NOTE TO USERS

uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

u Ottawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Hui Song

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Real Time Feedback Control Using Predictive States Estimation

TITRE DE LA THÈSE / TITLE OF THESIS

N. U. Ahmed

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

E. Gap

T. Yeap

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Real Time Feedback Control Using

Predictive States Estimation

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the degree of

Master of Science

Hui Song

Systems Science

University of Ottawa

Ottawa, Ontario, Canada K1N 6N5

January 2005

Your file Votre référence
ISBN: 0-494-11413-4
Our file Notre référence
ISBN: 0-494-11413-4

Acknowledgements

First of all, I would like to thank my advisor, Dr. Nasir Uddin Ahmed, for his immense help, excellent guidance and suggestion throughout my research. Without his patience and help, the work would not be done.

I wish to extend my sincere thanks to Mr.Cheng Li, Dr.Xinhuo Hua, Ms. YongJuan He and Ms. Hong Yan for their valuable comments, advice and encouragement.

Dedication

To my husband, Qiaofeng Guo, my sister and her husband, for their unconditional love, support and patience that has carried me through.

And I wish to thank my parents for their support and encouragement they have given me throughout my education.

Abstract

In this thesis, we present a real time feedback control strategy to optimize the dynamic performance of computer communication network. In previous studies [8-10], feedback delay, arising from communication delay, was shown to degrade system performance. Considering this negative impact of delay, we propose a new control law which predicts, in advance, the traffic and exercises control based on the predicted traffic.

In experiments, we apply the token bucket (TB) mechanism to construct a discrete dynamic system model [8-10], in which one multiplexor, linked to all the TBs, multiplexes the conforming traffic that have been policed at TBs. We demonstrate that the improvement of the system performance by presenting the simulation results corresponding to different stochastic traffic models. The experiments and analysis in this thesis provide valuable insight for the network researchers to do traffic optimal control [35].

Contents

List of Figures

List of Tables

Acronyms

CBR	Constant Bit Rate
FBC	Feed Back Control
FBCL	Feedback Control Law (simple)
LAN	Local Area Network
LMSE	Least Mean Square Error method
LRD	Long Range Dependence
MC	Monte Carlo Method
MPEG	Moving Picture Experts Group
PFC	Predictive Feedback Control
QoS	Quality of Service
SRD	Short Range Dependence
TB	Token Bucket
VBR	Variable Bit Rate

Symbols

C	Network link capacity (Constant)
e_i	The permission parameter for each traffic source
$g_i(t_k)$	Conforming traffic from the ith Token Bucket
$L_M(t_k)$	Traffic loss at the Multiplexor during time interval $[t_{k-1}, t_k)$
$L_T(t_k)$	Traffic loss at the Token Bucket during time interval $[t_{k-1}, t_k)$
$L_W(t_k)$	Waiting loss during time interval $[t_{k-1}, t_k)$
J	System cost according to the objective function
K	Total number of time intervals in a traffic sample
N_s	Total number of sample paths used while implementing MC method
N	Total number of individual traffic sources connecting to the system
$q(t)$	The Multiplexor state at time t
Q	Capacity of the multiplexor buffer
$r_i(t_k)$	Nonconforming traffic from the ith Token Bucket
T_i	Capacity of $i-th$ Token Bucket
T_d	Number of time unit delay
$u_i(t_k)$	Token generating rate of $i-th$ Token Bucket during time interval $[t_{k-1}, t_k)$
$V_i(t_k)$	Traffic arrival from $i-th$ traffic source during the time interval $[t_{k-1}, t_k)$
W_s	Observation Window Size
$X(t_k)$	the state of the dynamic system during $k-th$ time interval
α	Weight factor of the multiplexor loss

β	Weight factor of TB losses
γ	Weight factor of waiting loss
$\rho_i(t_k)$	State of $i-th$ Token Bucket during time interval $[t_{k-1}, t_k)$
τ	Time period (unit)
η	Network utilization

Chapter 1

Introduction

1.1 Motivation

Over the past two decades, the Internet has developed into a global, complex, fast and diverse communication system. The data flows are increasing every millisecond by enormous utilities across the networks, such as web surfing, teleconferencing, and E-commercial applications. Moreover, the quality of delivery service becomes critical with respect to packet loss, transmission delay and delay variation [4,22,25]. As a result, a number of control mechanisms have been proposed to efficiently manage and optimally control network traffic, i.e., allocating network resources, reducing delay, and minimizing congestion. The Token Bucket (TB) or Leaky Bucket (LB) algorithm is one of the widely employed control mechanisms which has been used to effectively shape and police traffic sources at the edge of access nodes [1-3, 8-10, 21].

A TB (or LB) is described by a pair of parameters: token generation rate u and bucket capacity T. It can be implemented as a traffic filter between the host and the network, or between the routers. The operation of TB can be simply stated as follows: the incoming tokens keep accumulating until they reach the capacity of the bucket (T); if the incoming

tokens exceed T, tokens in excess of the capacity will be dropped. If the Token Bucket acts as a traffic policer and there are sufficient tokens in the pool, the incoming traffic will be transferred to the network without delay and the same or any other equivalent amount of tokens will be removed from the token pool. If there are not sufficient tokens, the packets will be discarded without entering the networks. Therefore, the TB can limit the traffic transmission rate to the number of tokens available in the bucket [4-10,21].

Since TB (LB) algorithm was applied to real time transmission by IETF, it has been implemented as traffic control policy in wide areas such as admission control, congestion control and access control. Reviewing the study found in the literature, M. Butto, E. Caverolla, and A. Tonietti used LB to shape the incoming traffic and reduce the burstiness at the network boundaries. Thus each individual traffic source can be accommodated by an average bandwidth [1]. In order to effectively reserve network resources for users, P. Tang and T.Tai had investigated the derivation of the suitable TB parameters (T and u) based on the observation of traffic flows [5].

Similarly, applying TB algorithm at an access node of computer networks, Ahmed, Wang and Barbosa constructed a dynamic system model [8]. In this model, token buckets police incoming traffic, and one multiplexor serves all the superposition of the conforming traffic. System performance was investigated with a simple feedback control law. The results demonstrated improvement of QoS corresponding to the deterministic traffic. However, the study only gave a simple and scalable infrastructure for the study of TB, and feasible optimal feedback control laws had not been discussed in depth and token generation rate had not been optimized [8]. To further study the model, B. Li et al. employed the principles of dynamic programming and Genetic Algorithm (GA) to achieve an optimal control law (token generation rate) corresponding to the MPEG traffic traces [9]. However, for stochastic incoming traffic, this method may not be practical. In brief, in these papers

[8-9], there are some limitations that need to be improved and completed, such as:

- Although a simple concept of a traffic model was proposed, only one type of deterministic traffic model was fed to the system as its input traffic [8-9]. In fact, the real time network traffic exhibits short range dependence (SRD) and long range dependence (LRD), and most of traffic are stochastic.

- As mentioned before, the feedback control laws were not feasible. They were either too simple, because the optimization of token generation rate was not considered [8], or impossible to be implemented in practice, because the long period of enormous computation had been required before taking action [9]. In addition, the feedback control delay, arising from the communications, was not taken into consideration in these works.

In 2004, N.U. Ahmed, H. Yan and L.O. Barbosa further improved the model by feeding various types of stochastic traffic and considering all the details of the feedback control scenario [10]. The numerical results demonstrated that this system could be adapted to any kind of stochastic traffic. However, because of the feedback delay (arising from communications), system performance was highly degraded comparing with that in the absence of delay. Due to the time delay, the traffic information captured at the current time is the lagged or delayed version of the actual one. Provided with the delayed information, the controller cannot make an appropriate decision which leads to performance degradation. For that reason, in practice, the impact of delay cannot be ignored, and we have to take it into account and compensate it. However, in the above papers [8-10], the effort has not been made to deal with this problem and solve the impact of communication delay in feedback control.

Some research demonstrated that predicting the delayed information is one of the alternative methods to relax the impact of the communication delay [11-12,14-15]. Prediction

techniques, such as forecasting the future behavior of the traffic, can effectively reduce performance degradation caused by delay, and prevent network congestion by dynamic bandwidth allocation. S. Hu, A. Duel-Hallen and H. Hallen predicted the number of future users, and adjusted the channel capacity in advance to avoid system performance degradation [11]. Similar work had been done in satellite networks [12]. A. Pietrabissa and E. Guainella dynamically assigned the amount of bandwidth to each connection based on the estimation of network connection throughput. In the same way, A. Bhattacharya, A. G. Parlos, and A.F. Atiya focused on the single-step-ahead (SSP) and multi-step-ahead (MSP) traffic predictor to forecast MPEG-4 video traffic trace [14]. They had employed a nonlinear estimation tool; however, the method was applied off-line and wasn't adaptive. In reference [15], bandwidth requirements had been predicted to solve the admission control problem subject to the constant bit rate traffic. However, the authors were not sure whether this method was appropriate for other types of traffic.

Therefore, naturally inspired by prediction techniques, it would be very interesting that we apply such techniques to solve the problem encountered in these papers [8-10]. A real time feedback control scheme will be proposed in this thesis, and the delayed state information will be replaced with the predicted ones based on the past history measured online.

1.2 Objective

The key problem we address in this thesis is how to reduce the impact of feedback delay which leads to significant system performance degradation. The main objectives of this study include

1. developing an on-line traffic predictor that can capture major properties of network traffic: LRD and self similarity;

2. exploring the relationship of Hurst parameters and system performance;

3. proposing a predictive feedback control law which can be implemented on line and reduce the impact of feedback delay.

1.3 Contribution

In this study, we have proposed an on-line traffic predictor using the principle of Least Mean Square Error (LMSE) [13]. Our approach aims to solve system degradation caused by the lagged feedback information. The main contribution of the study differs from those methods proposed in previous studies in the following respects.

- Firstly, unlike the studies focused on one fixed type of traffic [8-9,14-15], our approach is not restricted to the underlying structure of traffic. The method is independent of the traffic model (statistics) and is useful for both SRD and LRD precesses.

- Secondly, comparing with the study of using off-line training [14], our algorithm only requires some matrix manipulations and takes less computation time. As a result, the method works well for online implementation.

- Thirdly, in the studies [8-10], the authors either considered the scenarios only in the absence of delay or presented the impact of delay without solution. In this thesis, we consider the scenario with information delay, and propose a predictive feedback control law, which compensates the feedback delay and greatly improves system performance.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 reviews the basic concepts used throughout the thesis, including traffic characteristics (long range dependence , short range dependence and self-similiarity), Quality of Service (QoS), TB algorithm and traffic policing and shaping. In addition, the methods used in the thesis are explained, e.g., fractional Brownian Motion simulation and Monte Carlo method.

Chapter 3 firstly describes a complete dynamic system model, including a single traffic model, a token bucket model and a multiplexor model. Then, in order to evaluate the overall system performance, an objective functional is introduced in section 3.4, and the utilization is defined in section 3.5.

Chapter 4 discusses control strategies. First, we analyze the feedback control with and without time delay. Secondly, we propose a predictive feedback control using the principle of least mean square error (LMSE).

Chapter 5 describes the configuration of the system, and some assumptions made for simulation experiments. Furthermore, the specifications of traffic traces are also presented in this chapter.

Chapter 6 analyzes and summarizes numerical results which evaluate the performance of LMSE predictor regarding to different values of prediction time delay and observation window size. Then, the performance of the complete system model is investigated using LMSE predictor.

Chapter 7 summarizes the achievements of the work and gives some recommendations and ideas for further research.

Chapter 2

Basic Concepts

The terminology and fundamental concepts related to this thesis are briefly introduced in this chapter. First, the definition of Quality of Service (QoS) is given in Section 1. Secondly, the characteristics of network traffic are covered in Section 2, including short range dependence, long range dependence and self-similarity. In Section 3 and Section 4, the TB algorithm and its usage are reviewed and extended to traffic policing and shaping. Fractional Brownian Motion is explained as a mathematical foundation of the traffic simulator in Section 5, and Monte Carlo method is introduced to compute the expected value of random numbers in Section 6.

2.1 Quality of Service

Quality of Service (QoS) in ISO (the International Standard Organization) standards is defined as a concept to specify how good the offered networking services are [24-25]. QoS can be characterized by a number of specific parameters. Generally, the parameters are associated with network service performance, such as end-to-end delay (latency), variation in end-to-end delay (jitter), packet/cell loss and bandwidth. For the purpose of

the satisfaction of QoS in communication networks, different control technologies are used to deliver traffic over the Internet. According to the level of the control over performance parameters, e.g., delay, loss, jitter, and/or bandwidth, end-to-end QoS service can be categorized into three basic levels:

- Best Effort Service: the first level of QoS, providing the basic connectivity with no guarantees, servicing the traffic flows by FIFO queues. The Internet is an example of best effort level of service, which is acceptable for email, most HTML, and FTP;
- Differentiated Service: the intermediate level of QoS, providing expedited process for some traffic, servicing QoS well to the large classes of data or aggregated traffic;
- Guaranteed Service : the highest level of QoS, providing a reservation of network resources to the specific classes of traffic, thus guaranteeing the delivery with required level of service [24-25].

2.2 Traffic Characteristics

In early days, it was common to apply Poisson process models (short range dependence) to characterize network traffic behavior. However, current networks integrate an increasing variety of traffic streams (applications), such as video, audio and data. As a result, the network traffic becomes more complex. Recent studies on actual traffic traces show that network traffic exhibits long range dependent or self-similar features [26-28]. Thus the traditional models (simple Poisson model) become inadequate to capture the diverse traffic characteristics in current networks. The trend to represent the characteristics of actual aggregated traffic becomes to synthesize a statistical model of long range dependence.

In the following section, some basic theoretical definitions will be reviewed to explain short range dependence, long range dependence and self-similarity.

2.2.1 Short Range Dependence and Long Range Dependence

Consider a sequence of traffic arrival $V(t_i)$ $(i = 0, 1, 2, \ldots)$, where each $V(t_i)$ is a discrete-time second order stationary stochastic process with mean μ, variance σ^2, and auto-correlation function

$$\gamma(k) = cov(V(t_i), V(t_{k+i})) = E[(V(t_i) - \mu)(V(t_{k+i}) - u)]/\sigma^2, \quad for \quad k = 0, 1, 2, \ldots.$$

According to γ_k, this second order stationary process can be classified into either short range dependent process or long range dependent process:

- Short Range Dependent Process, if

$$\sum_{k=-\infty}^{\infty} |\gamma(k)| < \infty; \tag{2.1}$$

- Long Range Dependent Process, if

$$\sum_{k=-\infty}^{\infty} |\gamma(k)| = \infty. \tag{2.2}$$

Here we are interested in long range dependent process. For long range dependent process, it turns out that $\gamma_{(k)}$ decays to zero as k increases, according to the power law as follows [21-22,29]:

$$\gamma(k) \sim const\ k^{-\alpha} \qquad as \quad k \to \infty, \quad for \quad 0 < \alpha < 1. \tag{2.3}$$

In summary, the auto-correlation function of the traffic with short range dependent property (SRD) falls off exponentially or faster. On the other hand, it decays much slower than exponential for long range dependent traffic, usually obeying some types of power law [30].

2.2.2 Self-Similarity

Comparing Figure 2.1(a) with Figure 2.1(b), we can clearly obtain the notion of self similarity. The curves at different time-scales resemble each other in Figure 2.1(a). However, in Figure 2.1(b), the process is more irregular and chaotic at large time-scales. Hence, for a self-similar process, the characteristic will retain the similarity regardless of the scale at which it is viewed. The scale can be space (length, width) or time.

With this introductory figure, the definition of self similarity is given subsequently. Let $V^{(m)}(t_i) \quad (i = 0, 1, 2, 3 \ldots)$ be an aggregated series obtained from the original $V(t_i)$ over non overlapping time intervals of size m, i.e.,

$$V^{(m)}(t_i) = \{V(t_{(i-1)*m+1}) + V(t_{(i-1)*m+2}) + \ldots + V(t_{(i*m)})\}/m.$$

Then the auto-correlation function is given by:

$$\gamma^m(k) = E[(V^{(m)}(t_i) - \mu)(V^{(m)}(t_{k+i}) - \mu)]/\sigma^2, \quad for \quad k = 0, 1, 2, \ldots.$$

$V(t_i)$ is self-similar if the following conditions hold:

$$\begin{aligned} \gamma(k) &\sim k^{-\alpha} \qquad as \quad k \to \infty, \quad for \quad 0 < \alpha < 1. \\ \gamma^m(k) &\sim \gamma(k), \quad as \quad k \to \infty, m \to \infty, \quad for \quad 0 < \alpha < 1. \end{aligned}$$

This implies that if k and m large enough, auto-correlation depends only on k, not m. That is, the correlation structure in the original process is preserved and repeated in the aggregated process.

