha_t\lambda_1}+\bar{C}_2\cdot\frac{\alpha_t\lambda_2-(\alpha_t\lambda_2)^{K+1}}{1-\alpha_t\lambda_2}\right]+\left(\bar{C}_1\lambda_1^{K+1}+\bar{C}_2\lambda_2^{K+1}\right)\left(1-\alpha_t^{\,2}\right)\alpha_t^{\,K}+JJJ\Big)$$

$$= J + JJ + JJJ \qquad \text{(B-16)}$$

here J, JJ and JJJ are the first, second and third terms respectively in the above equation, and:

$$\overline{N_{BG}} = J'\big|_{z=1} + JJ'\big|_{z=1} + JJJ'\big|_{z=1} \qquad \text{(B-17)}$$

where:

$$J'\big|_{z=1} = (1-\alpha_t)\left[\left(\frac{\bar{C}_1(\lambda_1-\alpha_t^{\,K}\lambda_1^{K+1})}{1-\alpha_t\lambda_1}\right)'\Bigg|_{z=1}+\left(\frac{\bar{C}_2(\lambda_2-\alpha_t^{\,K}\lambda_2^{K+1})}{1-\alpha_t\lambda_2}\right)'\Bigg|_{z=1}\right]$$

$$=\frac{(1-\alpha_t)(1-\beta_s)(\alpha_s+\beta_s-1)^2\left(1-\alpha^K(\beta_s+\alpha_s-1)^K\right)}{(2-\alpha_s-\beta_s)^2(1-\alpha_t(\beta_s+\alpha_s-1))}-\frac{(1-\beta_s)(\alpha_s+\beta_s-1)\left(1-\alpha_t^{\,K}\right)}{(2-\alpha_s-\beta_s)^2}+\frac{(1-\beta_s)\left(1-(K+1)\alpha_t^{\,K}\right)}{2-\alpha_s-\beta_s}+\frac{\alpha_t\left(1-\alpha_t^{\,K}\right)}{1-\alpha_t} \qquad \text{B-18}$$

$$JJ'\big|_{z=1} = (1-\alpha_t^{\,2})\alpha_t^{\,K}(\bar{C}_1\lambda_1^{K+1}+\bar{C}_1(K+1)\lambda_1^{K}\lambda_1'+\bar{C}_2\lambda_2^{K+1}+\bar{C}_2(K+1)\lambda_2^{K}\lambda_2')\Big|_{z=}$$

$$= \frac{\left(1-\alpha_t^{2}\right)\alpha_t^{K}}{(2-\alpha_s-\beta_s)}\left[\frac{(\beta_s-1)(\beta_s+\alpha_s-1)^{K+2}}{2-\alpha_s-\beta_s}+\frac{(1-\beta_s)(1-\alpha_s-\beta_s)}{2-\alpha_s-\beta_s}+(K+1)(1-\beta_s)\right] \tag{B-19}$$

$$'JJ|_{z=1} = \sum_{k_t=K+2}^{\infty}\left(\bar{C}_1'\lambda_1^{k_t}+\bar{C}_1k_t\lambda_1^{k_t-1}\lambda_1'+\bar{C}_2'\lambda_2^{k_t}+\bar{C}_2k_t\lambda_2^{k_t-1}\lambda_2'\right)\left(1-\alpha_t^{k_t-K+1}\right)\alpha_t^{K+\sum_{i=2}^{k_t-K}}$$

$$= \frac{\alpha_t^{k-1}}{2-\alpha_s-\beta_s}\sum_{m=2}^{\infty}\left[\frac{(1-\beta_s)(\beta_s+\alpha_s-1)^{K+m+1}}{2-\alpha_s-\beta_s}-\frac{(1-\beta_s)(\beta_s+\alpha_s-1)}{2-\alpha_s-\beta_s}+(K+m)(1-\beta_s)\right]\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m^2+m}{2}} \tag{B-20}$$

here $m = k_t - K$

B.3.2 The probability distribution of the state of the source when the token generator's tagged period terminates

Same as in section B.2.2, in order to analyze the number of cells lost after the token generator's On period during the random cycle, we need to know the probability of the source's different state when the token generator's On period ends. In this section we use the same PGF method to get these probabilities.

