INFORMATION TO USERS

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

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

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

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

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Prediction of Cell Loss Rate and Its Application to Connection Admission Control

Hamid-Reza Mehrvar

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

May 2001

ABSTRACT

Prediction of Cell Loss Rate and Its application to Connection Admission Control

Hamid R. Mehrvar

Concordia University, 2001

At a node of a broadband network, such as an ATM network, the prediction of quality of service plays an important role in formulating traffic control functions. The historical data from 1995 to 2001 has shown that data traffic has doubled each year and this trend is likely to continue. Assuming that traffic is generated from a number of data sources, we propose a new approach in predicting packet (or cell) loss rate, which is considered to be the quality of service of interest . The proposed approach not only does not rely on an assumption of a statistical model for the traffic patterns, but also closely approximates the cell loss rate in an output queue of a node. To do this, first, we identify a set of traffic parameters, as *traffic indicator*, that can describe the behavior of short-term, long-term or self-similar traffic. Then, we approximate the cell loss rate in terms of the traffic indicator using function approximation capability of a neural network system consisting of a linear combination of a number of sigmoid functions.

The proposed traffic indicator and cell loss approximator can be used for traffic engineering of broadband networks, e.g., ATM networks, to maximize the utilization of an output link. As an illustrative example, we propose a new connection admission control that predicts packet cell loss rate from the aggregate of two traffic indicators: one for the existing connections and the other for the new connection. If the predicted

cell loss rate for the aggregate traffic indicator is less than a pre-set threshold, the new connection is admitted. Under the assumption that the users do not require a tight bound on the cell loss rate, we showed that the proposed admission control is twice as efficient as the Equivalent Capacity.

DEDICATION

To my family,

with love and reverence

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisors, Prof. M. R. Soleymani and Prof. J. F. Hayes, for their support and encouragement. It has been a pleasure for me to work under their supervision. I greatly acknowledge the funding support provided to my research work by the Canadian Institute for Telecommunications Research (CITR) during my full-time status. I thank my supervisory committee, Dr. A. Benyamin Seeyar and Dr. J. W. Atwood, and the external examiner, Prof. John Mark, for their valuable comments and suggestions. I thank my friends for giving a warm and friendly atmosphere that was conducive to a fruitful research.

Contents

List of Acronyms

AAL5	ATM Adaptation Layer 5
ABR	Available Bit Rate
ANK	Adaptive Newton Kantorovich
AR	Auto-Regressive
ATBCL	Average Time Between Cell Losses
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband ISDN
BT	Burst Tolerance
CAC	Call Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CER	Cell Error Ratio
CI	Congestion Indication
CLP	Cell Loss Priority
CLR	Cell Loss Rate
CMR	Cell Misinsertion Rate
CTD	Cell Transfer Delay
DFGN	Discrete Fractional Gaussian Noise
EPD	Early Packet Discard
ER	Explicit Rate

FBA	Fixed Bandwidth Allocation
FBM	Fractional Brownian Motion
FFBM	Fast FBM
FFGN	Fast FGN
FGN	Fractional Gaussian Noise
FIR	Finite Impulse Response
FTP	File Transfer Protocol
GCRA	Generic Cell Rate Algorithm
GRA	Global Rational Algorithm
IDC	Index of Dispersion for Counts
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital network
LAN	Local Area Network
MCR	Minimum Cell Rate
MLE	Maximum Likelihood Estimation
MMF	Markov Modulated Fluid
MMPP	Markov Modulated Poisson Process
MPLS	Multi-Protocol Label Switching
MSE	Mean Squared Error
NAR	Non-linear Auto-Regressive
NARMA	NAR Moving Average

NPC	Network Parameter Control
NPEDF	Non-Preemptive Earliest Deadline First
NRT-VBR	Non-Real-Time VBR
OAM	Operation, Adminstration and Maintenance
OPNET	Optimum Network Engineering Tool
PCR	Peak Cell Rate
PPP	Point-to-Point Protocol
QoS	Quality of Service
PPD	Partial Packet Discard
PQ	Priority Queuing
RED	Random Early Detection
RSVP	Resource reSerVation Protocol
RT-VBR	Real-Time VBR
SAR	Segmentation And Re-assembly
Satcom	Satellite communications
SCR	Sustainable Cell Rate
SECBR	Severely-Errored Cell Block ratio
SSCOP	Service Specific Connection Oriented Protocol
RM	Resouce Management
TCP	Transmission Control Protocol

TPEDF	Tracking, Preemptive Earliest Deadline
UDP	User Datagram Protocol
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Connection
VS/DS	Virtual Source / Destination
WFQ	Weighted Fair Queuing
WWW	World Wide Web

List of Figures

List of Tables

Chapter 1

Introduction

At a node of a broadband network, such as an ATM switch, it is required to maximize the utilization of each one of the output links, while maintaining the quality of service (in terms of cell loss rate, delay, etc) of all services at acceptable levels. In this work, we assume that only data services utilize the network resources and, hence, the quality of service of interest is Cell Loss Rate[1] (CLR). This assumption is based on the observation that Internet traffic has doubled each year from 1995 to 2001 and the trend is likely to continue [1]. The platform that we considered is an ATM network, although the results are applicable to other broadband networks.

In order to maximize the utilization of an output link of an ATM switch, while maintaining an acceptable level for the CLR of all connections, user demands need to be controlled in three different forms. First, a decision must be made whether a user requesting a connection, can be accommodated. Second, once a connection is established, its flow should be monitored in order to avoid performance deterioration due to user violation from its negotiated traffic contract. Third, congestion control

[1]Packet loss rate, cell loss rate or loss probability all refer to the same quantity.

must be maintained to ensure that congestion at the node does not develop, or if the congestion occurs, to quench the source as quickly as possible. These three forms of traffic control schemes need to be applied at two different levels: at the connection set-up and during the connection cycle. Call Admission Control (CAC) is applied at the connection set-up, whereas congestion control schemes (open-loop and/or closed loop) are applied during the connection cycle.

1.1 Issues and Approaches

CLR prediction plays an important role in formulating traffic control functions, in particular, CAC. Without accurate CLR prediction, a network may adopt a conservative CAC function while guaranteeing the agreed upon CLR, and may lower the bandwidth utilization.

In the literature, the CLR in a finite buffer queuing system (with buffer size x) is often approximated by $P(Q \geq x)$, the tail of the queue length distribution, in the corresponding infinite buffer queuing system. This distribution is asymptotically exponential, i.e., $P(Q \geq x) \asymp Ae^{-\eta x}$ as $x \to \infty$. However, determining the asymptotic constant A is usually computationally intensive. In the case of equivalent capacity [2][3][4], this constant is ignored. This provides an easy way of allocating bandwidth, independent of the number of sources being multiplexed. However, this simplicity is the main weakness of this approach, as it does not exploit the statistical multiplexing gain.

Markov Modulated processes, which are considered to be rich stochastic processes, have attracted researchers' attention for system traffic modeling and performance

analysis. As an example of such process, Markov Modulated Poisson Process (MMPP), has been used to model several types of traffic sources such as variable bit rate videos and still images [4]-[8] fed into a statistical multiplexer. Despite the assumption of Poisson process for mathematical tractability, the results of MMPP are complex from the computational point of view. Another Markov-modulated process that has been extensively used to model various types of traffic is the Markov Modulated Fluid (MMF) process [9]. The advantage of this model over traditional queuing models is that the numerical complexity is independent of the buffer size. However, unlike the case of the MMPP in which the discrete nature of the cells is preserved, the fluid model is unable to capture the effect of cell variability for small buffer sizes. It can be viewed as an approximation to the MMPP for large buffer sizes. Furthermore, as in the case of the MMPP, when a large number of sources are being multiplexed, the computational complexity to estimate either the queue tail or loss probability can become prohibitively high.

As far as Long-Range Dependent (LRD) traffic modeling is concerned, Tsybakov *et al* [10] constructed a self-similar model by superposing an infinite number of independently and identically distributed (i.i.d) heavily-tailed on-off sources with Pareto sojourn time and derived a lower bound solution for the buffer overflow probability. The bound is not tight and the accuracy of the model is also questionable. On the other hand, Norros [11] proposed a simple model in which the input traffic is considered to be Fractional Brownian Motion (FBM) that is self-similar. For this model, he obtained an upper bound for the queue length distribution assuming that the queue scheme is First In First Out (FIFO) and link transmission rate is constant. The probability that the queue length is greater than x is shown to be bounded below by a

complementary error function that decays more slowly than the exponential form in the Markovian processes. Some experiments with actual traces of LAN traffic have also shown that in certain circumstances FBM models serve as a good approximation in estimating the queuing behavior [12]. In view of the practical significance of FBM, several attempts have been made to obtain a tighter bound for the performance of FBM traffic than the upper bound provided by Norros. The most interesting one relies on the fact that even though an FBM process has long range correlations in time, the frequency components of its time derivative are independent Gaussian variables. By setting up the problem in the Fourier domain, Narayan [13] has obtained an exact expression for asymptotic $P(V > x)$ for $1/2 < H < 1$, where H is the Hurst parameter. In certain cases, the FBM model can represent some traffic scenarios. However, for most traffic cases encountered in the broadband networks, FBM is considered to be at the extreme.

To avoid the problems of the previously introduced methods, other approximate approaches have also been proposed for approximating the CLR. For instance, in [33] approximations have been derived for the queue length distribution of a fluid multiplexer fed by a superposition of homogeneous on/off sources. This approach uses large deviation estimates of the queue length distribution obtained when the link capacity and the multiplexer buffer size scale (or diverge) with the number of sources sharing the multiplexer. In [15], the same approach is used to derive the large deviation estimates of the queue length distribution for more general heterogeneous arrival processes, such as superposition of Gaussian processes. Although the large deviation estimates give very good approximations of the CLR even under more practically interesting assumptions (e.g., reasonably small number of sources) while avoiding the

complexity of the fluid flow and the MMPP models, they still depend on the knowledge of the arrival process, which for most practical sources is yet to be determined. To avoid dependency on the assumption of the arrival process, the non-parametric approach [32] has been introduced. The approach uses the peak rate and the mean rate of the traffic sources to obtain the distribution of number of cells in an observation interval and from there to calculate an upper bound for the CLR. A comparison (as discussed in [14]) between a non-parametric model and a statistical model, such as the fluid flow model, shows that both models overestimate CLR, however, the fluid flow model outperforms the non-parametric one. Another approach that neither assumes the arrival process nor uses the declared parameters by user, such as the peak and the mean rates, is Global Rational Approximation (GRA) algorithm [34]. In this approach, CLR, as a function of number of sources (or buffer size) is approximated by a rational function of the form $R(x) = \frac{P_m(x)}{Q_n(x)}$ where P_m and Q_n are polynomial functions of degree m and n, respectively. The attractive feature of this approach is that it approximates CLR for a large-sized system using small-sized information. However, the accuracy of the approximation is closely related to the degrees of m and n. The computation time can sometimes be long since the fitting process is based on an iterative algorithm for determining suitable values for m and n. In addition, depending on the values of the pairs, the system of equations to be solved can sometimes encounter singularity problems.

Simple and yet accurate prediction of CLR is required in order to devise an efficient traffic (or admission) control scheme. The state-of-the art approaches assume *a priori* characterization of traffic for the convenience of mathematical tractability. However, it is quite difficult, if not impossible, to provide an accurate and tight statistical model

that adequately describes the traffic arriving at a queue or a multiplexer. Proper characterization of traffic avoids use of conservative approaches for CLR prediction and, hence, results in a better utilization of the network resources. It is well known that CLR in a multimedia environment depends on many unknown and unpredictable traffic parameters, such as burst and silence lengths distributions [16] and more generally, long term correlation. Recent findings from the analysis of data traffic [17] and VBR video [18] have shown long-range dependency in the traffic patterns. We believe that consideration of parameters that represent long-range dependency can improve the CLR approximation.

Most of the existing model-based approaches in CLR approximation are conservative due to either limitation of the assumption on the arrival process, or consideration of the generic parameters that are declared in the initial phase of a call set-up. The main focus of our research effort will be on the accurate prediction of CLR using a set of traffic parameters that characterize traffic. We propose to use a universal indicator of traffic by which CLR can be predicted. Such an indicator of traffic includes a set of parameters that impact CLR, and these parameters adequately describe the quality of Short-Range Dependent (SRD), Long-Range Dependent (LRD) and/or self-similar traffic[2]. The proposed approach for CLR prediction requires solutions to two main issues:

- *Identification*: which refers to finding a set of parameters, as an *indicator* of traffic, that are sufficient for CLR prediction. The desire is to have fewer parameters to reduce complexity.

[2]Markov Modulated sources are SRD, whereas thick-tailed distributed On-Off sources or FBM sources are LRD.

- *Approximation*: which refers to finding techniques that approximate CLR in terms of the traffic indicator.

To identify the indicator of traffic, we generated different traffic scenarios each by aggregating traffic from many bursty data sources. The generated traffic from each traffic scenario has been applied to a FIFO queue with a finite buffer capacity and the CLR has been measured for different traffic parameters (such as load, burstiness, etc). Our results (as discussed in Chapter 4) show that the indicator of traffic should include:

- Traffic load, which is related to the first moment of traffic,
- IDC(1), which is related to the second moment of traffic and it represents the value of the Index of Dispersion for Counts (IDC) calculated over a single interval,
- K, which is related to the slope of IDC curve in logarithmic scale and it refers to the degree of self-similarity (for FBM traffic K represents Hurst parameter), and
- IDC(L), which is the value of IDC at L interval and it represents the degree of burstiness in that interval.

We examined various approximation techniques to capture the functional relationship between the traffic indicator and the measured CLR. Due to the requirement of real-time CLR prediction, we are interested in the low complexity approaches. We have shown that for all traffic profiles, a neural network system consisting of a linear summation of a set of sigmoid functions can adequately capture CLR as a function of traffic indicator and buffer size.

As an illustrative example, we propose a new CAC strategy based on the proposed CLR approximator. In addition to the identification and approximation, the derivation of the projected traffic indicator for all connections is an issue for the formulation of the new CAC. This issue will be discussed in Chapters 6 and 7.

1.2 Thesis Contributions

Let us assume that

- traffic patterns are generated by data sources,
- CLR is the quality of service of interest,
- traffic sources do not require a tight bound on CLR, i.e., the traffic engineering functions assume predictive services over the network and
- the queues are FIFO with finite capacity.

The key questions that the proposed research attempts to answer are:

1. What are the sufficient parameters (as an *indicator* of traffic) by which CLR can be closely predicted?
2. How can one approximate the CLR as a function of the traffic indicator?
3. How well does the CLR approximator perform?
4. Given the CLR approximator, how can one derive the projected indicator of traffic for all the existing connections?

5. How can we formulate an admission strategy based on the traffic indicator? Or equivalently, how can the two indicators of traffic, i.e., for the existing connections and for the new one, be aggregated?

6. How effective is the proposed CAC procedure?

Answers to these questions constitute our main contributions.

1.3 Thesis Outline

The rest of this thesis is organized as follows. In chapter 2, we provide an overview of the state-of-the art admission control and CLR approximations in broadband ATM networks. In Chapter 3, we state the research problem, its objectives along with the justification of why we need to solve this problem. We break down the problem into three research issues: Identification, Approximation and Derivation.

In Chapter 4, we identify the indicator parameters that are universal and have dominant effect on the CLR. In Chapter 5, we examine various approximation techniques for the CLR function in terms of the identified traffic indicator and buffer size. A real-time derivation for the traffic indicator is discussed in Chapter 6, which provides the basis for the formulation of an admission strategy (discussed in Chapter7) using the proposed traffic indicator. Finally, thesis conclusions and some suggestions for future research are discussed in Chapter 8.

To complement the contents of the thesis Chapters, there are four appendices. Appendix A gives an overview of congestion control schemes in broadband ATM and IP networks, which complements the discussions in Chapter 2. Appendices B and C are related to MMPP and FBM traffic models, respectively. These traffic models

have been used to generate various traffic profiles and their mixtures. Appendix D provides a proof that a linear combination of a set of sigmoid functions can accurately approximate any continuous function, such as the CLR function.

Chapter 2

State-of-the-Art Connection Admission Control in Broadband ATM Networks

Generic traffic management functions can be classified as either traffic control or congestion control functions. A clear distinction of whether a traffic management mechanism is traffic control or congestion control is difficult. Traffic control functions are network functions that take actions to avoid congestion conditions. Congestion is defined as the state of the network elements in which the network is unable to meet the negotiated performance objectives for already established connections and/or new connections. A range of traffic and congestion control functions is necessary to maintain the quality of service of the connections over the network. These functions are divided into Connection Admission Control (CAC) and congestion control.

In this chapter we discuss CAC in ATM networks. Appendix B provides an overview of congestion control schemes in ATM and IP networks. The intent is to position the

proposed research with respect to the state-of-the-art traffic control functions.

2.1 Background

Since packet switched networks support a large number of bursty traffic sources, statistical multiplexing can be used to gain bandwidth efficiency allowing more traffic sources to share the bandwidth. But if a large number of traffic sources become active, traffic control is required to relieve the potential network congestion due to the limited capacity within the network. Provisioning links at a certain utilization, say 80%, does not solve congestion problem due to the random nature of traffic. On the other hand, setting the resource utilization at a low level, say 50%, so that a simple admission scheme can prevent congestion, is not an acceptable alternative because of under-utilizing the resources. Thus, the challenge in packet switching networks is to develop a framework that maximizes the resource utilization while controlling the traffic entering the network, so that the temporary periods of overload due to the stochastic nature of traffic do not turn into sustained periods of congestion, which deteriorate the network performance. In order to achieve efficient utilization of resources while QoS requirements (delay, loss, etc) of all applications are met, traffic control functions need to be performed both at the connection set-up and during a connection life-time. CAC is performed during the connection set-up whereas congestion control is performed during a life-time of a connection. Figure 2.1 shows this classification of traffic control functions. In the next sections we discuss connection admission control because of its relevance to this thesis. Congestion control schemes are discussed briefly in the Appendix A.

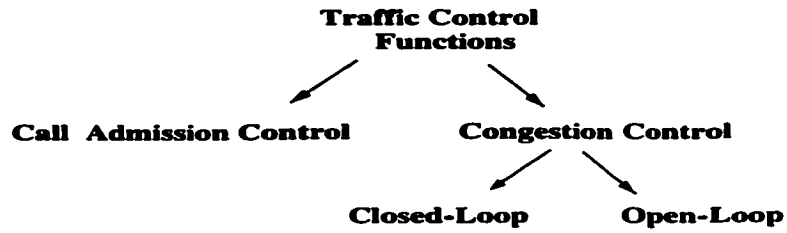

Figure 2.1: Classification of traffic control functions into admission and congestion control.

What makes the admission and congestion control problem different is the variety of traffic types to be accommodated each to be provided with a different quality of service. Thus, before addressing traffic control functions we discuss quality of service parameters and their relevance to each one of the services.

2.2 QoS Metrics

2.2.1 ATM QoS Parameters

The QoS metric in a connection-oriented service network differs from a connection-less service network. It can be categorized into two classes: the first class includes the *call control parameters* associated with connection-oriented service networks and the second class is the set of *information transfer parameters* defined in packet networks[1] as shown in figure 2.2. Call control parameters include connection *set-up delay*, connection *release delay* and *blocking probability*. Information transfer parameters are used by the network for CAC, finding a path that support its service requirement and so forth. These parameters can be classified into two classes of *transmission* and *networking* parameters.

[1]ATM networks are connection oriented packet switching networks.

Transmission parameters are:

- Cell Error Ratio (CER): CER is related to the bit error rate of the information in the cell. It is the ratio of cells in error to the total number of transmitted cells.
- Cell Misinsertion Ratio (CMR): CMR represents the number of misinserted cells during an interval. Misinsertion happens due to an error in the header, which causes a cell to appear in a connection flow that it does not belong to.
- Severely-Errored Cell Block Ratio (SECBR): SECBR is the ratio between the number of severely errored cell blocks and the total transmitted cell blocks. Practically, a cell block could be a sequence of cells between two OAM(Operation, Administration, Maintenance) cells.

Networking parameters are:

- Cell Transfer Delay (CTD): CTD is defined as the elapsed time between the transmission of the first bit of a cell and the reception of the last bit. This delay is composed of propagation delay, transmission delay, switching delay and queuing delay.
- Cell Delay Variation (CDV): CDV, also called *jitter*, gives the delay variability between two consecutive cells. ATM Forum specification differentiates between one-point CDV and two-point CDV. One-point CDV describes the variability in the pattern of cell arrival observed at a single measurement point with respect to the negotiated peak rate, whereas two-point CDV is the difference between the actual measured CTD between two points and a predefined reference CTD between the two points.
- Cell Loss Ratio: CLR is the ratio between the lost cells and the transmitted cells. Different applications have different CLRs, e.g., 10^{-3} for telephony and 10^{-9} for MPEG-2

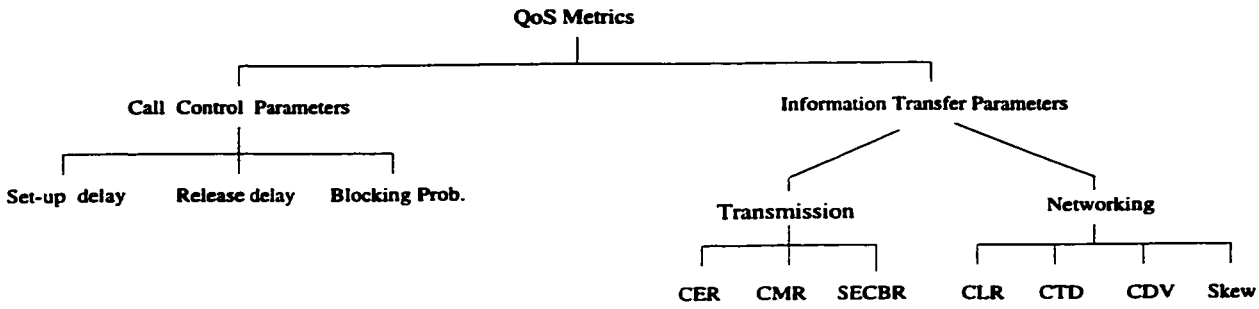

Figure 2.2: Quality of Service metrics.

video stream. A related parameter to CLR is the Average Time Between Cell Losses (ATBCL).

- Skew: This parameter shows the difference in presentation times of two objects, e.g., video and audio, which impacts the quality of the picture.

ATM Forum [19] specifies a subset of the above parameters as QoS parameters: CTD, CDV, CLR, CER, CMR and SECBR. Other QoS parameters such as connection establishment delay, blocking probability and skew may still be very useful.

The ITU-T Recommendation I.362 specifies six service classes among the various services and applications that an ATM B-ISDN might support. These are listed in Table 2.1. Class A and class B are stream type and real-time, whereas class C and class D are jitter-tolerant. Examples of class A, B, C and D are voice, compressed video, TCP/IP over ATM and SDMS over ATM, respectively. In class X, the traffic and timing relationship between the source and destination is defined by the user, whereas in class Y traffic characteristics can be changed after connection establishment. A QoS class specifies a set of performance parameters and the object values for the connections belonging to the class. There can be more than one QoS class associated with every service class in broadband networks.

Class	Bit Rate	Timing Relation*	Connection Mode	Example
A: stream	Constant	Required	connection-oriented	voice
B: stream	variable	Required	connection-oriented	video
C: data	variable	not Required	connection-oriented	bulk data
D: data	variable	not Required	connection-less	data(between LANs)
X: data	variable	not Required	connection-oriented	data
Y: data	variable	not Required	connection-oriented	data

*Timing between source and destination

Table 2.1: ITU-T Recommendation I.362 for B-ISDN service classes.

ATM Forum defined the following five QoS classes (or service categories):

a) Constant Bit Rate (CBR) category which requests a static amount of bandwidth and supports real-time applications requiring tight delay and constrained delay variations,

b) Real-Time Variable Bit Rate (RT-VBR) service which requires tight delay and constrained delay variations,

c) Non-Real-time VBR (NRT-VBR) which expects low CLR but has no delay variation requirement,

d) Available Bit-Rate (ABR) which changes its rate based on the network congestion information and does not require bounding the delay or delay variation but expects low CLR,

e) Unspecified Bit Rate (UBR) which does not require any delay constraints or CLR.

Each one of the above five classes specifies traffic parameters in terms of Peak Cell

QoS class	*Traffic Parameters*					*Guaranteed QoS*		
	PCR	CDVT	SCR	BT	MCR	CLR	CTD	CDV
CBR	•	•				•	•	•
RT-VBR	•	•	•	•		•	•	•
NRT-VBR	•	•	•	•		•	•	
ABR	•	•			•	•		
UBR	•	•						

Table 2.2: Traffic parameters and the required QoS for each one of the ATM service categories.

Rate (PCR), Cell Delay Variation Tolerance (CDVT), Sustainable Cell Rate (SCR), Burst Tolerance (BT) and Minimum Cell Rate (MCR). Table 2.2 shows the traffic parameters and the required QoS for each one of the ATM service categories.

2.2.2 IP QoS Parameters

IP networks have two different architectures for services: *Integrated* services and *Differentiated* services. Integrated services define two levels of services called *Controlled Load* and *Guaranteed QoS*. In addition, recent work in the Internet Engineering Task Force (IETF) has focused on defining scalable QoS capability for the Internet backbone using differentiated services. The IETF Integrated Services focus primarily on bounding absolute delay as the principal QoS parameters [21][20], as shown in Table 2.3. More recent standard activities in the IETF differentiated services working group extend the definition of QoS to include packet loss rate.

QoS parameter	Controlled Load	Guaranteed QoS
Maximum Delay Variation	Not specified	Automatically measured
Packet loss rate	Little or no congestion loss	enough buffer for zero congestion loss
Minimum delay	Large percentages of packets do not exceed minimum delay	Not specified
Average delay	Little or no queuing delay	Not specified

Table 2.3: IP's Integrated services QoS parameter: terminology and definitions.

2.3 Call Admission Control (CAC)

When a new call (or connection) is received at the network, the call admission procedure is executed to decide whether to accept or reject the call. A call is accepted if the network has enough resources to provide the QoS requirements for the new connection without affecting the QoS of the existing ones. Accordingly, two questions need to be answered: one is how the amount of bandwidth required by a new connection can be determined and second one is how we can make sure that service levels for the existing connections are not affected when multiplexed with the new connection. Any admission technique that tries to answer these two questions should function in real-time and attempt to maximize the utilization of network resources. The first step is to determine a set of parameters to describe the source activities adequately for the prediction of the performance of interest. This is still an open issue. Nevertheless, some CAC schemes, as in what follows, assume analytically tractable models such as multi-state Markovian models to characterize variable bit rate sources. Both ITU-T [22] and ATM Forum [19] require that the parameters specified in Table 2.2, i.e., traf-

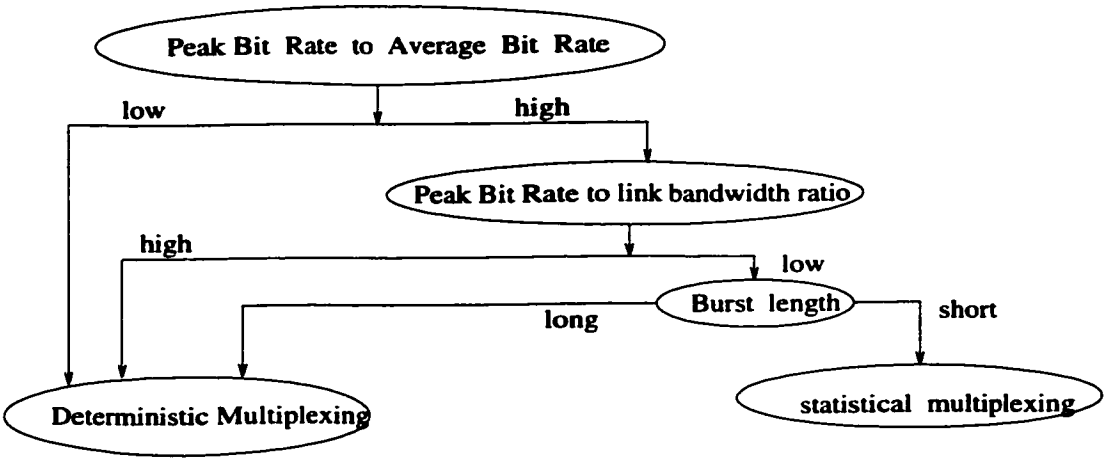

Figure 2.3: Deterministic multiplexing versus statistical multiplexing.

fic parameters, QoS requirement and QoS class (or service category) will be available to CAC.

In general, the bandwidth required by a connection can be allocated by two approaches: deterministic and statistical multiplexing, as shown in Figure 2.3. Deterministic approach allocates the peak rate and causes waste of bandwidth particularly for those connections with high peak to average bit rate ratios. Although this allocation can totally eliminate cell level congestion, there is nonzero probability that cell loss occurs when the number of active sources that simultaneously send cells to the queue exceed the buffer size. The deterministic approach goes against the philosophy of the ATM framework, which takes advantages of multiplexing capability. In statistical multiplexing, the allocated bandwidth to a VBR source is less than the peak rate, but necessarily greater than the average rate. This allocated bandwidth in some literature (e.g., [24]) is referred to as statistical bandwidth. As seen in Figure 2.3, there is a trade-off between deterministic multiplexing and statistical multiplexing. When the peak to the average ratio is high and peak rate to bandwidth ratio is low

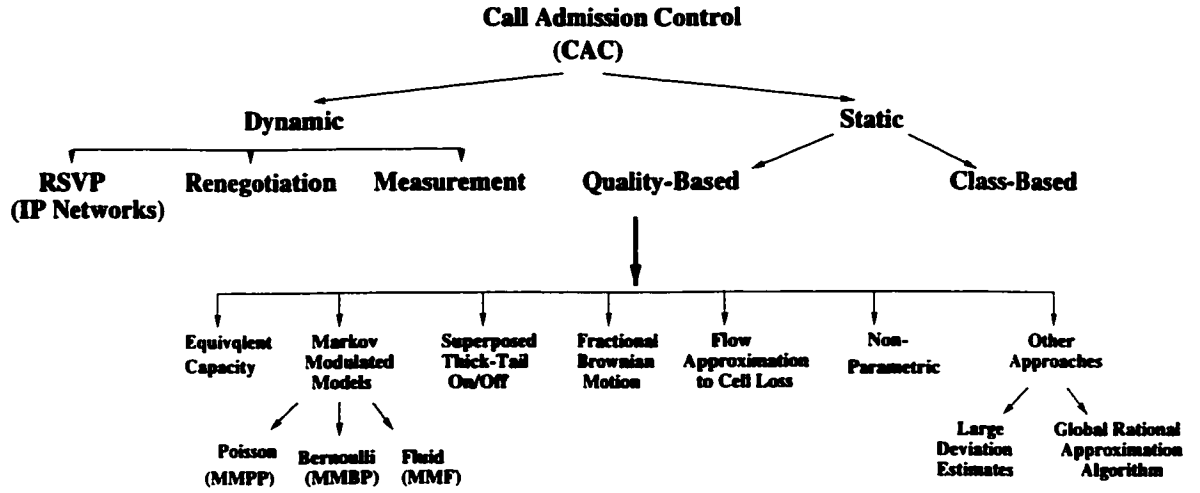

Figure 2.4: Classification of Call (connection) Admission Control.

and also burst length is short, statistical multiplexing is beneficial; otherwise, the deterministic approach is beneficial.

All the proposed CAC schemes in ATM networks are based on statistical multiplexing. They can be broadly classified into static and dynamic schemes. In the static scheme the decision is based on the static measures of either the ATM traffic descriptor (PCR, SCR and BT) supplied by the new connection or the maximum number of allowed connections for a given type of traffic. Dynamic schemes consider the dynamic changes in the network in an attempt to better utilize the network resources.

Static schemes are classified into *class-based* and *quality-based* schemes. Dynamic schemes are classified into *re-negotiation* and *measurement* schemes for ATM networks and Resource reSerVation Protocol (RSVP) for IP networks. Figure 2.4 shows this classification. In the following subsections, we address briefly all the schemes.

2.3.1 Class-based CAC

In this scheme, traffic is classified based on the required resources and the QoS requirements. For example, class 1 has bandwidth requirement of 1-10Mbps and class 2 has bandwidth requirement of 10-20 Mbps and so on. Due to the large number of classes on both bandwidth and QoS requirements, this approach may not be easy to implement.

Classification is made in several ways. For instance Galassi [23] tabulates the required bandwidth of all the connections that have the same characteristics. An entry $(i,\ j)$ in the table shows the bandwidth required for j class-i connections to meet their QoS requirements. A class-i connection is accepted if, for the total of j connections, the bandwidth specified in the table is available. In another scheme [25] (explained by [26]), the available bandwidth is distributed among various traffic classes. For every class i, n_i is the maximum number of class-i that can be accepted. The problem with this scheme is that connections of a certain class may be rejected when their number exceeds the maximum, even though other classes are not utilizing their allocated bandwidth. Bolla [27] has attempted to alleviate this problem by soft partitions.

The advantage of class-based schemes is their simplicity at connection acceptance. This is because the scheme uses tabulated results of off-line simulation or computation. The disadvantages are a low statistical multiplexing gain due to the classification and an increase in the storage as the number of classes grow. Low statistical multiplexing gain is due to the unused bandwidth of each class that can not be shared by other classes. For example, if a class 1 connection has a peak bandwidth between 10 to 20 Mbps, then a 12 Mbps connection causes 8 Mbps of unused bandwidth.

2.3.2 Quality-based CAC

Quality-based CACs attempt to predict the quality of service for all connections in order to decide whether to admit a new call. Most of the quality-based approaches in the literature have considered the CLR as the QoS of interest.

In the literature, the loss probability in a finite buffer queuing system (with buffer size x) is often approximated by $P(Q \geq x)$, the tail of the queue length distribution in the corresponding infinite buffer queuing system. For infinite buffer queuing systems, it has been shown in considerable generality that $P(Q \geq x)$ is asymptotically exponential, i.e.,

$$P(Q \geq x) \rightarrow Ae^{-\eta x} \qquad as \;\; x \rightarrow \infty. \tag{2.1}$$

Here η is a positive constant called the *asymptotic decay rate.* A is a positive constant called the *asymptotic constant.* The asymptotic decay rate in a finite buffer queuing system is the same as the one in the equivalent system with an infinite buffer (which is another reason why the tail is often used to approximate the CLR). However, it is usually computationally intensive to determine the asymptotic constant A; as a result, it is sometimes ignored. In the following we describe various quality-based CACs as classified in Figure 2.4.

Equivalent Capacity (Effective bandwidth)

The classical Equivalent Capacity approximation assumes that the constant A in (2.1) is 1, i.e., if we write it as

$$P(Q \geq x) \rightarrow e^{-\eta x + \log(A)} \qquad as \;\; x \rightarrow \infty$$

then

$$P_{loss} \approx e^{-\eta x} \qquad as \;\; x \to \infty \tag{2.2}$$

Although this approximation is logarithmically similar to the exact loss probability [28], the constant can often be a fairly small multiplicative factor. In that case the approximation would certainly not be useful for any practical loss probabilities. One of the main reasons that Equivalent Capacity has become very popular in the literature is that it provides an easy way of allocating bandwidth independent of the number of sources being multiplexed [2][3][4]. For example, let us consider the following bandwidth allocation problem:

For each source i, a constraint on the probability of loss P^i_{loss} [as defined by (2.2)] is given by

$$P^i_{loss} \leq Threshold, \tag{2.3}$$

then for a total capacity μ , how should the bandwidth be allocated such that the above constraint is met for each source?