For the historical reasons, we use the Hurst parameter $H = 1 - \alpha/2$ instead of α. The Hurst parameter characterizes the stochastic process as follows:

- If $1/2 < H < 1$ (which is equivalent to $0 < \alpha < 1$), the process has long-range dependence;

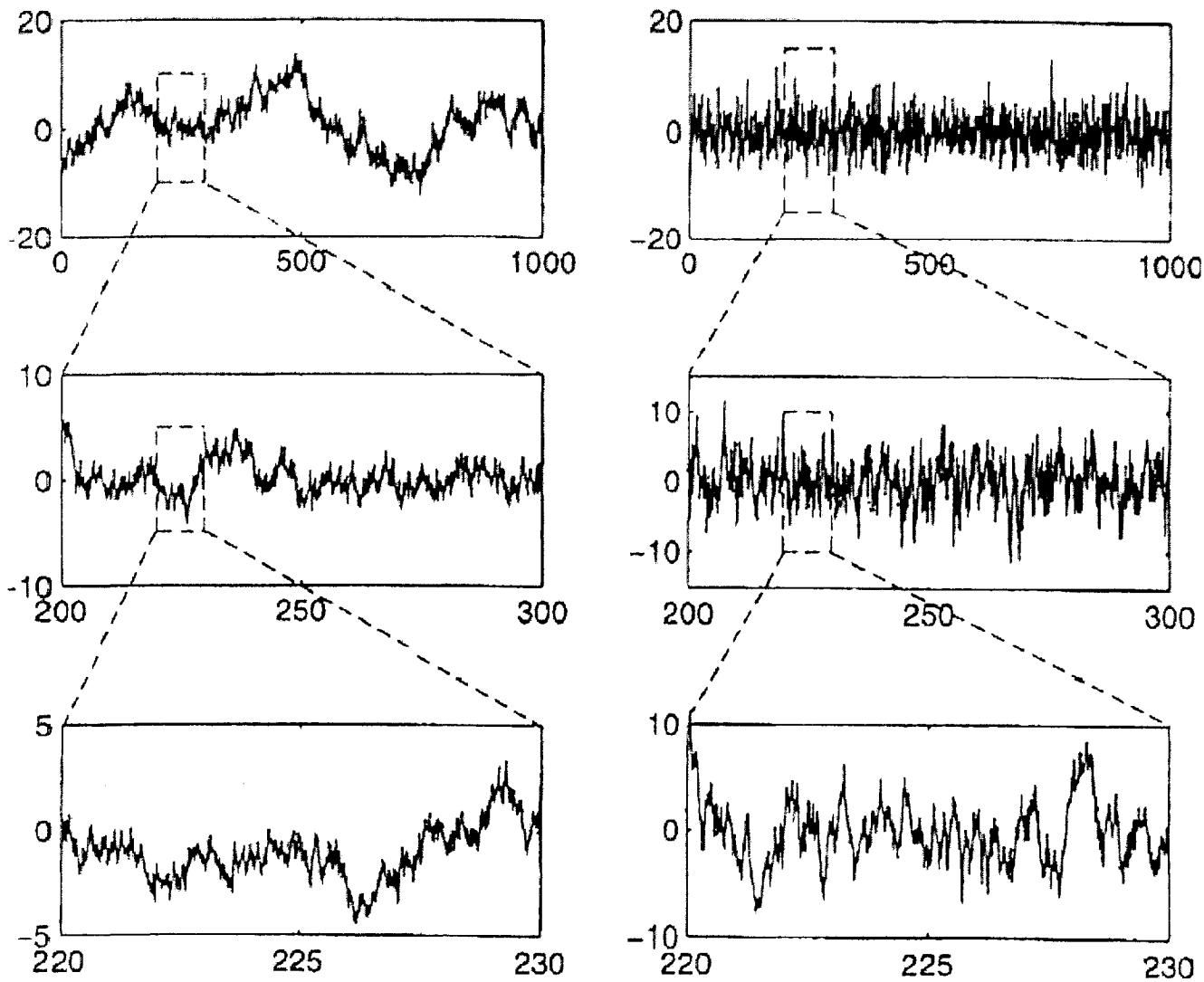


Figure 2.1: Comparison of a self-similar and a non self-similar processes in different time scale (Fig 8.3 in [21])

- If $H \leq 1/2$, we imply that $\alpha = 2(1 - H) > 1$, thus the process has short-range dependence.

2.3 Token Bucket Terminology

The token bucket algorithm, a variant and improved version of the leaky bucket algorithm [6], is one of the well-known traffic control mechanisms used nowadays. It has been implemented to regulate the transmission rate at the access node to the network [20].

According to TB mechanism, a TB model is composed of one token bucket (pool), with a predetermined capacity and a token generation rate [1-6]. There are two types of token bucket, unbuffered TB (see Figure 2.2) and buffered TB (see Figure 2.3) [3]. Simply implying from their names, the major difference is whether a data buffer has been added between the traffic source and the token bucket or not. The data buffer is usually used to hold the incoming traffic in the case of insufficient tokens available (explained in Section 2.4).

The operation of the TB is stated as follows: tokens flow into the bucket at the specified rate; if the number of the tokens reaches the bucket capacity while new tokens are generated, the newly generated tokens will be thrown away. Traffic flows through the TB to the network (depicted in Figure 2.2-2.3). Before being granted to access to the network, the traffic (or called packets) must obtain a number of tokens, which are equal to the same or any other equivalent number of the packet size. The packets will occupy the tokens in the order of their arrivals. Traffic that successfully obtains sufficient tokens will be passed on to the network instantaneously, and the corresponding number of tokens are removed from the token bucket simultaneously. In the case of failing to obtain enough tokens, there are two different situations corresponding to the type of the TB. For unbuffered

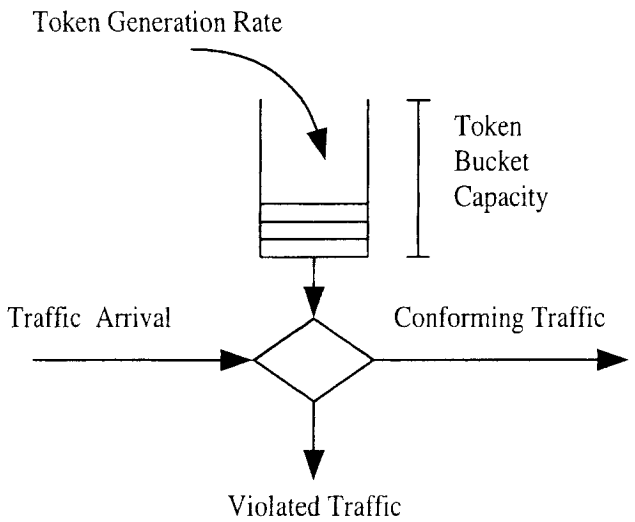


Figure 2.2: A Unbuffered Token Bucket Model

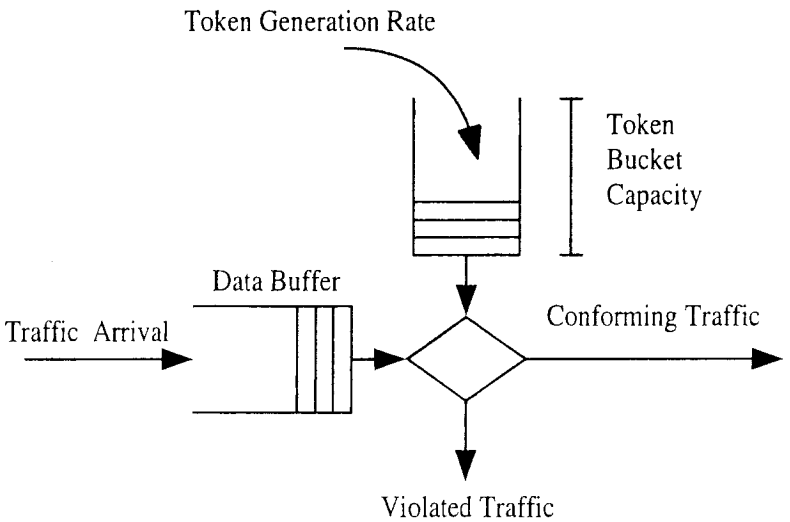


Figure 2.3: A Buffered Token Bucket Model

TB, the traffic will be discarded immediately without entering the network, whereas for buffered TB, the incoming traffic can be temporarily stored in the data buffer and wait until sufficient tokens available. If the traffic buffer has infinite capacity, there is no traffic loss during the transmission [1-7].

2.4 Traffic Policing and Shaping

A TB model can be used as a shaping or policing mechanism depending on the data buffer. For traffic policing implementation, there is only one token bucket without the data buffer (see Figure 2.2). For traffic shaping implementation, a data buffer is inserted (See Figure 2.3) to temporarily hold the incoming packets.

Both the shaping and policing schemes enable the network controllers to accommodate to the burstiness during traffic transmission and restrict the output rate to a predefined value. They share the same way to identify traffic descriptor violations. However, the differences between them are as follows:

- Traffic policers can limit the data source to a configurable rate at the network border, which ensures that each traffic source will not occupy more bandwidth than what has been allocated (see Figure 2.4 [4]). In short, it bounds the traffic rate below a specified value, and drops the packets immediately on traffic exceeding this rate;
- By using a data buffer, traffic shapers buffer the input traffic at the end of systems and then send it out to the network at a controlled rate (see Figure 2.5 [4]). In short, it smooths the traffic burstiness, re-parameterizes the traffic rate to comply with a predefined rate, and introduces the artificial delay before transmission.

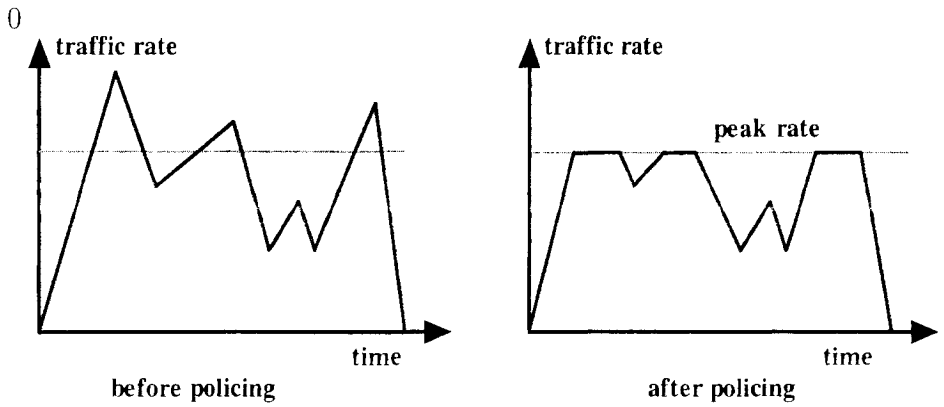


Figure 2.4: Policing Function

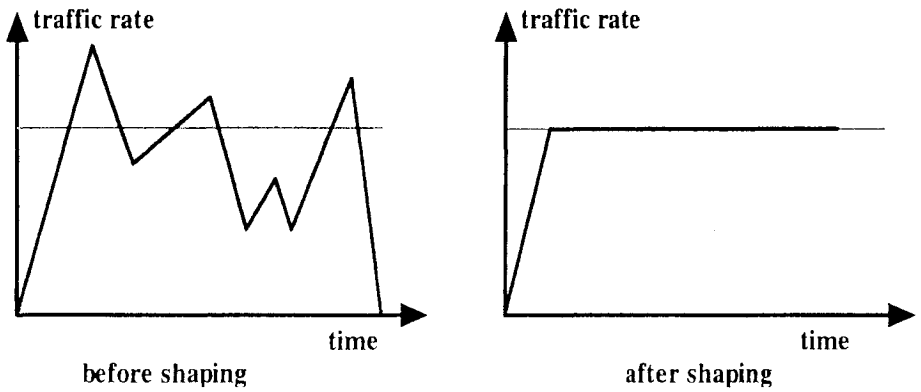


Figure 2.5: Shaping Function

TB is used as a traffic policer in the thesis.

2.5 Fractional Brownian Motion

Brownian motion was first observed by Robert Brown in 1827. He noticed that the motion of pollen grain particles suspending in fluid is an irregular and unceasing movement. This random movement was then called "Brownian motion" [23]. Here we introduce two types of Brownian motion: standard Brownian motion and fractional Brownian motion.

Standard Brownian Motion

First, we look at the mathematical explanation of standard Brownian motion. A standard Brownian motion or standard Wiener process over $[0, T]$, noted as $B(t)$, is a continuous function on $t \in [0, T]$ and satisfies the following three conditions:

1. B(0)=0 (with probability=1);
2. For $0 \leq s < t \leq T$, the random variable given by the increment $B(t) - B(s)$ is normally distributed with mean 0 and variance $t - s$:
3. For $0 \leq s < t < u < v \leq T$, the increments $B(t) - B(s)$ and $B(v) - B(u)$ are independent.

Fractional Brownian Motion

Fractional Brownian motion is a random walk process that has a defined Hurst exponent, denoted by $B_H(t)$, $t \in R$. With the Hurst parameter $H \in (0, 1)$, it is a continuous and centered Gaussian process, and satisfies the following conditions:

1. B(0)=0 (with probability=1);

2. For $0 \leq s < t \leq T$, the random variable given by the increment $B_H(t) - B_H(s)$ is normally distributed with mean 0 and variance $|t-s|^{2H}$;

3. $Cov[B_t^H \ B_s^H] = \frac{\sigma^2}{2}(t^{2H} + s^{2H} - \mid t-s \mid^{2H})$, where $s, t \geq 0, \sigma^2$ is the variance.

Furthermore, the constant H determines the covariance of the stationary increments. When $H > 0.5$, the increment is positive correlated, whereas when $H < 0.5$, the increment is negative correlated. When $H = 0.5$, then $B_H(t)$ coincides with the standard Brownian motion $B(t)$. The value of parameter H represents a measure of self-similarity in the traffic, and "burstiness". It becomes possible to provide a traffic model to describe the complex and self-similar characteristics of current network traffic.

2.6 Monte Carlo Method

The basis of "Monte Carlo method", along with its name, was apparently first formed during World War II in Los Alamos when Ulam, Nicholas Metropolis and John von Neumann applied this technique to build better atom bombs. With the appearance of faster and faster computers, Monte Carlo methods widely spread in fields as diverse as neutron transport problems, queuing theory, computer engineering and many others. By simulating random quantities, it can approximately solve the problems which cannot be evaluated analytically or are too complex to be calculated easily [31].

In this thesis, the method works specifically on solving the integration problem, the computation of an expected value of random variables and functions of random process. Consider a random variable of X and compute the expected value of an arbitrary function g of X, it can be calculated with

$$E(g(X)) = \sum_{x \in \chi} g(x) f_X(x)$$

if X is discrete, and

$$E(g(X)) = \int_{x \in \chi} g(x) f_X(x) dx$$

if X is continuous, where $f_X(x)$ is probability mass function or probability density function which is greater than zero on a set of values χ. Because sometimes the result cannot be obtained directly, Monte carlo estimator will approximate $E(g(x))$ by taking a large number of samples over the domain χ and averaging the values of $g(x_i)$:

$$\widetilde{g}_N(x) = \frac{1}{N} \sum_{i=1}^{N} g(x_i).$$

By the strong law of large numbers, if N gets large, $\widetilde{g}_N(x)$ shall be close to $E(g(x))$.

$$E(g(x)) = \lim_{N \to \infty} \widetilde{g}_N(x).$$

Chapter 3

System Model with Performance Measures

In this chapter, a complete system model is presented with token buckets, which are used as traffic policing elements in a computer network. In Section 1, based on a general traffic model for simulation implementation, a doubly stochastic Poisson process driven by fractional Brownian motion is developed as the input traffic source. In Section 2, the analytical description is given to a TB based model. Based on this single TB model, a complete system model is extended, which consists of N traffic sources, N TB models and one multiplexor in Section 3. The objective function and the utilization are defined in Section 4 to evaluate the dynamic system model.

The following symbols and their meanings will be used throughout the thesis:

1. $\{a \wedge b\} \equiv Min\{a, b\}$, $\{a \vee b\} \equiv Max\{a, b\}$, for $a, b \in R$;
2. $\{a \wedge b\} \equiv \{a_i \wedge b_i, i = 1, 2, \cdots, N\}$, $\{a \vee b\} = \{a_i \vee b_i, i = 1, 2, \cdots, N\}$, for $a, b \in R^n$;
3. For boolean function $I(S) = \begin{cases} 1, & \text{if the statement S is true} \\ 0, & \text{if the statement S is false.} \end{cases}$

3.1 Traffic Model

Over the Past two decades, a number of traffic models have been proposed and studied in traffic management area. Traditional characterization of the Internet traffic is based on Poisson process, which exhibits short range dependence, Bernoulli process or more generally doubly stochastic Poisson processes [26]. However, with the growth of the Internet and the increasing diverse of traffic, recent studies have shown that network traffic has self-similarity characteristic and long-range dependence (LRD) [27-28]. Self similarity means that a certain property of traffic behavior is preserved over space and/or time scales, and LRD is said to exhibit long term correlations that decay at rates slower than exponential. On the other hand, the correlation functions of traditional traffic models decay exponentially or faster.