Define:

$A_{k_t}(y)$ = the transient marginal PGF of the state of the source at the end of k_t'th slot during the token generator's tagged period

$A_{on}(y)$ = the marginal PGF of the state of the source at the end of the token generator's tagged period

from the initial conditions shown in equation (4-15) and (4-16), we have:

$$A_{k_t}(y) = Q_{k_t}(z,y)\Big|_{z=1} = \left(C_1\lambda_1^{k_t} + C_2\lambda_2^{k_t}\right)\Big|_{z=1}$$

$$= \frac{(1-\beta_s)(1-y)}{2-\alpha_s-\beta_s}(\beta_s+\alpha_s-1)^{k_t} + \frac{(1-\alpha_s)+y(1-\beta_s)}{2-\alpha_s-\beta_s} \qquad \textbf{(B-21)}$$

$$A_{on}(y) = \sum_{k_t=1}^{\infty} A_{k_t}(y)Pr(k_t)$$

$$=\frac{1}{2-\alpha_s-\beta_s}\left[(1-\beta_s)(\beta_s+\alpha_s-1)(1-\alpha_t)\cdot\frac{1-\alpha_t^{K}(\beta_s+\alpha_s-1)^{K}}{1-\alpha_t(\beta_s+\alpha_s-1)}+\left(1-\alpha_t^{2}\right)\alpha_t^{K}\left((1-\beta_s)(\beta_s+\alpha_s-1)^{K+1}+1-\alpha_s\right)\right]$$

$$+\ \frac{1}{2-\alpha_s-\beta_s}\left[(1-\alpha_s)\left(1-\alpha_t^{K}\right)+\alpha_t^{k-1}\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m^2+m}{2}}\left[(1-\beta_s)(\beta_s+\alpha_s-1)^{K+m}+(1-\alpha_s)\right]\right] \quad +$$

$$\frac{y}{2-\alpha_s-\beta_s}\left[(1-\beta_s)\left(1-\alpha_t^{K}\right)-(1-\beta_s)(\beta_s+\alpha_s-1)(1-\alpha_t)\cdot\frac{1-\alpha_t^{K}(\beta_s+\alpha_s-1)^{K}}{1-\alpha_t(\beta_s+\alpha_s-1)}+\left(1-\alpha_t^{2}\right)\alpha_t^{K}(1-\beta_s)\left(1-(\beta_s+\alpha_s-1)^{K+1}+\right.\right.$$

$$\left.\frac{y}{:-\alpha_s-\beta_s}\left[\alpha_t^{K-1}(1-\beta_s)\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m^2+m}{2}}\left(1-(\beta_s+\alpha_s-1)^{K+m}\right)+\left(1-\alpha_t^{2}\right)\alpha_t^{K}(1-\beta_s)\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)\right]\right. \textbf{(B-22)}$$

here $m = k_t - K$

Define:

$Pr(Off) = Pr$ (the source is Off when the token generator's tagged period ends)

$Pr(On) = Pr$ (the source is On when the token generator's tagged period ends) from the definition of the probability generating function and equation (B-22), we have:

$$Pr(Off) = \frac{1}{2-\alpha_s-\beta_s}\left[(1-\beta_s)(\beta_s+\alpha_s-1)(1-\alpha_t)\cdot\frac{1-\alpha_t{}^K(\beta_s+\alpha_s-1)^K}{1-\alpha_t(\beta_s+\alpha_s-1)}+\left(1-\alpha_t{}^2\right)\alpha_t{}^K\left((1-\beta_s)(\beta_s+\alpha_s-1)^{K+1}+1-\alpha_s\right)\right]+$$

$$\frac{1}{2-\alpha_s-\beta_s}\left[(1-\alpha_s)\left(1-\alpha_t{}^K\right)+\alpha_t{}^{K-1}\sum_{m=2}^{\infty}\left(1-\alpha_t{}^{m+1}\right)\alpha_t{}^{\frac{m^2+m}{2}}\left[(1-\beta_s)(\beta_s+\alpha_s-1)^{K+m}+(1-\alpha_s)\right]\right] \qquad \text{(B-23)}$$

$$Pr(On) = \frac{1}{2-\alpha_s-\beta_s}\left[(1-\beta_s)\left(1-\alpha_t{}^K\right)-(1-\beta_s)(\beta_s+\alpha_s-1)(1-\alpha_t)\cdot\frac{1-\alpha_t{}^K(\beta_s+\alpha_s-1)^K}{1-\alpha_t(\beta_s+\alpha_s-1)}+\left(1-\alpha_t{}^2\right)\alpha_t{}^K(1-\beta_s)\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)\right]$$

$$+\frac{1}{2-\alpha_s-\beta_s}\left[\alpha_t{}^{K-1}(1-\beta_s)\sum_{m=2}^{\infty}\left(1-\alpha_t{}^{m+1}\right)\alpha_t{}^{\frac{m^2+m}{2}}\left(1-(\beta_s+\alpha_s-1)^{K+m}\right)+\left(1-\alpha_t{}^2\right)\alpha_t{}^K(1-\beta_s)\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)\right] \text{(B-24)}$$