Equivalent Capacity provides a relatively simple answer to this question. It says that we can look at each source in isolation being fed into a queue with a certain capacity. Then, for source i , we can find the minimum capacity C_i such that the constraint given by (2.3) is met. The Equivalent Capacity approximation further states that as long as

$$\sum_i C_i \leq \mu, \tag{2.4}$$

none of the constraints $P^i_{loss} \leq Threshold$ will be violated.

Guerin *et al* [3] proposed an approximation for equivalent capacity with feasible real-time computation. They employed two different approaches: one based on the fluid

model[2] and the other based on the approximation that the stationary bit rate of N sources follows a Gaussian distribution with the mean and variance of the aggregate bit-rate.

a) *Gaussian Approximation*: In this method, a connection i is characterized by its average bit rate, m_i, and standard deviation, σ_i. The problem is to determine the required bandwidth of c_0 for n multiplexed connections so that the probability that the aggregate bit rate of all the connections exceeds c_0 is less than ϵ, where ϵ is a small positive number. Assuming that the aggregate bit rate follows a Gaussian distribution, c_0 can be derived (Appendix of [24]) as $c_0 \approx m + \alpha\sigma$, where m and σ are the mean and the standard deviation of the aggregate bit rate and α is the inverse of Gaussian distribution with one possible value given by

$$\alpha = \sqrt{2\ln(1/\epsilon) - \ln 2\pi}.$$

When a new connection arrives, new m, σ and c_0 are calculated. If c_0 is less than the link capacity, the connection is accepted. The Gaussian assumption for the aggregate bit rate is true if the stationary distributions of individual connections are also Gaussian. The approach doesn't fully explore the amount of achievable statistical gain because of the zero buffer assumption.

b) *Fluid Flow Approximation*: Let us assume each source is characterized by an independent On-Off source with the exponential sojourn time, peak rate of R, average rate of m and burst size of β. If a connection with the parameters (R, m, β) input to a link with buffer capacity of B, then the service rate that

[2]Fluid flow model [29] assumes the number of cells during On period is so large that it appears to be a continuous flow

gives buffer overflow probability of ϵ is

$$C_i = R\frac{z - 1 + \sqrt{(z-1)^2 + 4\rho y}}{2z} \tag{2.5}$$

with $z = -\ln(\epsilon)(1-\rho)\beta/B$, where ρ is the source utilization, i.e., m/R. In this framework, the total bandwidth of n connections is equal to the sum of the equivalent capacities of the individual connections, C_i:

$$C = \sum_{i=1}^{n} C_i.$$

However, C overestimates the required bandwidth for the aggregate traffic because interactions between individual connections have not been considered. To capture the effect of multiplexing, the Gaussian approximation is used together with the equivalent capacity. In particular, the total bandwidth required for the aggregate traffic of n connections, C', is given by

$$C' = \min(m + \alpha'\sigma, \quad \sum_{i=1}^{n} c_i),$$

where

$$m = \sum_{i=1}^{n} m_i, \quad \sigma^2 = \sum_{i=1}^{n} \sigma_i^2,$$

and the mean, m_i, and the variance, σ_i^2, of the connection bit rate are related to the connection parameters by

$$m_i = R_i\beta_i, \quad \sigma_i^2 = m_i(R_i - m_i).$$

When a new connection with parameters (R_{n+1}, m_{n+1}, β_{n+1}) arrives, new values of m and σ^2 are calculated and C_{n+1} is evaluated from (2.5). Then, C' is calculated for all $n+1$ connections and if it is less than the link bandwidth, the call is accepted. Combining the two approaches and choosing the minimum

value may still overestimate the capacity. Experiments in [30][14] indicate the inaccuracy of the equivalent capacity in some situations.

Given that the bandwidth required to meet the needs of each source is C_i, the *effective* bandwidth (or the equivalent capacity) of all sources is simply $\sum_i C_i$. Therefore, in the bandwidth allocation problem, the only computationally intensive part is to determine the capacity C_i for each individual source. This is why Equivalent Capacity appears to be such a tempting method to use for bandwidth allocation. However, it is the simplicity of (2.4) which also exposes its main weakness. Since the required capacities of connections add, it means that statistical multiplexing gain, which is so important to the success of these high-speed broadband networks, is not exploited. A little thinking will convince us that the only sources for which the bandwidth of connections add, as in (2.4), are Poisson sources. For all sources that are more variable than the Poisson source (such as Markov-Modulated sources), adding the capacities in this way will prove to be conservative and wasteful of network resources since the aggregate source will be "smoother" than the original sources. Moreover, if the sources are less variable than Poisson, adding the capacities would result in the system being overutilized and lead to violations of the negotiated QoS parameters.

Markov Modulated Models

A Markov-Modulated source is governed by an underlying continuous Markov chain with state space Σ that determines the current state of the source. In each state $i \in \Sigma$, the source transmits information at a rate λ_i according to a stochastic process, which we will call the modulated process. The sojourn times for each state are exponentially distributed with the mean $-1/m_{ij}$. When each sojourn time is over, the Markov chain

moves to a state $j \neq i$ with probability $-m_{ij}/m_{ii}$. Hence, m_{ij} is often referred to as the transition rate from state i to state j. The MMPP has been widely used to characterize different types of traffic such as voice, video, and still images citeElwalid93-[7]. In this model cells arrive at a multiplexer according to a Poisson process whose intensity depends on the state of a Markov process. The fact that the modulated process is assumed to be Poisson allows for mathematical tractability. Moreover, the Poisson approximation for the modulated process of the MMPP is fairly good when a large number of Markov-modulated sources are multiplexed, as is expected to be the case in the broadband networks. Heffes and Lucantoni [7] have studied the performance of a statistical multiplexer whose inputs consist of packetized voice and data sources. The server is allowed to be general. The superposition is approximated by a correlated MMPP which is chosen such that several of its statistical characteristics identically match those of the superposed process. The system is represented by MMPP/G/1 and matrix analytic methods are used to evaluate the system performance measures. This technique is shown to have good results in determining the mean delay in the queue, but it is not accurate in determining the queue delay distribution. In [5], Baiocchi *et al* provide a technique which studies multiplexed On-Off sources by approximating the aggregate input process by means of a suitably chosen two-state MMPP. Their model provides a good insight, but is unfortunately limited to the multiplexed On-Off case. In [8], Choudhury, Lucantoni, and Whitt provide an interesting three-term approximation to determine the tail probability of the waiting time for independently identically distributed (i.i.d.) On-Off sources, i.e.,

$$P(W \geq t) \approx \alpha_1 e^{-\eta_1 t} + \alpha_2 e^{-\eta_2 t} + \alpha_3 e^{-\eta_3 t}, \tag{2.6}$$

where α_1 and η_1 are the asymptotic constant and the asymptotic decay rate, respectively, in equation (2.1), while α_2, η_2, α_3 and η_3 are chosen to match different parameters and moments of the distribution $P(W \geq t)$. Assuming that service time is small enough in comparison to an average delay, we can approximately represent the probability of cell loss in the queue by the delay survivor function based on

$$Pr(Q > \frac{t}{h} = x) = Pr(W > t),$$

where h is the time to serve a cell. This approximation for the tail of MMPP can be computed much more easily than the exact tail, but it still suffers from the computational complexity involved in determining the asymptotic constant α_1.

Another Markov-modulated source that has been extensively used to model various types of traffic is the Markov Modulated Fluid (MMF) source. In MMF, information is generated and processed as a continuous flow (fluid) at a rate which depends on the state of the Markov process. The model gained widespread popularity as a result of the pioneering work by Anick, Mitra, and Sondhi [9]. The advantage of this model over traditional queuing models is that the numerical complexity is independent of the buffer size. However, unlike the case of the MMPP, where the discrete nature of the cells is preserved, the fluid model is unable to capture the effect of cell variability. Hence, the fluid model is usually inaccurate for small buffer sizes and can be viewed as an approximation to the MMPP for large buffer sizes. Furthermore, as in the case of the MMPP, when a large number of sources are being multiplexed, the computational complexity to estimate either the queue tail or loss probability can become prohibitively high.

Superposed Thick Tail On-Off Models

Tsybakov *et al* [10] constructed a self-similar model by superposing an infinite number of i.i.d heavily-tailed with Pareto sojourn time On-Off sources. They derived a lower bound for the buffer overflow as

$$P_{overflow} \geq \frac{c}{\alpha(\alpha-1)(E_\tau + E_k)^2}\left(\frac{x+a+1}{a}\right)^{\alpha+1}$$

where

$$E_\tau = \frac{\sum_{n=1}^{\infty} n^{-\alpha}}{\sum_{n=1}^{\infty} n^{-\alpha-1}}, \tag{2.7}$$

$$E_k = (1-e^{\lambda_1})^{-1} - 1, \tag{2.8}$$

$$a = (E_\tau + E_k)^{-1} - 1. \tag{2.9}$$

In the above set of equations x is the buffer size, α is the Pareto parameter related to the Hurst parameter, H, by $H = (3-\alpha)/2$ and λ_1 is the mean number of Poissonian active sources. Asymptotically, when x is large, $P_{overflow} \geq cx^{-\alpha+1}$ where c is a constant independent of x but dependent on λ_1 and α. The important conclusion is that for Pareto On-Off sources the overflow probability, $P_{overflow}$, can not decrease faster than hyperbolically with the growth of buffer size x.

Fractional Brownian Motion Model

Norros has considered a simple model where the input traffic is considered as Fractional Brownian motion (FBM), which is LRD. There is no exact solution, only an upper bound approximation method [11][31] was followed resulting in the following expression for the probability of the queue length:

$$P(Q > x) \approx \exp\left[-\frac{(C-m)^{2H}}{2k^2(H)\cdot a}\cdot x^{2-2H}\right] \tag{2.10}$$

where H is the Hurst parameter characterizing the long-range correlation in the input traffic, C is the service rate, m is the mean arrival rate, a is the variance coefficient and $k(H)$ is given by:

$$k(H) = H^H(1-H)^{1-H}.$$

From the above upper bound, equation (2.10), it is apparent that the CLR decays more slowly than the exponential form characteristic of Markovian processes. More recently, using queuing simulation experiments with actual traces of Ethernet LAN traffic, it has been shown that the effect of long range dependence traffic can often dominate the queuing behavior of experimentally measured traffic and that in certain circumstances FBM models serve as a good approximation in estimating the queuing behavior [12]. In view of its practical significance, several attempts have been made to obtain tighter bounds for the performance of FBM traffic than the provided upper bound in (2.10). The most interesting one relies on the fact that even though an FBM process has long range correlations in time, the frequency components of its time derivative are independent Gaussian variables. By setting up the problem in Fourier domain, Narayan [13] has obtained an exact expression for asymptotic $P(V > x)$ for $1/2 < H < 1$ as

$$P_{asy}(V > x) = \left[\frac{2^{1-1/H} d_0^{1/H-2}}{\sqrt{H-H^2}}\right] \theta \exp\left[(x+st_0)^2/2t_0^{2H}\right], \qquad (2.11)$$

where s is the difference between the serving capacity and the mean traffic rate and

$$d_0 = \frac{x+st_0}{t_0^H}$$

with

$$t_0 = \frac{xH}{s(1-H)}.$$

The parameter θ is a constant prefactor that has been determined for various values of H numerically. This parameter decreases as H increases, for instance for H=0.5, $\theta = 0.98$ and for $H = 1$, θ is 0.404.

Flow Approximation to Cell Loss

In this scheme [24], a VBR On-Off source is characterized by its peak bit-rate, p_i, and average bit-rate, m_i, with no assumption on the distribution of On-Off periods except that the probability of being in the active state is given by m_i/p_i. Consider N independent sources multiplexed on a link with the transmission rate of C cells/sec and a buffer size of B. Let r_i denote the random variable for cell distribution of connection i, then random variable $R = \sum_{i=1}^{i=N} r_i$ denotes the aggregate traffic from N connections. Thus, the flow model is a continuous cell stream with the arrival rate R and departure rate C. Denoting q as available buffer slot, the rate at which cells are lost, L, is given by

$$L = \sum_{X>C} (X - C) Pr(R = X)\{1 - Pr(q > 0 | R = X)\},$$

where X is the aggregate rate. The meaning of the equation is: if $X > C$ and the buffer is full, the rate at which cells are lost is equal to $(X - C)$. The key in applying this method is in the fact that as the active period increases while the peak and average rates are kept the same, the probability of available buffer slot reduces to zero for $X > C$. Then the above equation can be reduced as

$$\hat{L} = \sum_{X>C} (X - C) Pr(R = X),$$

which defines a supremum of L with respect to the burst length. Using $\hat{L}$, an upper bound for cell loss probability can be given as

$$P_l = \hat{L} / \sum_{i=1}^{N} m_i.$$

An approximation to P_l is given by

$$P_l = \frac{\prod_{n=1}^{N} (m_n/p_n)\{e^{-s^* p_n} - 1\} + 1}{s^* e^{-s^* C} \sum_{i=1}^{N} m_i},$$

with s^* obtained as the root of

$$\sum_{i=1}^{N} \frac{m_i e^{sp_i}}{(m_i/p_i)\{e^{sp_i} - 1\} + 1} - \frac{1}{s} - C = 0.$$

The calculated P_l can be used to decide whether to admit a new connection. The main drawback of this approach is that the size of buffer is not considered in making the decision.

Non-Parametric Approach

To avoid dependency on the assumption of arrival process, the non-parametric approach [32] has been considered as an alternative. The approach is based on the peak and the average cell rates of the connections and does not require any knowledge of the distribution of the cell arrival process. Consider a link with a transmission rate of C cells/sec and a cell transmission time (slot) of $1/C$. Let R_i' and a_i' denote the peak and the average cell rate of the connection i. Given an observation of r slots, the new peak and the average rates are defined as $R_i = \lceil rR_i'/C \rceil$ and $a_i = \lceil ra_i'/C \rceil$. If the multiplexed connections onto the link are provisioned a target CLR of P_{CLR}, then

$$P_{CLR} \leq U(n,\ r) = \frac{\sum_{k=0}^{\infty} \max(0,\ k - r)\theta_1^* \theta_2^* \cdots \theta_n(k)^*}{\sum_{k=0}^{\infty} k\theta_1^* \theta_2^* \cdots \theta_n(k)^*}$$

where $\theta_1^* \theta_2^* \cdots \theta_n(k)^*$ denotes n-fold convolution of $\theta_i(j)$s, with $\theta_i(j)$ denoting the distribution of the number of cells arriving (maximum or none) during r slots, i.e.,

$$\theta_i(j) = \begin{cases} a_i/R_i & if \quad j = R_i \\ 1 - a_i/R_i & if \quad j = 0 \\ 0 & \text{otherwise} \end{cases} .$$

Now, the new $(n+1)$th connection can be admitted if the calculated $U(n+1,\ r)$ is larger than the target cell loss rate, P_{CLR}, otherwise, it is rejected. The approach requires real-time calculation of $U(n,\ r)$ on a recursive manner that calls for a dedicated signal processor.

Other Approaches

Other approaches for the CLR approximation have also been proposed. For instance, in [33] approximations have been derived for the queue length distribution of a fluid multiplexer fed by a superposition of homogeneous On/Off sources. This approach uses large deviation estimates of the queue length distribution obtained when the link capacity and the multiplexer buffer size scale (or diverge) with the number of sources sharing the multiplexer. In [15], the same approach is used to derive the large deviation estimates of the queue length distribution for more general heterogeneous arrival processes such as superposition of Gaussian processes. Although the large deviation estimates give very good approximations of the CLR even under more practically interesting assumptions (e.g., reasonably small number of sources) while avoiding the complexity of the fluid flow and MMPP models, they still depend on the knowledge of the arrival process, which for most practical sources is yet to be determined.

Another approach that neither assumes an arrival process nor uses the parameter

declared in the traffic contract (as in the case of non-parametric) is global rational approximation algorithm (GRA) [34]. In this approach, the CLR as a function of number of sources (or buffer size) is approximated by a rational function of the form $R(x) = \frac{P_m(x)}{Q_n(x)}$, where P_m and Q_n are polynomial functions of degree m and n, respectively. The coefficients of P_m and P_n are determined by solving a set of linear equations: $R(x_i)=G(x_i)$, $i = 1, \cdots, m+n$, where the pairs $[G(x_i), x_i]$ represent small-sized system information, e.g., {CLR, number of users}. The attractive feature of the approach is to approximate CLR for a large-sized system using small-sized system information. Although it is efficient and accurate in many cases, it has the following drawbacks: 1) The accuracy of the approximation is closely related to the degrees m and n, which in turn determine the numbers of pairs $[G(x_i), x_i]$ that are required, i.e., $m+n$ pairs. The accuracy of GRA can be poor if m and n are not large enough; 2) The computation time can sometimes be long since the fitting process is based on an iterative algorithm to determine the suitable values for m and n. In addition, depending on the values of pairs, the system of equations to be solved can sometimes encounter singularity problems.

2.3.3 Renegotiation

In this scheme, when a source feels that the presented traffic descriptor during call admission no longer describes its traffic, it may choose to re-negotiate for more or less bandwidth by presenting a new traffic descriptor [35]. This approach results in a better utilization of the bandwidth at the expense of additional signaling overhead. For this reason, it has not been adopted by the ATM Forum.

2.3.4 Measurement

In a measurement scheme [36], similar to non-parametric approach, the admission decision is independent of the cell arrival process. It can be considered as a dynamic non-parametric approach as it measures the traffic descriptor of the existing connections in real-time when a new connection requests admission. Dynamic measurement can provide a better estimate of the QoS. Qiu and Knightly [37] proposed a measurement approach for the delay-sensitive traffic based on the measured maximal rate envelops of the aggregate flow. Lee and Song [38] proposed a variable window based measurement in which the window size changes based on the actual measured packet loss. The advantage of measurement approaches is that it considers the dynamicity of the network traffic in resource allocation and, hence, results in better utilization of the resources. Our proposed CAC, as we see, is similar to this approach. The main difference is that the proposed CAC uses a set of parameters that describe traffic behavior more adequately. This makes CAC to be more efficient at the expense of additional complexity.

2.3.5 IP's Resource reSerVation Protocol (RSVP)

RFC 2205 [39] defines how Internet services utilize the RSVP protocol to reserve resources and guarantee QoS over the best-effort IP infrastructure. Contrary to ATM, RSVP employs receiver-initiated, unidirectional reservation. RSVP requires that the receiver periodically refresh reservation requests. The network may deny a refresh request or simply stop providing the reserved capacity and QoS in the middle of a flow under certain conditions, whereas ATM avoids this situation. In other words, reservations and routing operate independently in IP, while in ATM, they operate

concurrently during the connection establishment interval.

Chapter 3

Research Topic

In broadband ATM networks, a connection traverses a set of switching nodes in the network. To set up a connection, network resources must be reserved at each node in order to guarantee the contracted quality of service. Figure 3.1 shows a network example in which a connection between the end-users A and B pass through several switches. A connection is established once all the switches have enough resources to handle the connection with the requested QoS. The set of rules (or procedures) that determine admissibility of a connection in an ATM switch are commonly termed as CAC. To maximize the utilization of network resources, hence the generated revenue, without affecting the QoS, CAC scheme has to be efficient. Prediction of QoS, e.g., CLR, plays an important role in formulating an efficient CAC.

In this chapter, we state the problem, its objective and why we need to solve this problem. We break the problem into three research issues that will be addressed in the next chapters. Let us first list the assumptions that we consider throughout this thesis.

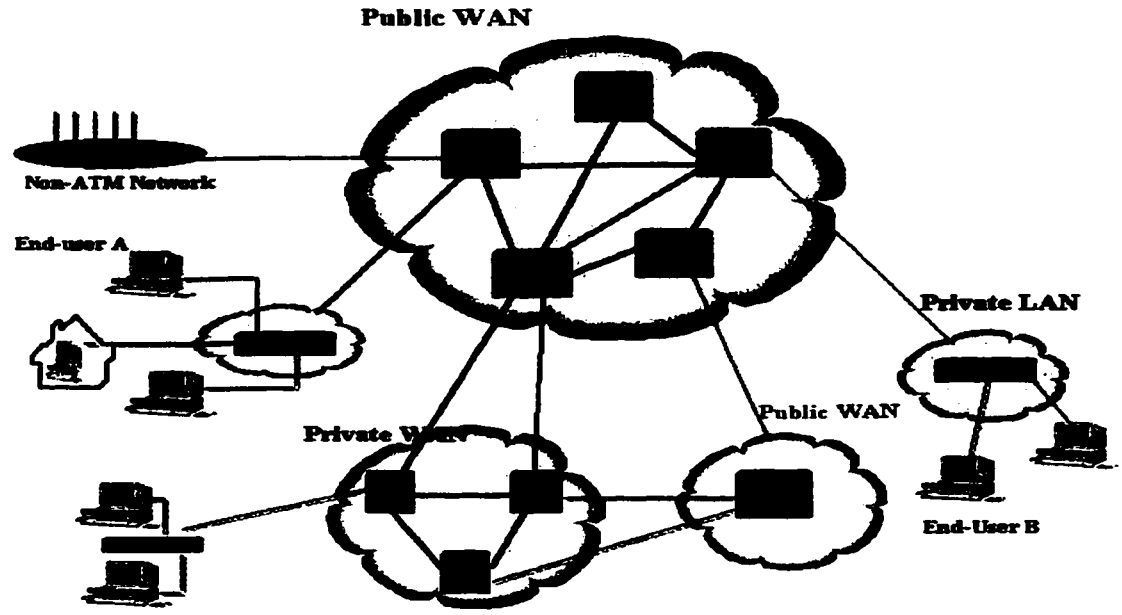

Figure 3.1: A network example.

3.1 Assumptions

We consider the following assumptions throughout the thesis:

- All the traffic sources are loss-sensitive data sources. We make this assumption as the historical patterns from 1995 to 2001 have shown that internet traffic has doubled each year, and this trend is likely to continue over the next decade [1].
- Traffic, being bursty, is queued at the output of each switch in order to allow statistical multiplexing gain.
- The CLR is the QoS of interest that needs to be maintained at an acceptable level for each connection. CAC procedure admits a connection if the CLR requirements of all the existing connections as well as that of the new one can be met.

- Classification of data sources is based on their CLR requirements, i.e., each output of a switch has a sub-queue for each CLR class. For simplicity, the proposed CAC scheme assumes that all data sources have the same CLR requirement.
- Queue scheduling scheme is FIFO.
- Queues are drained at a constant rate.
- The services that are carried by the network are *predictive*[1]. This is because the proposed CAC is a measurement based approach.

3.2 Problem Statement and the Objectives

The focus of this thesis is to approximate the CLR function for different traffic models at a node of a broadband networks. The objectives of this research is to maximize link utilization of a node through an efficient CAC scheme while maintaining the CLR of all connections at an acceptable level.

[1]Services can be categorized into three classes: *Guaranteed, probabilistic* and *predictive.* Guaranteed services require guaranteed QoS. For this service category, CAC procedure is simple. It accepts a connection if the requested peak-rate can be allocated. Probabilistic services require a tight bound on QoS. For these services, QoS-based CACs, discussed in Chapter 2, are more appropriate. Predictive services have more relaxed QoS requirements and, hence, measurement CACs can be used for these services in order to achieve higher bandwidth utilization. It has been shown that measurement-based CAC better utilize the bandwidth at the expense of relaxing the QoS requirements [40].

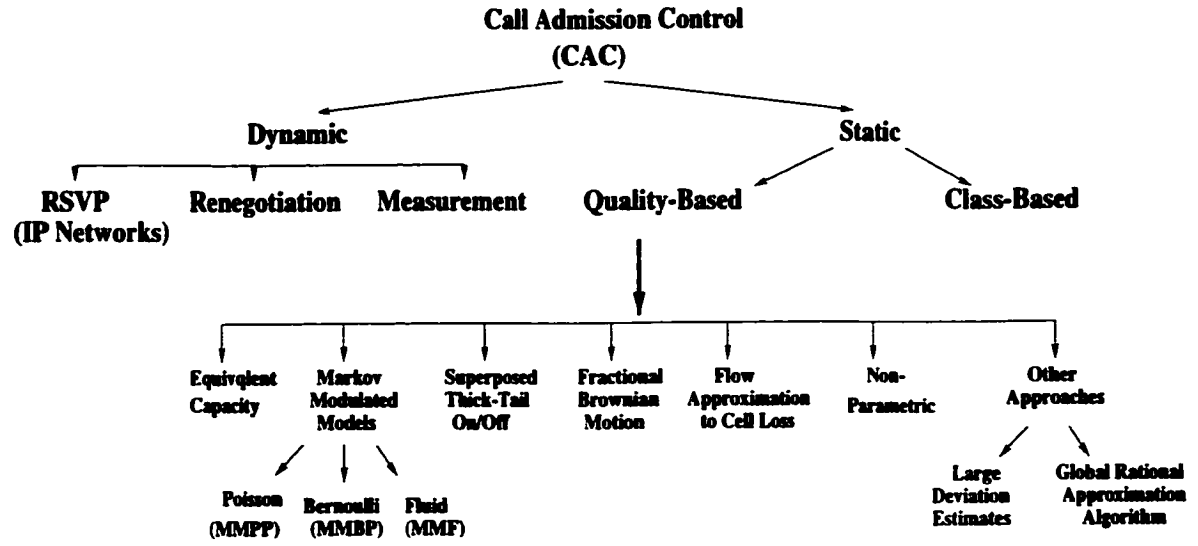

Figure 3.2: Classification of Call Admission Control (re-plotted from Chapter2).

3.3 Justification

Let us first examine the performance of the CAC schemes that were discussed in Chapter 2. Figure 3.2, which is a duplicate of Figure 2.4, shows the classification of various CAC schemes in broadband ATM and IP networks. They are classified into two main schemes: static and dynamic.

In a static scheme, the allocated bandwidth to a connection remains unchanged during the connection lifetime. The way the bandwidth is calculated divides static scheme into two main classes: Class-based and Quality-based. Class-based approaches have simplicity at connection acceptance, but require off-line computation and storage for the results. The required storage increases as the number of classes increase. The stored value may not track the real value due to the changes in the traffic patterns of a connection. Besides, the unused bandwidth from one class can not be shared by other classes. This results in a low statistical multiplexing gain. Quality-based

approaches predict the QoS metric, for example CLR, of all connections in order to decide whether to admit a new connection. Various methods such as Equivalent Capacity, Markov Modulated models, Superposed thick-tail On-Off, Fractional Brownian Motion, Non-parametric, Large deviation estimates and Global Rational Approximation methods have been used to approximate the CLR in an ATM multiplexer. All methods assume that loss probability in a finite buffer queuing system is approximated by the queue length distribution, which is considered to be asymptotically exponential (equation 2.1). In case of the Equivalent Capacity method, loss probability, P_{loss}, is approximated by $e^{-\eta x}$ and the bandwidth for each connection is calculated in isolation. The method assumes that the bandwidth of the connections can be added (see equation 2.4). This means that the Equivalent Capacity method is effective if each traffic source can be represented by a Poisson process. For variable bit-rate bursty sources, adding capacities is proven to be conservative and wasteful of network resources, since it does not fully explore the statistical multiplexing gain [18][42]. Markov Modulated Processes are rich stochastic process that model the variable bit rate traffic. In case of the Markov Modulated Poisson Process (MMPP), Choudhury *et al* [8] provided a three-exponential-term approximation that determines the tail probability of the waiting time for i.i.d On-Off sources, from which an approximation can be made for the queue tail probability. Although this approximation can be computed more easily than the exact tail computation, it suffers from computational complexity that is involved in determining asymptotic constant of α_1 (equation 2.6). Another Markovian Modulated Process is the Markov Modulated Fluid (MMF) model. The advantage of MMF over MMPP is that the numerical complexity is independent of the buffer size. However, unlike MMPP, the MMF model is unable to capture the effect of cell

variability and, hence, inaccurate for a small buffer size. Furthermore, when a large number of sources are being multiplexed, both MMPP and MMF models are too computationally complex to calculate the queue tail (or loss probability). In addition, MMPP and MMF models can represent the SRD traffic, and fail to represent LRD or self-similar traffic.

To overcome the shortcomings of Markov Modulated processes, other traffic models that are LRD and/or self-similar have been considered. For instance, Tsybakov *et al* [10] constructed an LRD and a self-similar model by superposing an infinite number of i.i.d heavily-tailed On-Off sources with Pareto sojourn time and derived a lower bound solution for buffer overflow probability that cannot decrease faster than hyperbolically as buffer size increases. However, the bound is not tight and the accuracy of the model is also questionable. Furthermore, the results discussed in [41] indicate that a 2-state Pareto Modulated Poisson Process (PMPP) source, constructed by aggregating many thick-tailed On-Off sources, does not trace actual data traffic of LAN. Another LRD and self-similar traffic model is Fractional Brownian Motion (FBM). Norros [11] has obtained an upper bound for the queue length distribution of FBM/D/1 queuing systems. The probability that queue length is greater than x is shown to be bounded below by the complementary error function which decays more slowly than the exponential form in the Markovian processes. Narayan [13] has obtained an exact expression for asymptotic queue length distribution (equation 2.11). Some experiments from actual measurements of real data show that FBM model can be used as a good approximation for LAN traffic [12]. However, there are traffic scenarios and services that have non-fractal property. Other approximate approaches in CLR prediction, such as using large deviation estimates [33][15], rely on the assumption of

arrival process, which is still unknown for many practical sources.

Unlike the previously discussed CLR prediction approaches, the Non-parametric approach [32], Flow approximation to Cell Loss approach [24], and Global Rational Approximation (GRA) algorithm [34] do not rely on the assumption of the arrival process. Both the Non-parametric approach and Flow approximation to Cell Loss use the declared parameters by source (such as peak rate, sustainable cell rate) to predict the CLR. As far as the performance is concerned, a comparison between the Non-parametric approach and Equivalent Capacity has shown that the Non-parametric approach achieves lower statistical multiplexing gain than the Equivalent Capacity [14]. This is because the Non-parametric approach is less sensitive to the buffer size and, hence, an increase in the buffer size does not result in a significant increase in the statistical multiplexing. Other sets of experiments in [14] have investigated the effect of the required CLR on the amount of statistical multiplexing. The amount of statistical multiplexing gain in the Non-parametric approach is more sensitive to the required CLR than that of the Equivalent Capacity. the Flow approximation to Cell Loss approach does not consider the impact of the buffer size in decision making and, hence, it is conservative compared to the Equivalent Capacity approach. The attractive feature of the GRA approach is that it approximates the CLR for a large-sized system using information from a small-sized system. However, if the degree of the polynomials of the numerator and denumerator is not large enough, it results in a poor approximation. In addition, the complex computation for the fitting process may encounter singularity.

Dynamic approaches, in general, try to utilize the network resources more efficiently

compared to the static ones. Renegotiation approach has not been supported by the ATM Forum because of the signaling overhead. A measurement-based approach is indeed a *dynamic* version of a non-parametric approach, as it uses real-time measurements of the ATM traffic descriptor. Although the measurement approach outperforms the non-parametric one, it still overestimates the required bandwidth. This is because the ATM traffic descriptor, i.e., the peak-rate, mean rate and burst size, cannot describe the correlation between the traffic patterns. RSVP, which has been proposed for Integrated Services over IP networks, not only is unscalable but also cannot guarantee delivering the QoS throughout the life-time of a connection.

Devising an efficient admission control that maximizes the utility of the resources remains unsolved, as all the state-of-the-art approaches for the CLR prediction tend to be conservative.

3.4 Research Issues

Accurate prediction of CLR is required in order to devise an efficient CAC. What all the previous studies have in common is the assumption of *a priori* characterization of the traffic for the convenience of mathematical tractability. Without proper characterization of the traffic, the network may adopt a conservative approach for the CLR prediction and hence underutilize the resources. It is well known that the CLR in multimedia environments depends on many unknown and unpredictable traffic parameters, such as the burst and silence lengths distributions [16] and more generally, long term correlation. Recent findings from the analysis of data traffic [17] and Variable-Bit-Rate (VBR) video [18] have shown long-range dependency in the traffic

patterns. With respect to the stated problem and the state-of-the-art approaches, there are three research issues involved:

1. **Identification**: which refers to forecasting the relevant traffic features (characteristics of traffic), or identifying *dominant* parameters of the traffic that impact the CLR. Representing these parameters as the *Traffic Indicator*, the first research issue is *identifying* the traffic indicator.

2. **Approximation**: which refers to finding a technique that approximates the CLR as a function of the traffic indicator. The technique needs to be accurate and less complex.

3. **Derivation**: which refers to finding the techniques and approaches that derive the projected traffic indicator in real time in order to perform admission control.

In the following chapters we address each one of these issues in detail.

Chapter 4

Identification of Traffic Indicator

4.1 Background

As discussed in Chapter 3, the first research question that we need to answer is to identify a set of traffic parameters that can describe the aggregate traffic to an output queue of a broadband switch. Moreover, the set should include necessary parameters that form the CLR predictor variable. Under the assumptions of Chapter 3, a solution to this issue requires analyzing the statistics of the aggregate traffic from many data sources that arrive to an output queue of an ATM switch.

4.2 Traffic Models

The traffic that arrives to an output queue of a switch is the aggregate traffic from many bursty data sources. This aggregate traffic can be represented by time series of $\mathbf{X}=\{\cdots, X_{i-1}, X_i, \cdots\}$, where X_i is the number of cells that arrive in the ith count interval, say one millisecond or one second. The aggregate traffic to a queue may represent short-term dependency, long range dependency or self-similarity in its traffic

patterns from time to time. Thus, we use a mixture of traffic models ranging from conventional ones to self-similar (or fractal) ones in order to generate the aggregate traffic. In the following subsections we discuss some of the SRD and LRD traffic models that can be used to model data and VBR sources.

4.2.1 Data Traffic Modeling

Recent studies on Local Area Network (LAN) and Wide Area Network (WAN) traffic suggest that the assumption that packet inter-arrival time is exponentially distributed is not always valid. In fact, a large number of papers (e.g., [17] and [42]-[45]) suggest that the modeling network traffic, such as Poisson processes, might result in under-estimating the burstiness of the packet arrivals and the required buffer sizes. Some WAN traffic studies [17][42] have provided evidence that the actual network traffic is self-similar, or in other words, its bursty nature is kept over a wide range of time scales. Poisson models are only valid when modeling the arrivals of TELNET or FTP user sessions but not the FTP or WWW (World Wide Web) data bursts. The following definitions for long range dependence and self similarity are adopted from the literature.

Definition 1: *A weakly stationary process (i.e., stationary as regards its second order statistics) is called "long-range dependent" if the correlation* $r(k)$[1] *between neighboring exclusive blocks does not asymptotically vanish when the block size is increased, i.e.,* $\sum_{k=1}^{\infty} r(k) = \infty$. *This correlation is zero for the Poisson process and decays exponentially towards zero for the finite space Markov processes.*

[1] Autocorrelation function $r(k)$ of an stationary process is defined as $E[X(t)X(t+k)]/E[X(t)X(t)]$, where $X(t)$ is a discrete time stochastic process or time series of number of cells at time instance t.