3.1.1 A General Model

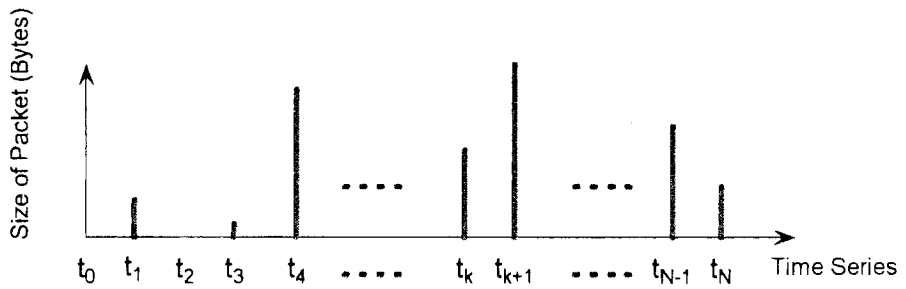


Figure 3.1: A General Traffic Model

In Figure 3.1, a general model is presented to simulate the incoming traffic [8-10]. A discrete time series represents the traffic arrivals $\{V(t_k), k = 0, 1, 2, \cdots, K\}$ generated by the users' applications, where $V(t_k)$ denotes the packet size (measured in bytes) arriving during non-overlapping time intervals $[t_k, t_{k+1})$. This can be modeled as a marked point

process where the event time is the point and the size of the packets is the mark. For simplicity, the time intervals are assumed of equal length.

3.1.2 Doubly Stochastic Poisson Process Model

A doubly stochastic Poisson process is one example of inhomogeneous processes whose rate varies stochastically. It can be viewed as two forms of randomness: that's associated with the stochastically varying rate, and that's associated with the underlying Poisson nature of the process even if its rate is constant. Let $N(t)$ be a homogenous Poisson process and let $\lambda(t), t \geq 0$ be a stochastic process independent of $N(t)$ and act as its intensity function. That is, $N(t)$ is a Poisson process conditional on $\lambda(t)$, which itself is a stochastic process.

The intensity function chosen in this thesis is a nonnegative function of FBM, denoted by $B_H(t)$. FBM itself is a self-similar process and constructed through the following integral transformation of standard Brownian motion $\{B(t), t \geq 0\}$.

$$B_H(t) = \int_0^t K_H(t-s)dB(s), t \geq 0, \tag{3.1}$$

where $K_H(t)$ is given by

$$K_H(t) = C_H t^{(H-\frac{1}{2})}, \qquad \frac{1}{2} < H < 1,$$

and C_H is any constant.

Thus, the intensity of Poisson process is obtained by

$$\lambda(t) \equiv |B_H(t)|, t \geq 0.$$

Therefore, a doubly stochastic Poisson process has been generated, which exhibits self-similar and LRD. A sample of DSPP trace is shown in Figure 3.2.

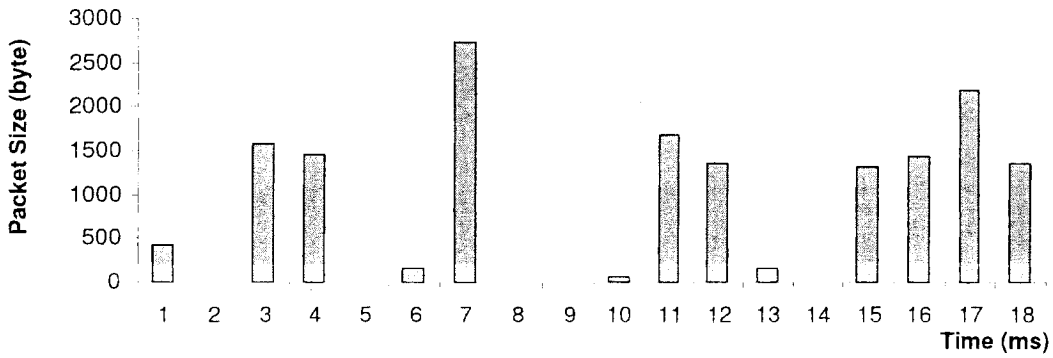


Figure 3.2: A DSPP Traffic Model

3.2 Basic Model for A Token Bucket

In this section, we review the token bucket model developed in previous works [8-10]. The TB model illustrated in Figure 3.3 is without a waiting buffer and acts only as a traffic policer in this thesis.

The operation of this dynamic model of TB can be simply stated as follows. The incoming tokens keep accumulating until they reach the capacity of the bucket T. The number of tokens offered during $k-th$ time interval is denoted by $u(t_k)$ and the number of tokens stored in the token pool is denoted by $\rho(t_k)$. If the incoming tokens exceed the capacity , tokens in excess of the capacity will be dropped. Thus, the acceptable tokens during $k-th$ time interval can be represented by $u(t_k) \wedge [T - \rho(t_k)]$. If the packet size of the arriving traffic $V(t_k)$ is less than the number of tokens available in the pool, the traffic is marked as conforming traffic and will be immediately passed on to the network for queuing up in the multiplexor. At the same time a number of tokens equal to the size of the packet are taken out of the token pool. Thus the state of TB at any time is determined by the algebraic sum of three terms: tokens left over from the previous time interval, new

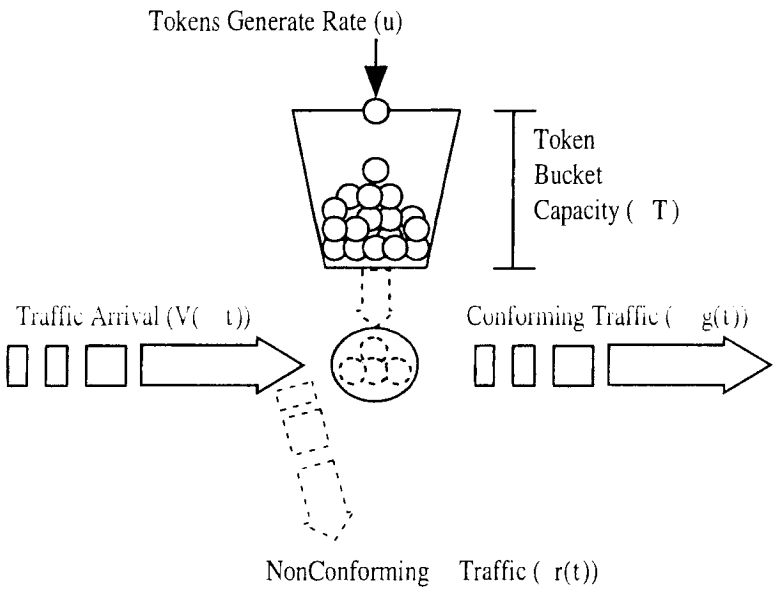


Figure 3.3: A Token Bucket Model

tokens freshly added during current time interval and tokens consumed during current time interval. As a result, the dynamics governing the status of the TB is given by the following expression:

$$\rho(t_{k+1}) = \rho(t_k) + \{u(t_k) \wedge [T - \rho(t_k)]\} - g(t_k), \tag{3.2}$$

where $g(t_k)$ denotes the conforming traffic given by:

$$g(t_k) = V(t_k) I\{V(t_k) \leq [\rho(t_k) + u(t_k) \wedge [T - \rho(t_k)]]\}. \tag{3.3}$$

Obviously, the conforming traffic is equal to the arriving packet $V(t_k)$ provided that the size of traffic arrival is smaller than or equal to the number of the tokens available in the pool. Otherwise, the traffic arrival has to be dropped at the token bucket, then $g(t_k)$ equals zero.

The nonconforming traffic is defined as the traffic dropped at the token bucket by the system, and it is given by

$$r(t_k) = V(t_k) - g(t_k). \tag{3.4}$$

3.3 A Complete System Model

A mathematical model is constructed to simulate a computer network. In this model, N token buckets serve N individual users (traffic streams), and all the conforming traffic, which have already been policed by the token buckets, are coupled to a multiplexor connected to an outgoing link with a (bandwidth) capacity C. This is illustrated in Figure 3.4.

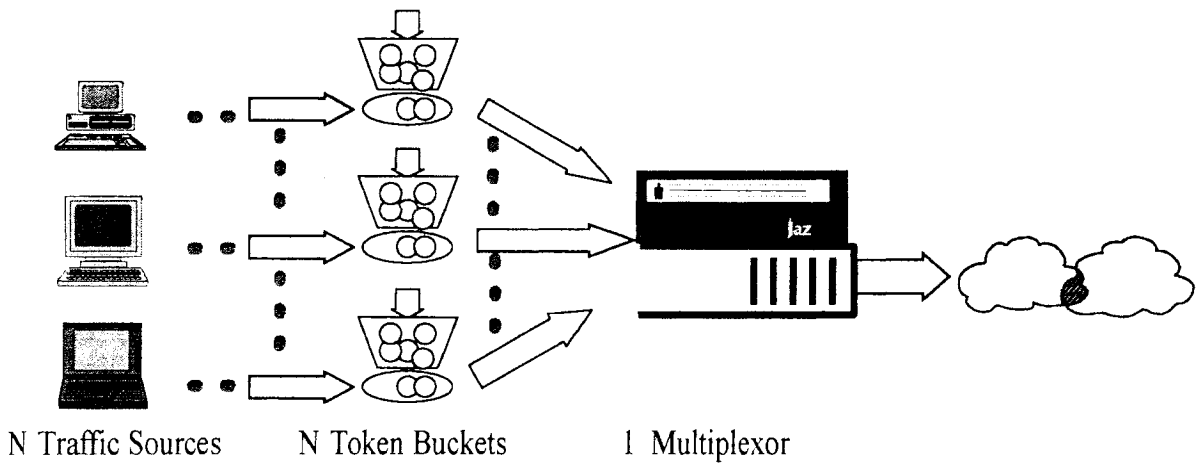


Figure 3.4: A Token Bucket Model

When the incoming traffic arrives at TBs, each TB implements its algorithm to police the packet arrival by dropping the nonconforming traffic and accepting the conforming traffic. All the conforming traffic are multiplexed and queued up to enter the multiplexor. As a matter of fact, not all conforming traffic from TBs can be accepted by the multiplexor because of the size limitation of the buffer size Q and the link capacity (speed) of the

accessing node. If the sum of these traffic exceeds the multiplexor size, some part of the conforming traffic may be dropped. The discarded traffic is defined as the traffic loss at the multiplexor.

The dynamics of a single TB is given by Equation (3.2). For a system consisting of N TBs, served by a single multiplexor, the associated variables are given by multi-dimensional vectors from R^N listed as follows:

$$\begin{cases} T \equiv (T_1, T_2, \cdots, T_N)' \\ \rho(t_k) \equiv (\rho_1(t_k), \rho_2(t_k), \cdots, \rho_N(t_k))' \\ V(t_k) \equiv (V_1(t_k), V_2(t_k), \cdots, V_N(t_k))' \\ u(t_k) \equiv (u_1(t_k), u_2(t_k), \cdots, u_N(t_k))' \\ g(t_k) \equiv (g_1(t_k), g_2(t_k), \cdots, g_N(t_k))' \\ r(t_k) \equiv (r_1(t_k), r_2(t_k), \cdots, r_N(t_k))'. \end{cases}$$

For the *i-th* TB, $T_i(t_k)$ and $\rho_i(t_k)$ represent its capacity and state, $u_i(t_k)$ is the token generation rate. $V_i(t_k)$ denotes the incoming traffic injected to the *i-th* TB. $g_i(t_k)$ and $r_i(t_k)$ denote the conforming traffic and the nonconforming traffic, respectively.

Since the sources are independent, the dynamics of the system of TBs is now defined by a system of N identical equations,

$$\rho_i(t_{k+1}) \;=\; \rho_i(t_k) + \{u_i(t_k) \wedge [T_i - \rho_i(t_k)]\} - g_i(t_k), \tag{3.5}$$

where $g_i(t_k)$ is the conforming traffic given by

$$g_i(t_k) = V_i(t_k) I\{V_i(t_k) \leq [\rho_i(t_k) + u_i(t_k) \wedge [T_i - \rho_i(t_k)]]\},$$

and $i = 1, 2, \cdots, N$.

Hence, the traffic losses at the i-th TB, or the nonconforming traffic of the i-th TB, can be expressed as

$$r_i(t_k) = V_i(t_k) - g_i(t_k).$$

The conforming traffic is accumulated in the multiplexor. Due to the limitation of queue buffer and the outgoing speed of the access node, not all the conforming traffic can be accepted by the multiplexor. During the k-th time interval, the traffic accepted by the multiplexor can be written as

$$\sum_{i=1}^{N} g_i(t_k) \wedge [Q - ([q(t_k) - C * \tau] \vee 0)]. \tag{3.6}$$

where the first term is the volume of all the conforming traffic provided by the TBs during time interval t_k, and the second term denotes the available buffer space after part of traffic has already been delivered into the outgoing link during time interval t_k. Hence, the smaller of the two parts represents the fresh traffic that will be accepted during the next time interval.

The state of the multiplexor, denoted by $q(t_k)$, is given by the volume of traffic waiting in the buffer for service during the k-th time interval as follows:

$$q(t_{k+1}) = [q(t_k) - C * \tau] \vee 0 + \sum_{i=1}^{N} g_i(t_k) \wedge [Q - ([q(t_k) - C * \tau] \vee 0)]. \tag{3.7}$$

The first term on the right side represents packets leftover from the previous (k-1)-th time interval, the second term represents the packets accepted by the multiplexor during the k-th time interval. This is a scalar equation governing the dynamics of the queue in the multiplexor.

Clearly, part of conforming traffic may be dropped at the multiplexor. Therefore, the traffic loss at the multiplexor is described as

$$L_M(t_k) \equiv \sum_{i=1}^{N} g_i(t_k) - \sum_{i=1}^{N} g_i(t_k) \wedge [Q - ([q(t_k) - C * \tau] \vee 0)]. \tag{3.8}$$

In summary, the dynamics of the whole access control protocol is governed by the system of $(N+1)$ difference equations (3.5) and (3.7). This leads to the following nonlinear state space model,

$$X(t_{k+1}) \equiv F(t_k, X(t_k), u(t_k), V(t_k)), \tag{3.9}$$

where $X \equiv (\rho, q)'$ denotes the state vector, u is the control vector, V represents the input traffic vector and F is the state transition operator determined by the expressions on the right hand side of equations (3.5) and (3.7).

3.4 Objective Function

An objective function is defined to specify which system feature should be minimized during the optimization. Obviously, traffic losses are the major concern in the evaluation of the system cost. In general, the traffic loss at the TBs during the k-th time interval is given by

$$L_T(t_k) \equiv \sum_{i=1}^{N} r_i(t_k) \equiv \sum_{i=1}^{N} [V_i(t_k) - g_i(t_k)],$$

while the loss at the multiplexor $L_M(t_k)$ during the same time interval is given by equation (3.8).

In addition to these losses, it is also important to include a penalty for the waiting time or time spent in the queue before served. For simplicity, the cost is assumed to be linearly proportional to the queue length. It is given by:

$$L_W(t_k) \equiv q(t_k).$$

Thus, the cost functional can be obtained by adding all these losses (cost) together. Since the incoming source (or user demand) is a random process, the average cost has to be computed as an expected value of the sum of all the costs. This is given by,

$$J(u) \equiv E\left\{\sum_{k=0}^{K}\alpha(t_k)L_M(t_k) + \sum_{k=0}^{K}\beta(t_k)L_T(t_k) + \sum_{k=0}^{K}\gamma(t_k)L_W(t_k)\right\}, \tag{3.10}$$

where u is the control law that determines the state of the system and hence the individual losses and finally the total cost. The functions α, β, γ are the relative weights or importance given to each of the three distinct losses, where the first one gives the average weighted loss at the multiplexor, the second for the loss at the TBs, and the last one is the penalty assigned to the average waiting time in the multiplexor (queue). The value of the functions chosen reflects the different concerns of network controllers.