B.3.3 Analysis of the cell loss probability in case UII.1

Define:

$l_2(z)$ = the PGF of the number of cells lost after the token generator's ON period during the random cycle

$\overline{N_{BG}}$ = the average number of cells lost during the source's tagged periods

similarly, we will have the following results:

$$l_2(z) = Pr(off)\cdot z^0 + Pr(on)\cdot\left(\frac{(1-\alpha_s)z}{1-\alpha_s z}\right) \qquad \text{(B-25)}$$

$$\overline{N_{BG}} = I_2(z)'\Big|_{z=1} = Pr(on)\cdot\left(\frac{(1-\alpha_s)z}{1-\alpha_s z}\right)'\Bigg|_{z=1} = Pr(on)\cdot\frac{1}{1-\alpha_s} =$$

$$\frac{1-\beta_s}{2-\alpha_s-\beta_s)(1-\alpha_s)}\left[\left(1-\alpha_t^{K}\right)-(\beta_s+\alpha_s-1)(1-\alpha_t)\cdot\frac{1-\alpha_t^{K}(\beta_s+\alpha_s-1)^{K}}{1-\alpha_t(\beta_s+\alpha_s-1)}+\left(1-\alpha_t^{2}\right)\alpha_t^{K}\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)+\right.$$

$$\frac{1-\beta_s}{2-\alpha_s-\beta_s)(1-\alpha_s)}\left[\alpha_t^{K-1}\sum_{m=2}^{\infty}\left(1-\alpha_t^{m+1}\right)\alpha_t^{\frac{m^2+m}{2}}\left(1-(\beta_s+\alpha_s-1)^{K+m}\right)+\left(1-\alpha_t^{2}\right)\alpha_t^{K}\left(1-(\beta_s+\alpha_s-1)^{K+1}\right)\right. \quad \text{(B-26)}$$

Define:

$P_{loss}(2)$ = cell loss probability in case UII.1

substitute equation (3-6), (B-17) and (B-26) into the above, we will have:

$$P_{loss}(2) = \frac{\overline{N_{off}}}{\overline{N_{on}}+\overline{N_{off}}} = \frac{\overline{N_{AB}}+\overline{N_{BG}}}{\overline{N_{AB}}+\overline{N_{BG}}+\overline{N_{BG}}} \qquad \text{(B-27)}$$

B.4 Analysis of the average CLP during a random cycle

Define

P_{loss} = the cell loss probability during a random cycle

Combine the probabilities of cases UI, UII.1 and UII.2, which were shown in equations (4-33), (4-34), (4-35) and (4-36) respectively, with the cell loss probabilities in cases UI and UII.1 which were shown in equation (B-16) and (B-27) into P_{loss}, we will have:

$$P_{loss} = P_{loss}(1)[Pr(\text{ FI}) + Pr(\text{FII})Pr(\text{FII.2 })] + P_{loss}(2)Pr(\text{FII})Pr(\text{FII.1 }) \qquad \textbf{(B-28)}$$

For variant token generator's parameter $\alpha_t(i)$, we can compare the forced and unforced protocols too in the following figure B.1, which is obtained from equations (B-28) and (4-38).

(Note: the infinite summations in equations (B-6, 9, 14, 22, 24 and 26), we have cintinued to add until the deviation is smaller than 1.0E-5.)