Definition 2: *A stochastic process X_t is called "self-similar" if it behaves up to a scaling factor in exactly the same way at all the time scales. The main parameter that characterizes a self-similar process is the Hurst parameter (H). X_t is called strictly self-similar [18], with Hurst parameter H, if:*

$$r(k) = \frac{1}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right], \qquad \text{for } 0.5 < \mathrm{H} < 1$$

and asymptotically self-similar when:

$$r(k) \approx \left[k^{2(1-H)}\right] \cdot L(k), \qquad \text{for } k \to \infty \text{ and } \lim_{t\to\infty} \frac{L(tx)}{L(x)} = 1, \quad \forall x > 0.$$

From the above definitions, it can be seen that a self-similar process is LRD as long as it has some positive correlation at all the time scales. This property is quite important to be considered in tele-traffic engineering, since LRD traffic patterns are the worst that could be expected. The fluctuations in LRD traffic are larger than those of SRD traffic and, hence, congestion build-up occurs more quickly. A lot of research has been carried out in order to develop methods that produce self-similar traffic, among them are the generation of n ($n \to \infty$) On-Off sources with heavy-tailed sojourn time distribution such as Pareto and the generation of sample paths of a fractional ARIMA [46] process. However, these approaches require a very large amount of computer processing power and take a very long time to come up with valuable results. Furthermore, results of [41] show that traffic generated by aggregation of n On-Off sources with Pareto sojourn time (known as Pareto Markov Modulated Poisson Process, PMPP) can not track the real traces of data traffic, whereas Fractional Brownian Motion (FBM) provides a closer match. FBM is a zero mean Gaussian process, $B_H(t)$, with the Hurst parameter H, and it is defined by

1. $E[B_H(t)] = 0$,

2. $B_H(0) = 0$,

3. $B_H(t+\delta) - B_H(t)$ is normally distributed, $N(0,\ \sigma|\delta|^H)$,

4. $B_H(t)$ has independent increments and

5. $E[B_H(t)B_H(s)]=\sigma^2/2\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right)$.

$B_H(t)$ is exactly self-similar, perfectly determined by H. In [31], FBM is defined to characterize the number of arrivals in the interval $(0,\ t)$:

$$N_t = mt + \sqrt{am}Z_t,$$

where m denotes the mean of the process, a is the coefficient of variation and Z_t is the normalized FBM with the Hurst parameter, H. FBM is an example of an exactly self-similar process that can be generated using fast algorithms. In the Appendix C, we detailed the approach that we used in this study to generate Fast Fractional Brownian Motion (FFBM) traffic samples.

4.2.2 VBR Traffic Sources

On-Off sources, as shown in Figure 4.1, have been widely used to model bursty traffic sources (e.g., [5][14]). With this model, traffic is switched off in the Off state, whereas in the On state traffic is generated deterministically at a constant rate A. For analytical tractability, in most literature, duration of On and Off periods has been assumed to be exponentially distributed and mutually independent with means β^{-1} and α^{-1}, respectively. Superposition of N On-Off sources results in an $N+1$ state birth-death process. The state space grows as the number of sources in the superposition is increased. As originally proposed by Heffes and Lucantoni [7], the

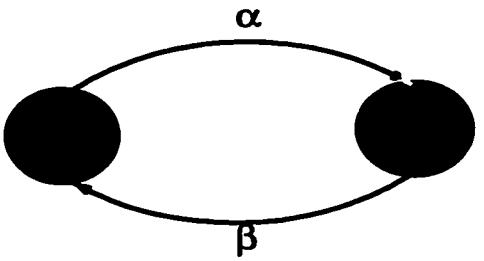

Figure 4.1: A single On-Off source that models a bursty traffic source.

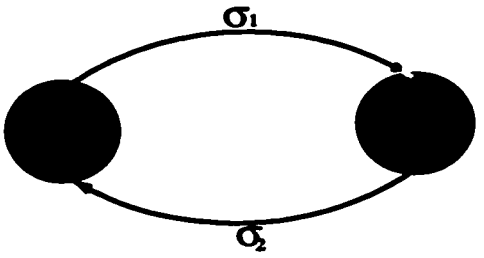

Figure 4.2: A 2-state MMPP that models aggregation of many On-Off sources.

aggregate packet arrival process from the superposition of many On-Off sources (with exponential sojourn time) may be represented by a doubly stochastic Poisson process that is modulated in a Markovian manner. They approximated the aggregate packet arrival process of N On-Off sources by a 2-state MMPP shown in Figure 4.2. The approximating MMPP model has been chosen in such a way that its statistical characteristics match those of the aggregate traffic from the On-Off sources. There are four parameters for the chosen 2-state MMPP; namely, the mean sojourn times (σ_1^{-1} and σ_2^{-1}) and the Poisson arrival rates of λ_1 and λ_2 in states 1 and 2, respectively. In Appendix B, we outlined a procedure that determines the four parameters of the MMPP from the characteristics of the original On-Off sources.

Video sources are divided into two main categories: *video conference* with uniform activity level, i.e., slow movements and *broadcast video* with more dynamic sequences.

a) **Modeling of video conferences**:

Video conferencing may be modeled by a model originally proposed by Maglaris *et al* [47]. In this model, each video source is represented by a continuous-time, discrete state Markov chain. The bit-rate from a source is quantized into M discrete levels of step-size γ. The model switches between various levels and spends exponentially distributed time at each level. Maglaris noted that the continuous time, discrete-state Markov chain, may be constructed from the superposition of M mini On-Off sources, where each mini-source is in one of the states, On or Off. Thus, superposition of video sources may also be approximated by a 2-state MMPP. The experimental results show that a two state MMPP source gives an excellent estimate of the covariance of teleconferencing the VBR video over a large number of frames [41]. Other traffic models that can represent video conferencing are Discrete Auto-Regressive (DAR) models and Gamma Beta Auto Regressive (GBAR) models. It has been shown that the DAR model does a good job of capturing the actual traces when a number of sources are multiplexed, whereas GBAR is well suited for single source video modeling, i.e. no interaction with other sources [48].

b) **Modeling of broadcast video** :

The results presented in [41] show that a simple 2-state MMPP fails to predict the covariance of highly correlated entertainment video or TV series. Moreover, Skelly [49] has found that an 8-state MMPP source can fit a TV-quality video segments. A simple DAR or GBAR model is not sufficient for the broadcast video, although the DAR model is used as a building block in a more complex model. Heyman [50] has shown that the number of bits per frame has a different autocorrelation function for the broadcast video than for video conferences. The

autocorrelation function for video conferencing decays geometrically to zero. For the broadcast video, autocorrelation function does not decay to zero. Moreover, the first frame after a scene change has significantly more bits than the other frames in the scene. Ramamurthy and Sengupta [51] observed that the correlation function declines more rapidly at the small lags than at the large lags, and that the time series can be described by a semi-Markov process, for which states have been identified by the bit rates for different types of scenes and one state for scene changes.

4.3 System Model in Traffic Indicator Identification

Figure 4.3 shows the system model that we used in order to generate various mixtures of Short Range Dependent (SRD) and Long Range Dependent (LRD) traffic that arrive to an output queue of a switch. Conventional Poisson traffic does not represent any correlation between the time series of **X**. Traffic generated by an MMPP model is SRD, whereas traffic generated by an FBM source is LRD and self-similar. Each traffic source independently sends its traffic, in terms of cells, to a finite queue. Thus, the time series of $\mathbf{X}=\{\cdots, X_{i-1}, X_i, \cdots\}$ represents the aggregate traffic to the queue. Each sample represents the total number of cells arrived in a time interval from all the sources. By changing the parameters of each model as well as the mixture ratios, we obtain different traffic profiles with different traffic characteristics. As shown in Figure 4.3, for each traffic profile, the CLR is measured for different queue sizes and the time series of **X**, and it is stored for further statistical analysis in this chapter.

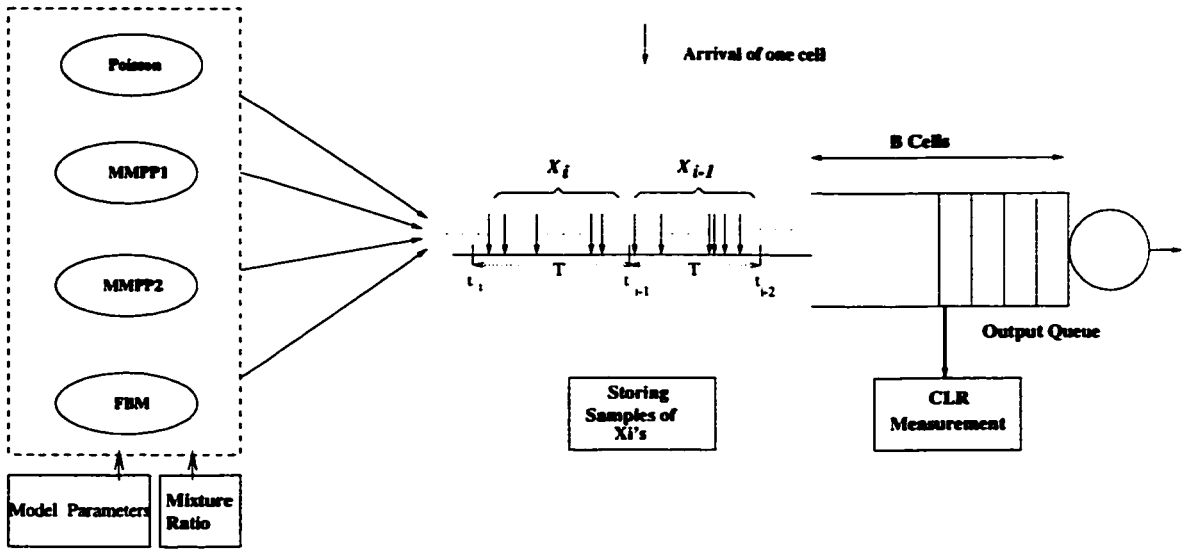

Figure 4.3: Simulation model to identify the traffic indicator.

4.4 Candidate Traffic Indicator

The commonly used parameters of $\mathbf{X}$ that may be thought to be considered are: $E[\mathbf{X}]$, $var(\mathbf{X})$ and the peak value of $\mathbf{X}$. The results reported in [53] have indicated that these parameters may be sufficient to characterize some applications, such as voice and video conference traces. However, they failed to characterize VBR and data sources. This is because $E[\mathbf{X}]$ and $var(\mathbf{X})$ are measured over a single time scale. As a result, we need to use measures that are applicable over multiple time scales. That is why the Index of Dispersion for Counts (IDC) is considered as a reliable measure of variability and it provides more information than variance, since it is calculated for multiple time scales. Let us assume Y_n is a random variable that represents the number of packets arrived in n consecutive intervals, i.e., a sample of Y_n is the sum of n samples of $\mathbf{X}$. Then, IDC is defined as [54]

$$IDC(n) = \frac{var(Y_n)}{E[Y_n]} = \frac{Var(Y_n)}{\lambda n} \qquad n \geq 1,\ 2,\ \cdots.$$

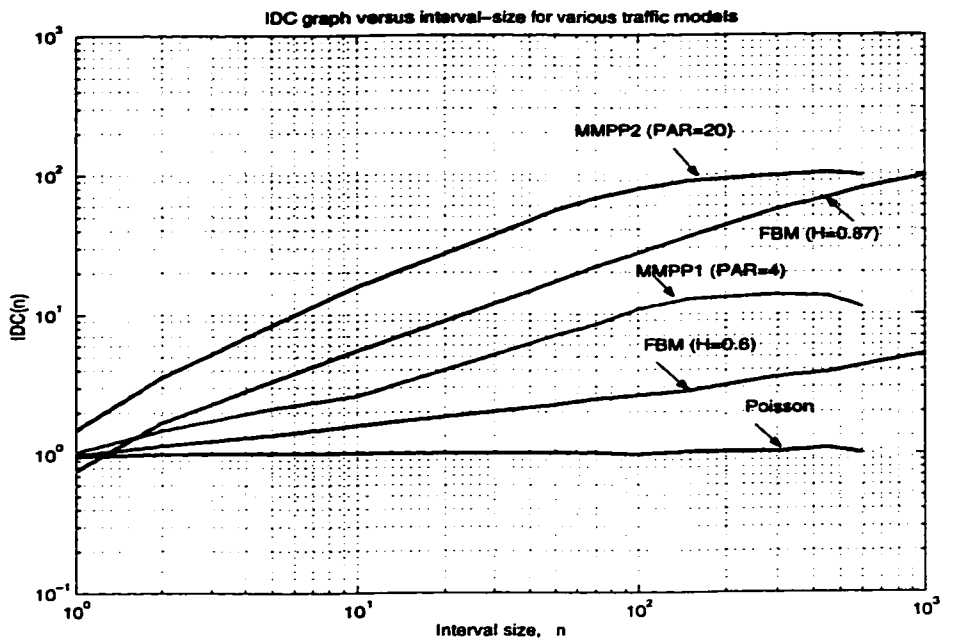

Figure 4.4: IDC curve for various traffic profiles.

For a long range dependent traffic, $Var(Y_n)$ experiences a faster growth as a function of n implying that IDC is a measure of burstiness. Figure 4.4 shows the IDC curves for Poisson, MMPP1, MMPP2 and FBM traffic models. MMPP1 and MMPP2 are two different MMPP sources with different peak-to-average ratios of 4 and 20, respectively. As seen in Figure 4.4, for a Poisson source IDC is a constant, i.e., showing no variability. For both MMPP1 and MMPP2 sources, the IDC curves reach a plateau at large values of n (or time), which indicates that the MMPP traffic is SRD. For FBM source IDC is increasing monotonically, which indicates that the FBM traffic is LRD. The IDC curves for the two plotted FBM sources are associated with two different burstiness levels (i.e., Hurst parameters).

Since time varying traffic arriving to a queue may represent different variability from time to time, we are interested in identifying a set of parameters from the IDC graph

that can describe the traffic behavior. Let us define

$$K \approx (1+s)/2$$

where s is the slope of the linear part of the IDC curve in a logarithmic scale, i.e.,

$$s \approx \lim_{(n_2-n_1)\to 0} \frac{\log(IDC(n_2)) - \log(IDC(n_1))}{\log(n_2) - \log(n_1)} \qquad 100 > n_2 > n_1 > 1$$

and

$$K = H \quad \text{for FBM traffic}$$

The value of K is between 0.5 and 1. The higher the value of K, the burstier the traffic. For conventional Poisson traffic, K is 0.5. For MMPP1 and MMPP2 sources, K equals 0.7 and 0.9, respectively.

Aside from the slope of IDC, there are two other parameters from the graph that may describe the behavior of a traffic profile. One is the initial value of IDC, i.e., IDC(1), and the other is the value of IDC calculated at a longer time interval L, say 300 time units. $IDC(1)$ is related to the second moment of traffic as

$$IDC(1) = \frac{E[\mathbf{X}^2] - E^2[\mathbf{X}]}{E[\mathbf{X}]},$$

whereas IDC(L) shows the amount of burstiness evaluated at the dependency interval L. It is given by:

$$IDC(L) = \frac{var(\mathbf{Y_L})}{E(\mathbf{Y_L})}, \quad L \geq 1,\ 2,\ \cdots, \tag{4.1}$$

$$\mathbf{Y_L} = \{\cdots,\ Y_L^{(k-1)},\ Y_L^{(k)},\ Y_L^{(k+1)},\ \cdots\}, \tag{4.2}$$

$$Y_L^{(k)} = \sum_{i=(k-1)L+1}^{kL} X_i, \quad k \geq 1,\ 2,\ \cdots \tag{4.3}$$

The slope of IDC curve as well as IDC(1) and IDC(L) change as the traffic becomes a mixture of various traffic models. For instance, Figures 4.5 and 4.6 show the IDC

curves for a mixture of Poisson sources with MMPP1 and MMPP2, respectively. Figure 4.7 shows the IDC curve of the aggregate traffic of a mixture of MMPP1 and MMPP2. Figures 4.8, 4.9, 4.10 and 4.11 show the IDC curves for the mixtures of MMPP1 and FBM (H=0.6), MMPP1 and FBM (H=0.8), MMPP1 and FBM (H=0.9) and MMPP2 with FBM (H=0.9), respectively.

It can be observed from Figures 4.7 to 4.11 that when the traffic profile changes, the IDC slope, IDC(1) and IDC(L) change. As a result, in addition to traffic load, we consider IDC(1), IDC(L) and K (which is related to the slope of the linear portion of the IDC curve), as the potential parameters that may describe a traffic mixture. The next question is to determine which one of these parameters impacts the CLR.

4.4.1 Effect of the Traffic Indicator Parameters in the CLR

To determine which one of the parameters of the traffic indicator impact the CLR, we use the model shown in Figure 4.3 for the CLR measurement. Traffic generated from a mixture of traffic models is applied to a queue that is served at a constant rate. We measure the CLR[2] versus the buffer size for each traffic mix scenario. The ratio of the arrival rate of the aggregate traffic, in terms of cells/sec, to the service rate is defined as traffic load.

Figures 4.12 and 4.13 show the measured CLR versus buffer size (in terms of cells) for MMPP1 and MMPP2, respectively, for different traffic loads. As seen, traffic load has significant impact on the CLR. On the other hand, when the traffic load is high and the traffic source is highly bursty, e.g., MMPP2, buffer size does not have any

[2]The confidence interval for all CLR measurements is 95%.

significant impact on the CLR. Let us consider the case where traffic load is constant and the mix ratio is changing. Figure 4.14 shows the CLR versus buffer size for various mix scenarios of MMPP1 and MMPP2 sources. Examining this curve in conjunction with IDC curves of Figure 4.7 reveals that the CLR is also a function of IDC slope and/or IDC(L). On the other hand, for a mixture of MMPP1 and FBM (H=0.8) the changes in the IDC slope are insignificant (see Figure 4.9). Thus the CLR changes, shown in Figure 4.15, are due to the value of IDC(L). Now, let us examine the IDC and the CLR of the mixture of MMPP2 and FBM. Figure 4.11 shows the IDC curve and Figure 4.16 shows the CLR at traffic load of 0.9. As seen, the IDC slope and the value of IDC(L) have both impacted the CLR. As a result of the above experiments, we consider an indicator of traffic that consists of four parameters: Load, K, IDC(1), and IDC(L), i.e.,

$$\underline{\mathbf{I}_G} = \{Load,\ IDC(1),\ K,\ IDC(L)\},$$

where $\underline{\mathbf{I}_G}$ is the global indicator of traffic that we propose as a CLR prediction variable. The next questions are:

how can we assess the performance of such a traffic indicator?

how can we predict the CLR using this indicator?

These questions are answered in the next chapter.

4.5 Confidence Interval for CLR Measurements

As the time series $\mathbf{X}$ generated by a traffic profile is a stochastic process, the question that is raised is how good is the measurement of the CLR? Since the true value of CLR is unknown, one cannot compare it to the correct answer. However, what

we can do is to postulate an interval, as confidence interval, and make a probabilistic statement about the chance that CLR is covered by that interval. Let random variable S represent the measured CLR for a given traffic load at a given buffer size. Using the following theorem from the literature, we can estimate $E(S)$ by $\overline{S}$.

Theorem: The estimator

$$\overline{S} = \frac{1}{n}(S_1 + S_2 + \cdots + S_n)$$

has the following properties

- (a) $E(\overline{S}) = E(S) = \mu$,
- (b) $var(S) = \sigma^2/n$,
- (c) For n sufficiently large $\overline{S}$ is approximately normal with the above mean and variance.

Parts (a) and (b) can be proved using the estimation theory discussed in the literature (e.g., [55]) and part (c) follows from the central limit theorem.

By the above theorem, CLR is normally distributed, i.e. CLR $\in N(\mu, \sigma/n^{0.5})$. Then,

$$Pr(-z_{1-\alpha/2} \leq \frac{\overline{S} - \mu}{\sigma/n^{0.5}} \leq z_{1-\alpha/2}) = 1 - \alpha, \tag{4.4}$$

where $(1 - \alpha)$ is the probability that a random variable from $Z \in N(0,\ 1)$ will fall in the interval $(-z_{1-\alpha/2},\ z_{1-\alpha/2})$. Equation (4.4) can be manipulated to the form

$$Pr(\overline{S} - (\alpha/n^{0.5})z_{1-\alpha/2} < \mu < \overline{S} + (\alpha/n^{0.5})z_{1-\alpha/2}) = 1 - \alpha. \tag{4.5}$$

Assuming

$$\begin{aligned} L &= \overline{S} - (\alpha/n^{0.5})z_{1-\alpha/2}, \\ U &= \overline{S} + (\alpha/n^{0.5})z_{1-\alpha/2}, \end{aligned} \tag{4.6}$$

equation (4.5) states that the probability that the random interval (L, U) contains μ is $(1 - \alpha)$.

In the following we obtain the confidence interval for the measured CLR of the three traffic models that we use in this thesis, i.e., MMPP1, MMPP2 and FBM. For every CLR of a traffic profile at a given buffer size, the confidence interval is calculated using $n = 10$ samples of CLR measurements each obtained by a different seed in the generation of the time series $\mathbf{X}$ for the specified traffic profile. It is noted that in each one of the n runs about 20×10^6 cells are generated.

Case 1: Confidence Interval for the CLR measurement of FBM

Let us consider an FBM source with Hurst parameter $H = 0.8$. Figure 4.17 shows the CLR measurements with the confidence coefficient of 99% at load 0.7, 0.75 and 0.8. The confidence intervals have been calculated using equation (4.6) with $n = 10$ samples, each obtained from different seed value in the generation of time series $\mathbf{X}$. Figure 4.18 shows the measured CLR and the confidence interval for load 0.9 and 0.95 with 99% confidence coefficient.

Case 2: Confidence Interval for the CLR measurement of MMPP1 Source

Figure 4.19 shows the CLR measurements with the confidence coefficient of 99% for MMPP1 traffic source. Similar to FBM case, the interval has been calculated using $n = 10$ measured samples. At higher traffic load, the variance of the measured n samples is considerably low. As a result, the confidence interval is smaller.

Case 3: Confidence Interval for the CLR measurement of MMPP2 Source

The CLR measurements of MMPP2 source with confidence coefficient of 99% has been ploted in Figures 4.20 and 4.21. Similar to the cases 1 and 2, the coffidence interval for each measured CLR is calculated using $n = 10$ samples of the random variable that represent the CLR for the given traffic load at the given buffer size.

4.6 Summary

In this chapter we first generated various traffic profiles by mixing both SRD and LRD traffic and then applied them to a finite FIFO queue that is served at a constant rate. For each traffic profile, we obtained the IDC curve versus interval-size and the measured CLR versus buffer size. Examining both the IDC and the CLR curves for various mixtures of traffic reveals that the traffic indicator should include Load, which is related to the first moment, IDC(1), which is related to the second moment, K, which is related to the IDC slope, and IDC(L), which represents the degree of burstiness evaluated at the dependency interval L. In the next chapter we examine the performance of this indicator in approximating the CLR function.

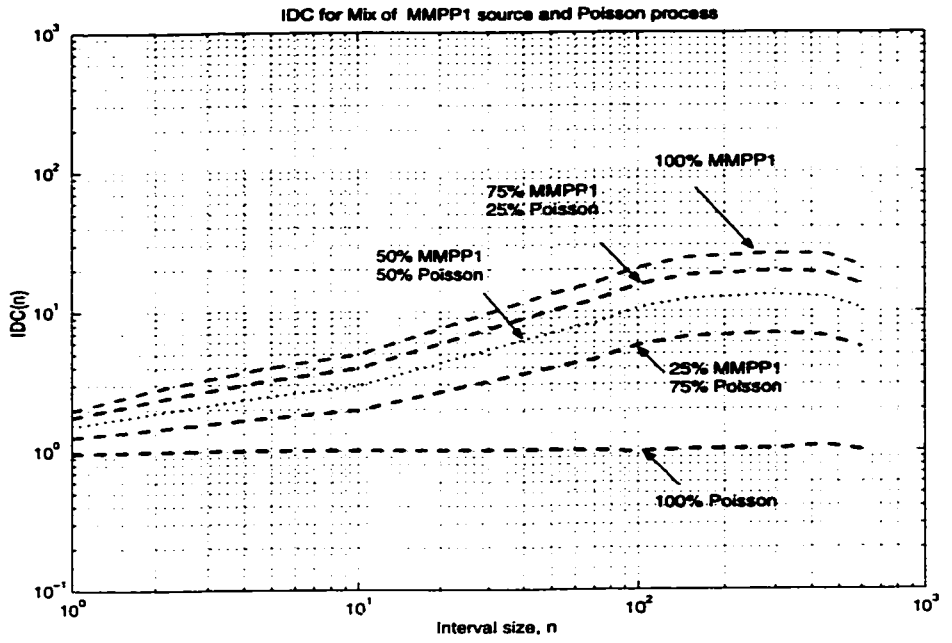

Figure 4.5: IDC curve for various mixtures of Poisson and MMPP1 sources.

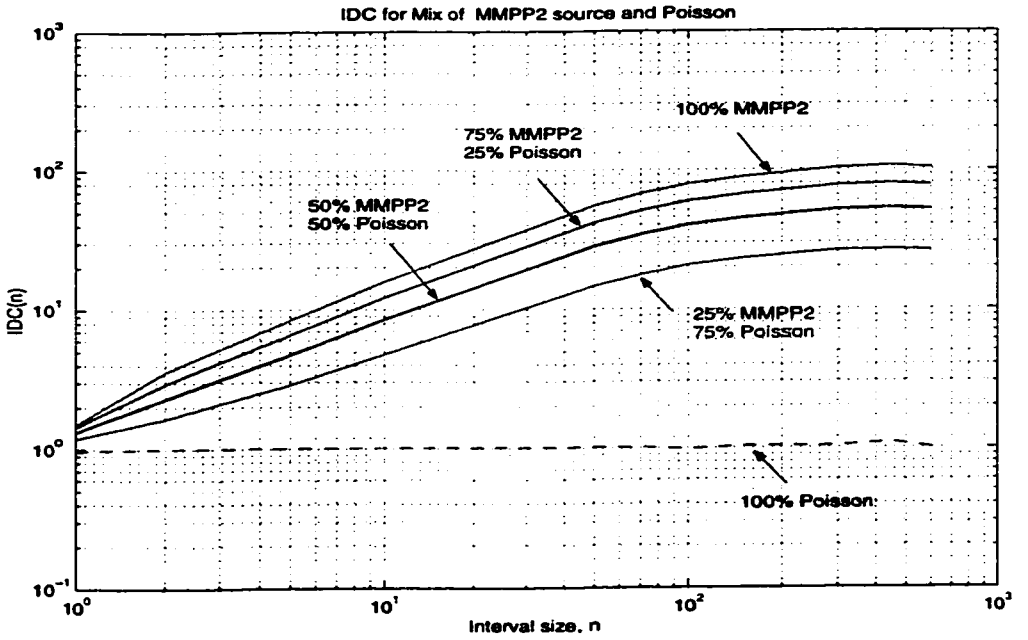

Figure 4.6: IDC curve for various mixtures of Poisson and MMPP2 sources.

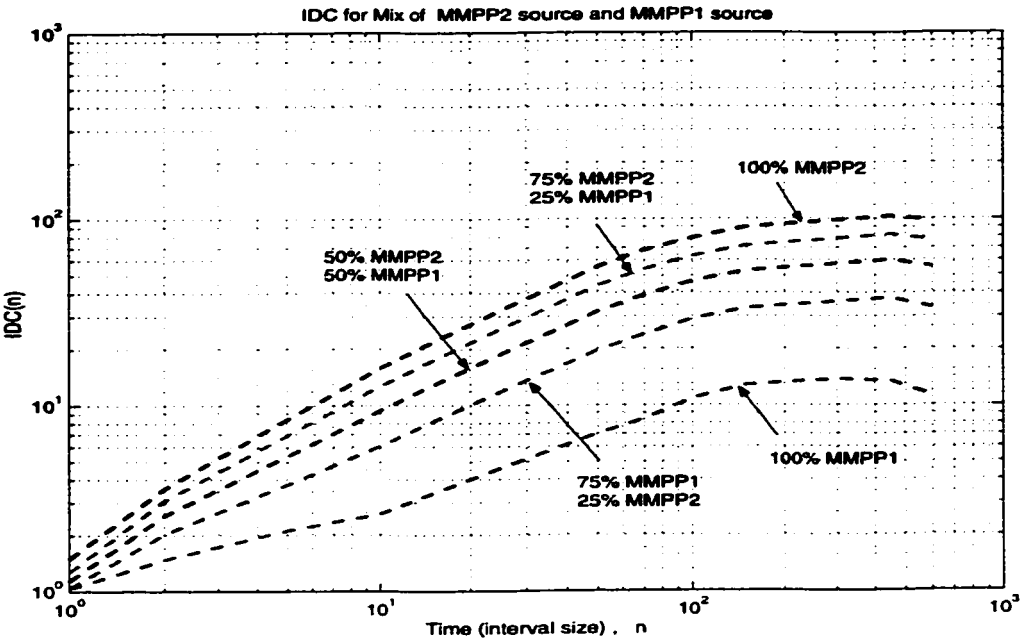

Figure 4.7: IDC curve for various mixtures of MMPP1 and MMPP2 sources.

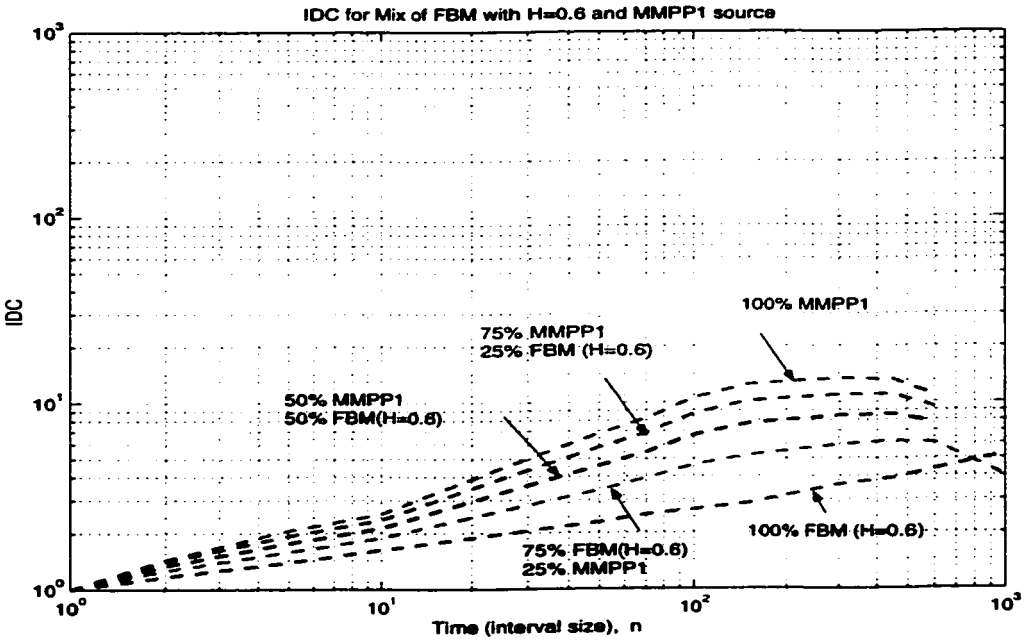

Figure 4.8: IDC curve for various mixtures of MMPP1 and FBM (H=0.6) sources.

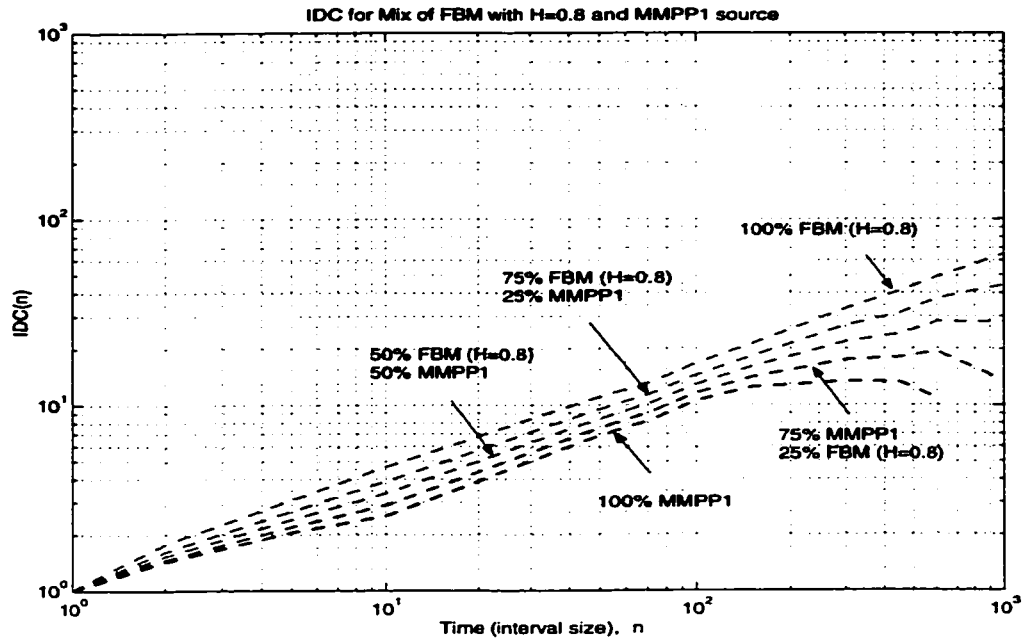

Figure 4.9: IDC curve for various mixtures of MMPP1 and FBM (H=0.8) sources.

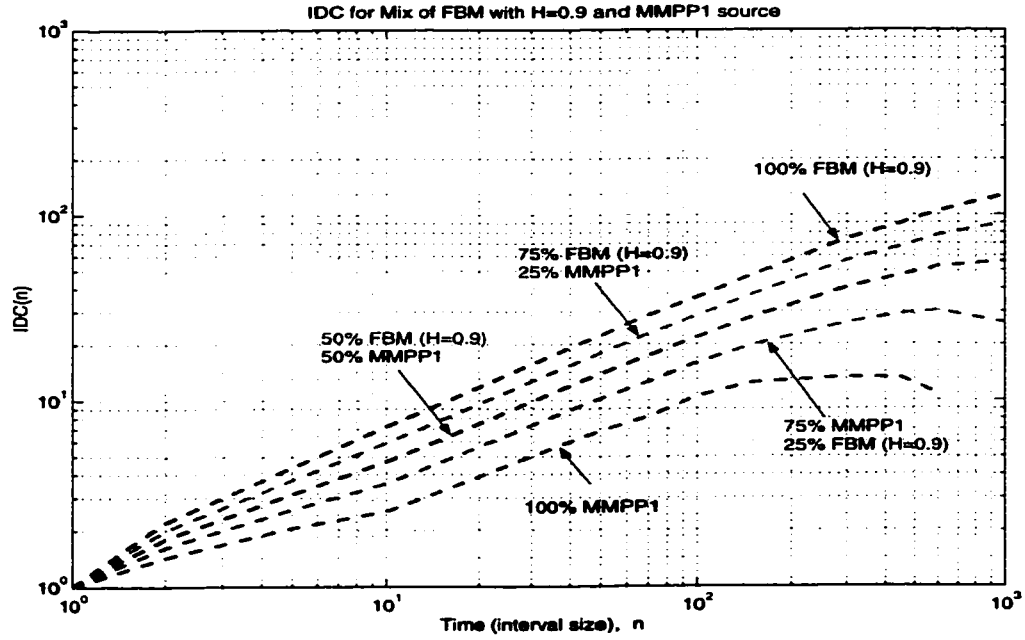

Figure 4.10: IDC curve for various mixtures of MMPP1 and FBM (H=0.9) sources.