Since the exact stochastic characterization of the traffic is not available or unknown, Monte Carlo method is employed to compute the expected values of the performance measures (explained in Section 2.6). Let N_s denote the number of samples used and let $\Omega \equiv \{w_j, j = 1.2.3.\cdots, N_s\}$ denote the elementary events or sample paths with finite cardinality N_s, the objective functional (3.9) is then given by:

$$J(u) \cong \frac{1}{N_s}\sum_{j=1}^{N_s}\left\{\sum_{k=0}^{K}\alpha(t_k)L_M(t_k,\omega_j) + \sum_{k=0}^{K}\beta(t_k)L_T(t_k,\omega_j) + \sum_{k=0}^{K}\gamma(t_k)q(t_k,\omega_j)\right\}. \tag{3.11}$$

3.5 Utilization

In addition to the objective functional, the network utilization is also used to evaluate the overall system performance. The total amount of data successfully transferred to the network is marked as the actual transferred traffic, while the maximum transfer capacity

is given by the link capacity. Utilization is then defined as ratio of the two, measured in percentage by:

$$\eta = \frac{\sum_{k=0}^{K}\sum_{i=1}^{N} V_i(t_k) - \left[\sum_{k=0}^{K} L_M(t_k) + \sum_{k=0}^{K} L_T(t_k)\right]}{C(t_K - t_0)} \times 100\%. \tag{3.12}$$

Chapter 4

Control Strategy- Feedback Control

Unlike open loop control which does not consider the status of network resources shared by competing users, the feedback control mechanism (closed loop control) exercises control based on available information at the current system state, and it can adapt to any change of network resources [16,19]. In Section 1 and 2, two scenarios of feedback control laws are studied to evaluate system performance of the dynamic model presented in Chapter 3. One scenario is the feedback control law without feedback delay, and the other one is the feedback control law with feedback delay. The results in our simulation have shown that the feedback control with time delay highly degrades system performance [10, 35]. In order to improve the degraded performance and reduce the impact of feedback delay, a predictive feedback control law is proposed in Section 3 corresponding to the scenario in the presence of delay.

4.1 Feedback Control in the Absence of Time Delay

For the system model presented in Chapter 3, a general feedback control law without communication (plus service) delay is of the form:

$$u(t_k) \equiv G\Big(V(t_k), \rho(t_k), q(t_k)\Big), k = 0, 1, 2, \cdots, K-1, \tag{4.1}$$

where G is a suitable function (to be determined) that maps the available data into a control action. The data may be given by the collection $\{V \in R^N, \rho \in R^N, q \in R^1\}$, which denotes the input traffic, the states of the TBs and the multiplexor, respectively.

In order to avoid cell losses at the multiplexor and monopoly by any users, a permission variable is included to compute the maximum permissible allocation for each user. It is given by:

$$\Theta_i(t_k) \equiv \left\{ \frac{V_i(t_k)}{\sum_{i=1}^{N} V_i(t_k)} \wedge \frac{e_i}{N} \right\}. \tag{4.2}$$

where $e_i \equiv 1, 2, \cdots, N$.

Clearly, each user's share is determined by the smaller of two fractions: the actual demand and the weight of permission assigned by the controller. If $1 \leq e_i < r$ and the actual demand is larger than $\frac{r}{N}$, the network provider withdraws the i-th user's full permission and allows other users to occupy the supplementary bandwidth. If $e_i = N$, the actual demand will be fulfilled. Thus by choosing an appropriate value of e_i, the network provider can control monopoly and even assign priorities.

Thus, the actual allocation for $i - th$ user is as follows:

$$A_i(t_k) = \Big\{ \Theta_i(t_k) * \Big[Q - \big[(q(t_k) - C * \tau) \vee 0 \big] \Big] \Big\} \wedge V_i(t_k). \tag{4.3}$$

Including the true allocation for each user, a simple feedback control law was suggested by Qun Wang et al. [8-10] as follows:

$$u_i(t_k) = G_i(V(t_k), \rho(t_k), q(t_k)) \equiv \{A_i(t_k) - \rho_i(t_k)\} I\{A_i(t_k) \geq \rho_i(t_k), A_i(t_k) = V_i(t_k)\}, \tag{4.4}$$

where A_i is given by the expressions (4.2-4.3).

4.2 Feedback Control in the Presence of Time Delay

In the presence of communication and service delay, if the same control law G is used, the actual control action would be different and given by the following expression, which is nothing but the delayed version u^d of the control u without delay (equation 4.1):

$$u^d(t_k) \equiv G\Big(V(t_{k-m_1}), \rho(t_{k-m_2}), q(t_{k-m_3})\Big), k = 0, 1, 2, \cdots, K-1, \tag{4.5}$$

where $m_i, i = 1, 2, 3$, denotes the number of time slots by which information reaching the controller is delayed. Note that, if the delay is less than one time slot ($0 \leq m_i < 1$, $i = 1, 2, 3$), the actual control action is simply considered as no delay. It is clear from the above expression (equation 4.5) that the current control is decided on the basis of past status information and therefore can not be expected to be as effective as the control without delay. In the presence of feedback delay, it is difficult to make an appropriate decision to effectively allocate network resources or prevent traffic congestion on the basis of delayed information, which results in the degradation of system performance [8-10].

4.3 Predictive Feedback Control

To reduce the impact of delay, traffic prediction, i.e. accurately providing the statistical characteristics of traffic, becomes one of the key issues in network control engineering.

Recently a number of traffic predictors have been proposed. However, some of those predictors either performed offline control [14] or focused only a particular type of traffic model [13,15-17]. The algorithm proposed in this thesis is called Predictive Feedback Control (PFC), which uses the principle of the LMSE technique. The experiments carried out by Ghaderi,Capka and Boutaba have demonstrated that the LMSE predictor can achieve the same accuracy as those fractional predictors [13]. Furthermore, the LMSE predictor is much simpler to implement and does not require excessive computation. In addition, this adaptive predictor is independent of the traffic model (statistics) and is useful for both short range dependent processes and long range dependent processes. Those characters make the LMSE method possible to be implemented online and more robustly.

4.3.1 Prediction Scheme-LMSE

The LMSE algorithm, one of most common used adaptive algorithms, was first introduced by Widrow and Hoff in 1959. Its practice is simple and has a reasonable performance.

Let $V(t_k), k = 0, 1, 2, \cdots, K$ denote the history of the traffic process measured in terms of bytes. The way to predict the incoming arrival is to use one segment of past history records which are a number of time units ahead of the current time. Thus, the predicted value can be expressed as a function of the given history records as follows:

$$\widehat{V}(t_k) = \sum_{r=T_d}^{W_s+T_d} \alpha_r V(t_{k-r}), \quad T_d \geq 0, \quad W_s \geq 0, \quad T_d + W_s < k, \tag{4.6}$$

where T_d denotes the number of time units ahead of the current time, and W_s denotes the length of the segment of past history, which is also called observation window size. In other words, T_d is the number of time units by which the samples are delayed and W_s is the number of past samples used to predict the future traffic. Here the vector α denotes

the weight or importance given to past samples observed.

The major objective of traffic prediction is to minimize the mean square difference (error) between the predicted traffic and the actual traffic measured. The choice of the weight vector α determines the level of prediction error. In order to determine the best weight vector, the estimation error is defined as follows:

$$J(\alpha) \equiv E\|V(t_k) - \widehat{V}(t_k)\|^2. \tag{4.7}$$

Then

$$\begin{aligned} J(\alpha) &= E\|V(t_k) - \sum_{r=T_d}^{T_d+W_s} \alpha_r V(t_{k-r})\|^2 \\ &= E\|V(t_k)\|^2 - 2\sum_{r=T_d}^{T_d+W_s} \alpha_r E(V(t_{k-r}), V(t_k)) \\ &\quad + \sum_{l=T_d}^{T_d+W_s}\sum_{r=T_d}^{T_d+W_s} \alpha_r\alpha_l E(V(t_{k-r}), V(t_{k-l})), \end{aligned} \tag{4.8}$$

where

$$(x, y) = \sum_i x_i y_i$$

denotes the standard inner product in R^N. If X, Y are two N-dimensional random vectors having finite second moments, then $E(X, Y)$ denotes the expected value of their inner product. Since there are no constraints on $\alpha \in R^d, (d = W_s)$, differentiating J with respect to α and setting it equal to zero, it can be obtained

$$\sum_{l=T_d}^{T_d+W_s} \alpha_l E(V(t_{k-l}, V(t_{k-r}))) = E(V(t_{k-r}), V(t_k)). \tag{4.9}$$

Let A and b denote the matrix and vector as defined below,

$$A \equiv \begin{pmatrix} E(V(t_{k-T_d}), V(t_{k-T_d})) & \cdots & E(V(t_{k-T_d}), V(t_{k-(T_d+l)})) & \cdots & E(V(t_{k-T_d}), V(t_{k-(T_d+W_s)})) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ E(V(t_{k-(T_d+l)}), V(t_{k-T_d})) & \cdots & E(V(t_{k-(T_d+l)}), V(t_{k-(T_d+l)})) & \cdots & E(V(t_{k-(T_d+l)}), V(t_{k-(T_d+W_s)})) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ E(V(t_{k-(T_d+W_s)}), V(t_{k-T_d})) & \cdots & E(V(t_{k-(T_d+d)}), V(t_{k-(T_d+l)})) & \cdots & E(V(t_{k-(T_d+d)}), V(t_{k-(T_d+W_s)})) \end{pmatrix}$$

$$b \equiv \begin{pmatrix} E(V(t_{k-T_d}), V(t_k)) \\ \vdots \\ E(V(t_{k-(T_d+l)}), V(t_k)) \\ \vdots \\ E(V(t_{k-(T_d+W_s)}), V(t_k)) \end{pmatrix}.$$

Then equation (4.8) can be written compactly as $A\alpha = b$, and if A is nonsingular, the solution is given by $\alpha = A^{-1}b$. Hence, the estimated traffic can be computed by the following expression,

$$\widehat{V}(t_k) = \sum_{r=T_d}^{T_d+W_s} (A^{-1}b)_r V(t_{k-r}), T_d \geq 0, W_s \geq 0, T_d + W_s < k, \tag{4.10}$$

where $(A^{-1}b)_r$ denotes the r-th component of the vector $A^{-1}b$.

4.3.2 Predictive Feedback Control

In order to minimize performance degradation caused by feedback delay, we can apply an appropriate control based on the estimation of future traffic instead of the delayed control.

Since the system states (ρ, q) are monitored and analyzed in real time, it is assumed that the delays for the system states are less than one time unit, and are not taken into consideration. However, the information of the incoming traffic will take time to be transmitted and processed. As we know, the delayed information will lead to inappropriate

control, which causes excessive packet losses and wastes network resources. In order to take the place of the delayed information, the predicted information is incorporated to the feedback control law for the incoming traffic, which is given by,

$$\widehat{u}(t_k) \equiv G(\rho(t_k), q(t_k), \widehat{V}(t_k)), \quad 0 < T_d < k-1, \; W_s \geq 0, \; T_d + W_s < k. \tag{4.11}$$

Chapter 5

Basic Data Used for Simulation Experiments

The implementation of the system approach is introduced in this chapter, as well as the assumptions of the system. The system parameters and configurations are interpreted in Section 1, which help to construct a mathematical system model for analysis. Then, two types of traffic model are presented in Section 2, which will be applied as the input traffic sources for the system model.

5.1 System Parameters and Configurations

For simplicity, the first assumption throughout the simulation is that only three traffic sources will be injected into the system. That is, three individual traffic traces, each of 4-second duration, are fed into three TBs and then the conforming traffic from TBs will be directed to a multiplexor. In addition, four more assumptions will be followed during the implementation:

- Three independent traffic traces share the same statistical characteristics and last

4 seconds;

- The packet size is measured in terms of bytes and one token is consumed for each byte that successfully passes to the networks;
- Three TBs are identical and have the same parameters;
- Feedback delay only occurs in the process of traffic information.

While calculating the objective function and considering the trade-off among different losses, it is necessary to assign different weights to different types of losses, such as losses at the TBs, losses at the multiplexor and losses associated with the waiting time in queue. If a heavier weight is assigned to one particular type of loss, then, this type of loss will be translated into a lower loss probability. It is possible for some traffic (packets) to be dropped at the TBs or the multiplexor due to the limitation of system resource. In this particular case, it is preferable to drop packets at the TBs rather than discarding them at the multiplexor. The reason for this is that if the packets have already been admitted to pass the Token Bucket, the system should successfully transmit them to the network instead of being dropped at the multiplexor. Otherwise, dropping packets at the multiplexor will unnecessarily waste the network resources by first accepting packets into the system and then rejecting them inside the system. Hence, concerning all the issues, the relevant weights have been assigned to the different types of losses as follows:

- $\alpha(t_k) = 10$, the weight assigned to losses at the multiplexor;
- $\beta(t_k) = 5$, the weight assigned to losses at TBs;
- $\gamma(t_k) = 0.3$, the weight assigned to the waiting losses in the queue.

The initial system states (TBs and multiplexor) are all set to zero as follows:

- $\rho_i(t_0) = 0$; for $i = 1, 2, 3$;

- $q(t_0) = 0$.

We follow the characteristics of a typical network system and assume that all the packets conform to the UDP/IP protocol in an Ethernet LAN. Therefore, in general, the packet size varies from 64 bytes to 1518 bytes. The rest of system parameters are listed in Table 5.1.

Parameters	**Bellcore Traffic**	**DSPP Traffic**
$T_i, i = 1, 2, 3$, *(TB Capacity)*	15180 Bytes	15180 Bytes
C *(Link Capacity)*	8 Mbps	8 Mbps
Q *(Buffer Size)*	45540 Bytes	45540 Bytes
τ *(Time Unit)*	0.005 Sec	0.005 Sec
K *(Number of Time Unit)*	800	800
M *(Sample Paths)*	600	1000
$e_i, i = 1, 2, 3$	3	3

Table 5.1: System Configuration and Parameters

5.2 Specification of Traffic Traces

Two types of traffic are studied throughout the thesis, one is Bellcore Ethernet traffic (BC-pAug89), downloaded from the Internet [33], and the other is a self-similar traffic, using a doubly stochastic Poisson process (DSPP) driven by FBM.

The traffic trace, produced from one or several sources over a long period, will be divided into small time units that do not overlap. Each partition contains the original information in units of bytes, and lasts 4 seconds. The data series will represent the volume of traffic

arriving at the TBs during each time unit.

5.2.1 Bellcore Traffic Trace

Bellcore traffic trace [33] was captured on an Ethernet segment at the Bellcore Morristown Research and Engineering Facility. This data set carried a major portion of the local traffic mixed with all the traffic between Bellcore and the Internet. The trace was measured at 11:25 on August 29, 1989, and lasted 3142.82 seconds. The trace captured one million Ethernet packets, whose size was between 64 bytes and 1518 bytes. Figure 5.1 shows part of the trace which is divided into 4- second segments.

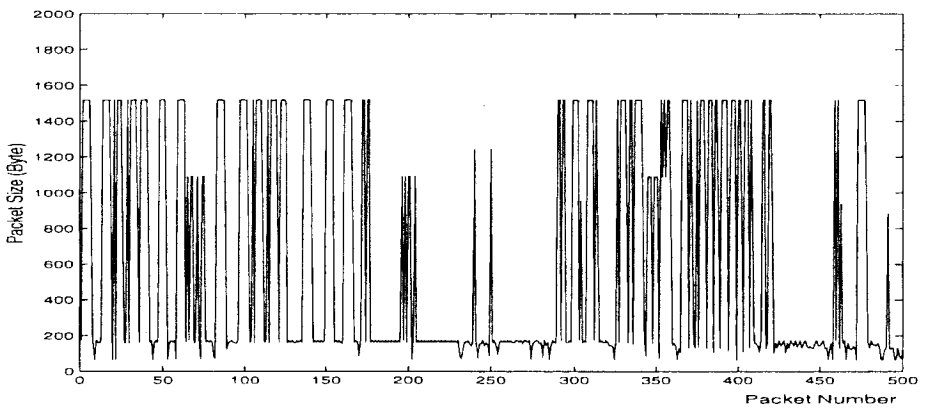


Figure 5.1: Bellcore Traffic Trace

5.2.2 DSPP Traffic Trace

The DSPP traffic trace is produced by driving a non-negative function of FBM, explained in section 3.1, as the intensity rate of a Poisson process. This trace will be used to simulate a self-similar traffic trace.

It has been reported that for the traffic stream generated from a video-conference application, the value of its parameter H usually lies between 0.6 and 0.75 [34]. Hence, for the first DSPP trace, the value of the Hurst parameter is chosen as 0.6. For the second DSPP trace, the first consideration is to take a higher Hurst parameter, so that the trace will appear "burstier" than the first one. Another consideration is to choose a parameter that relates the traffic simulation with an actual traffic trace measured on line. Thus, the Bellcore traffic trace (BC-pAug89) has been widely studied and its value of the parameter H lies in (0.79, 0.85). Based on the above facts, the value of the second parameter H is selected to be 0.8.

The value of C_H is chosen to be 15 throughout the thesis, which can influence the intensity of traffic $\lambda(t)$.