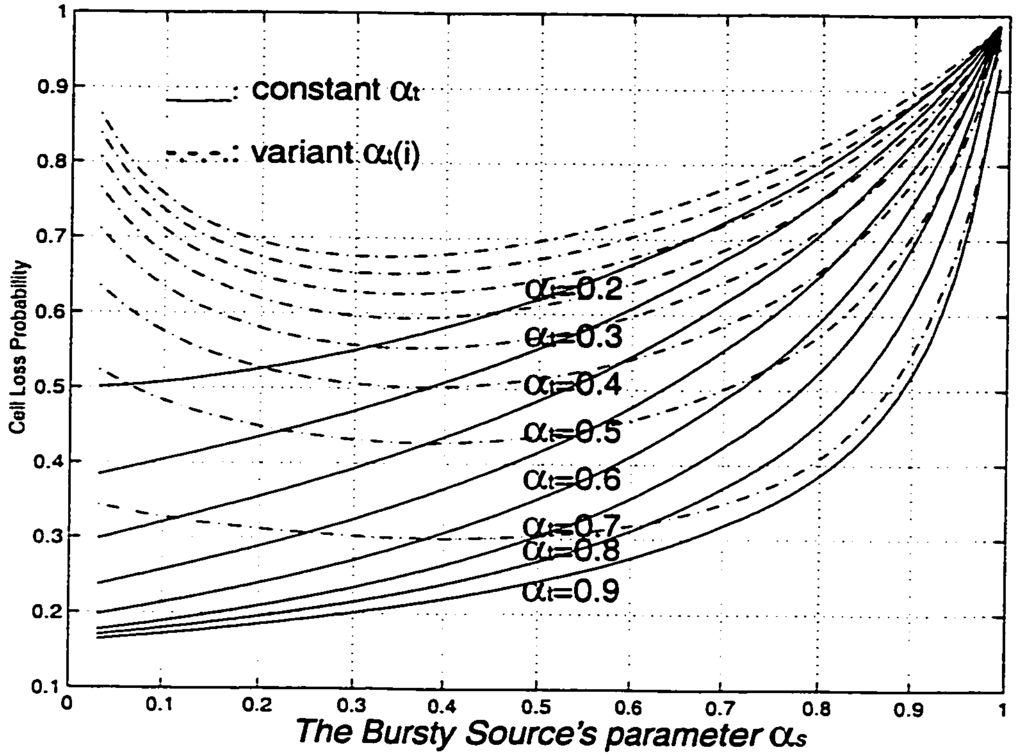


Figure B.1 The Comparison of CLP under the unforced protocol when $\beta_s = 0.95. \beta_t = 0.8. k = 4$

It shows that the variant parameter $\alpha_t(i)$ gives a higher CLP, the same results have been observed in the forced protocol too in figure A.1.

References

[1] Byeong Gi Lee, Minho Kang and Honghee Lee, *Broadband Telecommunications Technology,* Artech House, c1996.

[2] L.G. Cuthbert and J.C. Sapanel, *ATM: The Broadband Telecommunications solution*, Institution of Electrical Engineeers, c1993.

[3] Raif O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Arthec House, c1994.

[4] Faouzi Kamoun, "A New Approach in The Transient Analysis of ATM Multiplesers with Bursty Sources", Ph. D. thesis, Concordia University, 1995.

[5] Yuan Chen, "Departure Process Characterization Analysis of a Leaky Bucket Scheme", Master thesis, Concordia University, 1996

[6] James Martin, et. al., *Asynchronous Transfer Mode: ATM Architecture and Implementation*, Prentice Hall, 1997

[7] Richard D. Gitlin, Jeremiah F. Hayes and Stephen B. Weinstein, *Data Communications Principles*, Plenum Press, c1992.

[8] Z. H. Liang and M. Mehmet Ali, "The Modified Leaky Bucket Mechanism for Bursty Sources", IEEE CCECE 98, Vol. 2.

[9] Nanying Yin and Micheal G. Hluchyj, "Analysis of the Leaky Bucket Algorithm for On-Off Data Sources", IEEE Globecom 1991, pp 9.2.1-9.2.7

[10] Micheal G. Hluchyj and Nanying Yin, "On the Queueing Behavior of Multiplexed Leaky Bucket Regulated Sources", IEEE Infocom 1993, pp. 6a.3.1-6a.3.8

[11] Micheal G. Hluchyj and Nanying Yin, "A Second-Order Leaky Bucket Algorithm to Guarantee QOS in ATM Networks", IEEE Infocom 1996, pp 1090-

1096.

[12] H. Bruneel, “Queueing Behavior of Statistical Multiplexers with Correlated Inputs”, IEEE. Trans. Commum, Vol. 36, No. 12, pp. 1339-1341, 1988.