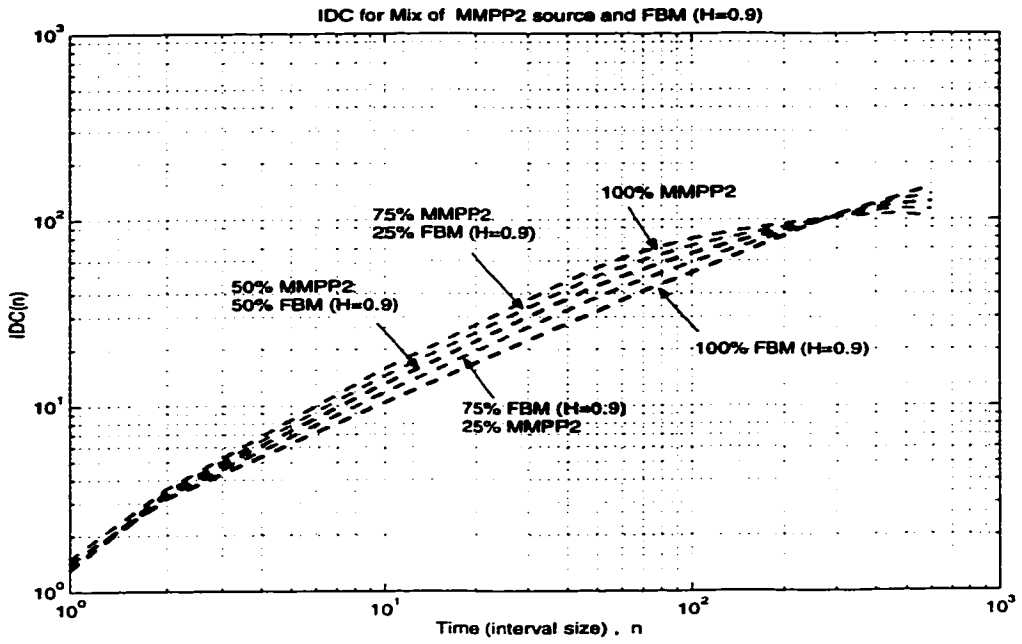

Figure 4.11: IDC curve for various mixtures of MMPP2 and FBM (H=0.9) sources.

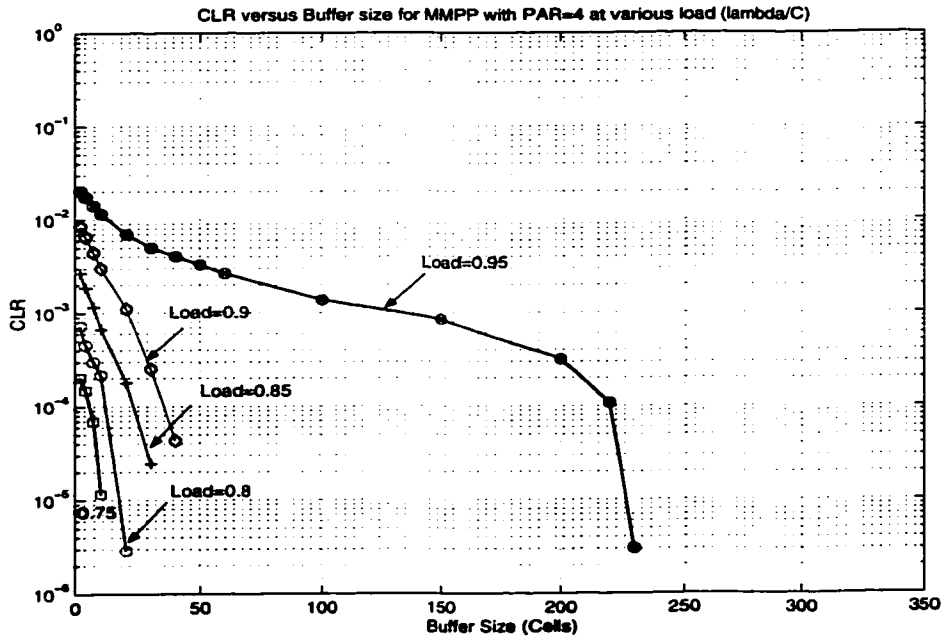

Figure 4.12: Measured CLR versus buffer size for MMPP1 source.

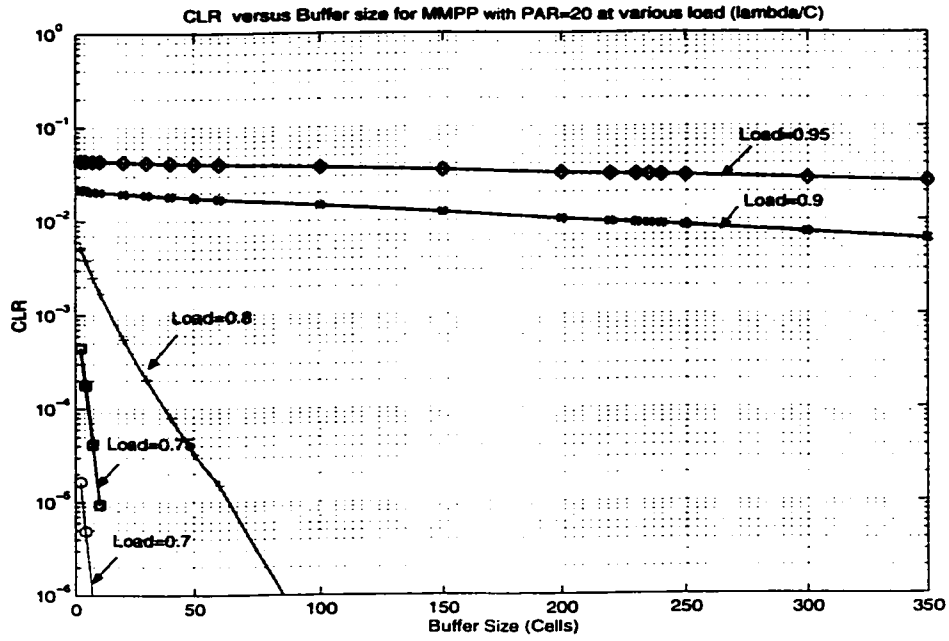

Figure 4.13: Measured CLR versus buffer size for MMPP2 source.

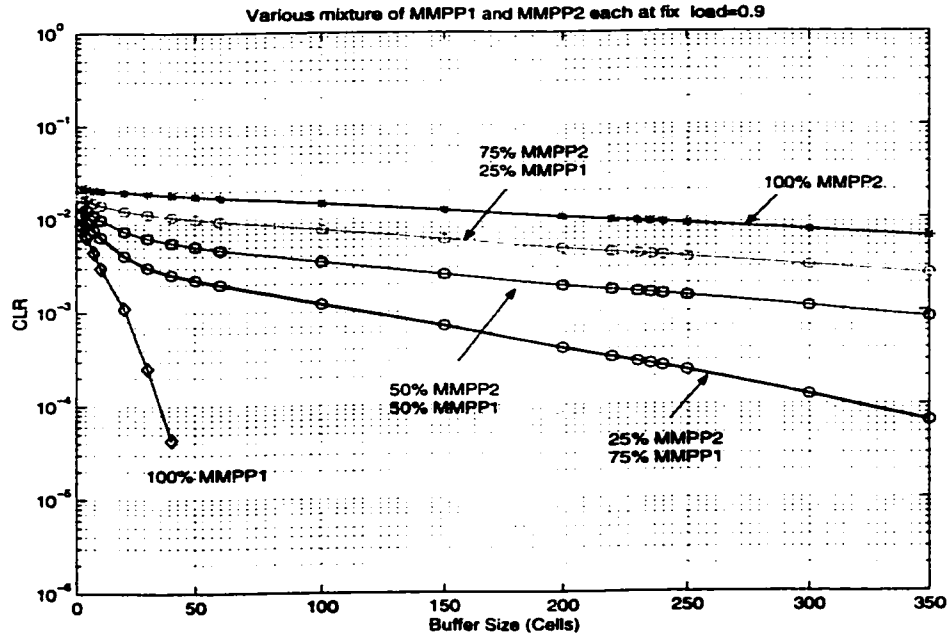

Figure 4.14: Measured CLR versus buffer size for various mixtures of MMPP1 and MMPP2 sources.

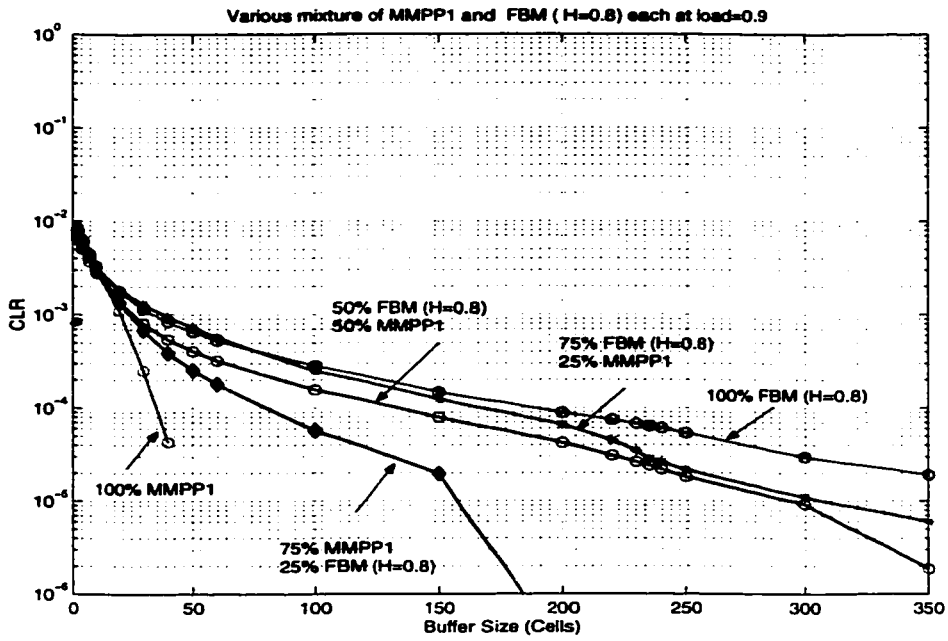

Figure 4.15: Measured CLR versus buffer size for various mixtures of MMPP1 and FBM (H=0.8) sources.

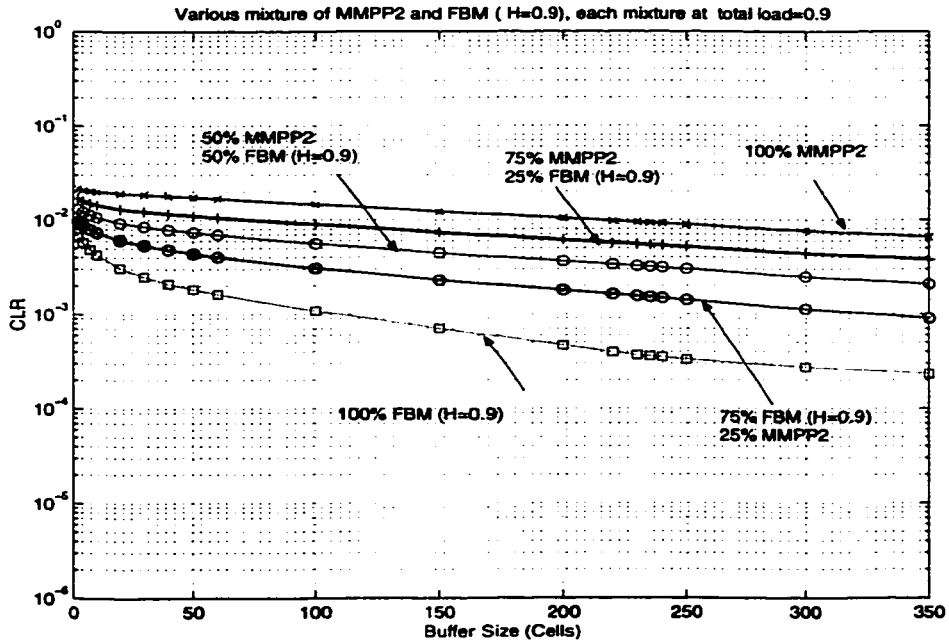

Figure 4.16: Measured CLR versus buffer size for various mextures of MMPP2 and FBM (H=0.9) sources at load=0.9.

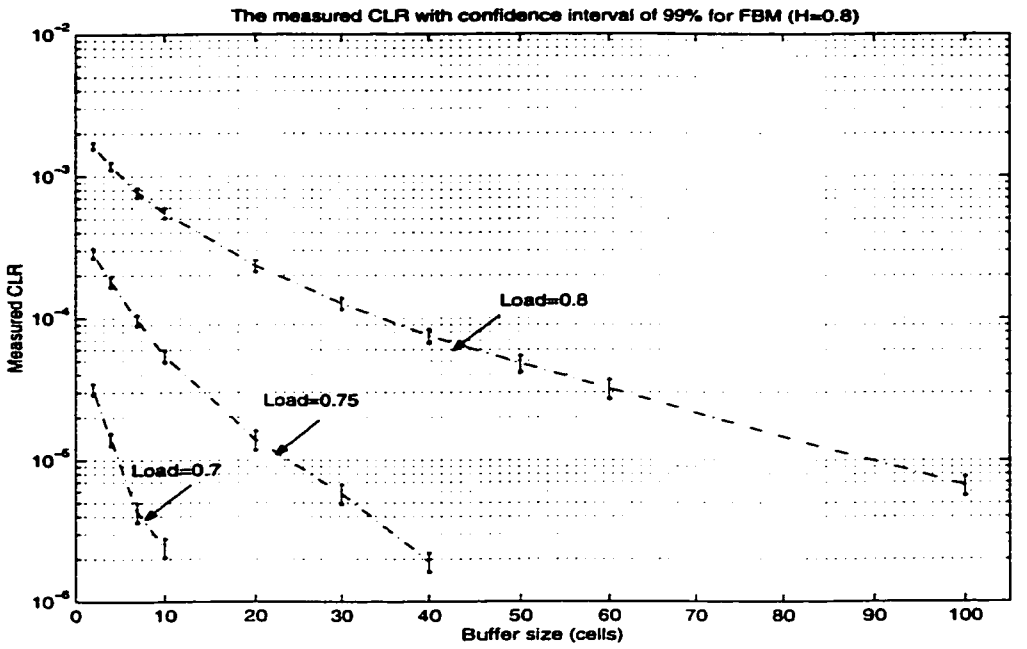

Figure 4.17: The measured CLR versus buffer size with confidence coefficient of 99% for FBM traffic (H=0.8) at different traffic load.

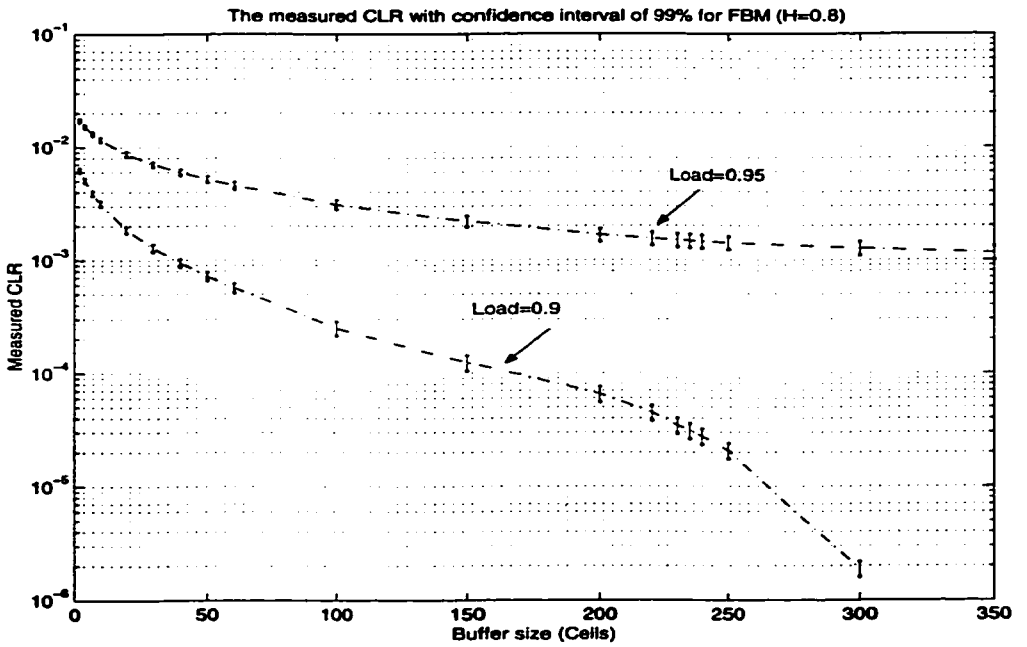

Figure 4.18: The measured CLR with confidence coefficient of 99% for FBM traffic (H=0.8) for traffic load of 0.9 and 0.95.

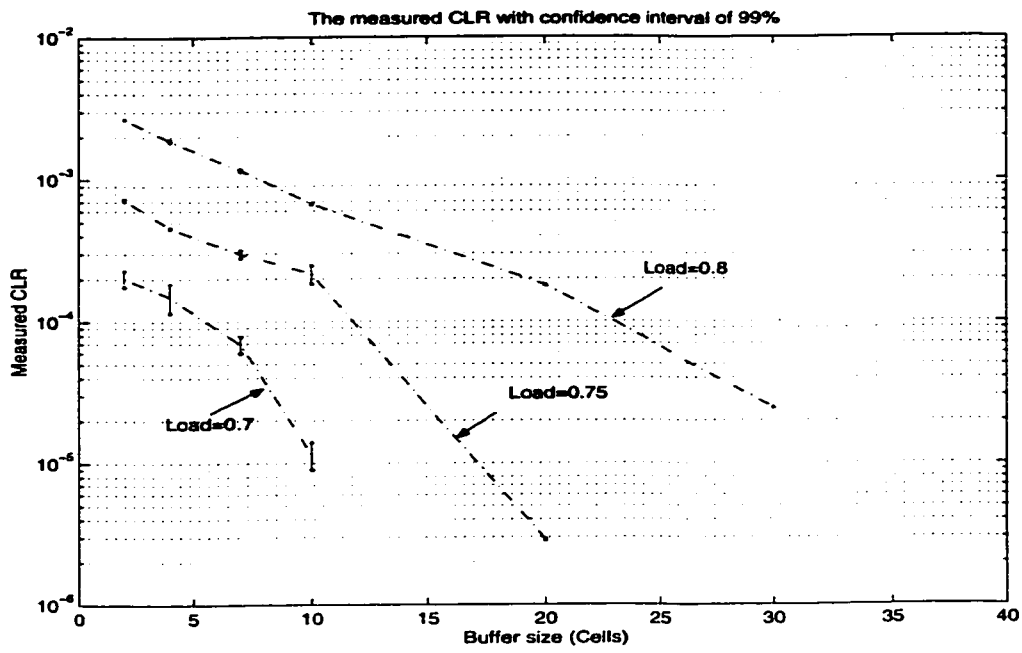

Figure 4.19: The measured CLR with confidence coefficient of 99% for MMPP1 source.

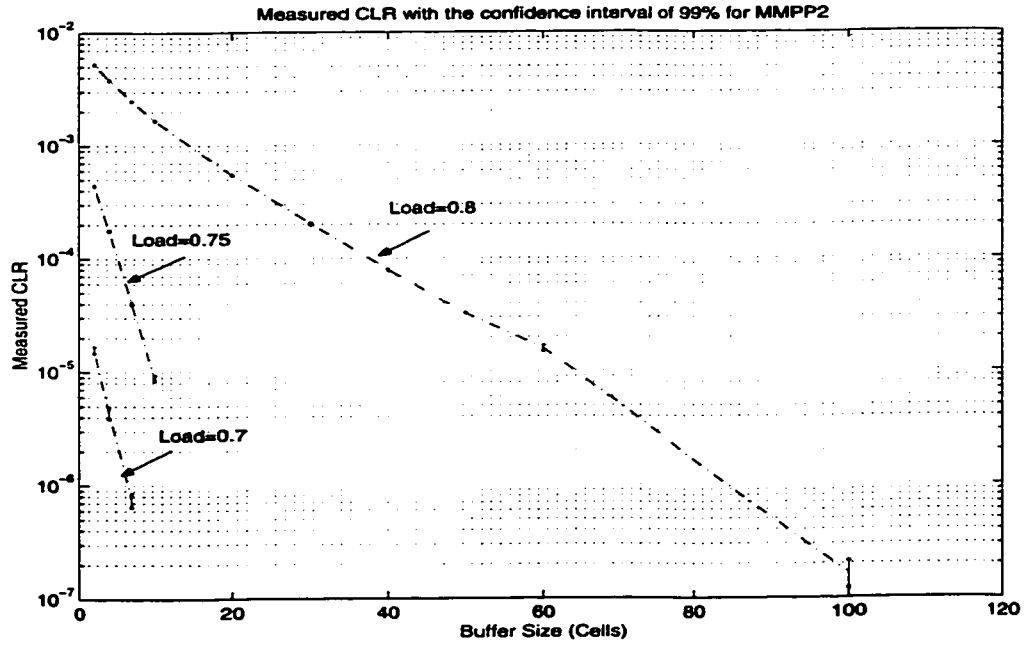

Figure 4.20: The measured CLR with confidence coefficient of 99% for MMPP2 traffic for traffic load of 0.9 and 0.95.

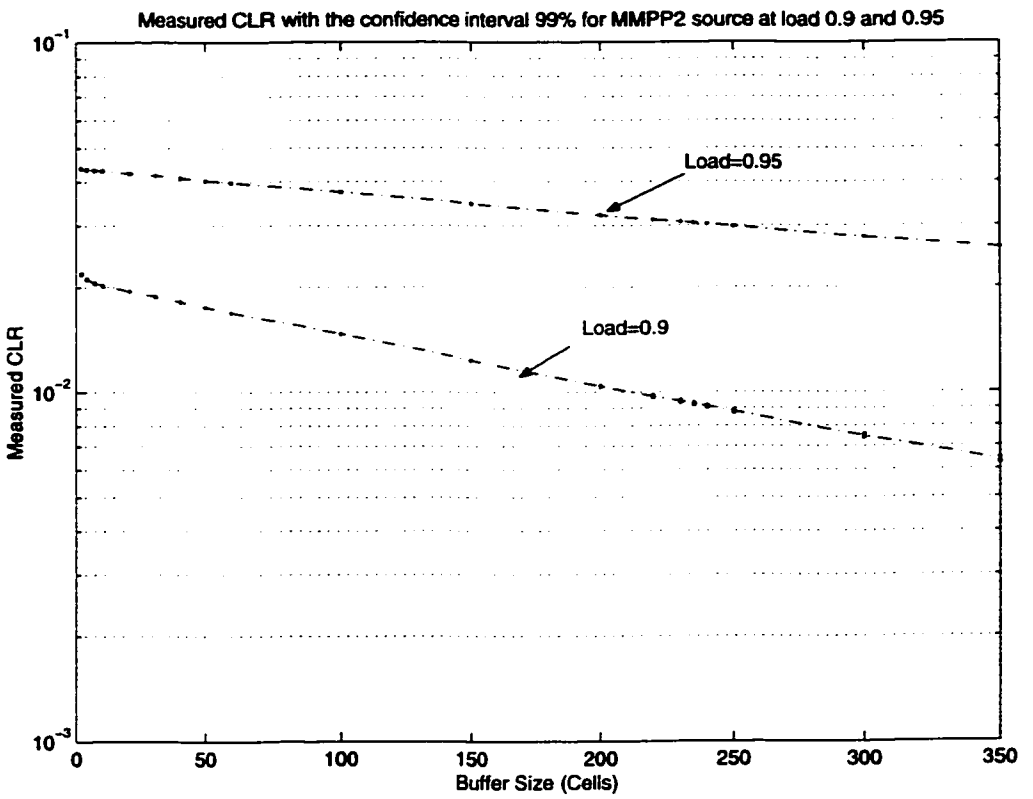

Figure 4.21: The measured CLR with confidence coefficient of 99% for MMPP2 traffic for traffic load of 0.9 and 0.95. As the variance of each measurement is small the confidence interval is small.

Chapter 5

CLR Approximation Using Traffic Indicator

5.1 Background

As discussed in the previous chapters, in the literature CLR has been approximated by $P(Q > x)$, the probability that queue exceeds x. There are two main shortcomings associated to this approximation. First, CLR assumes a finite buffer space whereas $P(Q > x)$ assumes an infinite buffer space. Second, the derived queue overflow probability assumes a traffic model, for the convenience of mathematical tractability, that may not follow the real traffic traces. For instance, the derived $P(Q > x)$ [52] based on a Poisson model underestimates the CLR when data sources are bursty. When an MMPP source is used to model one or many data sources that are SRD, the analytical result of $P(Q > x)$ overestimates the CLR. For superposed tick-tail On-Off sources, the derived $P(Q > x)$ only provides a lower bound for CLR [10], while using the analytical results for the FBM traffic model provides an upper bound for CLR

[13][31]. Table 5.1 summarizes the results of various approaches.

In searching to obtain an approach that closely approximates the CLR, in Chapter 5 we identified a set of parameters that impact the CLR and they can be used as indicator of traffic. The candidate traffic indicator consists of four parameters: Load, which is related to the first moment of traffic, IDC(1), which is related to the second moment of traffic, K, which is related to the slope of IDC curve versus time, and IDC(L), which is the IDC value at the dependency interval L. In this chapter, we propose an approach that uses traffic indicator for CLR approximation [97]. Since approximating CLR using a minimum number of parameters is desirable, we also consider traffic indicators that contain a subsut of parameters that were identified in Chapter4. We show that all the parameters of the set is required for a close CLR approximation.

5.2 A Proposed Simple Approach in CLR Approximation

The problem of CLR approximation as a function of the traffic indicator can be abstracted as a task of approximating an unknown function from a set of input-output pairs, say, $\mathcal{T}=\{(\underline{\mathbf{I}}_G^p,\ P_{loss}^p);\ p=1,\ \cdots,\ P\}$, where P is the number of pairs. It is hypothesized that an input vector $\underline{\mathbf{I}}_G^p=\{Load^p,\ IDC^p(1),\ K^p\ IDC^p(L)\}$ and an output sample P_{loss}^p are related by an unknown function f

$$P_{loss}^p = f(\underline{\mathbf{I}}_G^p,\ B^p) + e^p \quad \forall p$$

Model	CLR Approximation	Comments
Poisson	$P(Q>x)=1-(1-\rho)\sum_{n=0}^{x}\frac{\rho^n(n-x)^n}{n!}e^{-\rho(n-x)} \qquad 0\le\rho\le 1$	Underestimation
MMPP	$P(Q>x)=P(W>t), \qquad x=t/h$ $P(W \ge t) \approx \alpha_1 e^{-\eta_1 t} + \alpha_2 e^{-\eta_2 t} + \alpha_3 e^{-\eta_3 t}$	Complex & limitation of results
Superposed On-Off (Pareto)	$P(Q>x) \ge \frac{c}{\alpha(\alpha-1)(E_c+E_k)^2}\left(\frac{x+\alpha+1}{\alpha}\right)^{\alpha+1}\sum_{n=1}^{\infty} n^{-\alpha}$ $E_k=(1-e^{-\lambda_1})^{-1}-1 \qquad \alpha=(E_c+E_k)^{-1}-1 \qquad E_c=\frac{\sum_{n=1}^{\infty} n^{-\alpha-1}}{\sum_{n=1}^{\infty} n^{-\alpha}}$	Lower-bound
FBM	$P(Q>x)=\left[\frac{2^{1-H^{-1}} d_0^{H^{-1}-2}}{\sqrt{H-H^2}}\right]\theta \exp\left[(x+st_0)^2/2t_0^{2H}\right]$ $d_0=\frac{x+st_0}{t_0^H} \qquad t_0=\frac{xH}{s(1-H)} \qquad s=1-\rho$	Asymptotic Result
	$P(Q>x) \approx \exp\left\{-\frac{(C-m)^{2H}}{2k^2(H).IDC(1).m}x^{2-2H}\right\}$ where $k(H)=H^H(1-H)^{1-H}$	Upper-bond Result

Table 5.1: A summary of the state-of-the-art CLR approximation for various models.

where B^p is the buffer size and e^p is a random vector due to the imprecise measurement of P_{loss}^p and $\underline{\Gamma_G^p}$. The task of function approximation is to find $\hat{f}$, an approximation of f, such that the metric of an approximation error, e.g., Mean Squared Error (MSE) is minimized.

In a classical approximation theory, arbitrary non-linear functions are approximated by a linear combination of nonlinear basis functions. As examples of such classical methods, Spline fitting, projection-based approximation and Fourier methods can be mentioned. On the other hand, classical techniques for approximation can be embedded naturally in the computational framework of a multi-layer feed-forward network

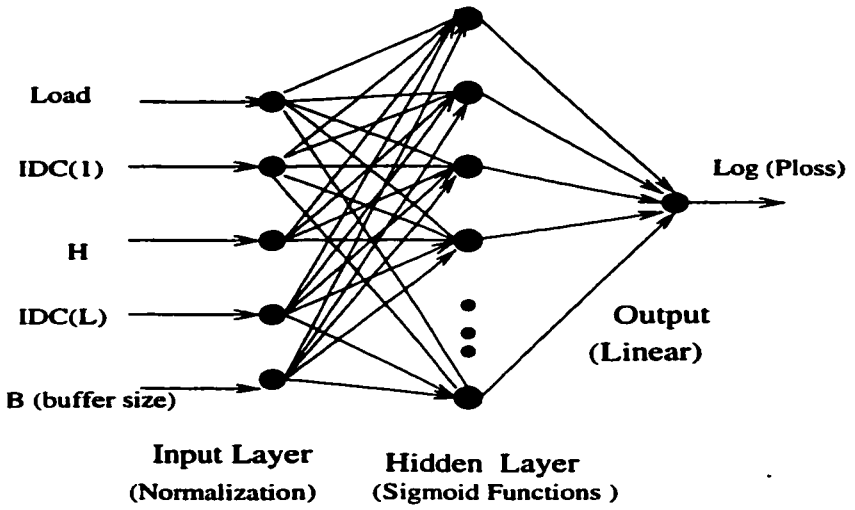

Figure 5.1: A three layer feed-forward neural network architecture (superposition of sigmoid functions) that approximates CLR (output) as a function of the traffic indicator.

as shown in Figure 5.1. It has been demonstrated [56] both analytically and constructively that function representation by multilayer feed-forward neural networks is more general since it includes many representation schemes of the classical approximation methods.

Figure 5.1 shows a three layer feed-forward neural networks system that we use in this case study. The three layers are: input, hidden and output layers. There is no processing neuron (node) at the input layer. The middle layer in Figure 5.1 is called hidden layer as its outputs are not directly accessible. Each hidden layer node simply implements one basis function, and the output layer (or node) linearly combine the outputs of the hidden layer nodes. For a given traffic indicator, say $\underline{\mathbf{I}_G^j}$= $\{Load^j,\ IDC^j(1),\ K^j\ IDC^j(L)\}$, the output of a hidden layer node, say ith node, is

given by

$$O_i = h(W_1 Load + W2IDC(1) + W3K + W4IDC(L) + W5B)$$

where

$$\mathbf{W} = [W_1\ W_2\ W_3\ W_4\ W_5]$$

is the weight vector and

$$\mathbf{Y} = [Load\ IDC(1)\ K\ IDC(L)\ B]$$

is the input vector. The function $h(\mathbf{WY}^t)$ is often referred as an *activation* function. Its domain is the set of activation rules, *net*, of the neuron model. Each neuron (node) in the hidden layer performs the operation of summation of its weighted input, or the scalar product computation to obtain *net*. Subsequently, it performs the nonlinear operation *h(net)* through its activation function. The activation function that we used is a sigmoid function defined as

$$h(net) = \frac{1}{1 - e^{z \cdot net + b}} \tag{5.1}$$

where $z > 0$ is propositioned to the neuron gain determining the steepness of the continuous function $h(net)$ near *net=0* and b is the bias value of neuron.

The reasons that we select a single hidden layer feed-forward neural network with the sigmoid activation function (as the basis function) to approximate the CLR function are:

1. It has been proven (see the Appendix D) that a linear combination of sigmoid functions can approximate any continuous function as long as there is no limit on the number of sigmoid terms.

2. Examining Table 5.1 reveals the existence of exponent terms in the CLR function. This means that a solution to CLR approximation not only needs to capture the exponential term (as in the case of Poisson and MMPP [8]), hyperbolic term (as in the case of superposed thick-tail on-off sources [10]) and e^{-x^2} term (as in the case of FBM [13][31]), but also needs to capture a general solution for a mixed traffic that is identified by a traffic indicator. As we will see in the next subsection, a single hidden layer feed-forward neural network with sigmoid activation function is capable of capturing the CLR function.

5.3 Performance Evaluation of the Proposed Approach

Figure 5.2 shows the model that we used to collect the statistics of the measured CLR and the measured indicator of traffic from the collected time series of $\mathbf{X}$. Two sets of samples were collected: one for training, i.e., adjusting the weights of the system of Figure 5.1, and the other one for testing the performance of the approximated function.

In Chapter 4 we determined a set of parameters, which can form the traffic indicator, is $\{Load, IDC(1), K, IDC(L)\}$. To assess the effectiveness of each one of these parameters, we determine the amount of improvement that each one can introduce in approximating CLR. The performance metric is MSE in approximation.

It is clear that the first and second moments of traffic (which are related to load and IDC(1)) need to be considered. We consider the following four cases (as shown in Table 5.2) derived by including and/or excluding $IDC(L)$ and H. For each case, the

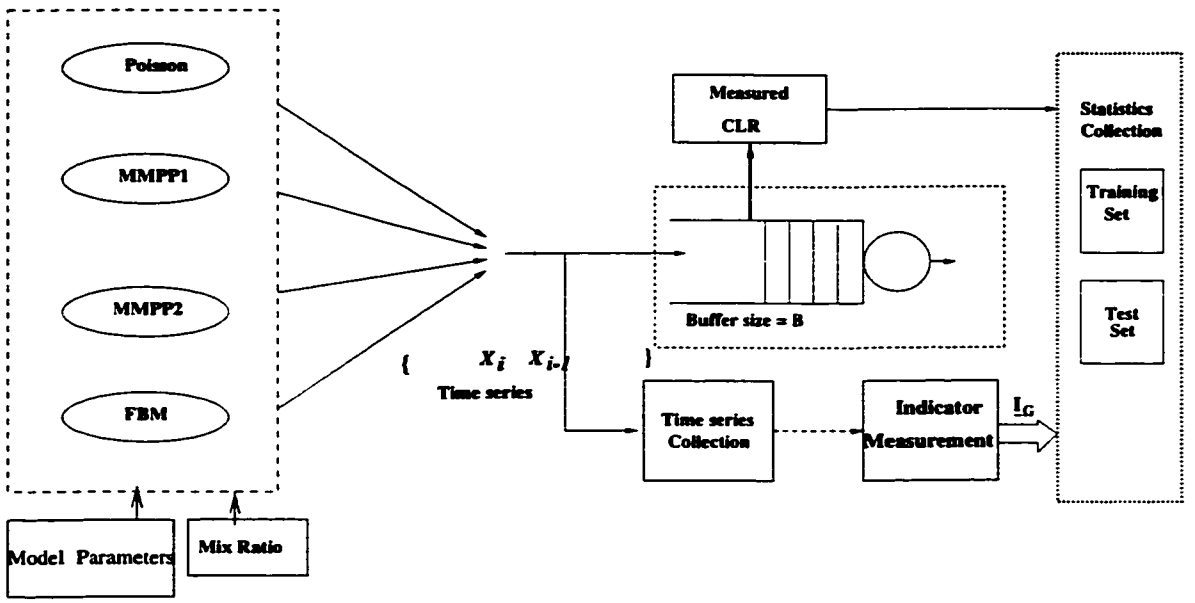

Figure 5.2: Simulation model for performance evaluation of the CLR approximator.