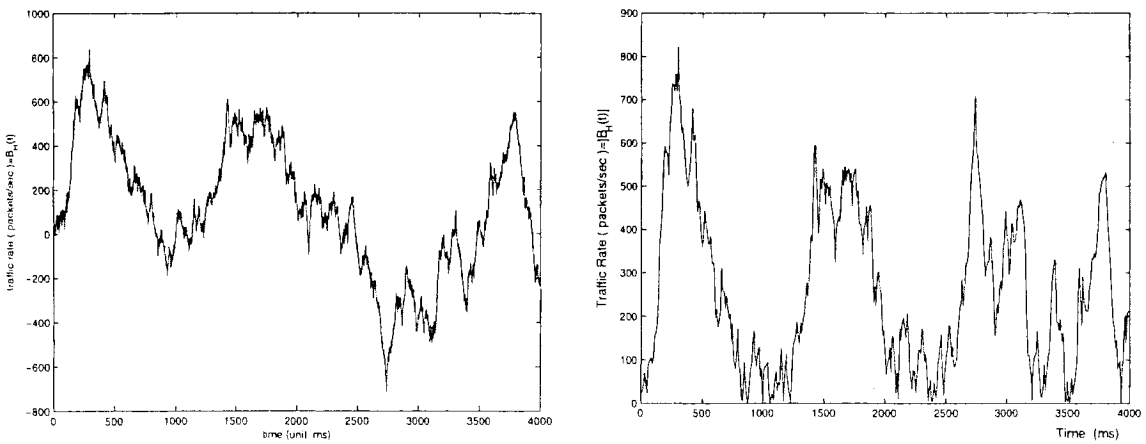


Figure 5.2: Hurst Parameter = 0.6 (a) $B_H(t)$ (b) $\lambda(t)$=$|B_H(t)|$

Figure 5.2 depicts the intensity rate of a DSPP traffic trace with the Hurst parameter $H = 0.6$. Figure 5.2a is the value of FBM, denoted by $B_H(t)$, obtained by the equation (3.1), and Figure 5.2b is the intensity rate $\lambda(t)$, i.e. the absolute value of FBM ($|B_H(t)|$).

Using the data in Figure 5.2(b) as the input rate of a Poisson process, we are able to produce a doubly stochastic Poisson process, which exhibits self-similar property. The sample of the trace, lasting 4 seconds, is plotted in Figure 5.3.

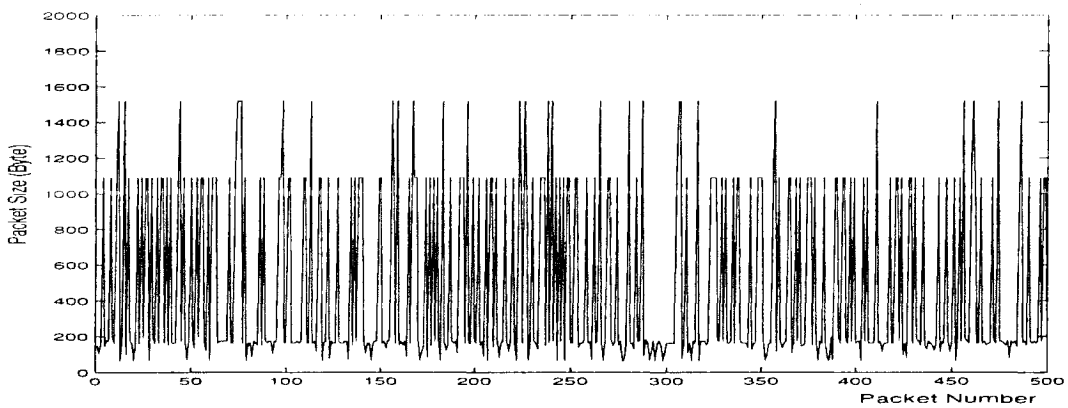


Figure 5.3: DSPP Traffic Trace (H = 0.6)

Figure 5.4 shows another set of the intensity rate with $H = 0.8$. The sample of DSPP traffic trace with $H = 0.8$ is plotted in Figure 5.5

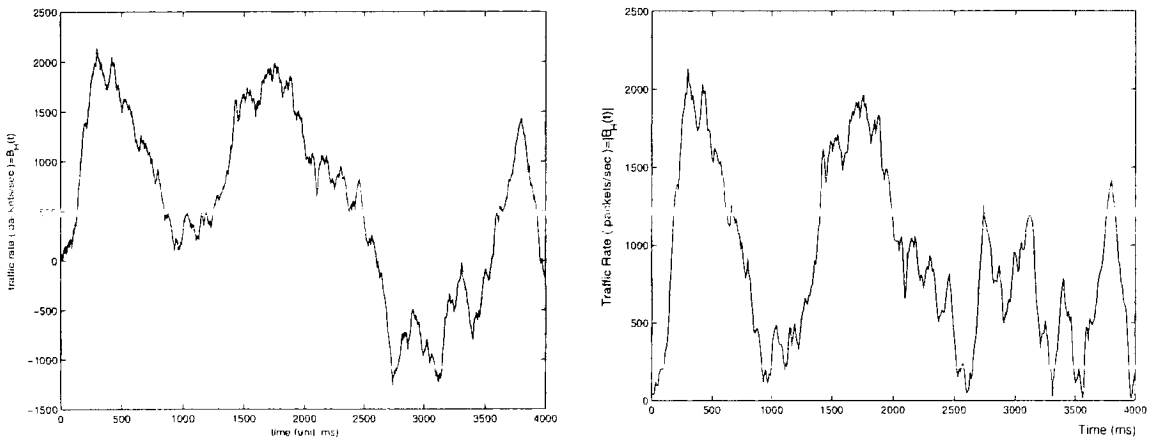


Figure 5.4: Hurst Parameter = 0.8 (a) $B_H(t)$ (b) $\lambda(t)$=$|B_H(t)|$

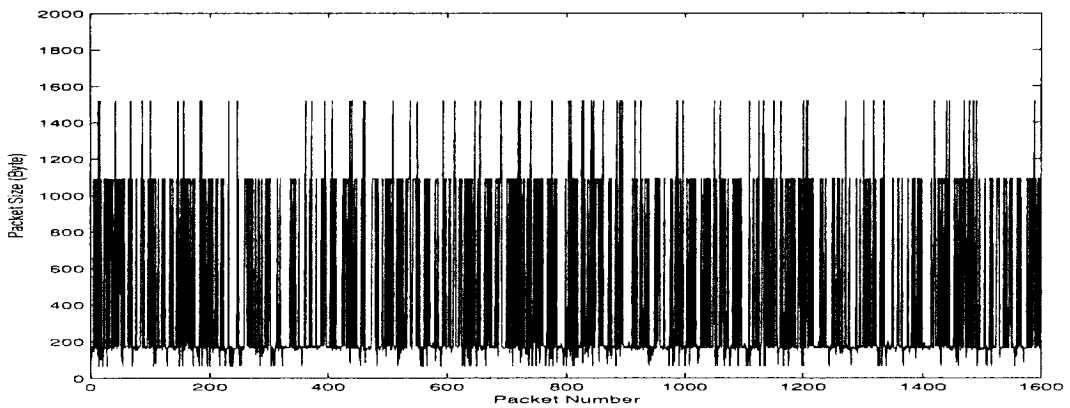


Figure 5.5: DSPP Traffic Trace (H = 0.8)

Chapter 6

Numerical Results and Analysis

In this chapter, numerical results are produced from the experiments corresponding to the Bellcore traffic and the DSPP traffic. The performance of the LMSE predictor is studied with different values of observation window size and prediction time delay. Since prediction time delay varies with the communication delay which causes delayed control actions, the numerical results corresponding to required prediction time delays can be used to evaluate the performance of the system described by the equations (3.5), (3.7) and (4.10). Hence, the LMSE predictor will be incorporated into a feedback control law. Then, based on the different input traffic, it will be shown that how the observation window size and prediction time delay influence the system response and performance .

6.1 Performance of the LMSE Predictor

The performance of the LMSE Predictor is evaluated for different observation window sizes and prediction time delays, subject to the Bellcore traffic and the DSPP traffic. In addition, the numerical results presented here explore the relationship between the Hurst parameter and the prediction errors.

6.1.1 Criterion of LMSE Performance

To illustrate the dependence of estimation error on the observation window size W_s and the prediction time delay T_d, the following equation is used to calculate the expected value of the estimation error:

$$E(T_d, W_s) = \sqrt{(E\|\widehat{V}(t_k) - V(t_k)\|)^2} = \sqrt{\left(\frac{1}{N_s}\sum_{j=1}^{N_s}\left(\widehat{V}(t_k, w_j) - V(t_k, w_j)\right)\right)^2},$$

where w_j denotes the j-th sample path and N_s denotes the number of sample paths used.

The inverse of Signal-to-Noise Ratio (E_{NSR}) is used to evaluate the quality of prediction results. This is written as:

$$E_{NSR} \equiv SNR^{-}1 = \frac{\sum E(T_d, W_s)^2}{\sum(V(t_k))^2} = \frac{\left(\frac{1}{N_s}\sum_{j=1}^{N_s}\left(\widehat{V}(t_k, w_j) - V(t_k, w_j)\right)\right)^2}{\sum(V(t_k))^2}. \quad (6.1)$$

The smaller the E_{NSR}, the more accurate the prediction result.

6.1.2 Prediction Performance with Bellcore Traffic Trace

In this section, the prediction results will be analyzed by feeding the Bellcore traffic with different observation window sizes and prediction time delays.

(A) Comparison of Prediction Performance

Figure 6.1 shows the comparison between the actual traffic and the prediction result corresponding to one set of multi-step predictor with a fixed observation window size $W_s = 3\ \tau$. Figure 6.1A is the real traffic obtained from Bellcore Lab, and Figure 6.1B

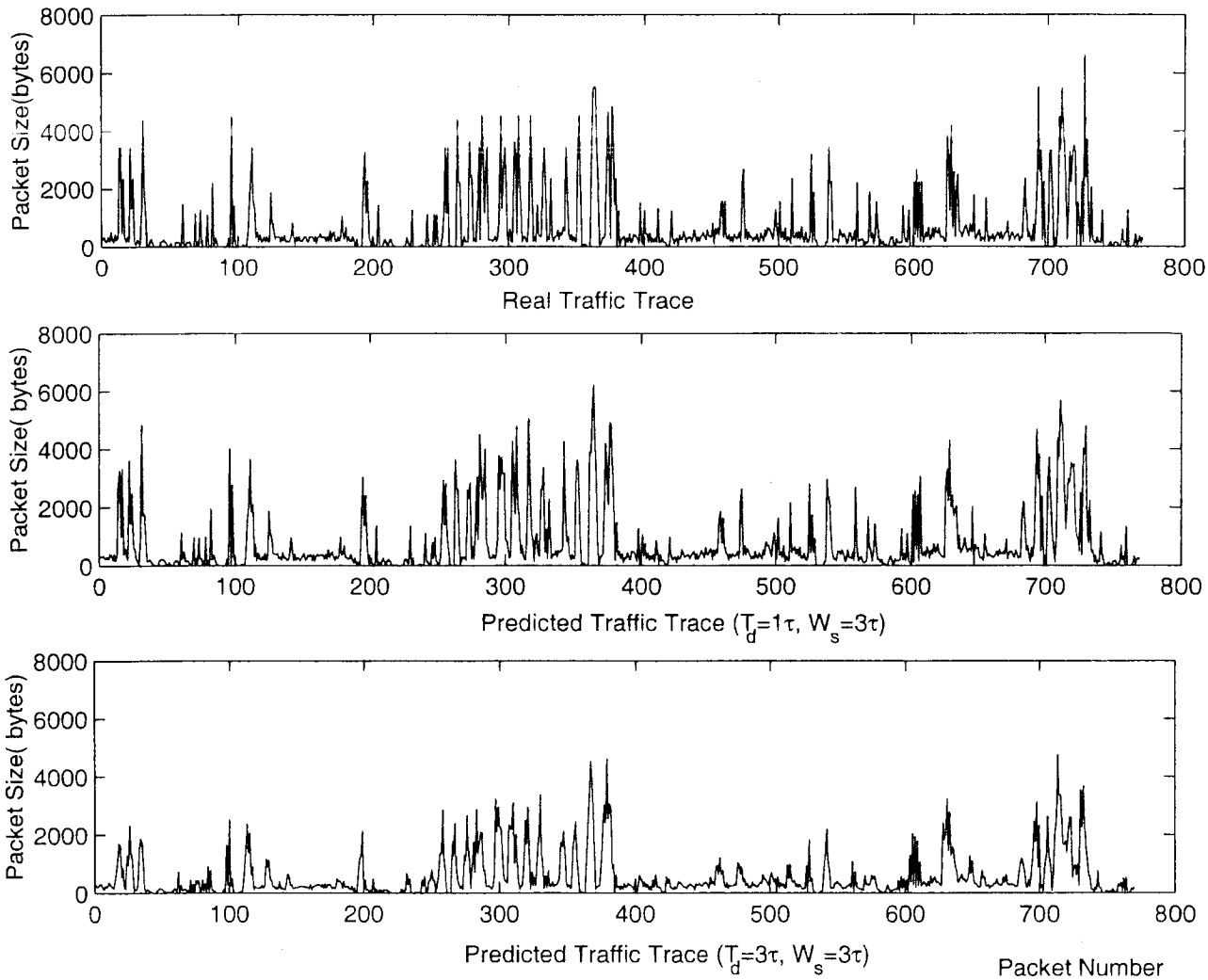


Figure 6.1: Bellcore Traffic: Actual Trace VS Predicted Trace

and 6.1C depict the estimated traces computed by the LMSE method corresponding to different prediction time delays. Clearly shown in Figure 6.1, the predicted traffic trace with the lower prediction time delay $T_d = 1\ \tau$ appears more similar to the original one than that with time delay $T_d = 3\ \tau$. Therefore, when doing a prediction, the further the history information is behind the current state, the less accurate the prediction result will be.

(B) Dependence of E_{NSR} on Observation Window Size

Figure 6.2 shows the plots of E_{NSR} as a function of observation window sizes for fixed prediction time delays as parameters. It is clear that for a fixed prediction time delay, the error decreases with the increase of (observation) window size. This is expected because the more past records are used to predict the future traffic, the more accurate the predicted output will be. On the other hand, it is also clear from this figure that, for a fixed window size, prediction error increases with the increase of prediction time delay.

Furthermore, for a fixed prediction time delay, as the window size increases, the prediction error tends to reach a lower limit, but still greater than zero. This means that by simply increasing the window size we cannot expect to improve the performance beyond a certain limit.

(C) Dependence of E_{NSR} on Prediction Time Delay

In Figure 6.3, E_{NSR} is plotted as a function of prediction time delay for fixed values of observation window size. It is clear from these plots that for any fixed window size, E_{NSR} increases with the increase of prediction time delay required. This is also expected.

Again for a fixed prediction time delay, as the window size increases the prediction error

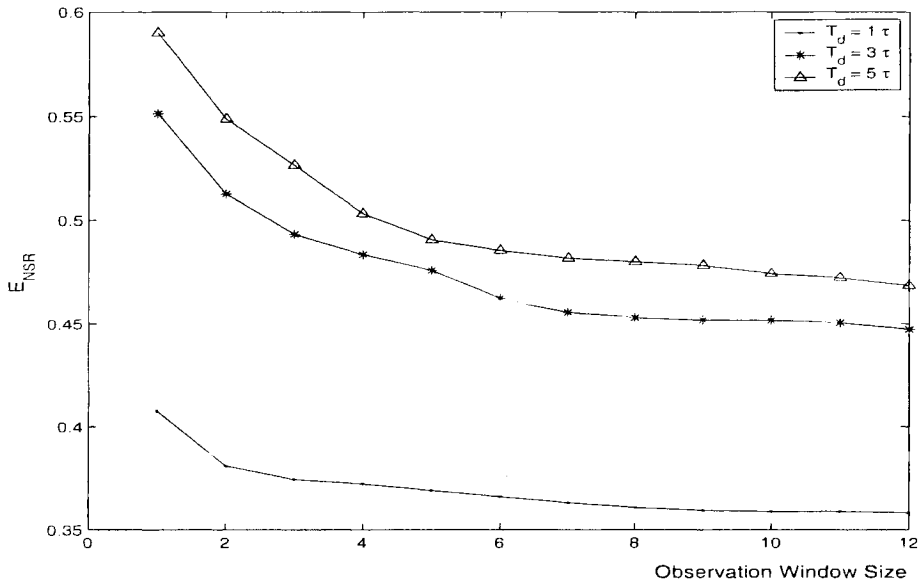


Figure 6.2: Prediction Error vs W_s - Bellcore Traffic

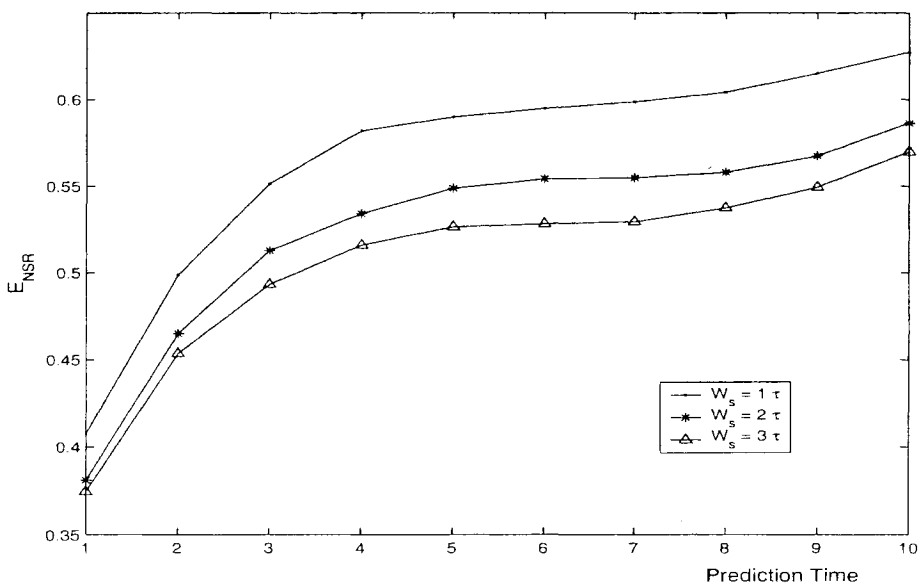


Figure 6.3: Prediction Error vs T_d - 1 (Bellcore Traffic)

decreases. However, the level of decrease will slow down with large window size as illustrated in Figure 6.4. Clearly, there is no large difference existing among those 3 curves with increasing window sizes.