[13] Jianqin Weng, Ioannis Lambadaris and Micheal Devetsikiotis, “Fuzzy Leaky Bucket Congestion Control in ATM Networks with Markovian and Self-Similar Traffic”, IEEE ICC 97, pp. 838-843.

[14] Partho Pratim Mishra, “Effect of Leaky Bucket Policing on TCP over ATM Performance”, IEEE ICC 96, pp 1700-1706.

[15] Pedro I. Sanchez and Ravi R. Mazumdar, “Source Policing for ATM Systems Based on Peakedness Rate”, IEEE Infocom 96, Vol. 1, pp 3d.1.1-3d.1.8.

[16] Joseph S.M. Ho, Huseyin Uzunalioglu and Ian F. Akyildiz, “Cooperating Leaky Bucket for Average Rate Enforcement of VBR Video Traffic in ATM Networks”, IEEE Infocom 95, pp. 10a.4.1-10a.4.8.

[17] Athanasios V. Vasilakos and Antonios F. Atlasis, “Effectiveness of the LB-SELA Policing Mechanism in an ATM Network Node”, IEEE Globecome 95, Vol. 2, pp 627-631.

[18] Tetsuya Takine, Bhaskar Sengupta and Toshiharu Hasegawa, “A Conformance Measure for Traffic SHaping in High-speed Networks with an Application to the Leaky Bucket”, IEEE Infocom 94, Vol. 2, 4b.2.1-4b.2.8.

[19] Daniel Chonghwan lee, “Effects of Leaky Bucket Parameters on the Average Queueing Delay: Worst Case Analysis”, IEEE Infocom 94, Vol. 2, 4b.3.1-4b.3.8.

[20] M. Sidi, W. Z. Liu, I. Cidon and Gopal, “Congeation Control Through Input Rate Regulation”, IEEE Globecom 89, pp 1764-1768.

[21] K. Sohraby and M. Sidi, “On the Performance of Bursty and Correlated

Sources Subject to Leaky Bucket Rate-Based Access Control Schemes", IEEE Infocom 91, pp 426-434.

[22] Dimitris Logothetis and K. Travedi, "Transient Analysis of the Leaky Bucket Rate Control Scheme Undeer Poisson and On-Off Sources", IEEE Infocom 94, pp 4b.4.1-4b.4.8.

[23] Sabine Wittevrongel and Gerwig Bruneel, "Output Traffic Analysis of a Leaky Bucket Traffic Shaper Fed by a Bursty Source", IEEE ICC 94, pp 1581-1591

[24] Chikara OHTA et al, "Peak Rate Regulation Scheme for ATM Networks and Its Performance", IEEE Infocom 93, Vol. 2, pp 6a.4.1-6a.4.10.

[25] Arthur W. Berger, "Performance Analysis of a Rate-Control Throttle where Tokens and Jobs Queue", IEEE Journal of JSAC, Vol. 9, February 1991.

[26] Erwin P. Rathgeb, "Modeling and Performance Comparison of Policing Mechanisms for ATM Networks", IEEE Journal of JSAC, Vol. 9, No. 3, April 1991.

[27] Milena Butto, Elisa Cavallero and Alberto Tonietti, "Effectiveness of the 'Leaky Bucket' Polcing Mechanism in ATM Networks", IEEE Journal of JSAC, Vol. 9, No. 3, April 1991.

[28] Nanying Yin, "Analysis of a Rate-Based Traffic Management Mechanism for ABR Service", IEEE Globecom 95, pp. 1076-1082.

[29] Jaehong Park, Yongseok Yoon and Cheeha Kim, "An Effective Mean Rate Policing Scheme: Moving Ceiling Scheme", IEEE Globecom 95, pp. 1071-1075.

[30] M. Talla et al, "A New Congestion Control Scheme for Hybrid ATM/TDMA Netwroks", IEEE Globecome 95, pp 286-290.

[31] Kai Y. Eng, Mark J. Karo and Richard D. Gitlin, "Memory- and Channel-Sharing Techniques for Congestion Control in ATM Networks", IEEE Infocom 93,

pp 3a.3.1-3a.3.8.

[32] Lorne G. Mason, Anne Pelletier and Jacques Lapointe, "Towards Optimal Policing in ATM Networks", IEEE Infocom 95, pp. 10a.3.1-10a.3.10.

[33] Pramod Pancha and Magda El Zarki, "Leaky Bucket Access Control for VBR MPEG Video", IEEE Infocom 95, pp. 6d.3.1-6d.3.8.