CLR function is approximated by

$$P_{loss_i} \approx f_i(\mathbf{I_{Gi}},\ B) \quad i = 1,\ 2,\ 3,\ 4.$$

where $\mathbf{I_{Gi}}$ is the indicator of traffic for case i, as shown in Table 5.2 and B is the buffer size. Function $f(\cdot)$ is the superposition of sigmoid functions,

$$f_i(\mathbf{I_{Gi}},\ B) = \zeta_i + \sum_{k=1}^{M} \frac{\alpha_{k_i}}{1 + e^{-(\mathbf{I_{G_i}} \odot \mathbf{W_{k_i}} + \omega B + \gamma_{k_i})}},$$

in which $\odot$ denotes the scalar product and

M : Number of sigmoid functions or the equivalent number of neurons,

$\mathbf{W_{k_i}}$: Vector that contains the weight of each input,

ω : Weight of buffer size,

γ_{k_i} : The bias value for the function,

α_{k_i} : The gain of the kth sigmoid function and

ζ_i : Function constant.

Case	Traffic Indicator	Parameters
1	$\mathbf{I_{G_1}}$	{Load, $IDC(1)$ }
2	$\mathbf{I_{G_2}}$	{Load, $IDC(1)$, K }
3	$\mathbf{I_{G_3}}$	{Load, $IDC(1)$, $IDC(L)$ }
4	$\mathbf{I_{G_4}}$	{Load, $IDC(1)$, K, $IDC(L)$ }

Table 5.2: Various cases for the set of parameters of the traffic indicator that can be considered for CLR approximation.

$\mathbf{W_{k_i}}$, γ_{k_i}, α_{k_i} and ζ_i (all for the case i) are obtained in such a way that the MSE of the case i, between the actual output and the approximated one is minimized, i.e.,

$$\left\{\frac{1}{n}\sum_{l=1}^{n}\left[\log(P_{loss_i}^{(l)}) - \log(f_i^{(l)}(\mathbf{I_{Gi}},\ B))\right]^2\right\} \quad \text{is minimized,}$$

where n is the total number of samples over which the optimization is performed.

Let us drop the subscript i. The most widely used algorithm to minimize MSE is back-propagation (described in [58]). A set of input vectors, i.e., $\{\underline{input_1},\ \underline{input_2},\ \cdots, \underline{input_n}\}$ are applied to the system of Figure 5.1. The weights are adjusted based on the corresponding target vectors, $\{\underline{taget_1},\ \underline{target_2},\ \cdots,\ \underline{target_n}\}$ at the output. The procedure stops when a certain stop-criterion is satisfied. The input and output are

$$\{\underline{input_1},\ \cdots, \underline{input_n}\} = \begin{bmatrix} Load^{(1)} & Load^{(2)} & \cdots & Load^{(n)} \\ IDC(1)^{(1)} & IDC(1)^{(2)} & \cdots & IDC(1)^{(n)} \\ K^{(1)} & K^{(2)} & \cdots & K^{(n)} \\ IDC(L)^{(1)} & IDC(L)^{(2)} & \cdots & IDC(L)^{(n)} \\ B^{(1)} & B^{(2)} & \cdots & B^{(n)} \end{bmatrix} \tag{5.2}$$

$$\{\underline{target_1},\ \cdots,\ \underline{target_n}\} = \left\{\log P_{loss}^{(1)} \quad \log P_{loss}^{(2)} \quad \cdots \quad \log P_{loss}^{(n)}\right\}. \tag{5.3}$$

Input vector is connected to the kth sigmoid function with $\mathbf{W_k}$ (Figure 5.1). For a given input vector, say $\underline{input} = \{input_1, input_2, \cdots, input_5\}$, where $input_1$ denotes the Load value, $input_2$ denotes the value of $IDC(1)$, $input_3$ denotes the value of K, $input_4$ denotes the value of $IDC(L)$, and $input_5$ denotes the size of buffer, B. The output of kth sigmoid function is given by:

$$O_k = g(\sum W_{sk} input_s + \gamma_k) \tag{5.4}$$

where $g(x) = 1/(1 + e^{-x})$. Let the target state of the output be t. Output error can be defined as

$$E = \frac{1}{2}(t - O)^2 \tag{5.5}$$

where $O = \sum_{k=1}^{M} \alpha_k O_k + \zeta$.

Let us consider a neuron whose output is O_j and it receives the output of previous node l with gain W_{lj}. The gradient descent algorithm adapts the weights according to the gradient error, i.e.,

$$\Delta W_{lj} \propto -\frac{\partial E}{\partial W_{lj}} = -\frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial W_{lj}}. \tag{5.6}$$

Specifically, we define the error signal as

$$\sigma_j = -\frac{\partial E}{\partial O_j}. \tag{5.7}$$

With some manipulation, we can get

$$\Delta W_{lj} = \epsilon\ \sigma_j\ O_l \tag{5.8}$$

where ϵ is learning rate. For the output O, which is a single linear neuron (linear summation of sigmoid neurons), σ_j is given by

$$\sigma = (t - O), \tag{5.9}$$

and, hence,

$$\Delta\alpha_k = \epsilon\sigma O_k \quad k = 1, \cdots, M.$$

For the output of sigmoid functions,

$$\sigma_j = O_j(1 - O_j)\sigma\alpha_j \quad j = 1, \cdots, M. \tag{5.10}$$

In order to improve the convergence characteristics, we can introduce a momentum term with a momentum gain τ and an adaptive learning rate to equation (5.8):

$$\Delta\alpha_k(m+1) = \epsilon(m)\sigma O_k + \tau\Delta\alpha_k(m) \quad k = 1, \cdots, M \tag{5.11}$$

$$\Delta W_{lj}(m+1) = \epsilon(m)\sigma_j O_l + \tau\Delta W_{lj}(m) \quad j = 1, \cdots, M, \quad l = 1, 2, 3, 4 \tag{5.12}$$

where m represents the iteration index or *epoch* that each input is presented to the system.

In the following subsections, we first evaluate the performance of the CLR approximation with respect to the selection of traffic indicator. After selecting the traffic indicator, we compare the performance of the proposed CLR approximator with those proposed in the literature.

5.3.1 Performance Evaluation of the Traffic Indicator

Let us assume that CLR is approximated using an arbitrary number of (for example 40) sigmoid functions [1]. Two sets of statistics are collected using the system model of Figure 5.2: training set and test set. We used the training set to adjust the weights and the coefficients of the neural network approximator of Figure 5.1. The test set is used to assess the performance of the approximator. Each set consists of

[1] A higher number of sigmoid functions improves slightly the accuracy at the expense of complexity.

a thousand samples of input and output pairs , which are the actual measured CLR, constructed from various traffic profiles: Poisson, MMPP, FBM and their mixtures such as (Poisson, MMPP1), (MMPP1, MMPP2), (MMPP1, FBM), (MMPP2, FBM), etc. Figure 5.3 shows the convergence of the MSE as a function of the epoch (iteration) in adjusting the weights and the coefficients. The stop criteria has been set to be either an MSE of 10^{-2} in predicting $|\log(CLR)|$ or an iteration index of 2×10^5, whichever of the two comes first. We observe that at the epoch of 2×10^5, the value of MSE for case 4 is 1/5 of that of case 1 and 1/2 of that of case 3. We examine the approximated functions for all the four cases by applying the test set. Figures 5.4 to 5.10 show the performance of the approximator with respect to the measured CLR for all the four cases. Figure 5.4 shows the predicted CLR for all four traffic indicators when the traffic profile is the mixture of MMPP1 and MMPP2. As seen, for this traffic profile both $\mathbf{I_{G_3}}$ and $\mathbf{I_{G_4}}$ closely approximate CLR. Figures 5.5, 5.6 and 5.7 compare the predicted CLR for mixed traffic of MMPP1 and FBM for different values of both K and load. From these figures, it is seen that only $\mathbf{I_{G_4}}$ closely approximate the CLR. For the case of FBM traffic, as shown in Figure 5.9, both $\mathbf{I_{G_3}}$ and $\mathbf{I_{G_4}}$ are good indicators to approximate the CLR. However, examples shown in Figures 5.8 and 5.10 for MMPP2 and MMPP1, respectively, indicate that only $\mathbf{I_{G_4}}$ can closely approximate the CLR. As a result of the above discussion, we conclude that using $\mathbf{I_{G_4}}$={ Load, IDC(1), K, IDC(L)} as the traffic indicator leads to a close prediction of CLR for various traffic profiles.

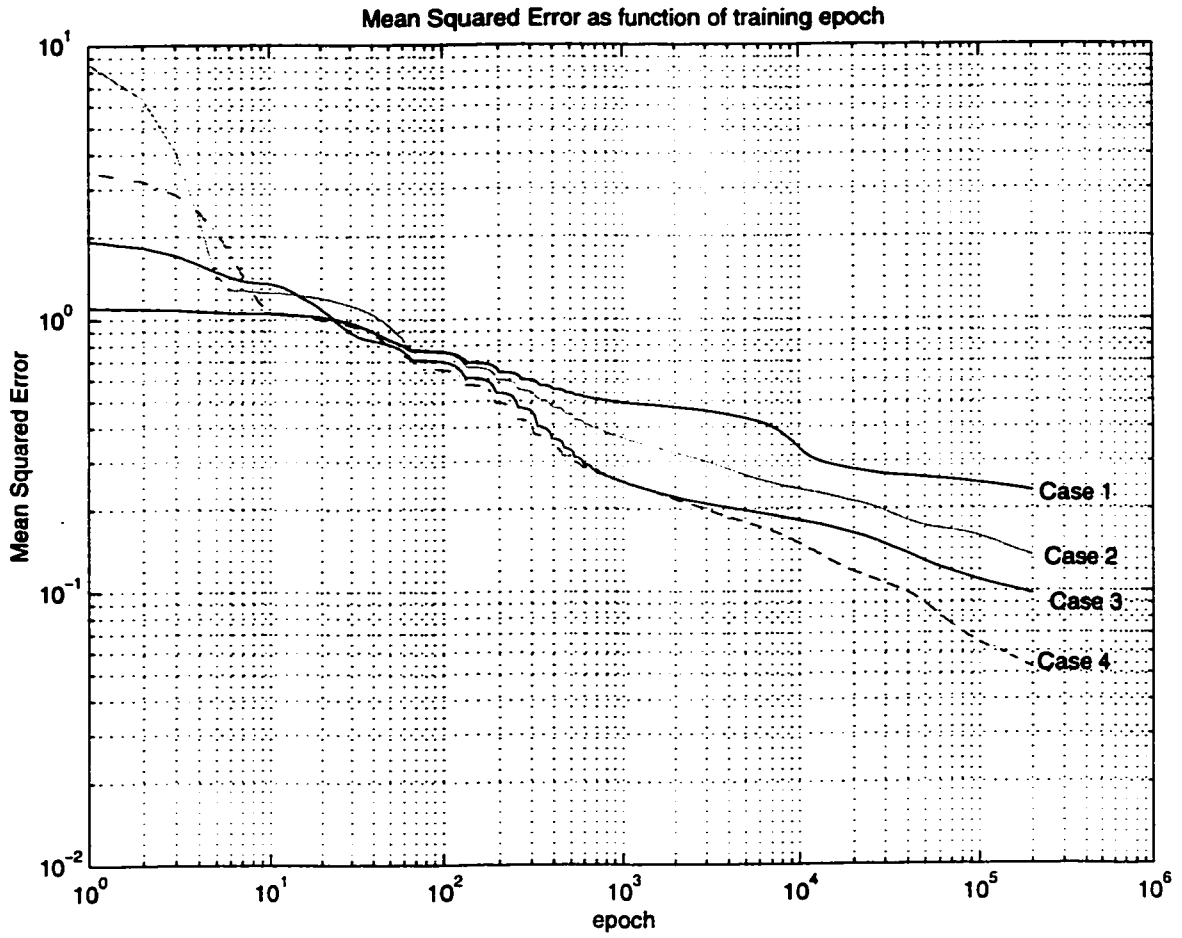

Figure 5.3: Mean Square Error (MSE) versus the epoch. The training set, which is the same for all four cases, consists of one thousand input (traffic indicator and buffer size) and output (CLR) pairs constructed from traffic profiles of Poisson, MMPP1, MMPP2, FBM (with different H) and their two-by-two mixtures.

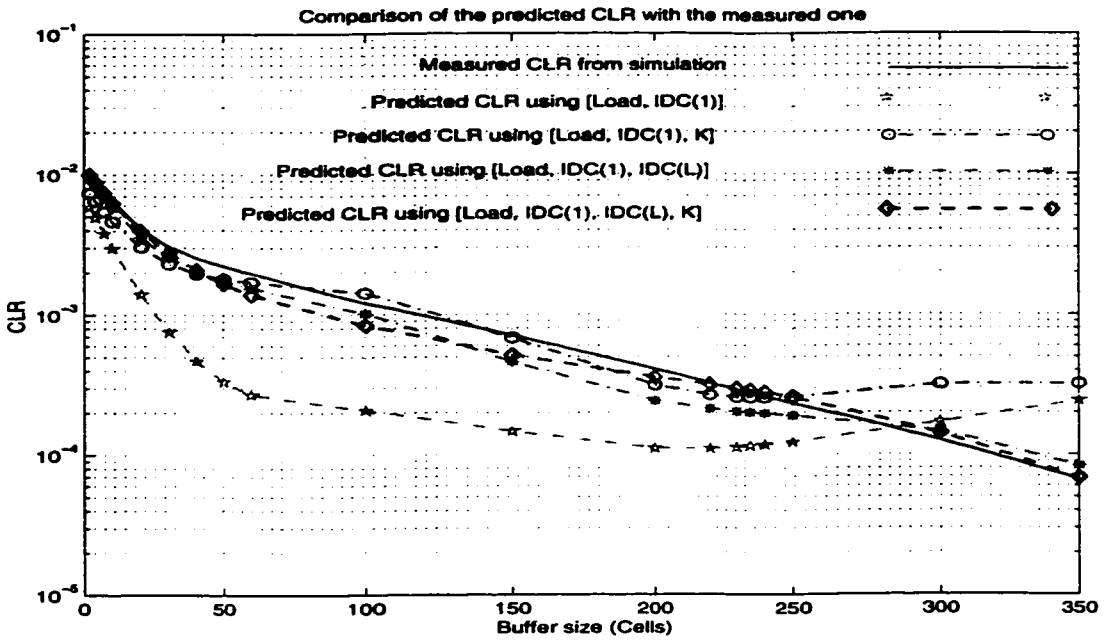

Figure 5.4: An example of predicted CLR for various traffic indicators when the input traffic profile is a mixture of MMPP1 and MMPP2.

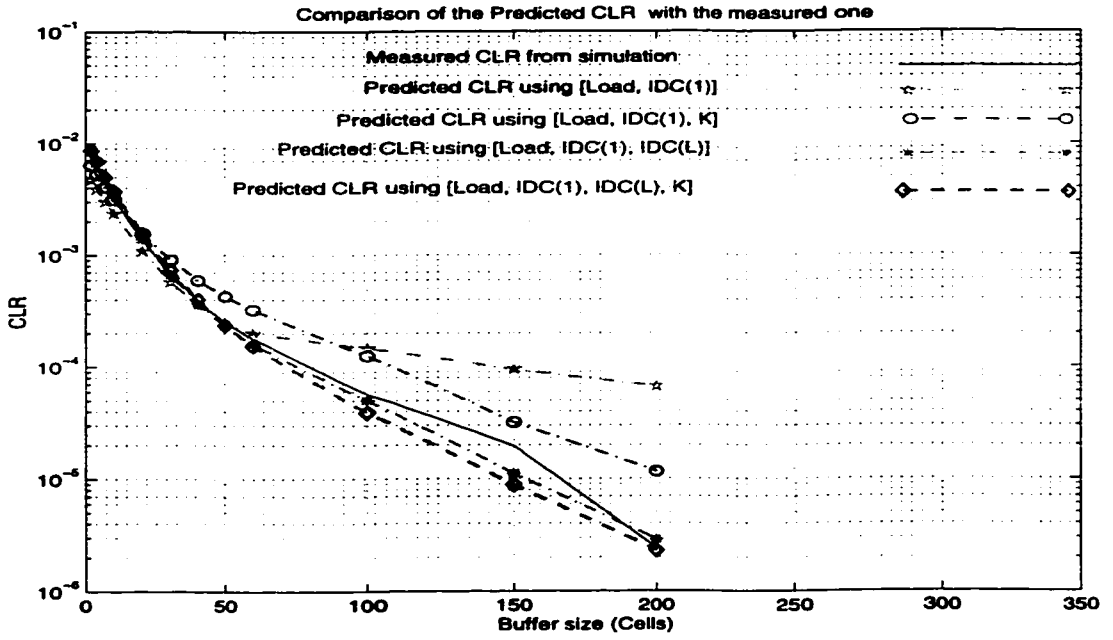

Figure 5.5: Predicted CLR for various traffic indicators when the input traffic profile is a mixture of MMPP1 and FBM (H=0.8) and load=0.9.

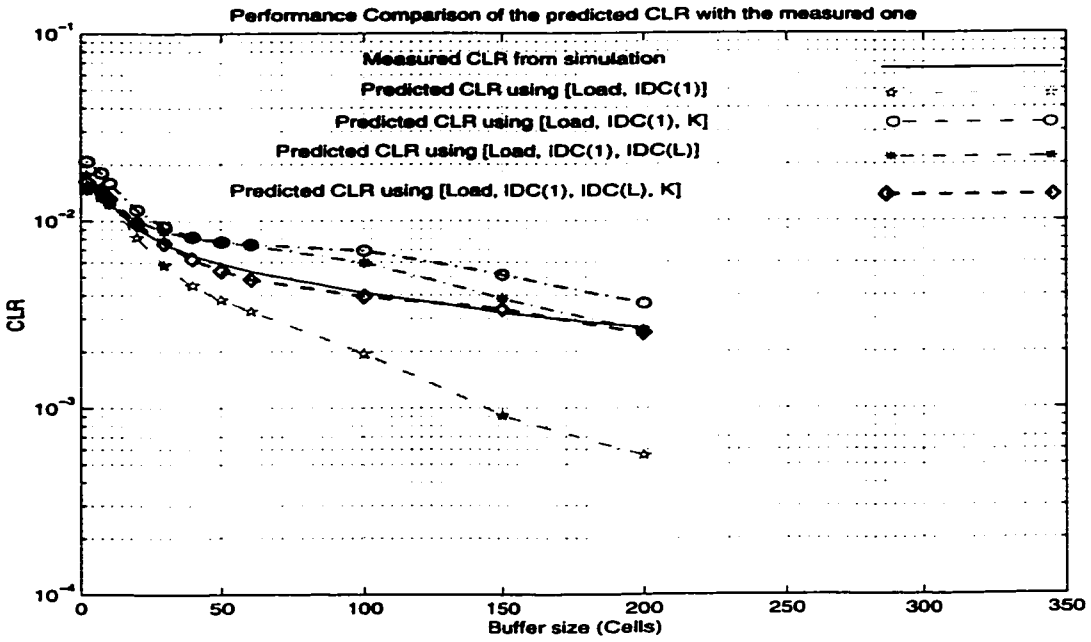

Figure 5.6: An example of predicted CLR for various traffic indicators when the input traffic profile is a mixture of MMPP1 and FBM (H=0.9) and load=0.95.

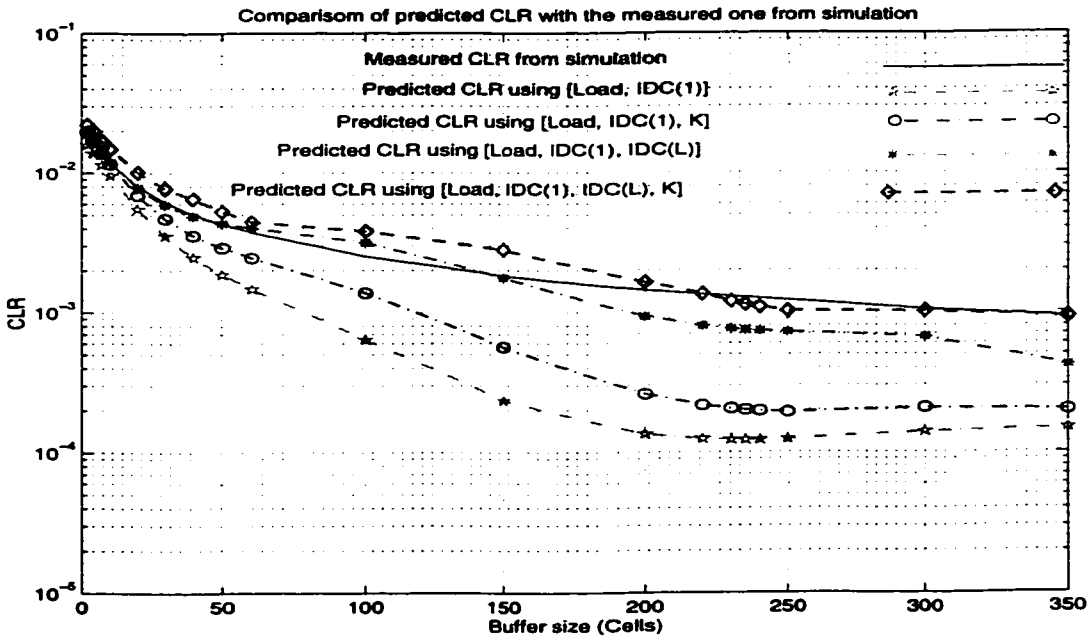

Figure 5.7: An example of predicted CLR for various traffic indicators when the input traffic profile is a mixture of MMPP and FBM (H=0.8) and load=0.95.

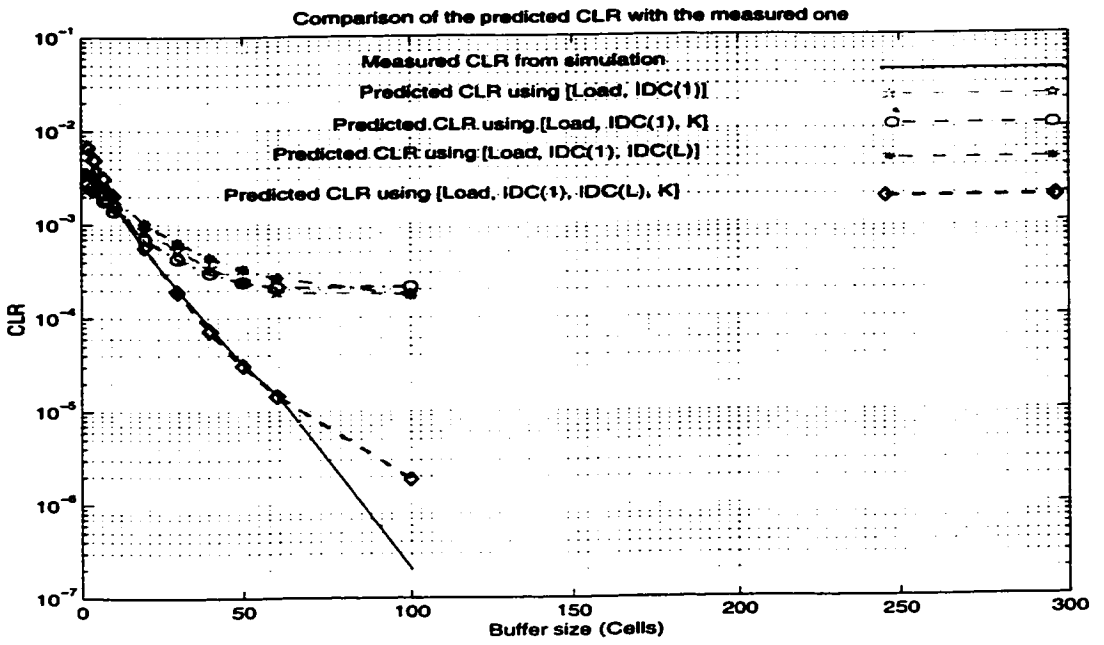

Figure 5.8: An example of predicted CLR for various traffic indicators when the input traffic profile is an MMPP2 source at load=0.85.

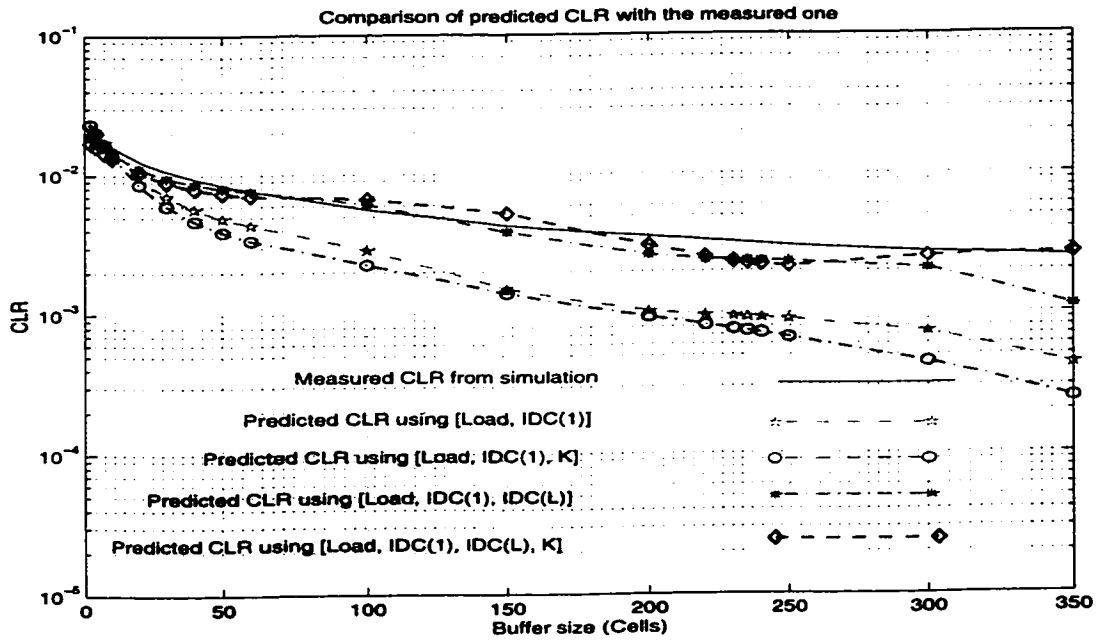

Figure 5.9: An example of predicted CLR for various traffic indicators when the input traffic profile is FBM with H=0.8.

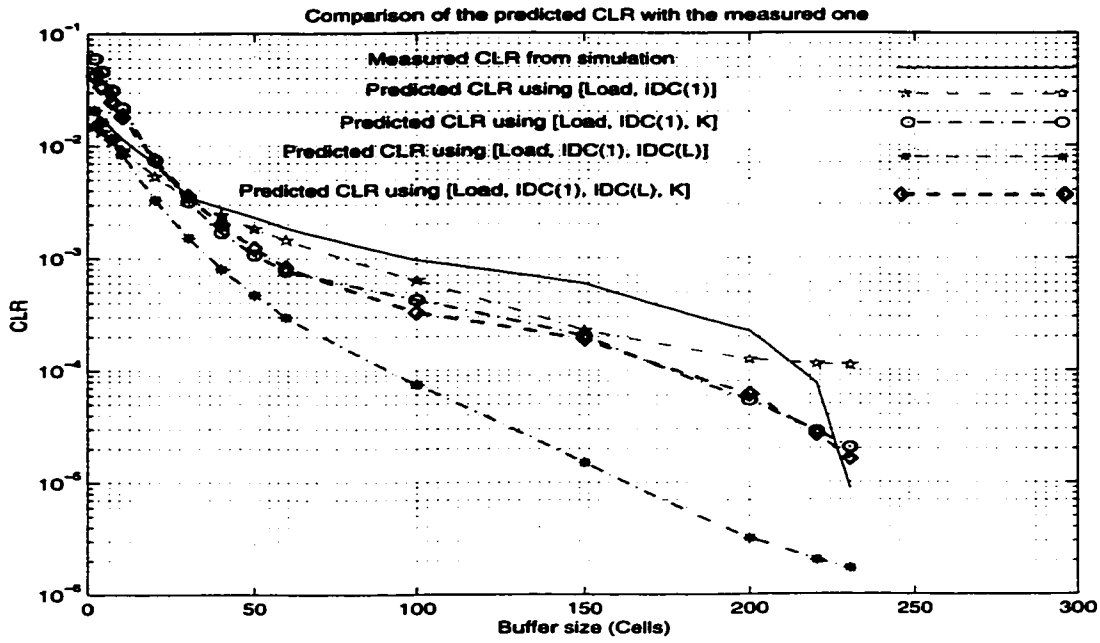

Figure 5.10: An example of predicted CLR for various traffic indicators when the input traffic profile is an MMPP1.

5.3.2 Performance Evaluation of the Approximator

In this subsection, we compare the performance of the CLR approximator that uses $\mathbf{I}_{\mathbf{G}_4}$ with the analytical results of FBM and MMPP.

Let us first examine the queue length distribution of FBM. According to [31], there is no exact formula for the queue length distribution of FBM. Only an upper-bound approximation method exists as

$$P(Q > x) \approx \exp\left[-\frac{(C-m)^{2H}}{2k^2(H)\cdot IDC(1)\cdot m}\cdot x^{2-2H}\right] \tag{5.13}$$

where C is the service rate, m is the mean arrival rate ($Load = m/C$) and $k(H)$ is given by [31]:

$$k(H) = H^H(1-H)^{1-H}.$$

Now, let us look at an infinite FIFO queue that is served at a constant rate and it

is fed by a 2-state MMPP source, i.e., MMPP/D/1 queue. The stationary transition probability matrix is given by [52]

$$\mathbf{P} = \begin{bmatrix} A_0 & A_1 & A_2 & A_3 & \cdots \\ A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & A_0 & A_1 & A_2 & \cdots \\ 0 & 0 & A_0 & A_1 & \cdots \\ 0 & 0 & 0 & A_0 & \cdots \\ \vdots & & & & \vdots \end{bmatrix} \tag{5.14}$$

where

$$A_m = \int_0^\infty P(m,\ t)dQ(t) \tag{5.15}$$

and $P(m,\ t)$ is the probability matrix of m cells arriving in time t and $Q(t)$ is the probability distribution of service time of the queue. Let us assume that $\mathbf{x}$=($\mathbf{x_0}$, $\mathbf{x_1}$, $\cdots$) is the steady-state probability (queue length distribution) vector of P. For a 2-state MMPP, $\mathbf{x_i}$s are row vectors of 2 dimension. In order to obtain $\mathbf{x}_i$, it is required to solve the fundamental matrix $\mathbf{G}$, the transition probability matrix of first passage time, from the following non-linear matrix equation [59]

$$\mathbf{G} = \sum_{n=0}^{\infty} A_n G^n. \tag{5.16}$$

Several fast algorithms have been proposed to solve these equations (for example [60]). For a stable system when the $Load < 1$ we have

$$\mathbf{gG} = \mathbf{g}, \quad \mathbf{g}e = 1$$

where e is unity column vector. $\mathbf{x_0}$ is given by

$$\mathbf{x_0} = (1 - Load)\mathbf{g}.$$

The steady-state equation is given by $\mathbf{xP} = \mathbf{x}$ as

$$\mathbf{x_i} = \mathbf{x_0 A_i} + \sum_{\nu=0}^{i} \mathbf{x}_{\nu+1} \mathbf{A}_{i-\nu}.$$

One of the solution methods is Adaptive Newton-Kantorovich (ANK) algorithm[2][61]. ANK works with the following iteration

$$\begin{aligned} \mathbf{x_i} &= \left(\mathbf{x_0} \overline{\mathbf{A_i}} + \sum_{m=1}^{i-1} \mathbf{x}_m \overline{\mathbf{A}}_{i+1-m} \right) (\mathbf{I} - \overline{\mathbf{A}}_1)^{-1}, \quad i \leq M \\ \mathbf{x_i} &= \left(\mathbf{x_0} \overline{\mathbf{A_i}} + \sum_{m=i-M}^{i-1} \mathbf{x}_m \overline{\mathbf{A}}_{i+1-m} \right) (\mathbf{I} - \overline{\mathbf{A}}_1)^{-1}, \quad i > M \end{aligned}$$

where M is selected such that $\left|\overline{A}_{M+1}\right| < \epsilon$ and

$$\overline{\mathbf{A}}_i = \sum_{\nu=i}^{\infty} \mathbf{A}_\nu \mathbf{G}^{\nu-i}.$$

Using $\mathbf{x}$, an upper bound for CLR in a finite queue with size B can be obtained as

$$CLR_{analytical} = \sum_{i=B}^{\infty} \mathbf{x}_i \mathbf{e}^T. \tag{5.17}$$

Now, we consider four cases among many possible cases and compare the predicted CLR using the proposed approach with that of the analytical approach. We assume the service rate is C=100 cells/time-interval.

Case 1: *MMPP1 as traffic profile*

Let us consider that the traffic profile is generated by only an MMPP1 source that has been mapped from N=51 On-Off sources (see Appendix B) each with peak-to-average ratio PAR=4, peak-rate A=5 cells/interval, On-time sojourn time $\alpha = 0.284$ and Off-time sojourn time $\beta = 0.154$. Figure 5.11 shows the analytical result of the CLR approximation for MMPP1 source using equation

[2]Computaional complexity of the ANK solution can be embeded into a framework of a neural network system [62].

(5.17). The measured indicator of traffic, $\mathbf{I}_G$, for this MMPP1 source is $Load$ = 0.9, IDC(1)=1.04, IDC(L)=2.7 and H=0.64. Using this indicator of traffic, CLR is predicted by the proposed CLR approximator for different queue sizes. As shown in Figure 5.11, when compared to the analytical results, the proposed approximator closely predicts the CLR. However, there is a significant difference between the calculated CLR by using (5.17) and the measured CLR. This is because equation (5.17) provides an upper bound for CLR assuming that the services over the network are probabilistic[3].

Case 2: *MMPP2 as traffic profile*

Now, let us assume that traffic is generated by an MMPP2 source, equivalent to aggregating N=8 On-Off sources with peak-to-average ratio PAR=20, peak-rate A=33 cells/interval, and the same values for α and β as in the case 1. Figure 5.12 depicts the analytical results by using ANK iteration and equation (5.17). The measured indicator of traffic, $\mathbf{I}_G$, for the traffic generated by this MMPP2 source is $Load$ = 0.9, IDC(1)=1.5, IDC(L)=102.6 and K=0.9. This indicator has been applied to the proposed CLR predictor and the results have been compared with both the measured CLR and the analytical one as shown in Figure 5.12. Similar to case 1, it is observed that the analytical approach overestimates the CLR, while the proposed approach closely follows the measured one.

Case 3: *FBM as traffic profile*

As another example, let us assume that the traffic profile is generated by an FBM source[4] with H=0.8, mean $m = Load \cdot C$ (0.9 × 100 cells/interval) and

[3]As mentioned in Chapter 3, the proposed approach assumes predictive services.