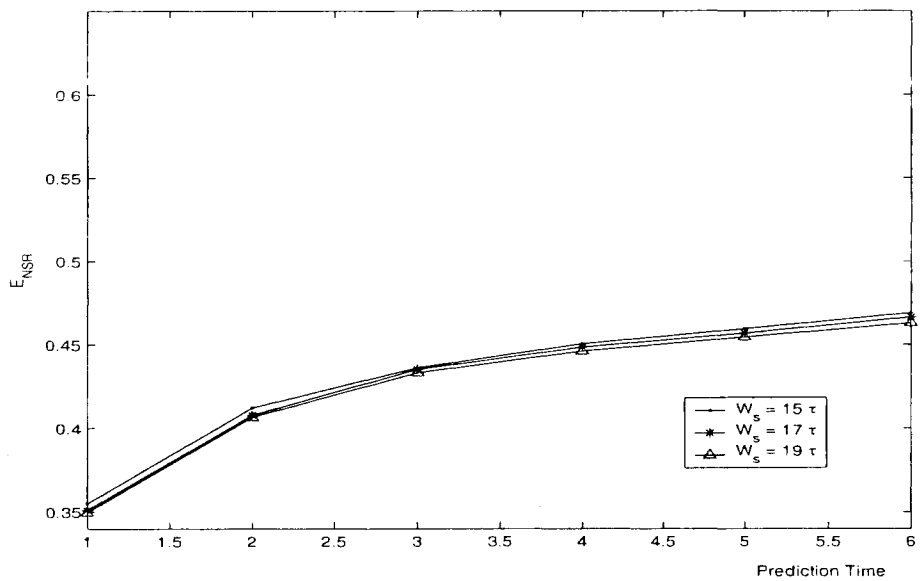


Figure 6.4: Prediction Error vs T_d - 2 (Bellcore Traffic)

6.1.3 Prediction Performance with DSPP Traffic Traces (H = 0.6, 0.8)

In this section, the predictor performance has been evaluated corresponding to DSPP traffic traces with the Hurst parameter equal to 0.6 (DSPP-1) and 0.8 (DSPP-2), respectively. The issue of how the Hurst parameter influences the prediction results has also been discussed by using the increasing observation window sizes and different prediction time delays.

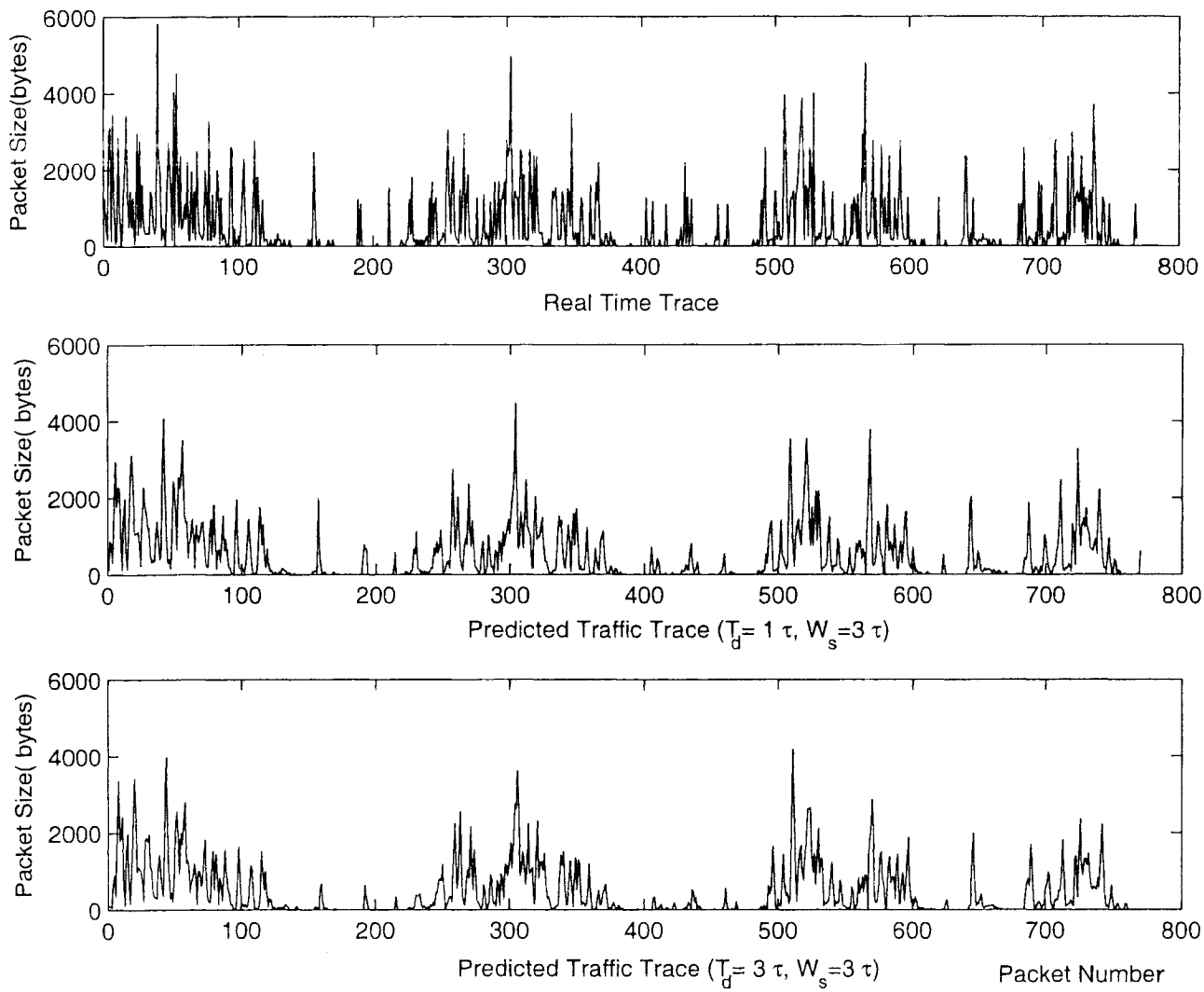


Figure 6.5: DSPP Traffic - 1 ($H = 0.6$): Actual Trace VS Predicted Trace

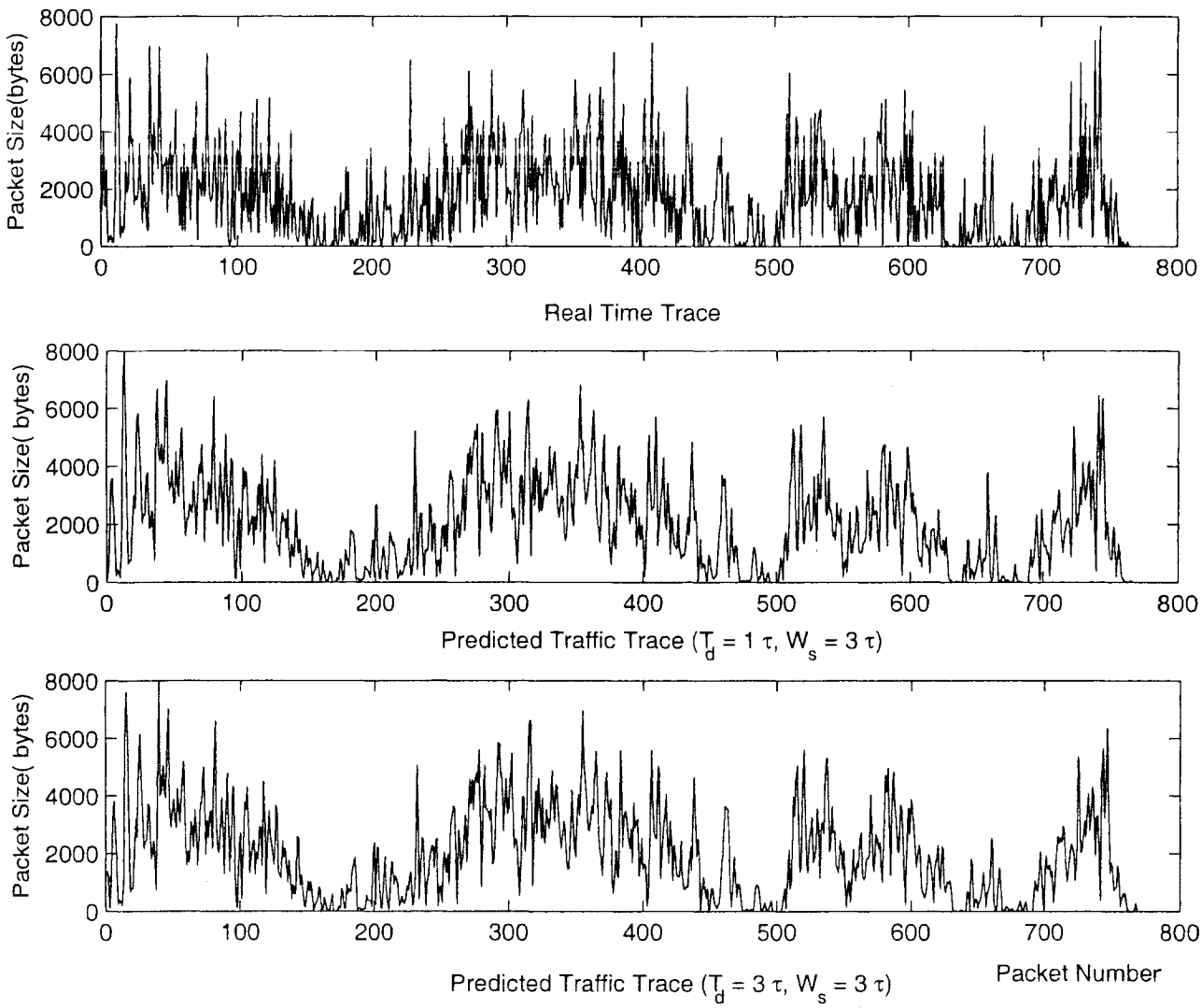


Figure 6.6: DSPP Traffic - 2 ($H = 0.8$): Actual Trace VS Predicted Trace

(A) Comparison of Prediction Performance

Here, the LMSE technique is applied to forecast two sets of the DSPP traces. Figure 6.5 and Figure 6.6 show the numerical results for the traces with the Hurst parameter $H = 0.6$ and $H = 0.8$, respectively. These two figures provide a good view to make comparison between the actual traffic and predicted traffic traces using one step ahead and three steps ahead predictor for two sets of traces. Clearly, the forecasted values in Figure 6.5B, which are obtained by using one step ahead predictor, appear closer to the actual values in Figure 6.5A than those values in Figure 6.5C, which are obtained by applying three steps ahead predictor. This is also true for the trace in Figure 6.6 for the Hurst parameter $H = 0.8$. Therefore, based on a fixed number of history records, it demonstrates that if the records are closer to the current state, the prediction result will be more accurate.

(B) Dependence of E_{NSR} on Observation Window Size

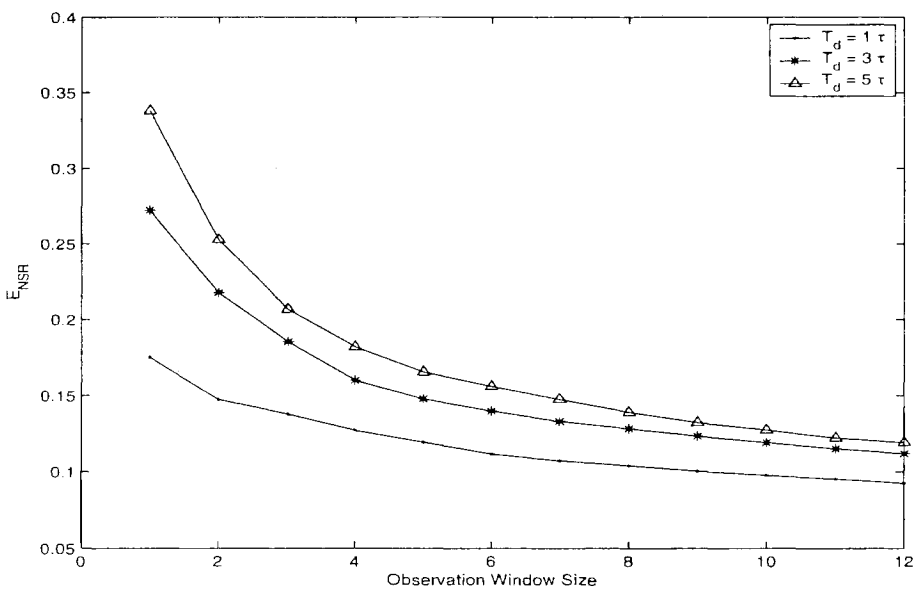


Figure 6.7: Prediction Error vs W_s - DSPP-1 Traffic (for $H = 0.6$)

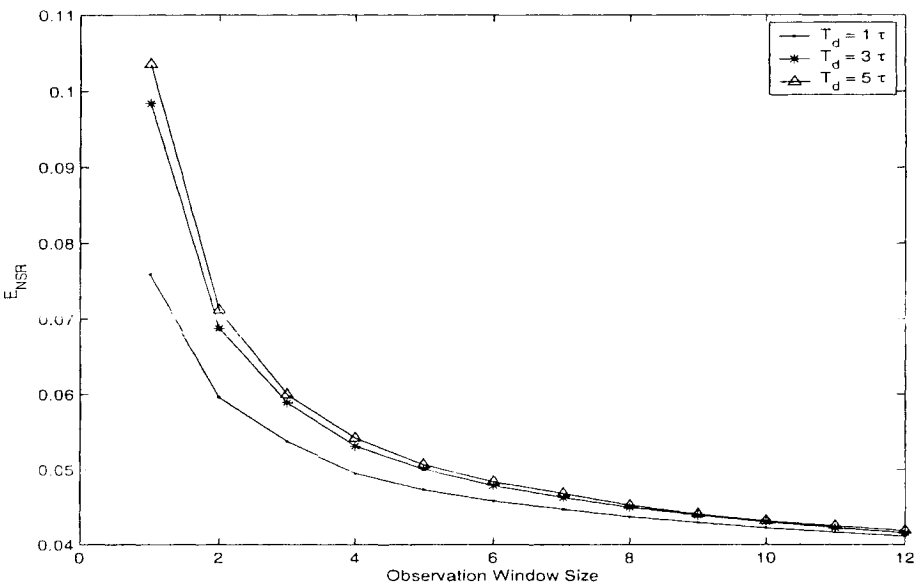


Figure 6.8: Prediction Error vs W_s - DSPP-2 Traffic (for $H = 0.8$)

It is clear from Figure 6.7 and 6.8, plotted for two different Hurst parameters, that the prediction error decreases when the observation window size increases for a fixed prediction time delay. This is similar to what has been observed in the case of the Bellcore traffic (Figure 6.6). The more history records are used for traffic prediction, the more accurate the result will be.

It's also noted that with the increase of the Hurst parameter, E_{NSR} decreases. Figure 6.9 demonstrates the relationship between E_{NSR} and the Hurst parameter with a fixed prediction time delay and three different window sizes. The result illustrates that the long range dependence (LRD) property exists in those traces. The larger the Hurst parameter, the stronger the correlation with past information, which results in reduced prediction error with increasing observation window sizes.

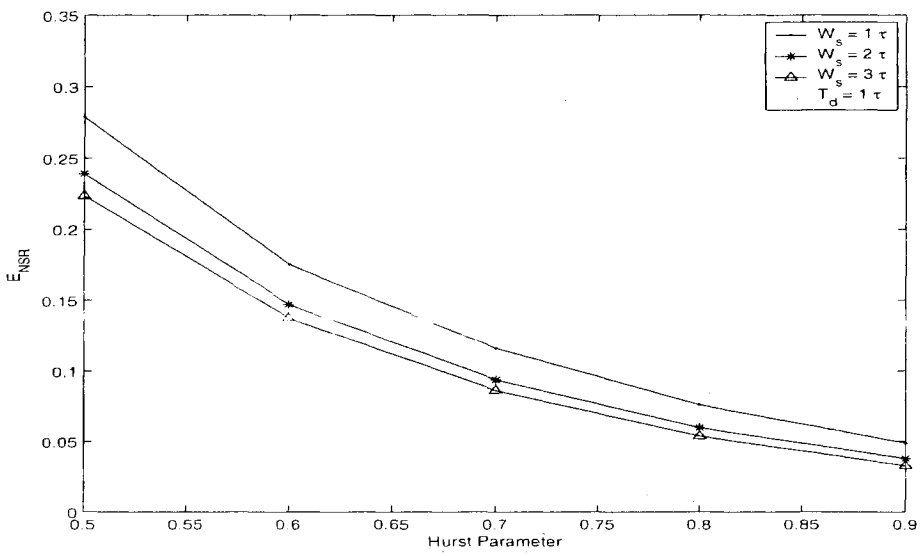


Figure 6.9: Prediction Error vs Hurst Parameter - DSPP Traffic

(C) Dependence of E_{NSR} on Prediction Time Delay

Figure 6.10 and 6.11 offer another insight to analyze how the prediction time delay affects the prediction accuracy. These two figures describe E_{NSR} as a function of prediction time delay based on fixed values of observation window size with the Hurst parameter $H = 0.6$ and $H = 0.8$. Clearly, for any fixed window size, E_{NSR} increases with the increase of prediction time delay and appears to reach a plateau, which appears similar in the previous case for the Bellcore Traffic. It's also noted that E_{NSR} is smaller with larger Hurst parameters, which is further illustrated in Figure 6.12. It demonstrates the relationship between E_{NSR} and Hurst parameter dependence of prediction time delay with fixed observation window size.

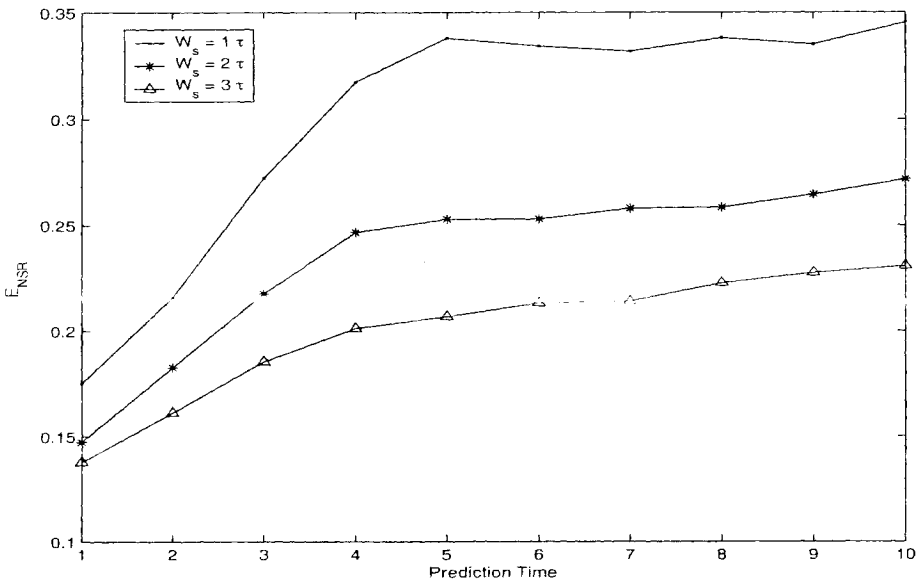


Figure 6.10: Prediction Error vs T_d - DSPP-1 Traffic (for $H = 0.6$)

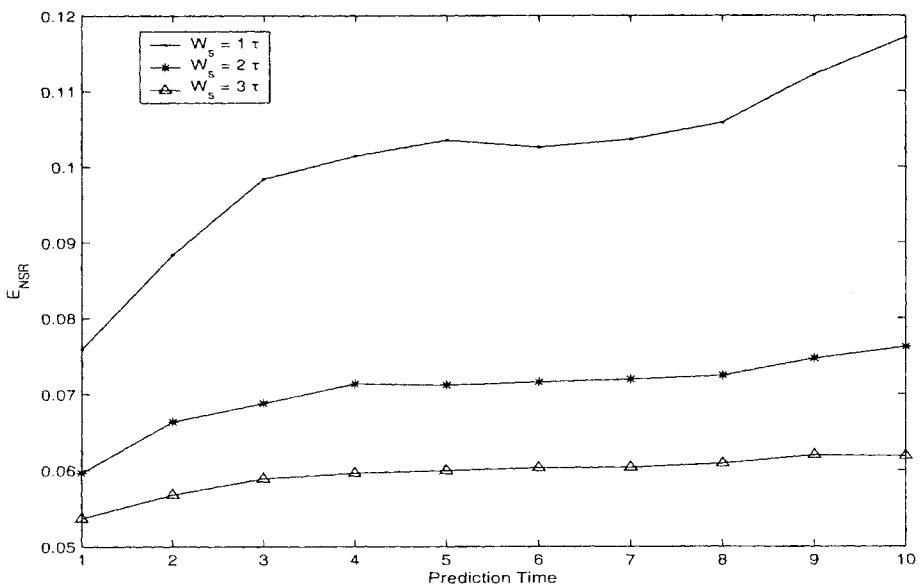


Figure 6.11: Prediction Error vs T_d - DSPP-2 Traffic (for $H = 0.8$)

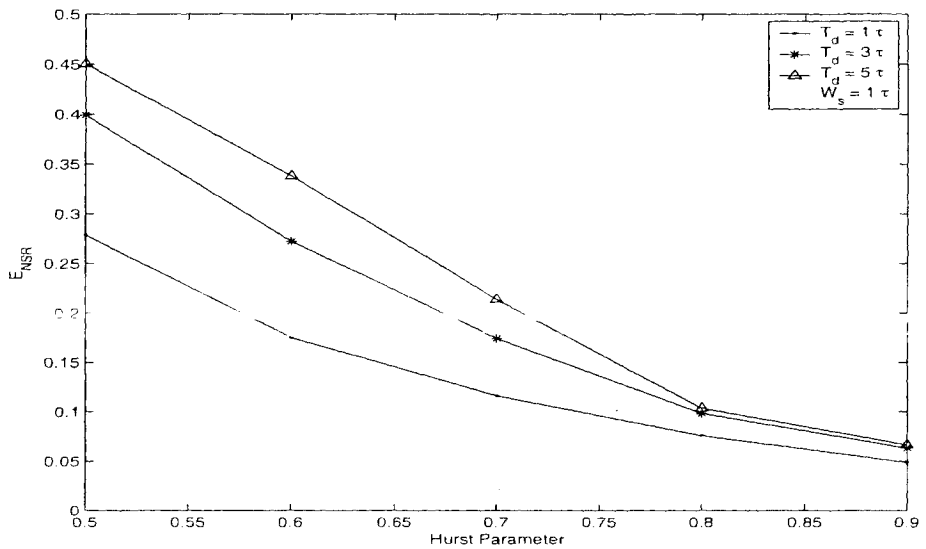


Figure 6.12: Prediction Error vs Hurst Parameter - DSPP Traffic

6.2 System Performance with Predictive Feedback Control

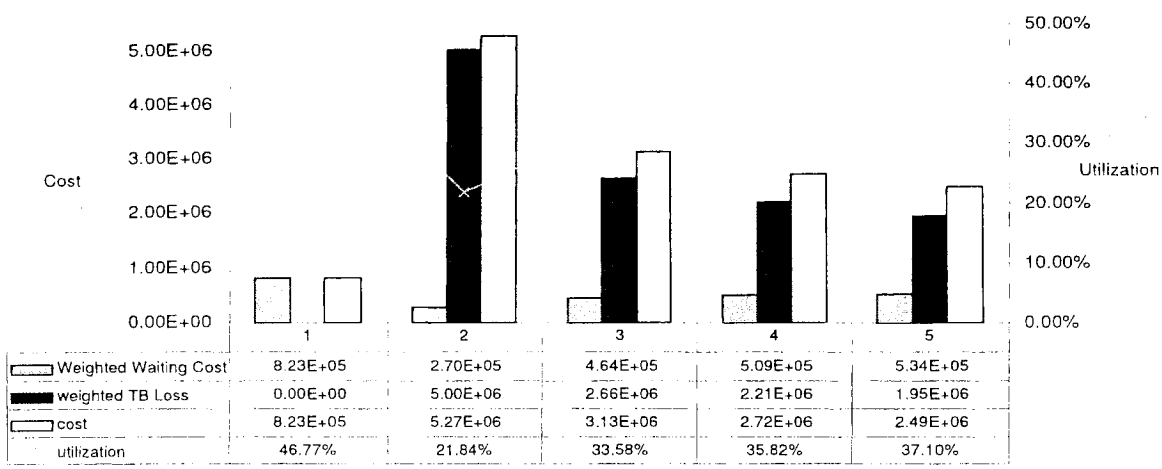
As described in Chapter 1 and Chapter 4, communication delay in the network, resulting in the delay in control actions, adversely affects the system performance [10]. In this thesis, by forecasting the future traffic and state information, the impact of time delay has been compensated. Thus, appropriate control actions, based on the prediction information instead of delayed ones, will be taken to improve network utilization and reduce the packet losses. In this section, simulation results illustrate the improvement of the overall system performance by using predictive feedback control.

6.2.1 Dependence of System Performance on Control Policies

In the experiment, five feedback control cases, listed below, have been considered to evaluate the performance of predictive feedback control laws corresponding to two types of traffic (Bellcore traffic and DSPP traffic). The numerical results, shown in Figure 6.13-6.15, are produced and analyzed corresponding to the following five control laws.

- Case 1: feedback control (without delay, $T_d = 0$)
- Case 2: feedback control (with delay, $T_d = 1$)
- Case 3: feedback control (predictive feedback control, $T_d = 1$, $W_s = 1$)
- Case 4: feedback control (predictive feedback control, $T_d = 1$, $W_s = 6$)
- Case 5: feedback control (predictive feedback control, $T_d = 1$, $W_s = 11$)

(A) Bellcore Traffic Trace



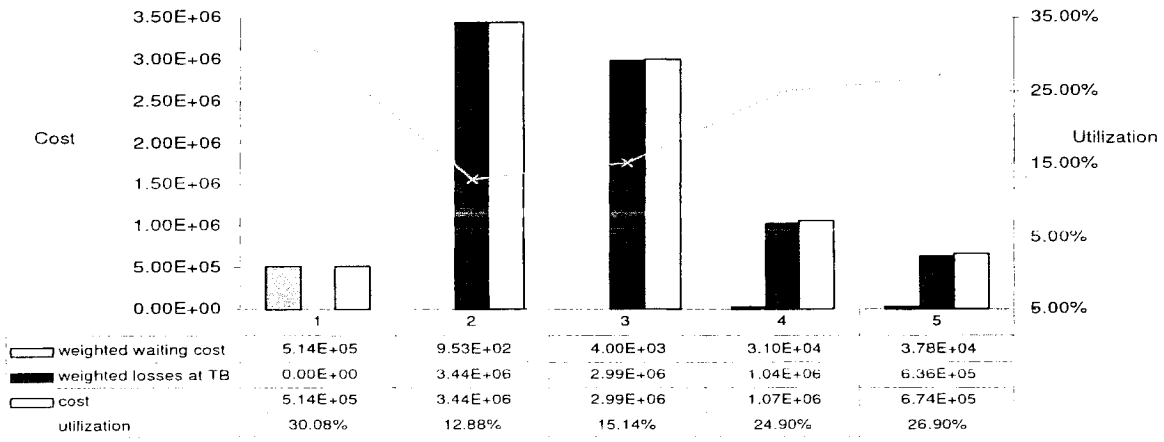
	1	2	3	4	5
Weighted Waiting Cost	8.23E+05	2.70E+05	4.64E+05	5.09E+05	5.34E+05
weighted TB Loss	0.00E+00	5.00E+06	2.66E+06	2.21E+06	1.95E+06
cost	8.23E+05	5.27E+06	3.13E+06	2.72E+06	2.49E+06
utilization	46.77%	21.84%	33.58%	35.82%	37.10%

Figure 6.13: System Performance (Costs and Utilization, cases 1-5)- Bellcore Traffic

In Figure 6.13, the weighted TB losses, the waiting-time losses, the total cost and the utilization are listed for comparison. It is clear from Figure 6.13 that feedback control without (communication) delay (case -1), achieves the minimum cost and the highest utilization. The worst situation occurs in Case-2 with one unit of communication delay. Provided with the delayed (traffic) information, the controller can not supply the required number of tokens to match the incoming traffic. This leads to significant packet losses at the Token Bucket resulting in degradation of performance and utilization. On the other hand, by using the predictive feedback control law (Cases 3-5), it is possible to reduce the performance degradation significantly as shown. In all these cases we use predictive feedback control laws with increasing window sizes. It is clear (case-3) that the usage of this control law substantially improves the performance despite communication delay. This is further improved by using larger window sizes as seen in cases-4-5. System utilization given by the thin curve shows that utilization is highest in the absence of communication delay and lowest in its presence (case-2). It then increases if predictive feedback control is used with increasing window size (cases 3-5).

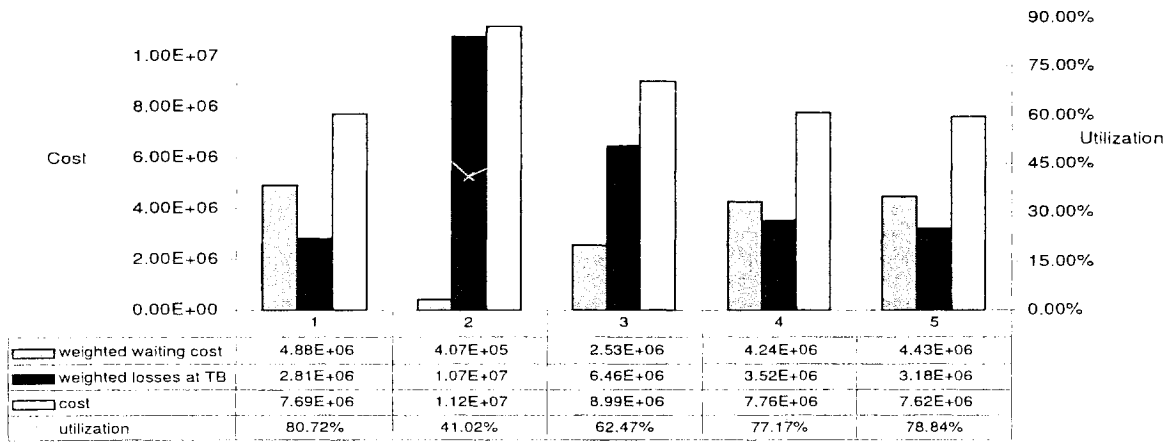
(B) DSPP Traffic Trace

Performance results, shown in Figure 6.14-6.15, are yielded corresponding to DSPP traffic traces with $H = 0.6$ and $H = 0.8$ using the same 5 cases as in Bellcore traffic. The results have a similar general pattern as those of Bellcore traffic shown in Figure 6.13. Among 5 cases, case-1 maintains the lowest system cost and the highest utilization, and the worst case is case-2 without any control optimization. With predictive feedback control law, system performance is getting better with increasing window sizes in cases 3-5. The loss at TB is progressively reduced with the increase of (observation) window size, and the utilization is improved drastically with the growth of (observation) window size. In summary, the predictive feedback control provides a good estimation of future traffic, which is helpful in minimizing the packet losses and improving the utilization.



	1	2	3	4	5
weighted waiting cost	5.14E+05	9.53E+02	4.00E+03	3.10E+04	3.78E+04
weighted losses at TB	0.00E+00	3.44E+06	2.99E+06	1.04E+06	6.36E+05
cost	5.14E+05	3.44E+06	2.99E+06	1.07E+06	6.74E+05
utilization	30.08%	12.88%	15.14%	24.90%	26.90%

Figure 6.14: System Performance (Costs and Utilization, cases 1-5)- DSPP1



	1	2	3	4	5
weighted waiting cost	4.88E+06	4.07E+05	2.53E+06	4.24E+06	4.43E+06
weighted losses at TB	2.81E+06	1.07E+07	6.46E+06	3.52E+06	3.18E+06
cost	7.69E+06	1.12E+07	8.99E+06	7.76E+06	7.62E+06
utilization	80.72%	41.02%	62.47%	77.17%	78.84%

Figure 6.15: System Performance (Costs and Utilization, cases 1-5)- DSPP2

6.2.2 Dependence of Cost on Observation Window Size

The cost function is a measure of overall system performance, which is plotted as a function of window size for 3 different values of communication delay.