[4]Generation of FBM traffic has been discussed in the Appendix C.

variance coefficient of $m \cdot IDC(1)$ (90 cells/interval $\times$ 1). Figure 5.13 compares the analytical result given by equation (5.13) with both the measured CLR and the predicted one using the proposed approach. The measured indicator of the traffic is $Load$ = 0.9, IDC(1)=1.0, IDC(L)=28 and H=0.8. As shown in Figure 5.13, equation (5.13) provides an upper bound for CLR, whereas the predicted CLR using the proposed approach closely follows the measured CLR.

Case 4: *Mixture of MMPP1 and FBM as traffic profile*

As an example of a mixed scenario, let us consider a traffic profile that consists of 50% MMPP1 traffic and 50% FBM traffic. For this traffic mix the measured traffic indicator is $Load$ = 0.9, IDC(1)=0.98, IDC(L)=21.5 and K=0.78. Figure 5.14 compares the predicted CLR with the measured one. Now, let us assume that CLR is approximated by either analytical result of FBM or that of MMPP. Let us first apply the measured traffic indicator to equation (5.13) and evaluate it by substituting C=100 cells/interval, $m = Load \cdot C$, IDC(1)=0.98 and K=0.78. The results, plotted in Figure 5.14, show that using analytical results of the FBM model to approximate the CLR of the mixed traffic overestimates the CLR. Now, let us look at the performance of the analytical result of MMPP model. By trial and error, we determined that the measured traffic indicator can result from aggregating N = 37 On-Off sources with A = 7 cells/interval, PAR=4, α = 0.284 and β = 0.154. Mapping the On-Off sources to the equivalent MMPP source (using the procedure in the Appendix B), we can solve for the CLR using equation (5.17). As shown in Figure 5.14, using the analytical results of MMPP to approximate the CLR of the mixed traffic overestimates the CLR at small buffer sizes, whereas it underestimates the CLR at large buffer

sizes. On the other hand, for the mixed traffic, FBM approximation of CLR result in an overestimation of CLR.

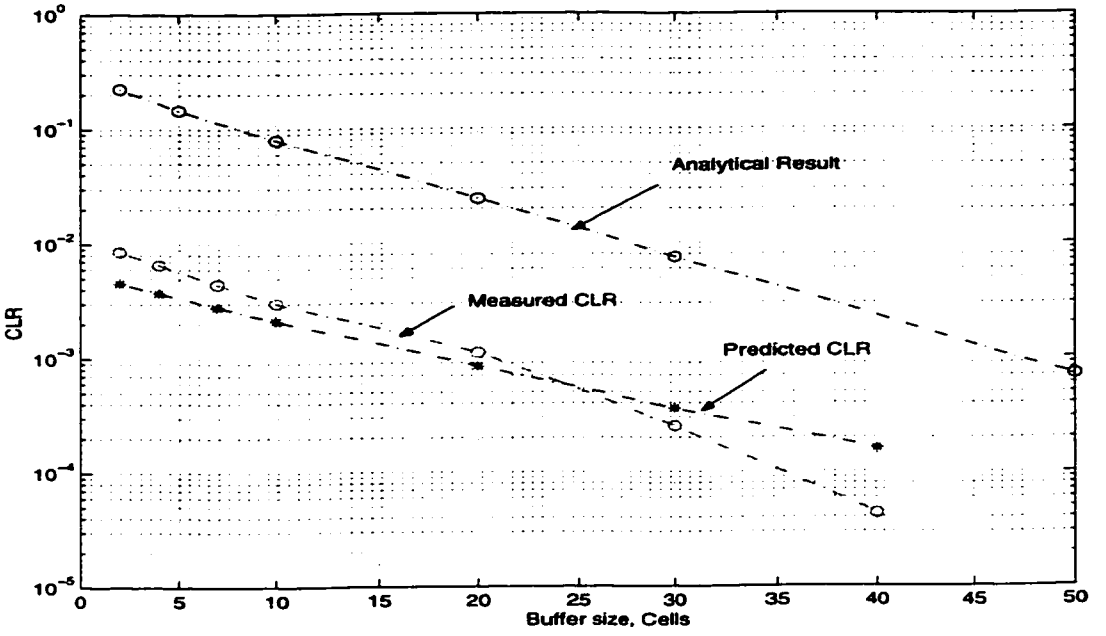

Figure 5.11: Comparison of the predicted CLR with the analytical result of MMPP1 source discussed in case 1.

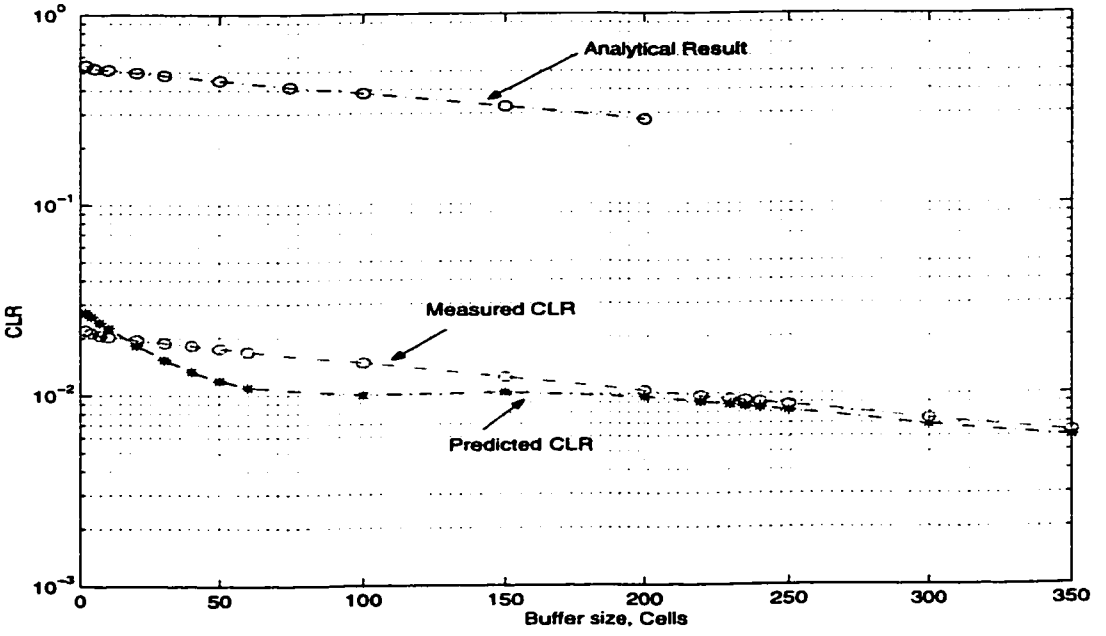

Figure 5.12: Comparison of the predicted CLR with the analytical result for the MMPP2 source discussed in case 2.

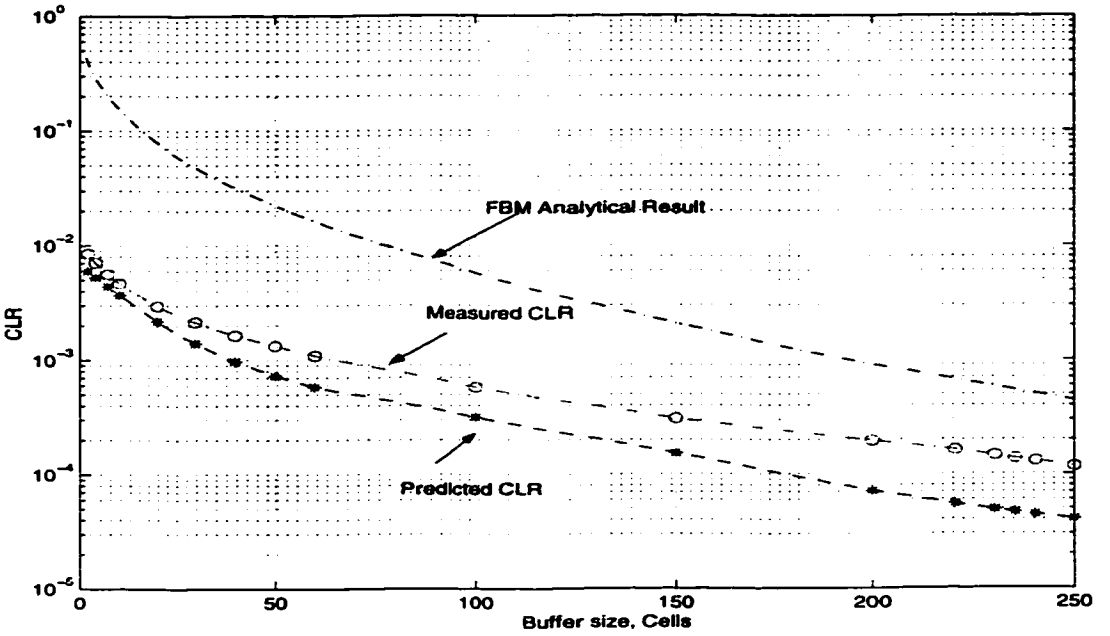

Figure 5.13: Comparison of the predicted CLR with the analytical result for the FBM source discussed in case 3.

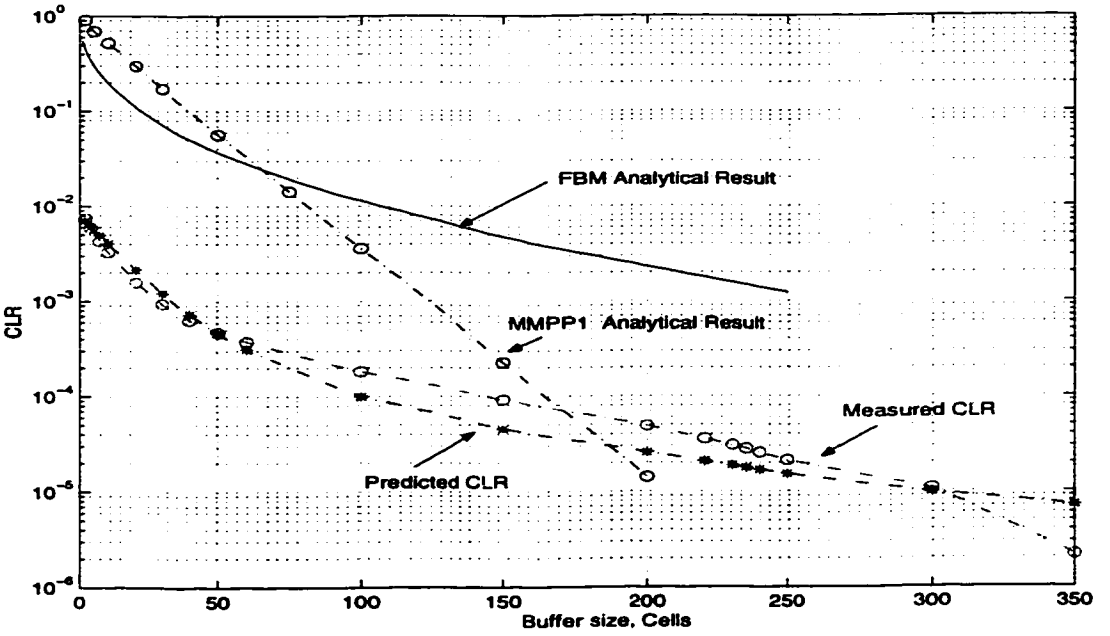

Figure 5.14: Comparison of the predicted CLR with the analytical approaches of FBM and MMPP1 sources discussed in case 4.

5.4 Summary

In this chapter, we selected a neural network system consisting a linear combination of a set of sigmoid functions to approximate the CLR from the candidate traffic indicator whose parameters were identified in Chapter5. We considered four traffic indicators each constructed from a subset of the identified parameters. The results indicate that all four parameters of the candidate indicator (i.e., {Load, IDC(1) , K and IDC(L)})are necessary for a close CLR prediction. Results also indicate that the learned CLR function in terms of the selected traffic indicator closely follows the measured CLR and outperforms the results of analytical approaches.

Chapter 6

Real-time Derivation of Traffic Indicator

The derivation of the parameters of the traffic indicator can be viewed as an estimation problem. The parameters of $\underline{\mathbf{I}_G}$ can be estimated from the traffic samples using a set of estimators, i.e.,

$$\underline{\mathbf{I}_G} = \mathbf{Z}(X_i,\ X_{i-1},\ \cdots,\ X_{i-s}),$$

where $\mathbf{Z} = \{Z_1(\cdot), \cdots,\ Z_4(\cdot)\}$ is a set of estimators for the four parameters of $\underline{\mathbf{I}_G}$ and $\mathbf{s}=\{s_1,\ s_2,\ s_3,\ s_4\}$ is a vector that represents the orders of the estimators. In the following sections, we describe these estimators in detail.

6.1 Load and IDC(1) Measurement

Load and IDC(1) can be obtained by measuring the mean $\mu_{\mathbf{X}}= E[\mathbf{X}]$ and the variance $\sigma_{\mathbf{X}}=E[(\mathbf{X} - \mu_{\mathbf{X}})^2]$ of the time series $\mathbf{X}$ over a large window of samples as

$$\textit{Load} \ = \ Z_1(\mathbf{X}) = \frac{1}{C}E[\mathbf{X}]$$

$$IDC(1) \;=\; Z_2(\mathbf{X}) = \frac{\sigma_{\mathbf{X}}}{E[\mathbf{X}]}, \tag{6.1}$$

where C is the transmission rate in terms of cells/sec. In [63], we discussed an alternative approach in measuring load at time i using a Finite Impulse Response (FIR) filter as

$$Load_i = \frac{1}{C}\sum_{j=0}^{N-1} W_j X_{i-j},$$

where i is the current time interval. N is chosen based on the frequency of the traffic load variation. W_i's are calculated dynamically using Least Mean Square algorithm [63]. The FIR approach is effective when traffic load and its variance are the only parameters of the traffic indicator. Since the proposed traffic indicator includes both K[1] and $IDC(L)$, which both represent traffic burstiness, we use a large number of samples to measure the mean rate and the standard deviation of the traffic. The measured values of the mean rate and the standard deviation are used to estimate load and IDC(1).

6.2 Techniques to Measure K and IDC(L)

For LRD and self-similar traffic, the parameter K represents the Hurst parameter. Thus, before we address the proposed approach in measuring K and IDC(L), we first discuss classical methods in the measurement of the Hurst parameter. Then, we discuss the shortcomings of the classical approaches followed by the proposed approach [64].

[1]FOR FBM or a self-similar traffic $K = H$, where H is the Hurst parameter. For other $K = 0.5(1 + s)$ where s is the slope of the linear part of the IDC curve.

6.2.1 Classical Approaches in Measurement of the Hurst Parameter

In the following subsections we discuss several methods that have been used in the literature to estimate the Hurst parameter.

IDC Calculation

For a given time interval of length m, IDC is given by the variance of the number of arrivals during the interval of length m divided by the expected value of that same quantity:

$$IDC(m) = \frac{var(\sum_{j=1}^{m} X_j)}{E\left[\sum_{j=1}^{m} X_j\right]}, \tag{6.2}$$

where X_j represents the number of cells arrived to the queue in the ith time interval. To calculate IDC(m), a large number of samples of X_j are collected and divided into blocks of m so that each block is the sum of m samples. The variance and the average of these blocks are calculated and substituted into the equation (6.2). The Hurst parameter is estimated by calculating (6.2) for two different intervals m_1 and m_2 ($m_2 > m_1$), obtaining the slope in logarithmic scale and evaluating 0.5(1+*slope*).

Adjustment Range for Hurst Estimation

Suppose during the time interval (0, T'), the time series of X_1, X_2, $\cdots$, $X_{T'}$, have arrived into the queue. The mass diagram, which is basically a plot of the cumulative samples up to time t, is given by

$$S_t = \sum_{i=1}^{t} X_i \quad \text{for } t = 1, 2, \cdots, T'.$$

The average rate during the time interval (0, T') is $\overline{X}_{T'} = S_{T'}/T'$. Now, let us assume that the cells are served steadily at the average rate. Assuming the initial value of

the number of cells in the queue is X_0, at time t the queue has $X_0 + S_t - t\overline{X}_{T'}$ cells. Attention is focused on the maximum and the minimum of the queue values, which are:

$$D_{max} = X_0 + \max_{1 \leq t \leq T'}(S_t - t\overline{X}_{T'})$$

and

$$D_{min} = X_0 + \min_{1 \leq t \leq T'}(S_t - t\overline{X}_{T'}),$$

respectively. The required buffer that does not allow queue overflow is $R(X_1, X_2, \cdots, X_{T'}) = D_{max} - D_{min}$ which is called adjusted range of flows. The statistic $\frac{R}{S}$, where S is the standard deviation of $(X_1, X_2, \cdots, X_{T'})$, assumes further importance due to the work of Hurst on long-term dependency. Hurst found that $\frac{R}{S}$ could be approximately equated to the asymptotic result $\left(\frac{T'}{2}\right)^H$, where $\hat{H}$ is the estimated the Hurst parameter. As a result, H can be found by

$$\hat{H} \approx \frac{\log R/S}{\log T'/2}. \tag{6.3}$$

Hurst Estimation using Maximum Likelihood Estimation (MLE)

MLE could also be employed to estimate the Hurst parameter. The method assumes that the traffic patterns follow a Discrete Fractional Gaussian Noise (DFGN) model (or FBM). Let X_i, $i = 1, 2, \cdots, n$ be an observed time series with $E[X_i] = \mu$ and $cov[X_i, X_{i-k}] = Y_k$. Because DFGN is multivariate normally distributed, its log

likelihood is

$$L(H,\ \mu,\ Y_0) = -\frac{1}{2}\ln \underline{\Gamma}_n - \frac{1}{2}\begin{bmatrix} X_1-\mu \\ X_2-\mu \\ \cdot \\ \cdot \\ \cdot \\ X_n-\mu \end{bmatrix} \underline{\Gamma}_n^{-1} \begin{bmatrix} X_1-\mu \\ X_2-\mu \\ \cdot \\ \cdot \\ \cdot \\ X_n-\mu \end{bmatrix} \tag{6.4}$$

where $\underline{\Gamma}_n$ and Y_0 are covariance matrix and the variance of X_i, respectively. $\underline{\Gamma}_n$ is given by

$$\underline{\Gamma}_n = Y_0\,[C(i-j,\ H)]_{n\times n}\,,$$

in which $C(s,\ H)$ is given by equation (C.1) in the Appendix C. Efficient estimates of μ and Y_0 are obtained from

$$\hat{\mu} = \overline{X}_n = \frac{1}{n}\sum_{t=1}^{n} X_t$$

and

$$Y_0 = C_0 = \frac{1}{n}\sum_{t=1}^{n}(X_t - \overline{X}_n)^2.$$

MLE of H can be obtained by evaluating $L(H,\ \mu,\ Y_0)$ for $H = 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9$ and using inverse quadratic interpolation provided that n is not too large, say $n \leq 200$.

Other Approaches

Recently, an Abry-Veitch wavelet-based estimator has been proposed for on-line estimation of the Hurst parameter [65] which seems promising as it requires low memory and less computations. In addition to wavelet method, other approaches such as line length method and Fourier filtering have been deployed for Hurst estimation [66].

However, Lee *et al* [66] were not able to give a fair comparison of the performance of the these methods, because the Hurst parameters of the sets of samples that they used were unknown.

6.2.2 Performance Comparison of Classical Approaches

Since MLE method assumes a self-similar traffic model, we compare the performance of the classical approaches using the FBM model. When IDC calculation is used in Hurst estimation, the results (as plotted in Figure 6.1) show that the estimated H is fairly close to the actual Hurst parameter that was used by the FBM traffic generator in the Appendix C. For any time interval m, the Hurst parameter has been measured by obtaining a logarithmic slope at m, i.e.,

$$\hat{H} = 0.5\left(1 + \lim_{\Delta m \to 0} \frac{\log[IDC(m+\Delta m)] - \log[IDC(m-\Delta m)]}{\log[(m+\Delta m)] - \log[(m-\Delta m)]}\right).$$

The number of samples of $\mathbf{X}$ (or window size) affect the IDC or Hurst calculation. Figure 6.2 shows this effect for a given m. The results highlight that in order to obtain a good estimation of H, the number of samples that we need to consider is at least 8000. For this reason, a real-time estimation of the Hurst parameter using the IDC slope may not be effective due to the computational overhead.

When the adjusted range approach is deployed, there will be a bias value in the Hurst estimation. Figure 6.3 shows, for instance, the estimated values versus the real ones for $T = 2000$. For $0.5 < H < 0.75$, the bias value is positive, whereas for $H > 0.75$, it is negative.

MLE approach can potentially offer a good estimation if n could be chosen fairly large,

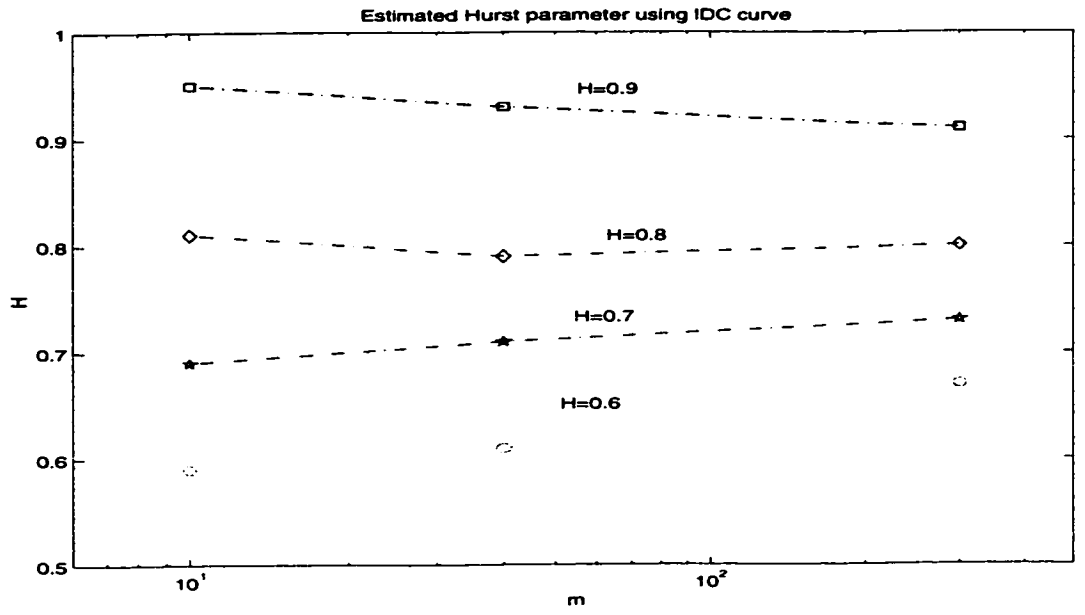

Figure 6.1: Estimated Hurst parameter using IDC slope.

i.e. $n >> 200$. However, large n results in a very complex and impractical inverse matrix calculation. Another disadvantage of the MLE approach is the assumption that the traffic is FBM, which might not be the case in broadband networks.

6.2.3 Proposed Approach for Measurement of K and IDC(L)

On-line measurements of the proposed parameter, K, and IDC(L) have two main requirements:

1) the newly arrived sample can be merged with the existing processed data, rather than requiring complete re-computation;

2) the technique needs to be efficient enough to implement at the rate samples arrive.

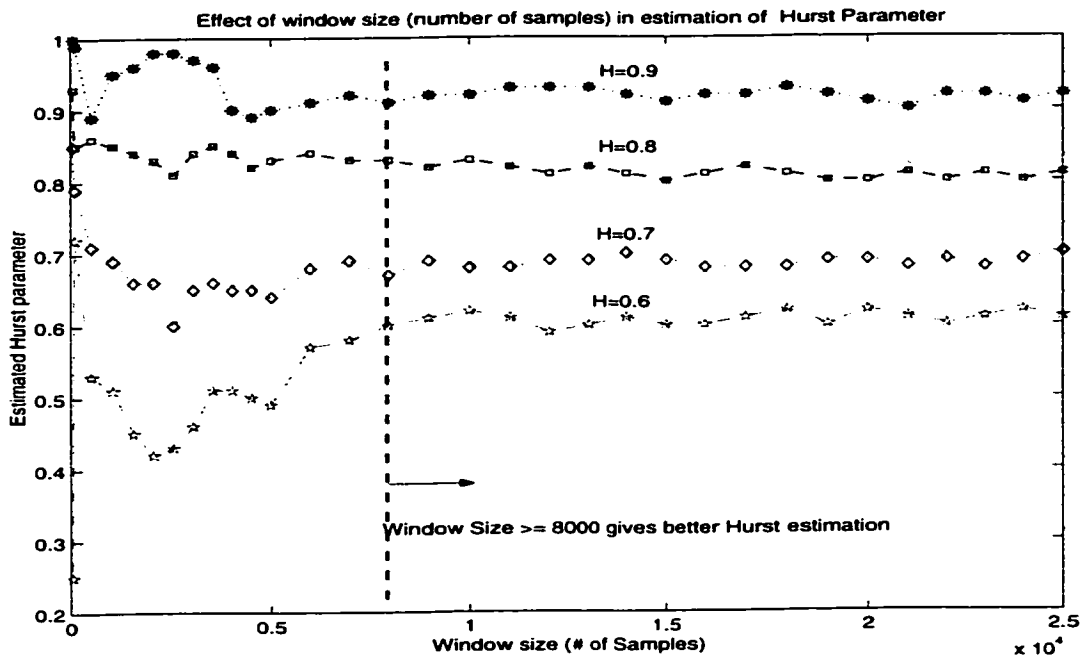

Figure 6.2: Effect of number of samples in IDC calculation for a given m.

The first requirement is critical for on-line estimation, whereas the second one is related to the necessary computing power.

Now let us examine the classical approaches. None of the approaches comply with the above requirements. Hurst estimation using MLE suffers from both the computational complexity and the assumption of the FBM traffic model. Rescaled adjustment, with asymptotic result, gives an estimate with a bias value that is undesirable. IDC calculation gives a better estimate for a fairly large window of observation, however, it lacks efficiency. To make IDC calculation more efficient and compliant with the above requirements, we propose to use a Neural Network system to learn the IDC function as well as to learn the correlation between the time series of traffic [64]. This is due to the following potential benefits that can be achieved when a neural network system is used:

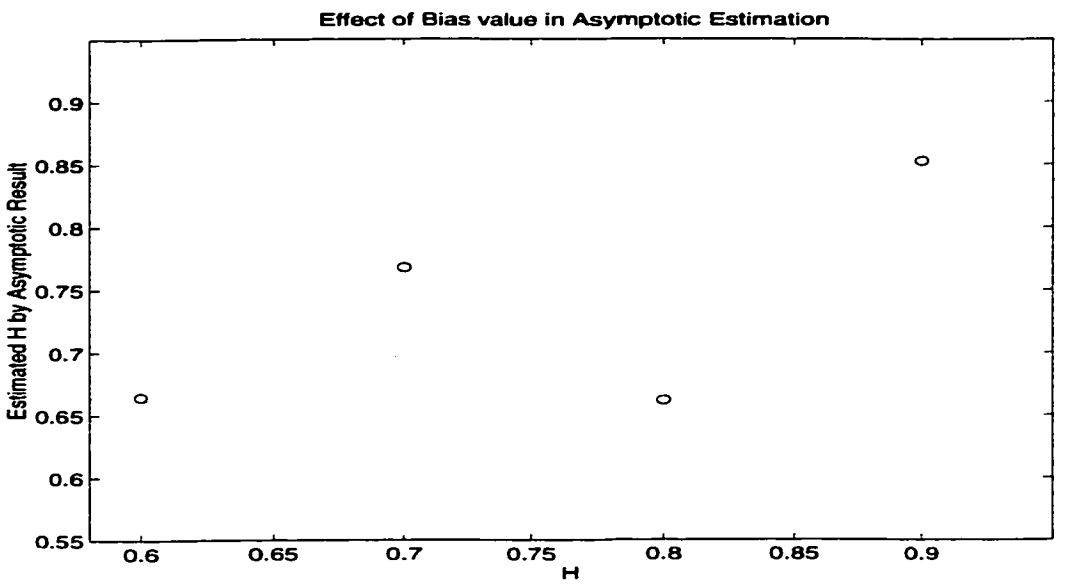

Figure 6.3: Estimated Hurst parameter when $\frac{R}{S}$ is equated to asymptotic result given by Hurst.

- *a) Adaptive learning*: Neural network can learn the functional relationship between the input time-series and output IDC value(s) during the course of network operation and, hence, it can *approximate* the IDC function.

- *b) High computation rate*: This is due to the parallelism of hardware implementation of neural networks. In general, computation time is independent of the neural network dimension and the number of input and outputs.

- *c) Generalization on learning:* A neural network has the ability to generalize learning to what has never been seen.

As a result, neural network can overcome the shortcomings of direct IDC calculations and comply with the two requirements.

In the Appendix D, it has been demonstrated that a neural network with a set of sigmoid functions is capable of approximating any continuous real function of n real variables with support in the unit hypercube. The function of the K is

$$K = 0.5\left(1 + \frac{d}{d[\log m]}\log[IDC(m)]\right)\Big|_m \approx 0.5\left(1 + \frac{\Delta[\log IDC(m)]}{\Delta[\log m]}\right)\Big|_m$$

where

$$\begin{aligned} IDC(m) &= \frac{var(\sum_{j=1}^{m} X_j)}{E\left[\sum_{j=1}^{m} X_j\right]} \\ \Delta\log[IDC(m)] &= \log[IDC(m_2)] - \log[IDC(m_1)] \\ \Delta\log(m) &= \log(m_2) - \log(m_1). \end{aligned}$$

m_1 and m_2 (where $m_2 > m_1$) are two time intervals. The parameter K is a continuous function of samples of X_j, $j = 1, \cdots, J$, where J is the number of samples that is required in order to make a good estimation of K. From the Figure 6.2, it can be seen that the selection of $J \approx 8000$ results in a close estimation of the value of K, for all values between 0.5 and 1.

Now, let us assume J samples of $\mathbf{X}$ are used in estimation of K. Consider two estimated values at time $i-1$ and i, i.e. $\hat{K}_{i-1}$ and $\hat{K}_i$. Samples of X_{i-J-1} to X_{i-1} and X_{i-J} to X_i have been used to estimate $\hat{K}_{i-1}$ and $\hat{K}_i$, respectively. Since both estimations have inputs from X_{i-J} to X_{i-1} in common, it is possible to use previous estimated sample $\hat{K}_{i-1}$ in conjunction with X_{i-J-1} and the newly arrived sample X_i to estimate K_i. As a result, the function of K has three variables and it can be expressed as

$$\hat{K}_i = f(X_i,\ X_{i-J},\ \hat{K}_{i-1}),$$

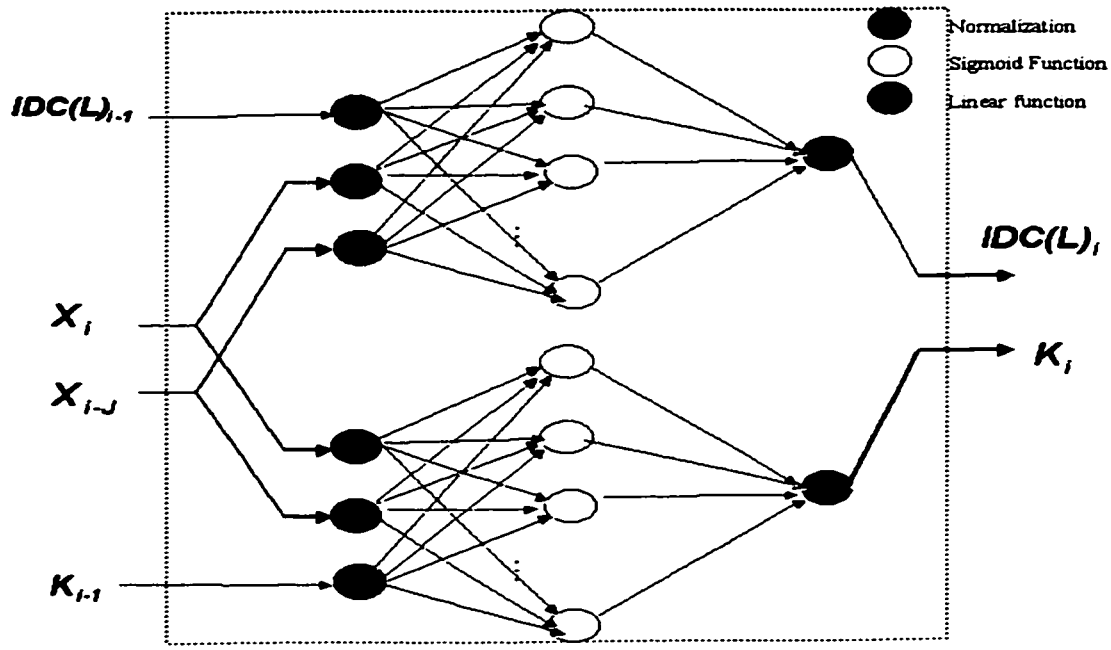

Figure 6.4: A neural network system consists of two sets of linear combinations of sigmoid functions that approximate K and IDC(L).

which can be approximated by a single hidden layer neural network[2]. Similarly, the function of IDC(L) can be expressed as

$$I\hat{D}C(L)_i = f(X_i,\ X_{i-J},\ I\hat{D}C(L)_{i-1}).$$

Figure 6.4 shows a neural network system that approximates K and IDC(L)[3]. Each parameter is approximated by a linear combination of sigmoid functions.

[2]Narendra [67] has argued that, in general, under weak conditions on the function $f(\cdot)$, a multi-layer neural network can be constructed to approximate such mapping over compact sets.

[3]In order to avoid propagation of an estimation error over a long period of time, periodically, the actual value of K is calculated and fed into the estimator.

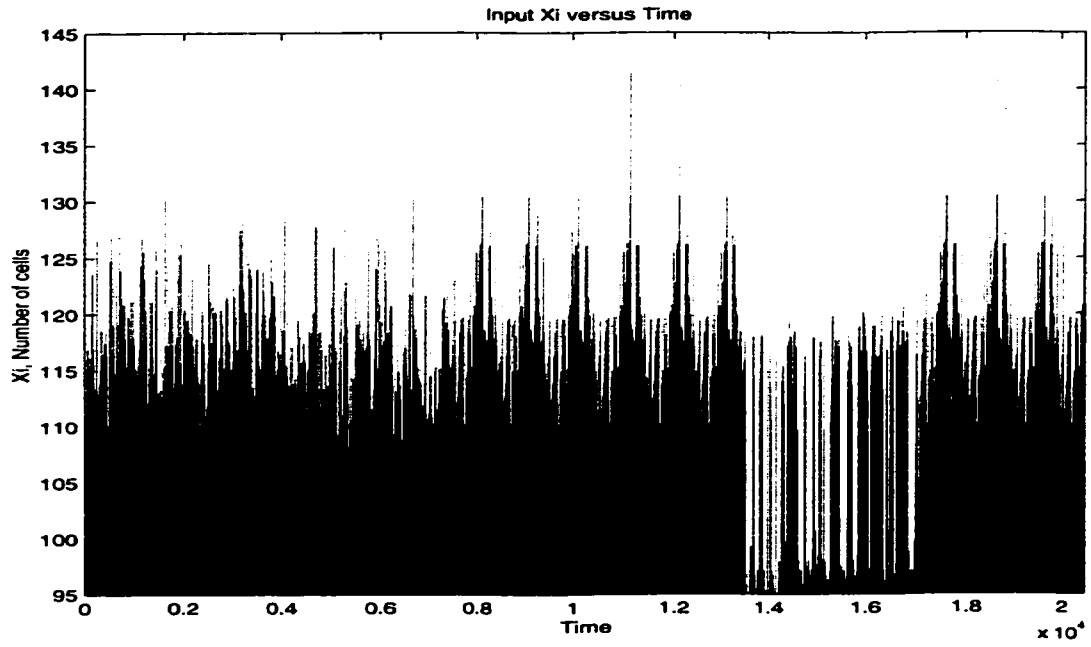

Figure 6.5: Time series of $\mathbf{X}$ that has been used for training of the approximator.