(A) Bellcore Traffic Trace

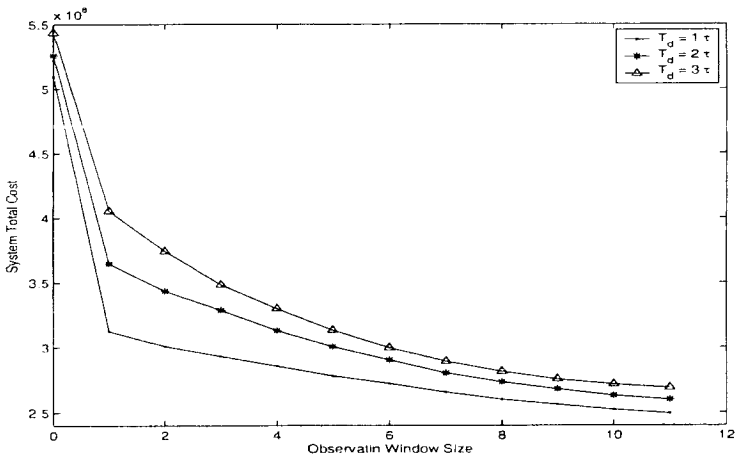


Figure 6.16: System Cost vs W_s (for $T_d = 1, 2, 3$)- Bellcore traffic

Figure 6.16 shows the plots of total cost as a function of the observation window size (W_s) with the fixed feedback delay (T_d). It is clear from this figure that system cost decreases with increasing (observation) window size for any given communication delay and it increases with increasing delay for any fixed window size.

(B) DSPP Traffic Trace

Again in Figure 6.17 (a,b), the system cost is plotted as a function of the window size for two different values of Hurst parameters. It is clear that these results show a similar pattern as those of Bellcore traffic shown in Figure 6.16. Comparing Figure 6.17a and

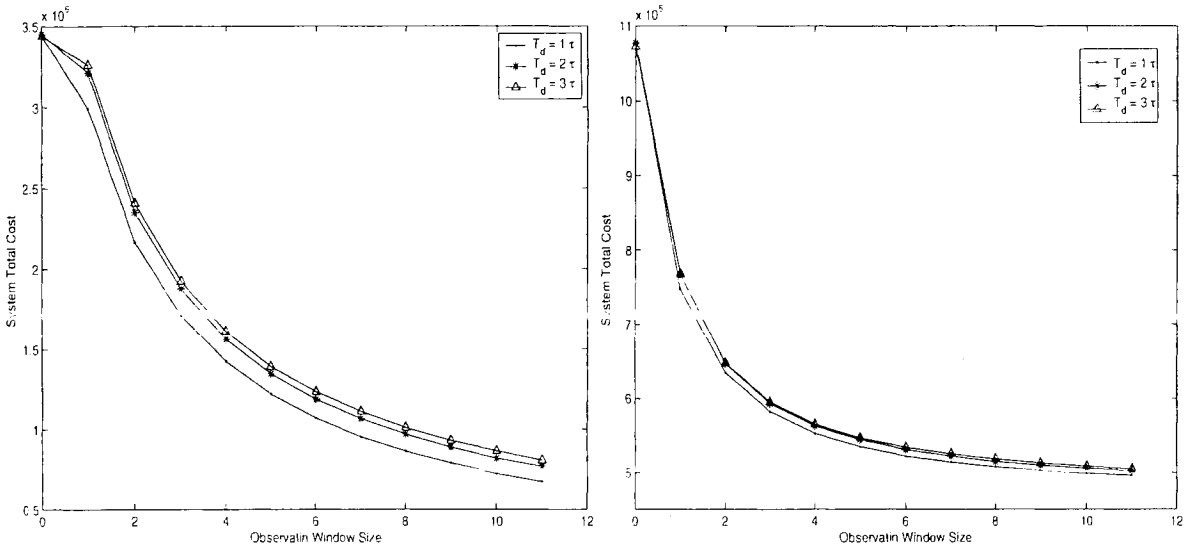


Figure 6.17: System Cost vs W_s - DSPP Traffic (for $T_d = 1, 2, 3$) (a) H=0.6 (b) H=0.8

6.17b, it is observed that the cost reduction, with the increase of observation window size, is again more pronounced for larger Hurst parameters. This is due to the fact that the process with larger Hurst parameter has stronger correlation with the past, and hence larger window size contains more useful information for a more accurate prediction of the future traffic.

6.2.3 Dependence of Utilization on Observation Window Size

System utilization depends on the volume of traffic successfully transmitted to the network. The utilization presented here is plotted as a function of observation window size for three different values of communication delay.

(A) Bellcore Traffic Trace

Figure 6.18 gives another aspect of system performance. It is clear from these curves

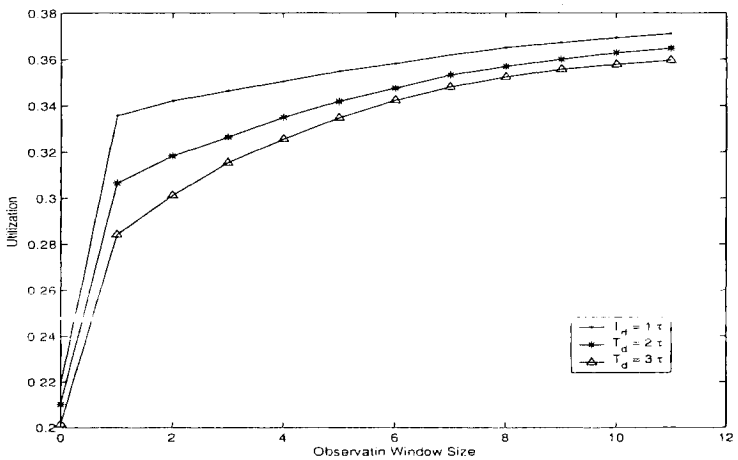


Figure 6.18: Utilization vs W_s–Bellcore Traffic

that utilization increases with increasing window size by feeding the same input traffic. Noted that as the window size increases to a certain level, the growth of utilization slows down. It illustrates that by simply increasing the window size, the packet losses cannot be prevented beyond a certain limit. Again this is due to the same reason as mentioned in section 6.1.2B. In addition, among three different values of prediction time delay, the prediction accuracy is lowest with the largest prediction time delay, which also leads to the lowest utilization.

(B) DSPP Traffic Trace

Dependence of utilization on DSPP traffic traces is plotted in Figure 6.19. Clearly, as the window size increases, utilization increases, which has been observed also in Figure 6.18 for Bellcore traffic. The reason is that, by using predictive feedback control with increasing window sizes, we improve the prediction accuracy and then reduce packet losses at TBs. As a result, the total volume of traffic transmitted to the network increases, and then the utilization increase.

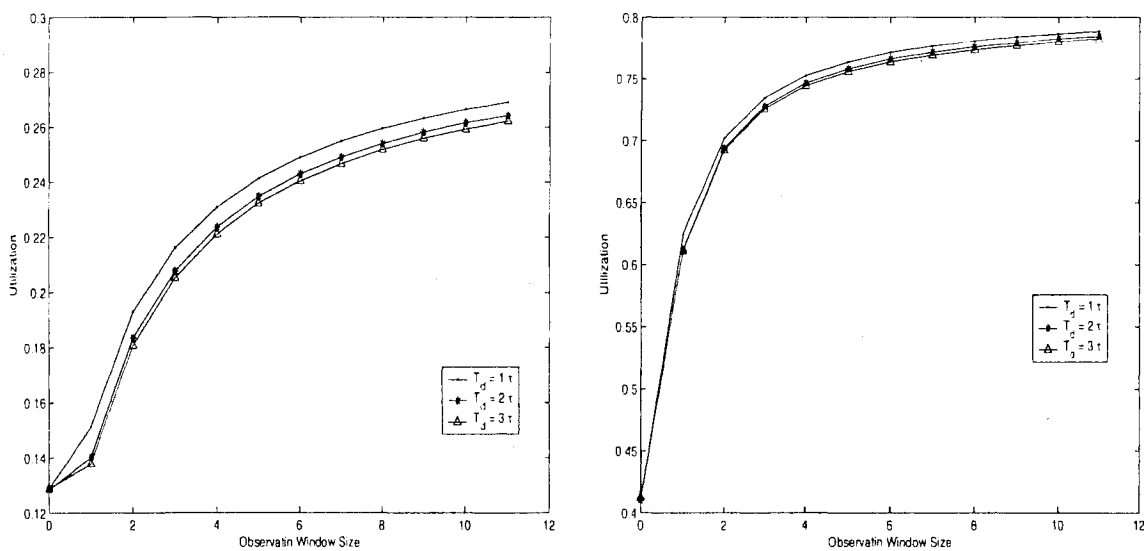


Figure 6.19: Utilization vs W_s -DSPP Traffic (for $T_d = 1, 2, 3$) (a) H=0.6 (b) H=0.8

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have proposed and analyzed a predictive feedback control law, which is applied to reduce the impact of feedback delay. The control law has been incorporated in the system describing the dynamics of the token bucket access control mechanism and the multiplexor.

During the implementation of the predictive feedback control, we have started by an in-depth analysis of the prediction accuracy dependence of prediction time delay and observation window size. From the analysis, we are able to derive some useful guidelines for the settings of these two variables, and we also find that by simply increasing the observation window size for any fixed prediction time delay, the performance will be not beyond a certain limit. Furthermore, we also study the relationship between the Hurst parameter of the traffic and the prediction performance. It has been found that the process with larger Hurst parameter exhibits long range dependence and strongly affects the prediction performance. Finally, with new predictive feedback control law, we analyze the improvement of the utilization and system cost, such as losses at the TBs, losses at

the multiplexor and waiting time in the queue of the multiplexor. From the simulation result, we have been able to get an insight into the dependence of system performance on observation window size, corresponding to different values of Hurst parameters.

In summary, according to the numerical simulation results presented in chapter 6, this LMSE control law effectively improves the overall system performance and prevents network instability. It also leads to a better understanding of the impact of Hurst parameters on network performance. With this method, the impact of feedback delay has been compensated without causing performance degradation.

7.2 Future Work

The further study of the predictive control law could be of interest in the following issues:

1. To improve the system performance and compensate the prediction error, there is a need of studying the buffered version of the token bucket in our system model;

2. Another challenging topic is to test the predictive feedback control law in a real time environment.

Bibliography

[1] M. Butto, E. Caverolla, and A. Tonietti, *Effectiveness of The Leaky Bucket Policing Mechanism in ATM Networks*, IEEE J. Selected Areas in Communications, 9(3):335-342. April 1991.

[2] S.L. Wu and W.S. E. Chen, *The Token-bank Leaky Bucket Mechanism for Group Connections in ATM Networks*, In Proceeding of ICCCN96, Rockville, MD., October 1996.

[3] Natarajan Gautam, *Buffered and Unbuffered Leaky Bucket Policing: Guaranteeing QoS, Design and Admission Control* , Telecommunication Systems 21:1, 2002, pp.35-63.

[4] Cisco on-line http://www.cisco.com/univercd/cc/td/doc/product /software/ios120/12cgcr/qos$_c$/qcpart4/qcpolts.htm.

[5] P. Tang and T.Tai, *Network Traffic Characterization Using Token Bucket Model*, In Proceedings of IEEE Infocom'99, New York, March 1999.

[6] J.Turner, *New directions in communications (or which way to the information age?)*, IEEE communications Magazine, Vol.24, 1986, pp.8-15.

[7] on-line http://qbone.internet2.edu/bb/Bucket.doc

[8] N.U. Ahmed, Q. Wang and L.Orozco Barbosa, *A Systems approach to modeling the Token Bucket Algorithm in computer Networks*, Mathematical Problems in Engineering: Theory, Methods and Applications, 2002, 8(3), pp.265-279.

[9] N.U. Ahmed, B. Li and L.Orozco Barbosa, *Optimization of Computer Network Traffic Controllers using a Dynamic Programming/Genetic Algorithm Approach.* (submitted)

[10] N.U. Ahmed, H. Yan and L.Orozco Barbosa, *Performance Analysis of the Token Bucket Control Mechanism Subject to Stochastic Traffic*, Dynamic of Continuous, Discrete and Impulsive Systems Serious B: Applications & Algorithms, November 2004, pp.363-391.

[11] S. Hu, A. Duel-Hallen, H. Hallen, *Long-Range Prediction Makes Adaptive Modulation Feasible for Realistic Mobile Radio Channels*, Proc. of 34rd Annual Conf. on Infor. Sciences and Systems, March 2000, Vol I, pp.WP4-7-4-13.

[12] A. Pietrabissa and E. Guainella, *TCP-Friendly Bandwidth-on-Demand Scheme for Satellite Networks*, ASMS Conference 2003.

[13] Majid Ghaderi, *On the relevance of Self-Similarity in Network Traffic Prediction*, Tech. Rep. CS-2003-28, School of Computer Science, University of Waterloo, October 2003.

[14] A. Bhattacharya, A. G. Parlos, and A.F. Atiya, *Prediction of MPEG-Coded Video Source Traffic Using Recurrent Neural Networks*, IEEE Transaction on Signal Processing, Vol.51, No.8, Aug. 2003.

[15] M. Ghaderi, J. Capka and R. Boutaba, *Prediction-Based Admission Control for DiffServ Wireless Internet*, In Proceedings of IEEE Vehicular Technology Conference, (VTC'2003), Oct. 2003.

[16] Y. Gong and I. F. Akyildiz, *Dynamic Traffic Control Using Feedback and Traffic Prediction in ATM Networks*, Proc. of IEEE INFOCOM, 1994, pp.91-98.

[17] Y. Shu, Z. Jin, L. Zhang and L. Wang, *Traffic prediction using FARIMA models*,in Proceeding of IEEE/ICC'99, Vol.2, Jun. 1999, pp.891–895.

[18] G. Gripenberg and I. Norros, *On the prediction of fractional brownian motion*, Journal of Applied Probability, Vol.33, 1996, pp.400-410.

[19] B. J. Vickers, M. Lee and T. Suda,*Feedback Control Mechanisms for Real-Time Multipoint Video Services*, IEEE Journal on Selected Areas of Communication, Vol.15, Apr. 1997.

[20] Andrew S. Tanenbaum, *Computer Networks. Third Edition*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1996.

[21] W. Stallings, *High-Speed Networks and Internets: Performance and Quality of Service*, Prentice-Hall International, Inc, pp. 220-247, 2002.

[22] J. Beran, *Statistics for Long-Memory Processes*, Chapman & Hall: New York, 1994.

[23] A.Einstein, *Investigations on the Theory of the Brownian Motion*, Dover, USA, 1956.

[24] Cisco on-line,*http://www.cisco.com/univercd/cc/td/doc /cisintwk/ito_doc/qos.htm*

[25] [ITU-T] ITU-T Rec. E.800, *Terms and Definitions Related to the Quality of Telecommunication Services*, Blue Book, 1988.

[26] V.Paxson and S. Floyd, *Wide-Area Traffic: The Failure of Poisson Modeling*, IEEE/ACM Transactions on Networking, Vol.3 No.3, pp. 226-244, June 1995.

[27] M.E. Crovella and A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*, IEEE/ACM Transactions on Networking, 5(6):835-846, December 1997.

[28] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*, IEEE/ACM Transactions on Networking, Vol.2, No.1, February 1994, pp.1-15.

[29] D.R.Cox, *Long -Range Dependence: A Review*, Statistics: An Appraisal (H. David and H. David, eds.), Ames, Iowa: Iowa State Universtiy Press, 1984, pp.55-74.

[30] Fowler, T.B., *A Short Tutorial on Fractal and Internet Traffic*, The telecommunications Review, Vol.10, 1999, pp.1-15.

[31] A.Dubi, *Monte Carlo Applications in Systems Engineering*, John Wiley, Sons Ltd., NY, USA 2000.

[32] J. Banks, John S. Carson II, Barry L. Nelson, *Discrete-event System Simulation*, Second Edition, Prentice Hall, Upper Saddle River, New Jersey 07458

[33] http://ita.ee.lbl.gov / html /contrib /BC.html.

[34] M.garrett and W.Willinger, *Analysis, modeling and generation of self-similar VBR video traffic*, In Preceedings SIGCOMM'94 . London, England, September 1994.

[35] N.U. Ahmed and H. Song, *Real Time Feedback Control Using Predictive States Estimation*, (submitted)