6.2.4 Performance Evaluation of the Proposed Approximator

To examine the performance of the proposed approximator of Figure 6.4, we generate two sets of data. One set for training the neural network (i.e. adjusting the weights, bias values, etc.) and the other set to test the performance of the trained approximator. Each set consists of traffic patterns generated by MMPP1, MMPP2, FBM with H=0.6 and FBM with H=0.9 and their mixtures. Figure 6.5 shows time series of $\mathbf{X}$ that has been used as input of the training signal. Figures 6.6 and 6.7 show the values of K and and IDC(L) of $\mathbf{X}$ as the output of the training set.

Once again, back-propagation is used to train the neural network (or adjust the coefficients of the approximator). The average mean-square error for an input-output pair has been plotted in Figure 6.8 versus training epoch. Note that the converged value of MSE for IDC(L) is higher than that of K. This is because IDC(L) ranges

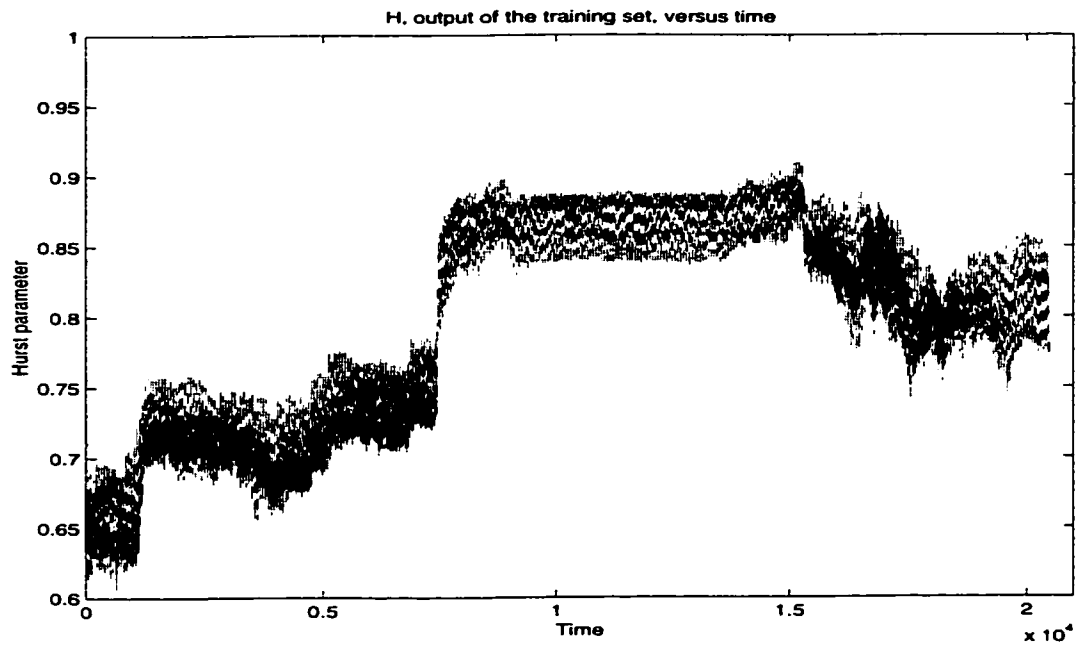

Figure 6.6: The value of K for the time-series $\mathbf{X}$ that has been used as the output of the training set.

from 1 to 200, whereas the value of K ranges from 0.5 and 1.

Figure 6.9 shows the time series of the mix traffic that has been used to test the performance of the approximator. Figures 6.10 and 6.11 shows the output of the neural network compared to the actual one. Figures 6.12 and 6.13 (which are indeed two portions of Figure 6.10) show closer comparison of the approximated K with the actual one. Similarly, Figures 6.14 and 6.15 show a closer comparison of the approximated IDC(L) with the actual one.

6.3 Summary

In this chapter, we addressed the real-time estimation of the parameters of the traffic indicator. The four parameters of the traffic indicator are estimated from the time

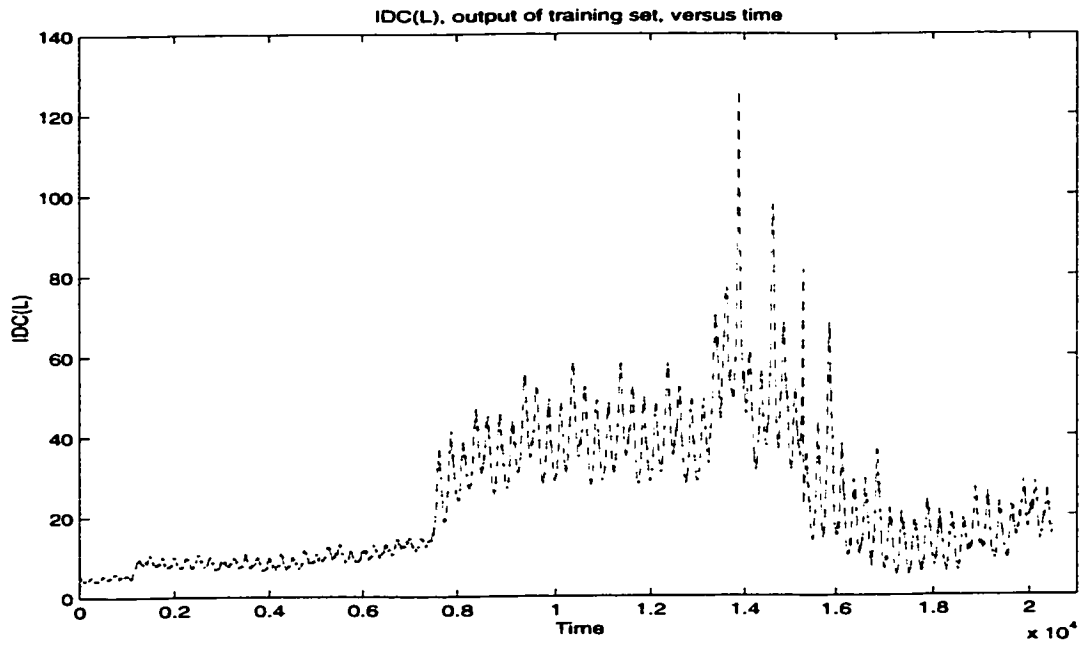

Figure 6.7: IDC(L) of the time-series $\mathbf{X}$ that form the output part of the training set.

series of cell counts. Load and IDC(1) are estimated from the mean and the variance of the time series. Classical approaches in estimating IDC(L) and the Hurst parameter (or more generally K) suffer from imprecision, computational complexity or inefficiency. To cope with the shortcomings of the classical approaches, we deployed a neural network system consisting of a set of sigmoid activation functions that learn the functional relationship between the time series $\mathbf{X}$ and IDC(L) (or K). To reduce the number of inputs and, hence, reduce the complexity, we used the previously estimated sample in estimating new samples. To avoid error propagation as a result of using the old estimated value, periodically, the actual calculated sample of IDC(L) (or K) are fed into the neural network to reset the propagated error. The results show that the estimated values of the K and IDC(L) using a neural network system closely follow the actual calculated values while estimation can be performed real-time with

less computational complexity.

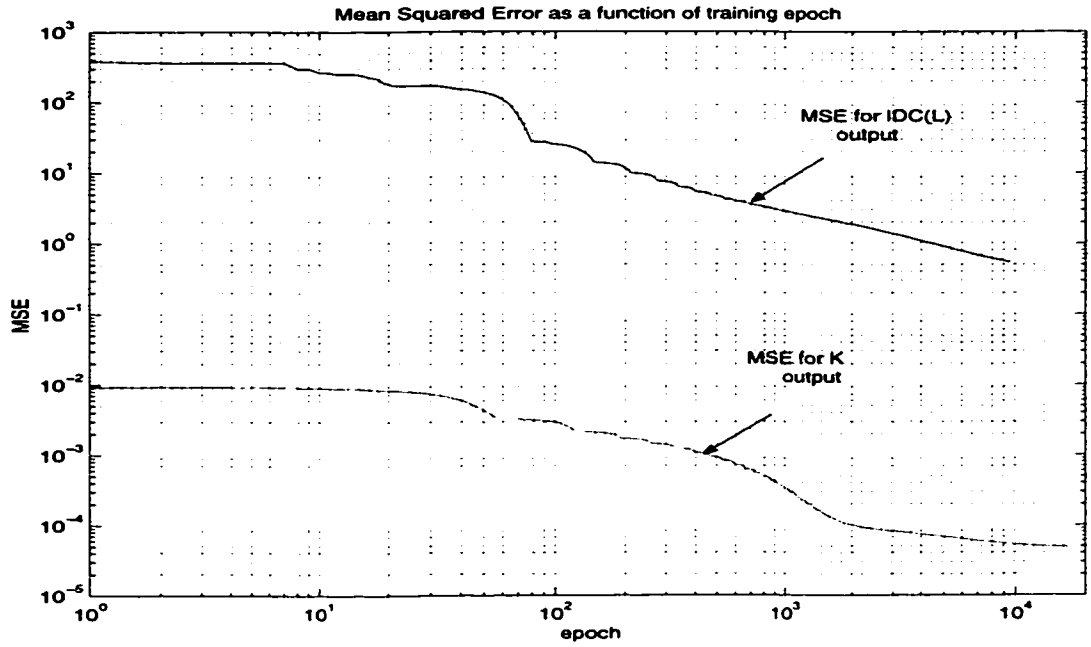

Figure 6.8: Average MSE of the outputs as a function of training epoch.

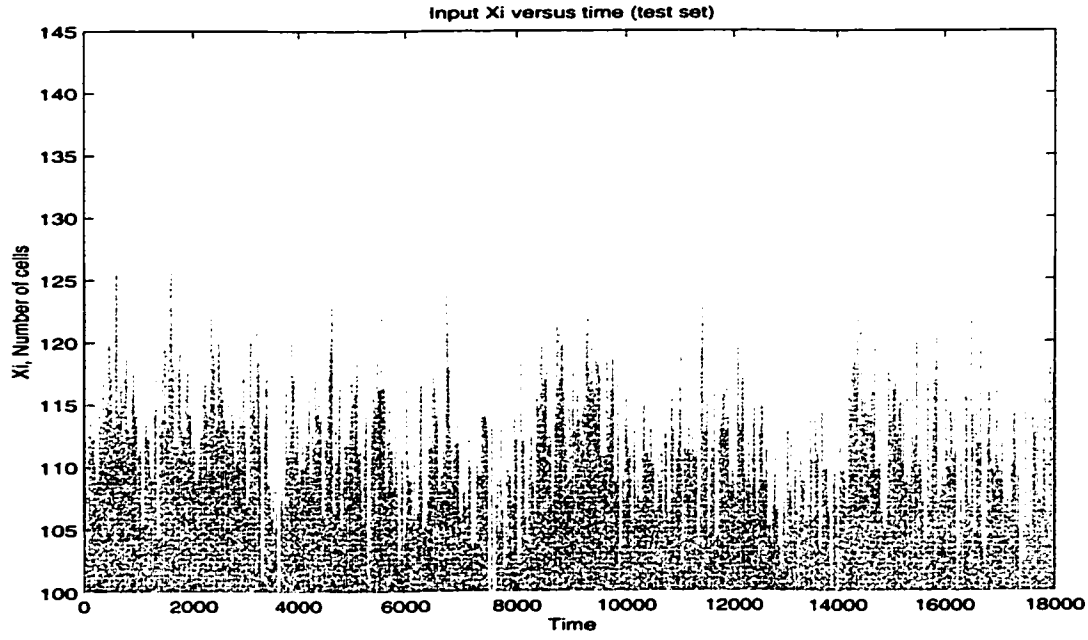

Figure 6.9: Time series of $\mathbf{X}$ that has been used to test the approximator.

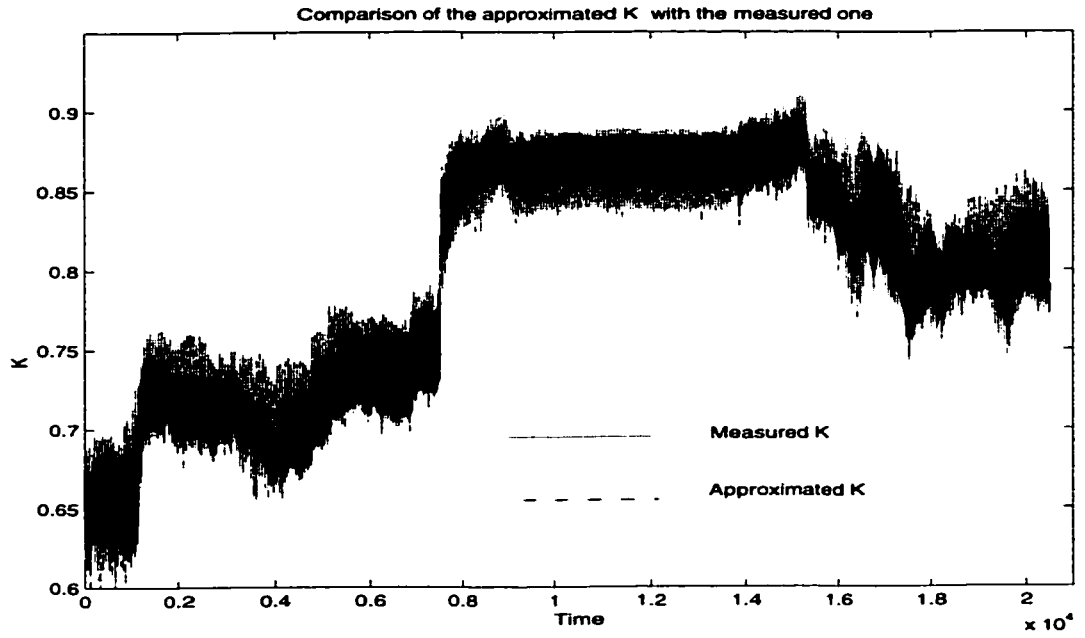

Figure 6.10: Comparison of the approximated value for K with the measured one.

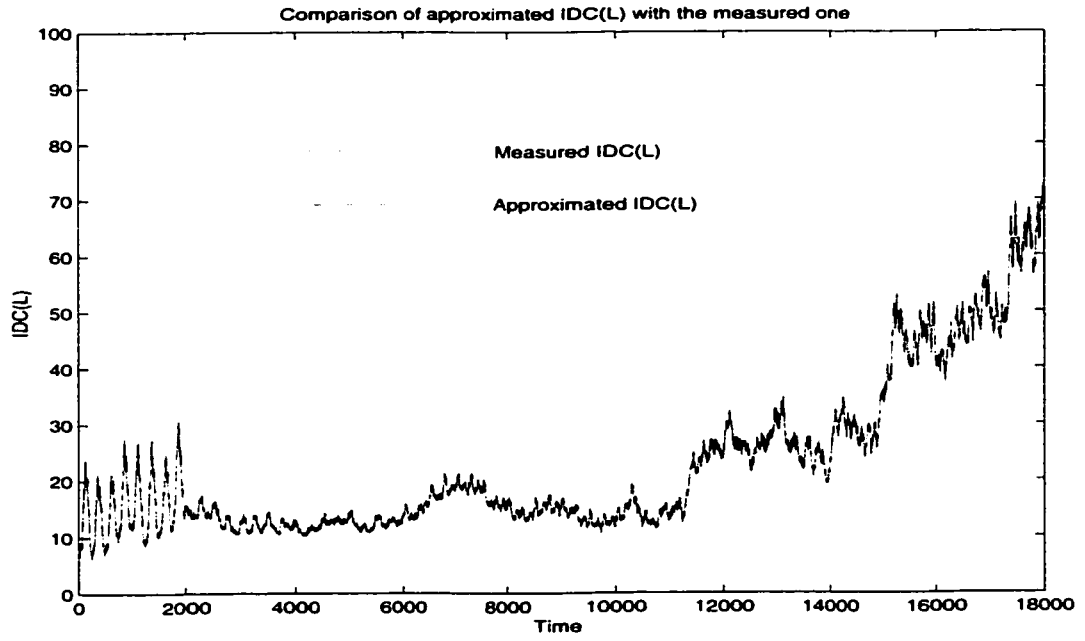

Figure 6.11: Comparison of the approximated IDC(L)with the measured one.

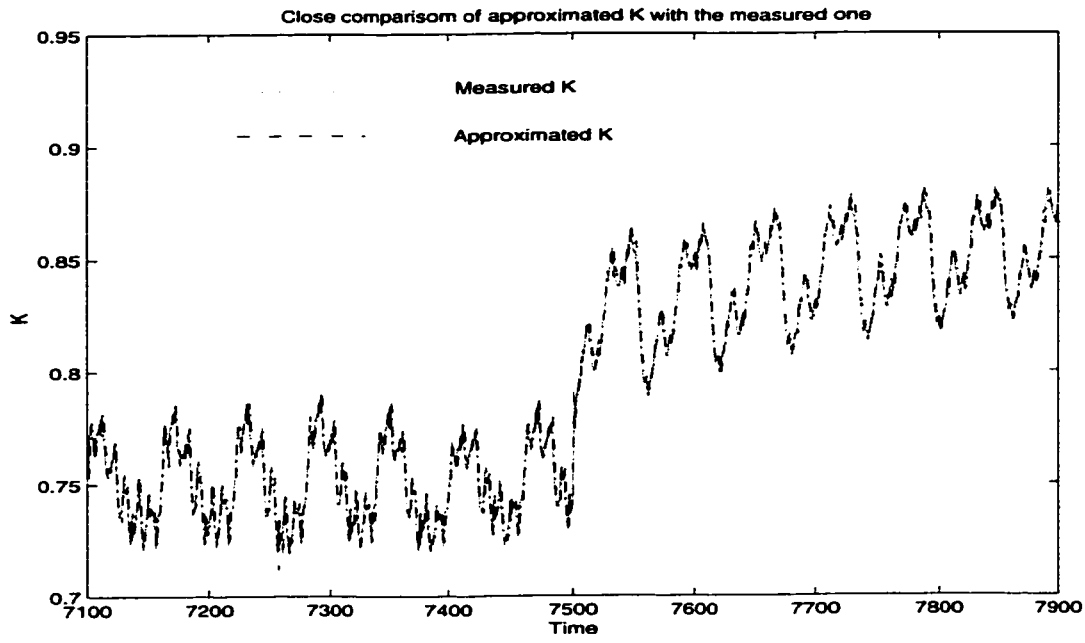

Figure 6.12: A close comparison of the approximated value for K with the measured one for the shown time scale.

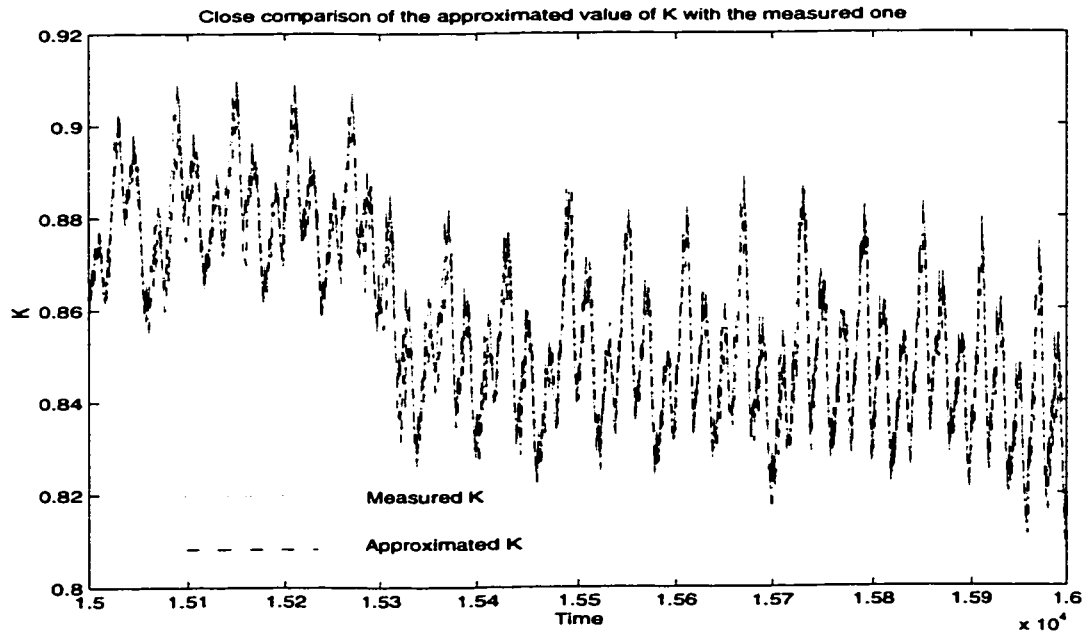

Figure 6.13: A close comparison of the approximated value for K with the measured one for the shown time scale.

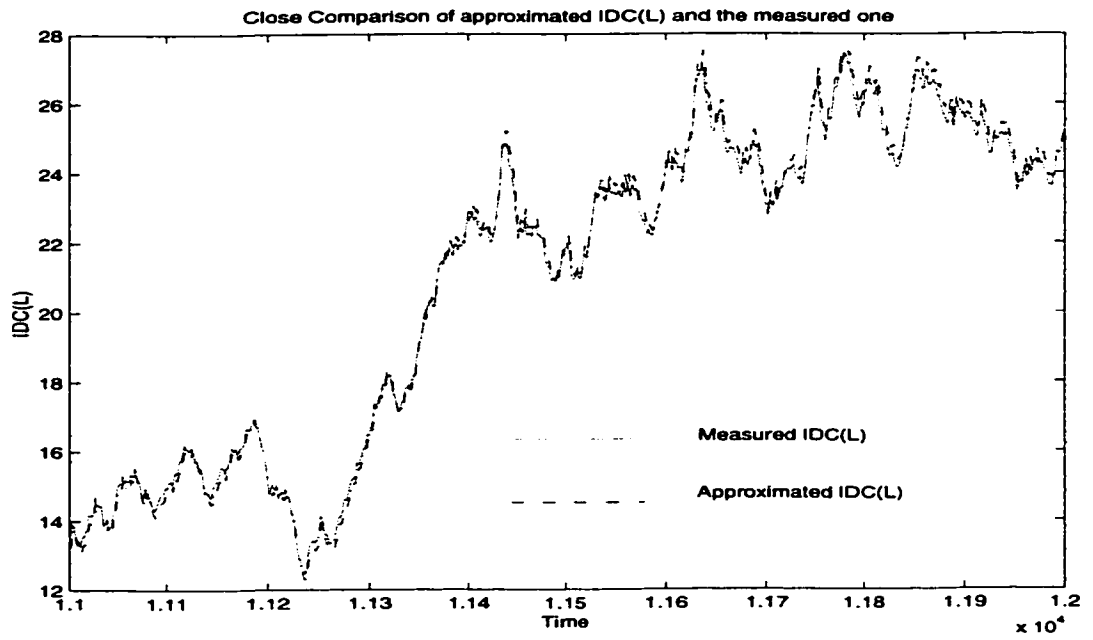

Figure 6.14: A close comparison of the approximated IDC(L) with the measured one for the shown time scale.

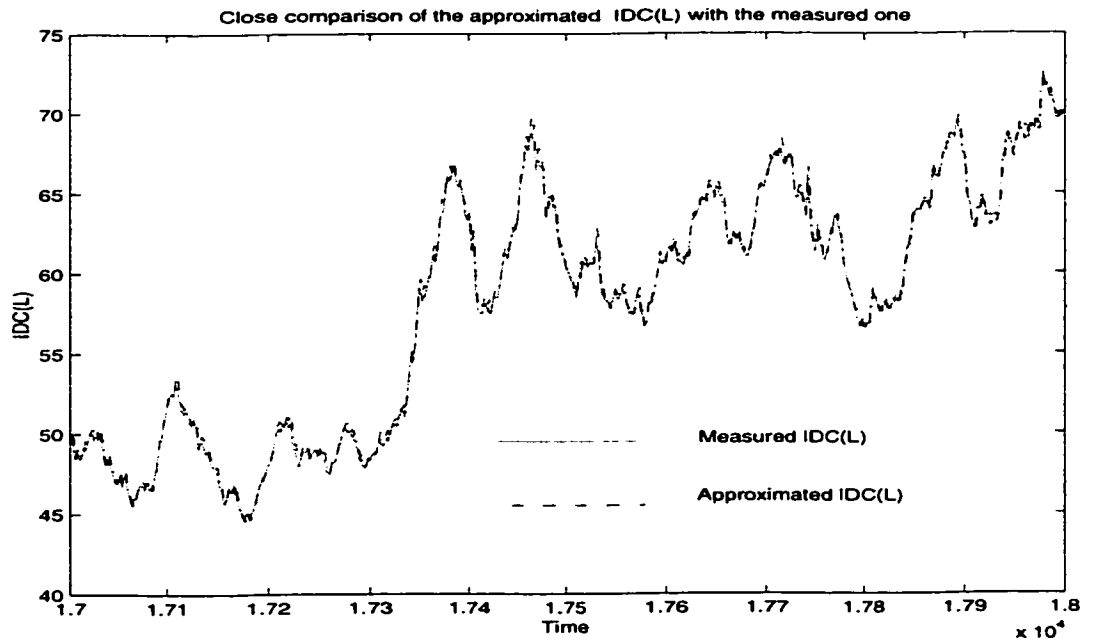

Figure 6.15: A close comparison of the approximated IDC(L) with the measured one for the shown time scale.

Chapter 7

Application of the Proposed CLR Approximator in ATM CAC

An ATM connection may traverse a set of switching nodes in the network. To set up a connection, resources need to be reserved at each queuing point to guarantee the QoS. Under the assumption of Chapter 3, in this Chapter we give an illustrative example on application of the proposed CLR approximator in formulating an admission strategy. We also assume that traffic contract includes the parameter K in addition to the mean rate and the peak rate. The parameter K has been obtained from off-line characterization of the traffic source.

Ideally, an admission decision can be viewed as a solution to a queuing problem. To decide whether a connection can be admitted, the queue need be solved to predict the CLR for all the connections including the new one. If the predicted CLR is less than an acceptable threshold, the new connection is accepted. In ATM networks, a solution to this problem has remained an open issue due to its complexity, specially when a complex traffic model is assumed.

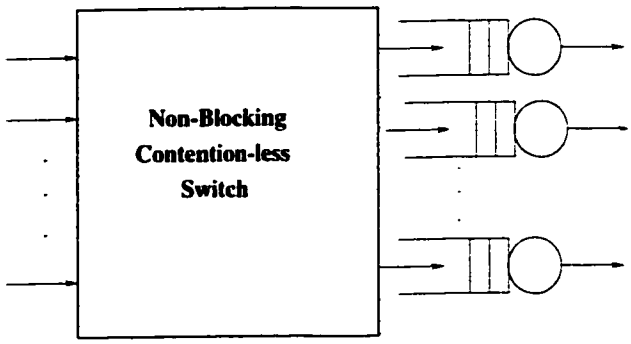

Figure 7.1: An ATM switch with output queuing.

In this chapter, we use the results of the previous chapters on both real-time measurements of the traffic indicator and the CLR approximation to propose a new admission strategy that maximizes link utilization while maintaining the CLR less than a set threshold. We then assess the performance of the proposed admission scheme using OPNET simulation and compare its performance with that of the Equivalent Capacity scheme.

7.1 Introduction

Once a switch receives a connection admission request to an output, it executes an admission procedure to decide whether to accept or reject the connection. It is accepted if the CLR of all connections (including the new one) can be maintained at an acceptable level. Let us consider a non-blocking contention-less ATM switch of Figure 7.1. CLR occurs in an output queue when the the multiplexed traffic from all the inputs sources is bursty and there is a limited buffer capacity. Let us assume that traffic of all sources that arrives to the queue requires the CLR of less than a predefined threshold, for instance 10^{-6}. The question that CAC tries to answer is: would the CLR exceed the threshold if a new connection were added to the existing

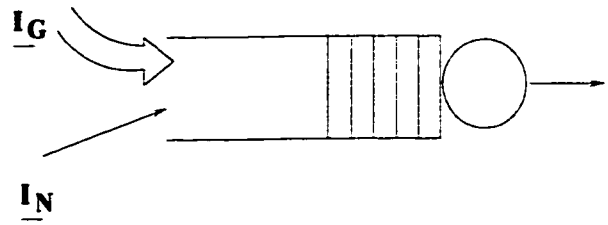

Figure 7.2: Admission procedure attempts to solve a queuing problem with two traffic indicators.

ones? If the answer is yes, the connection is rejected; otherwise, it is accepted. The engineering challenge requires developing an admission strategy that explores fully statistical multiplexing gain and, hence, maximize link utilization while maintaining CLR for all connections below the threshold. As discussed in Chapter 3, accurate prediction of CLR for the aggregate traffic of both the existing connections and the new one is the key in developing an efficient admission strategy. For the non-blocking switch of Figure 7.1, the CAC algorithm is applied to each output queue. If we isolate each queue from the rest of the switch, we will obtain the queuing model of Figure 7.2. The model is known as multiplexer (or ATM multiplexer) in which a number of traffic sources feed a finite queue that is being served at a constant rate.

The proposed approach first characterizes the existing connections by a traffic indicator $\underline{I_G}$ and the new connection by a traffic indicator $\underline{I_N}$[1]. Then, it solves the queue for the aggregate indicator of traffic. The solution uses the learning capability of a neural network in capturing the functional relationship between the aggregate traffic indicator and the measured CLR as discussed in Chapter 5.

[1]An alternative static approach that we previously proposed uses traffic classification and maps the total number of connections from all classes to a traffic indicator using a fuzzy inference system [92].

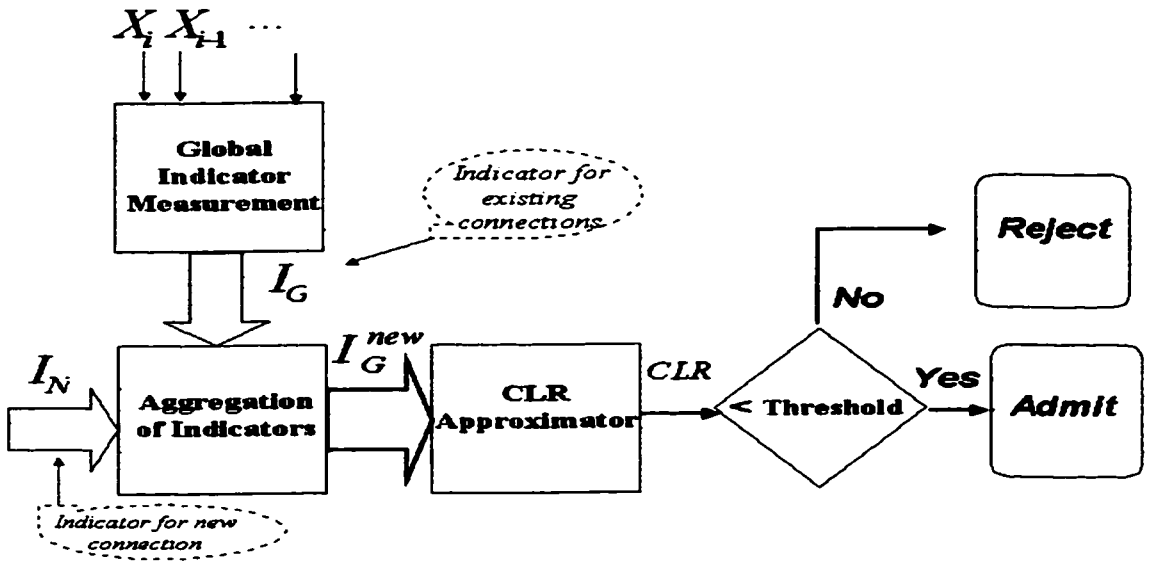

Figure 7.3: Proposed admission procedure that uses the traffic indicator in CLR prediction.

7.2 CAC Procedure

Figure 7.3 shows the proposed CAC strategy. $\mathbf{I_G}$ is the measured indicator of traffic that represents the existing connections to the queue. $\mathbf{I_N}$ is the traffic indicator for the new requested connection. The CAC procedure is as follows:

a): A user requests a connection to a queue and declares the traffic indicator of the new connection, i.e. $\underline{\mathbf{I}}_N$.

b): Traffic indicator for the requested output queue, i.e., $\underline{\mathbf{I_G}}$, is measured from the time series $\mathbf{X}$ using approaches discussed in Chapter 6.

c): The two traffic indicators, $\underline{\mathbf{I_G}}$ and $\underline{\mathbf{I}}_N$, are aggregated using a set of equations that will be described in a sequel, i.e., equation (7.12).

d): The CLR is predicted from the aggregate traffic indicator and the given buffer size.

e): If the CLR is less than the predefined threshold, say 10^{-6}, the new connection is admitted; otherwise, it is rejected.

In the following sections, we first describe the aggregation of two traffic indicators and then give an illustrative example in which we compare the performance of the proposed CAC with that of equivalent capacity.

7.3 Deriving the Aggregate Traffic Indicator

Let us first assume that the new connection has been admitted to the queue. In the ith time interval, the time series of the total traffic to the queue can be represented by

$$X_i = X_{gi} + X_{ni}$$

where X_{gi} and X_{ni} represent the number of cells that arrived in the ith interval from all the existing connections and the newly admitted one, respectively[2]. Aggregate traffic load is given by

$$\begin{aligned} Load &= E[X_i]/C = (E[X_{gi}] + E[X_{ni}])/C \\ &= (\lambda_g + \lambda_n)/C \end{aligned} \tag{7.1}$$

where λ_g and λ_n are the average cell rates for all the existing connections and the new connection, respectively, and C is the link transmission rate. IDC value of the

[2]The subscript g indicates the global indicator of the traffic and the subscript n represents the new connection.

aggregate traffic for the interval-size of k is given by

$$\begin{aligned} IDC(k) &= \frac{var(\sum_{i=1}^{k} X_i)}{E[\sum_{i=1}^{k} X_i]} \\ &= \frac{var\left(\sum_{i=1}^{k}(X_{gi}+X_{ni})\right)}{E\left(\sum_{i=1}^{k}(X_{gi}+X_{ni})\right)} \\ &= \frac{var(Y_{gk}+Y_{nk})}{k(\lambda_g+\lambda_n)}. \end{aligned}$$

Since $Y_{gk} = \sum_{i=1}^{k} X_{gi}$ and $Y_{nk} = \sum_{i=1}^{k} X_{ni}$ are not correlated to each other,

$$IDC(k) = \frac{var(Y_{gk}) + var(Y_{nk})}{k(\lambda_g+\lambda_n)}. \tag{7.2}$$

On the other hand, IDC for the existing connections and the new one are given by

$$IDC_g(k) = \frac{var(\sum_{i=1}^{k} X_{gi})}{E[\sum_{i=1}^{k} X_{gi}]} = \frac{var(Y_{gk})}{k\lambda_g} \tag{7.3}$$

$$IDC_n(k) = \frac{var(\sum_{i=1}^{k} X_{ni})}{E[\sum_{i=1}^{k} X_{ni}]} = \frac{var(Y_{nk})}{k\lambda_n}. \tag{7.4}$$

The aggregate of K is given by

$$\begin{aligned} K_{aggregate} &= \frac{1}{2}(1 + IDC\ slope) \\ &= \frac{1}{2}\left(1 + \lim_{\Delta k \to 0} \frac{IDC(k_2) - IDC(k_1)}{\Delta k}\right) \end{aligned} \tag{7.5}$$

where $\Delta k = |k_2 - k_1|$ and $k_2 > k_1$. Replacing (7.2) into (7.5) results

$$\begin{aligned} K_{aggregate} &= \frac{1}{2}\left[1 + \lim_{\Delta k \to 0} \frac{1}{\Delta k}\left(\frac{var(Y_{gk_2}) + var(Y_{nk_2})}{k_2(\lambda_g+\lambda_n)} - \frac{var(Y_{gk_1}) + var(Y_{nk_1})}{k_1(\lambda_g+\lambda_n)}\right)\right] \\ &= \frac{1}{2}\left[1 + \lim_{\Delta k \to 0} \frac{1}{\Delta k}\left[\frac{var(Y_{gk_2})}{k_2(\lambda_g+\lambda_n)} - \frac{var(Y_{gk_1})}{k_1(\lambda_g+\lambda_n)} + \frac{var(Y_{nk_2})}{k_2(\lambda_g+\lambda_n)} - \frac{var(Y_{nk_1})}{k_1(\lambda_g+\lambda_n)}\right]\right] \\ &= \frac{1}{2}\left[1 + \lim_{\Delta k \to 0} \frac{1}{\Delta k}\left[\frac{\lambda_g}{\lambda_g+\lambda_n}\left(\frac{var(Y_{gk_2})}{k_2\lambda_g} - \frac{var(Y_{gk_1})}{k_1\lambda_g}\right)\right.\right. \\ &\qquad \left.\left. + \frac{\lambda_n}{\lambda_g+\lambda_n}\left(\frac{var(Y_{nk_2})}{k_2\lambda_n} - \frac{var(Y_{nk_1})}{k_1\lambda_n}\right)\right]\right] \\ &= \frac{1}{2}\left[1 + \frac{\lambda_g}{\lambda_g+\lambda_n}\zeta_g + \frac{\lambda_n}{\lambda_g+\lambda_n}\zeta_n\right] \end{aligned} \tag{7.6}$$

where ζ_g and ζ_n are defined as

$$\zeta_g = \lim_{\Delta k \to 0} \frac{1}{\Delta k} \left(\frac{var(Y_{gk_2})}{k_2 \lambda_g} - \frac{var(Y_{gk_1})}{k_1 \lambda_g} \right) \tag{7.7}$$

$$\zeta_n = \lim_{\Delta k \to 0} \frac{1}{\Delta k} \left(\frac{var(Y_{nk_2})}{k_2 \lambda_n} - \frac{var(Y_{nk_1})}{k_1 \lambda_n} \right) \tag{7.8}$$

Using (7.3), the parameter K for the existing connections is given by

$$\begin{aligned} K_g &= \frac{1}{2} \left[1 + \lim_{\Delta k \to 0} \frac{1}{\Delta k} (IDC_g(k_2) - IDC_g(k_1)) \right] \\ &= \frac{1}{2} \left[1 + \lim_{\Delta k \to 0} \frac{1}{\Delta k} \left(\frac{var(Y_{gk_2})}{k_2 \lambda_g} - \frac{var(Y_{gk_1})}{k_1 \lambda_g} \right) \right] \\ &= \frac{1}{2} [1 + \zeta_g] \end{aligned} \tag{7.9}$$

Similarly, using (7.4), the parameter K for the new connection is given by

$$K_n = \frac{1}{2} [1 + \zeta_n]. \tag{7.10}$$

Combining (7.6), (7.9) and (7.10) result in

$$K_{aggregate} = \frac{\lambda_g K_g + \lambda_n K_n}{\lambda_g + \lambda_n}. \tag{7.11}$$

Now, let us assume that the traffic indicator $\underline{\mathbf{I}_G} = \{Load_g,\ K_g,\ IDC_g(1),\ IDC_g(L)\}$ has been measured and the traffic indicator of the new connection has been given by $\underline{\mathbf{I}_N} = \{Load_n,\ K_n,\ \sigma_n\}$, where $Load_n = \frac{\lambda_n}{C}$ (with λ_n as the mean cell rate and C the link transmission rate), σ_n is the cell rate standard deviation and K_n is the declared burstiness parameter for the new connection. Then, the parameters of the aggregate traffic indicator can be given as

$$\begin{aligned} Load_{aggregate} &= (\lambda_g + \lambda_n)/C \\ K_{aggregate} &= \frac{\lambda_g K_g + \lambda_n K_n}{\lambda_g + \lambda_n} \\ IDC_{aggregate}(1) &= \frac{\lambda_g IDC_g(1) + \sigma_n^2}{\lambda_g + \lambda_n} \\ IDC_{aggregate}(L) &= \frac{var(\sum_{i=1}^{L} (\mathbf{X}_{gi} + \mathbf{X}_{ni})}{E[\sum_{i=1}^{L} (\mathbf{X}_{gi} + \mathbf{X}_{ni}]} \approx \frac{L\sigma_n^2}{\lambda_n + \lambda_g} + IDC_g(L). \end{aligned} \tag{7.12}$$

The approximation of the last equation assumes $\lambda_g >> \lambda_n$.

7.4 Performance Assessment of the Proposed CAC

We use simulation using OPNET to assess performance of the proposed CAC. Let us assume that there are three different VBR classes that are entered to the network. Let us consider that the connection admission requests from VBR1, VBR2 and VBR3 classes to a given output of a switch are generated based on Poisson models with average rates of 100, 10 and 40 calls/count-interval, respectively. The holding time, or the time a connection stays in the network on average, is exponentially distributed with average of 180, 240 and 30 count-intervals for VBR1, VBR2 and VBR3 classes, respectively. That means if the count interval is assumed to be one second, the average holding time for VBR1 sources is 3 minutes, for VBR2 sources is 4 minutes and for VBR3 sources is 30 seconds. During the holding time, each VBR source is modeled by an On-Off source with exponential sojourn time. Table 7.1 shows the peak rate, R, average rate, m, and burst size, β, for each On-Off source. In addition to these parameters, we assume that each source declares our defined parameter K. Under the assumption of On-Off, the variance, σ_n is given by [24]

$$\sigma_n = (m(R-m))^{0.5}$$

The Equivalent Capacity approach uses these parameters to allocate bandwidth to each one of the classes. Table 7.1 also shows the traffic indicator for each class of traffic assuming that $C = 1.22 \times 10^5$ cells/sec (or 51.84 Mbps).

A connection request from each class is submitted to the admission control. Once a request is admitted, it uses link capacity for an exponentially distributed holding time

Class	*Parameters of Equivalent Capacity*			*Traffic Indicator* $\underline{\mathbf{I_N}}$		
	R	m	β	$Load = \frac{m}{C}$	K_n	$\sigma_n = (m(R-m))^{0.5}$
	cells/sec	cells/sec	cells			
VBR1	132.5	53.01	2.33×10^2	0.0004345	0.6	64.9
VBR2	43773	4377.36	6.74×10^4	0.0359	0.75	41338
VBR3	6910	354.52	1.06×10^4	0.002906	0.9	1524.5

Table 7.1: Traffic parameters of three VBR classes that were used in the case study.

during which cells are generated based on the parameters described in Table 7.1. The first graphs of Figures 7.4, 7.5 and 7.6 show the number of connection requests as a function of simulation time for VBR1, VBR2 and VBR3 classes, respectively. The second and the third graphs of these figures show the admitted number of connections and the rejected number of connections, respectively. Figures 7.7 shows the total number of admitted VBR1 sources versus simulation time that are in the system and use the output resources. Figures 7.8 and 7.9 show the same quantity for VBR2 and VBR3, respectively. The aggregate traffic indicator is obtained from the measured indicator $\underline{\mathbf{I_G}}$ and the declared indicator $\underline{\mathbf{I_N}}$ using the sets of equations given in (7.12). Figures 7.10, 7.11, 7.12 and 7.13 show the aggregate indicator of traffic, i.e. load, K, IDC(1) and IDC(L), respectively, versus the simulation time. Figure 7.14 show the approximated CLR for all the existing connections as a function of simulation time. Note that in all of the figures, the y-axis quantities are sampled every one second of the simulation. The first 1000 sec of simulation has been considered as warm-up.

7.4.1 Performance Comparison with Equivalent capacity

Figure 7.15 shows the offered load, i.e. traffic load from all the requested connections, versus the carried load, i.e. the load from the admitted connections, for the simulation period. In the simulation, the link transmission rate of 51.84 Mbps (or 1.22×10^5 cells/sec) has been considered. As seen, the maximum achieved carried load during the simulation time is 0.77. The number of connections that corresponds to this carried load is $N_{\text{VBR1}} = 108$, $N_{\text{VBR2}} = 17$ and $N_{\text{VBR3}} = 39$. Now let us calculate the required bandwidth using the Equivalent Capacity approach. From Chapter 2, the required bandwidth for Equivalent Capacity is given by

$$C' = \min(m' + \alpha'\sigma', \quad N_{\text{VBR1}} \cdot C_{\text{VBR1}} + N_{\text{VBR2}} \cdot C_{\text{VBR2}} + N_{\text{VBR3}} \cdot C_{\text{VBR3}})$$

where

$$\begin{aligned} m' &= N_{\text{VBR1}} \cdot m_{\text{VBR1}} + N_{\text{VBR2}} \cdot m_{\text{VBR2}} + N_{\text{VBR3}} \cdot m_{\text{VBR3}} \\ \sigma'^2 &= N_{\text{VBR1}} \cdot \sigma^2_{\text{VBR1}} + N_{\text{VBR2}} \cdot \sigma^2_{\text{VBR2}} + N_{\text{VBR3}} \cdot \sigma^2_{\text{VBR3}} \\ \alpha' &= \sqrt{2\ln(1/CLR) - \ln 2\pi}. \end{aligned}$$

C_{VBR1}, C_{VBR2} and C_{VBR3} are obtained from

$$\begin{aligned} C &= R\frac{z - 1 + \sqrt{(z-1)^2 + 4\rho y}}{2z} \\ z &= -\ln(CLR)(1-\rho)\beta/B \\ \rho &= m/R \end{aligned}$$

by substituting the values of R, m, β and ρ from table 7.1 for each class of traffic. B is the buffer size in terms of cells and it has been assumed to be 200 in the simulation.

The numerical results of Equivalent Capacity for a single connection as well as multiple connections from each class have been tabulated in Table 7.2. As seen, the required

bandwidth to accommodate $N_{VBR1} = 108$, $N_{VBR2} = 17$ and $N_{VBR3} = 39$ connections is 4.106×10^5 cells/sec. A comparison between this bandwidth and the bandwidth that we considered in the simulation, i.e. 1.22×10^5 cells/sec, reveals that Equivalent capacity needs at least three times more bandwidth in order to accept $N_{VBR1} = 108$, $N_{VBR2} = 17$ and $N_{VBR3} = 39$ connections.

Let us consider another example. For the period of the simulation, on average there are $N_{VBR1} = 160$, $N_{VBR2} = 9$ and $N_{VBR3} = 64$ connections in the system, as shown in Figures 7.7, 7.8 and 7.9, respectively. Equivalent capcity needs a bandwidth of 1.3×10^4 cells/sec for $N_{VBR1} = 160$, 2.3×10^5 cells/sec for $N_{VBR2} = 9$ and 8.3×10^4 cells/sec for $N_{VBR3} = 64$. The total required bandwidth sums to 2.56×10^5 which is double the link bandwidth of 1.22×10^5 cells/sec. As a result, we conclude that on average Equivalent Capacity has half the efficiency of the proposed CAC. There are two reasons for this significant improvement. One is that Equivalent Capacity sums the required bandwidth of each connection and, hence, it does not exploit the statistical multiplexing gain. The other reason is that Equivalent capacity is used for probabilistic services, which require a tight bound on CLR, whereas the proposed approach target predictive data services, which do not require a tight bound on CLR.

7.5 Summary

In this chapter, we proposed an admission control scheme based on the proposed indicator of traffic and the CLR approximator. The proposed admission strategy is highly efficient for predictive services. It maximizes link utilization by allowing as many as possible connections to the network while maintaining an acceptable CLR for

Class		Required Bandwidth (cells/sec)
VBR1	One connection	1.4216×10^2
	108 connections	9.5406×10^3
VBR2	One connection	4.376×10^4
	17 connections	3.494×10^5
VBR3	One connection	6.901×10^3
	39 connections	6.1243×10^4

Table 7.2: The required bandwidth for each class of traffic when bandwidth is allocated by the Equivalent Capacity approach.

all connections. It outperforms the Equivalent Capacity approach. In the illustrative example that we considered, on average, Equivalent capacity has half the efficiency of the proposed CAC. This is because Equivalent Capacity can not maximize the statistical multiplexing gain since it considers each connection in isolation. The advantage of Equivalent Capacity over the proposed approach is that it guarantees a tight bound on CLR.

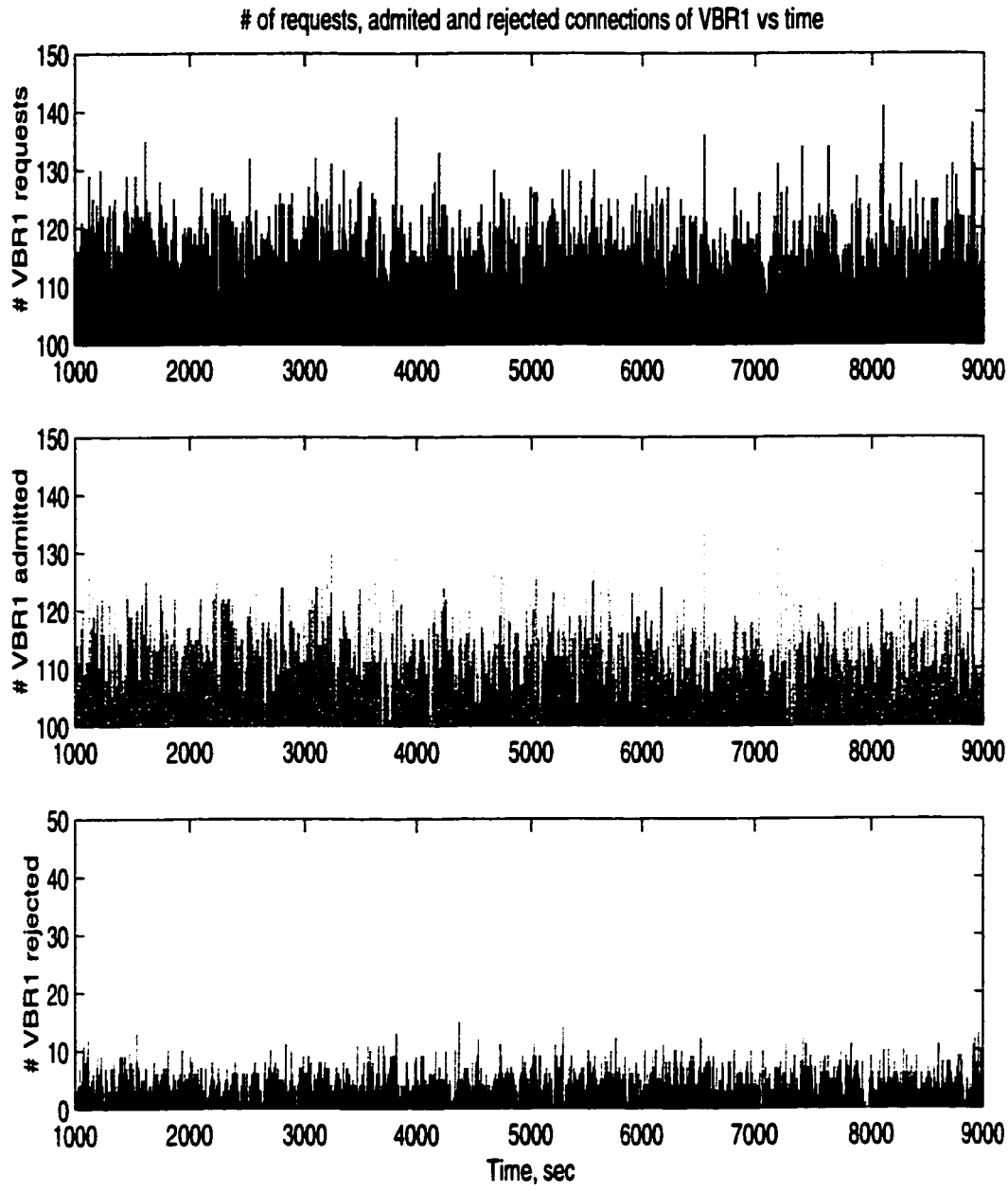

Figure 7.4: The number of requested, admitted and rejected connections for VBR1 class of traffic versus simulation time.

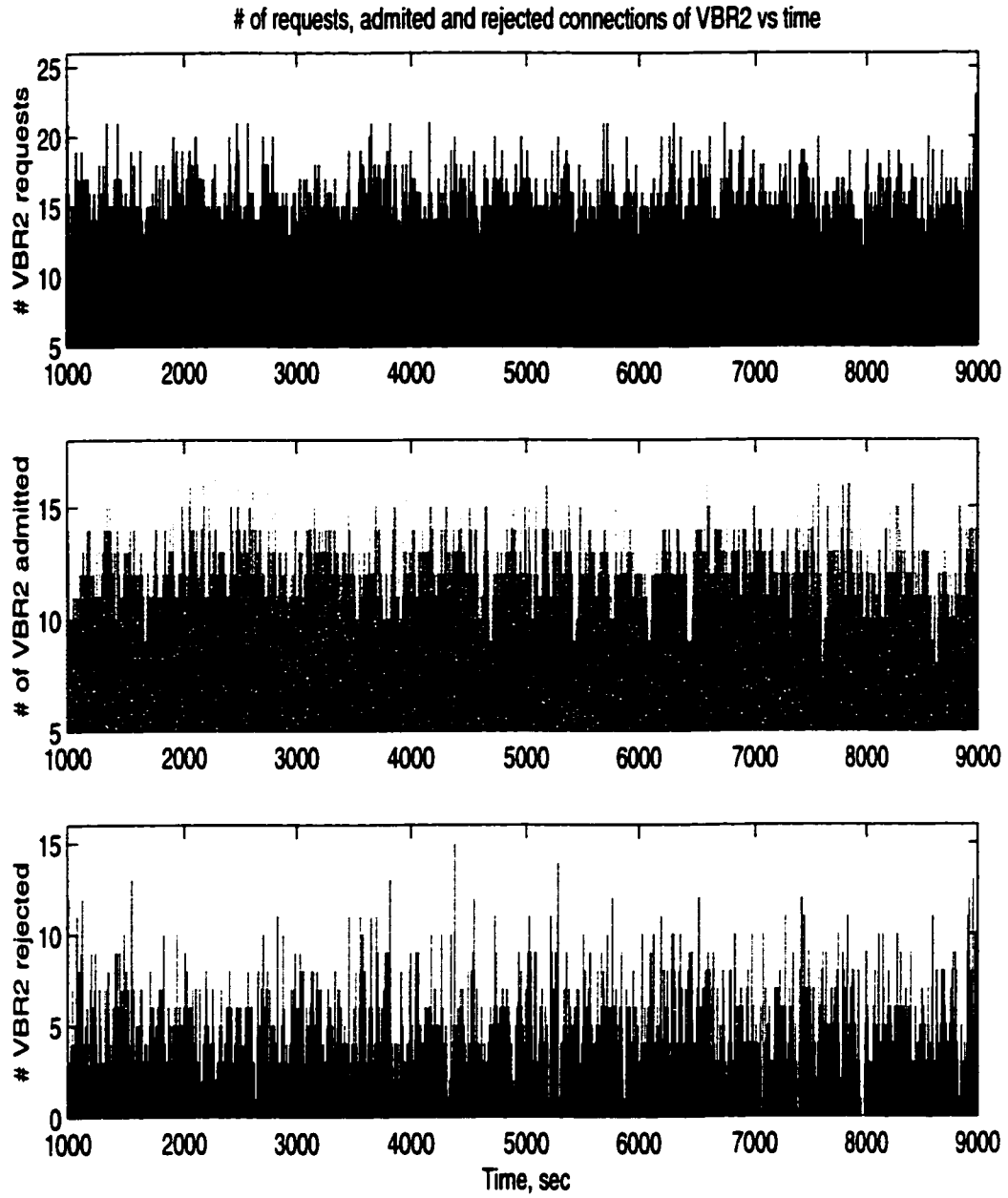

Figure 7.5: The number of requested, admitted and rejected connections for VBR2 class of traffic.

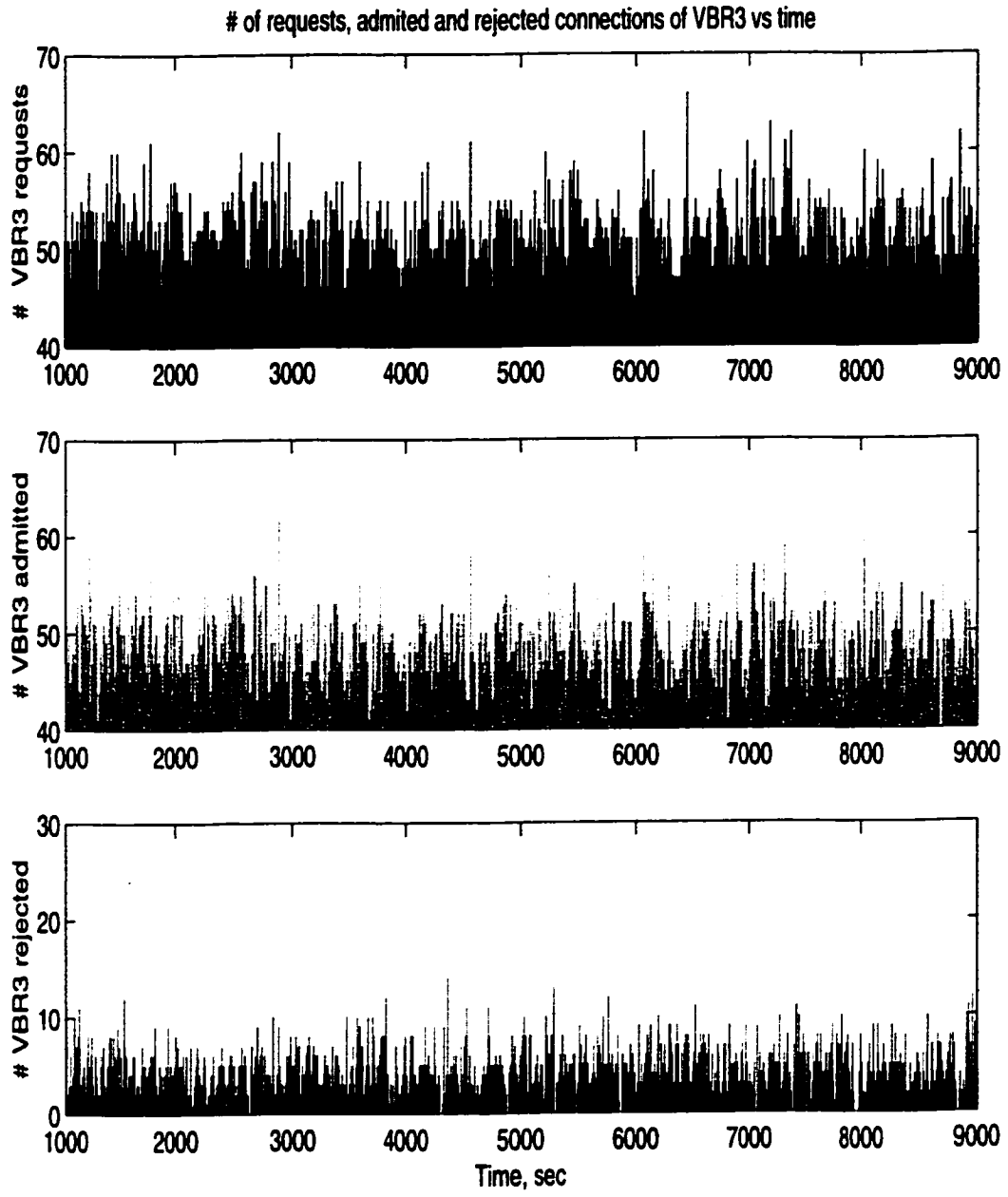

Figure 7.6: The number of requested, admitted and rejected connections for VBR3 class of traffic.

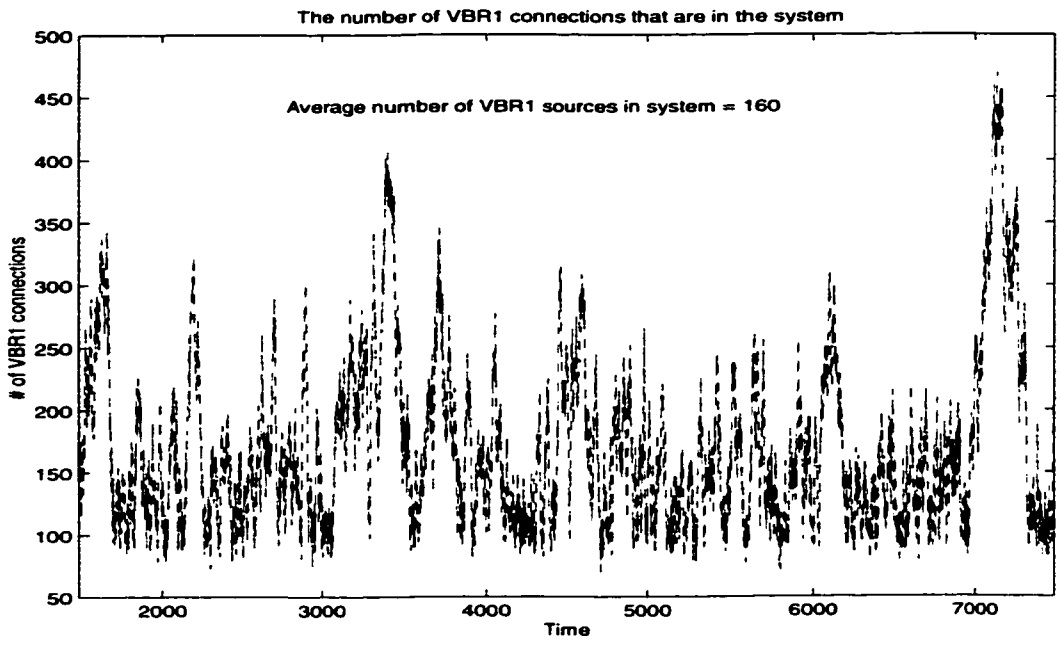

Figure 7.7: The number of VBR1 that are in the system versus simulation time.

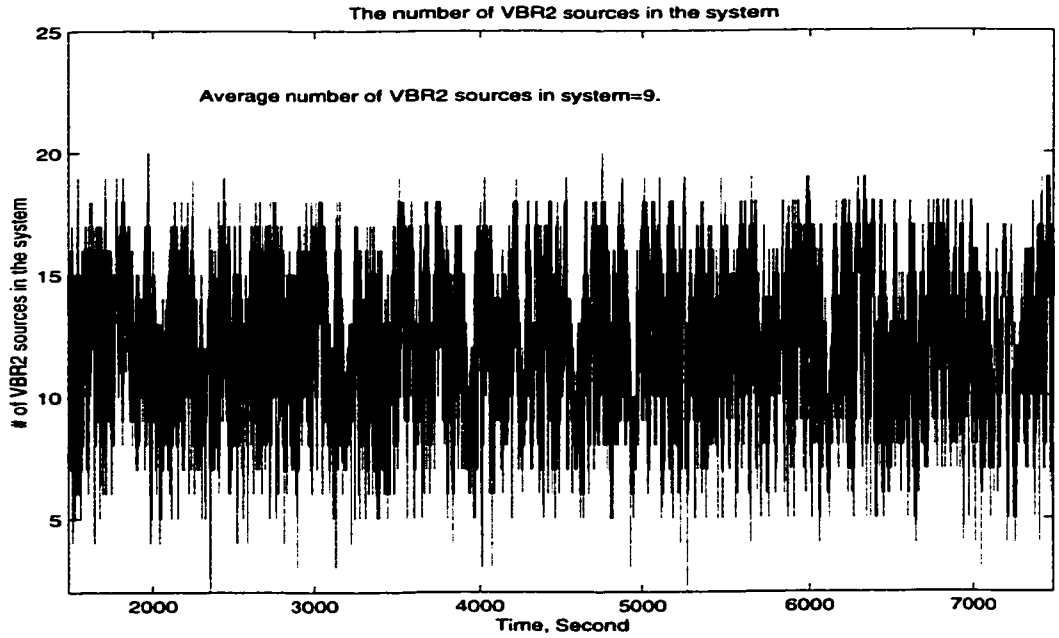

Figure 7.8: The number of VBR2 that are in the system versus simulation time.

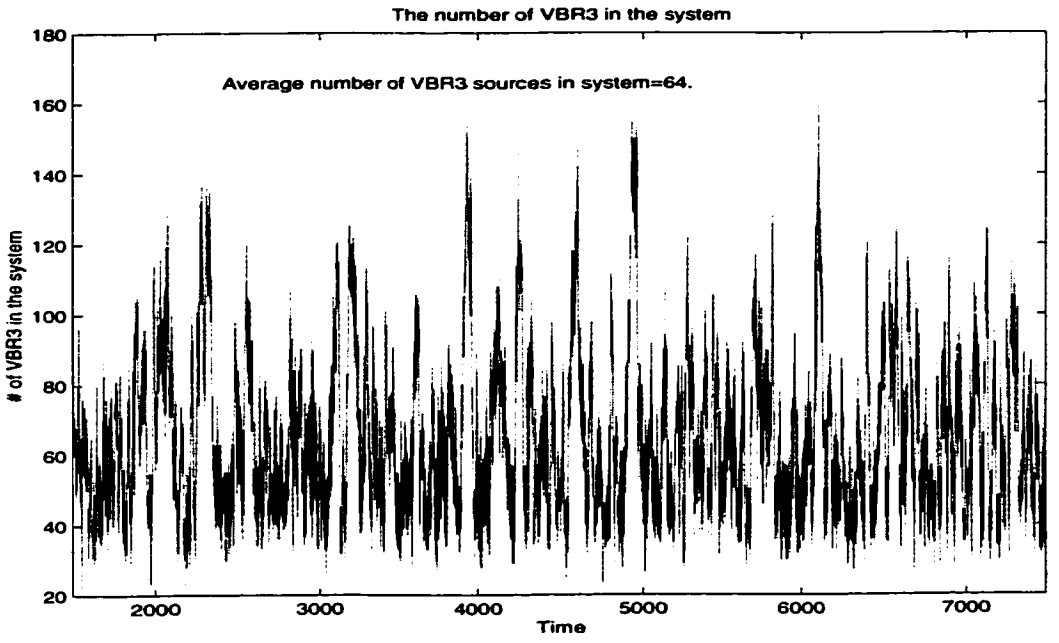

Figure 7.9: The number of VBR3 that are in the system versus simulation time.

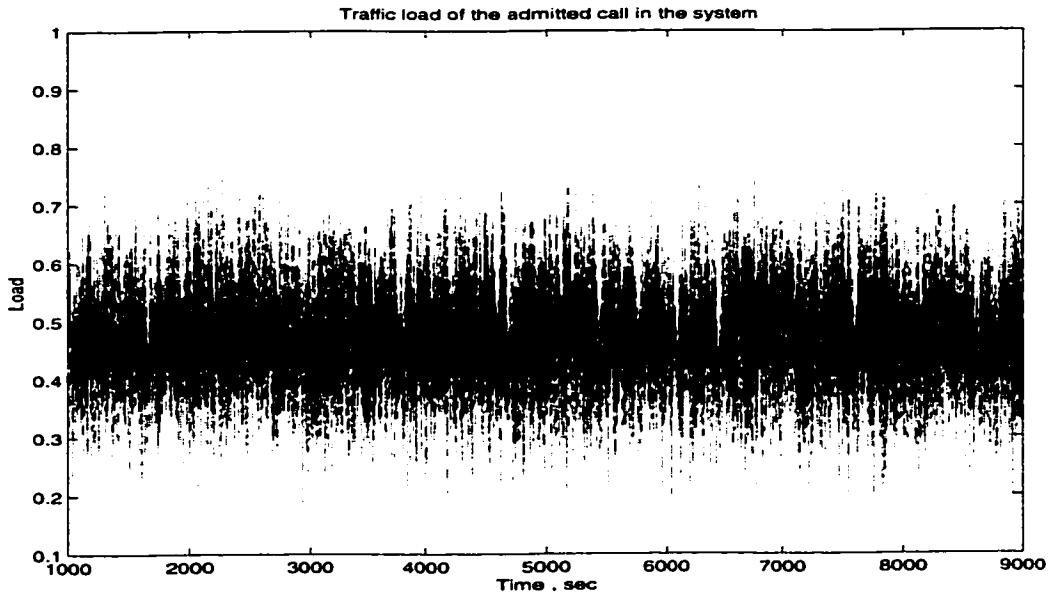

Figure 7.10: Measured traffic load versus simulation time.

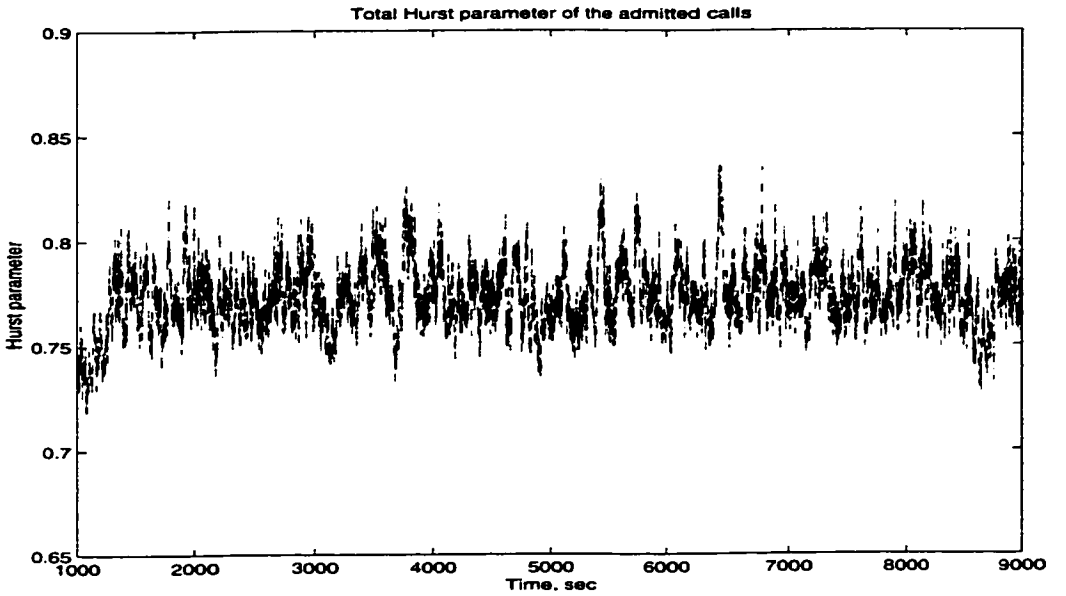

Figure 7.11: Measured parameter K of all connections versus simulation time.

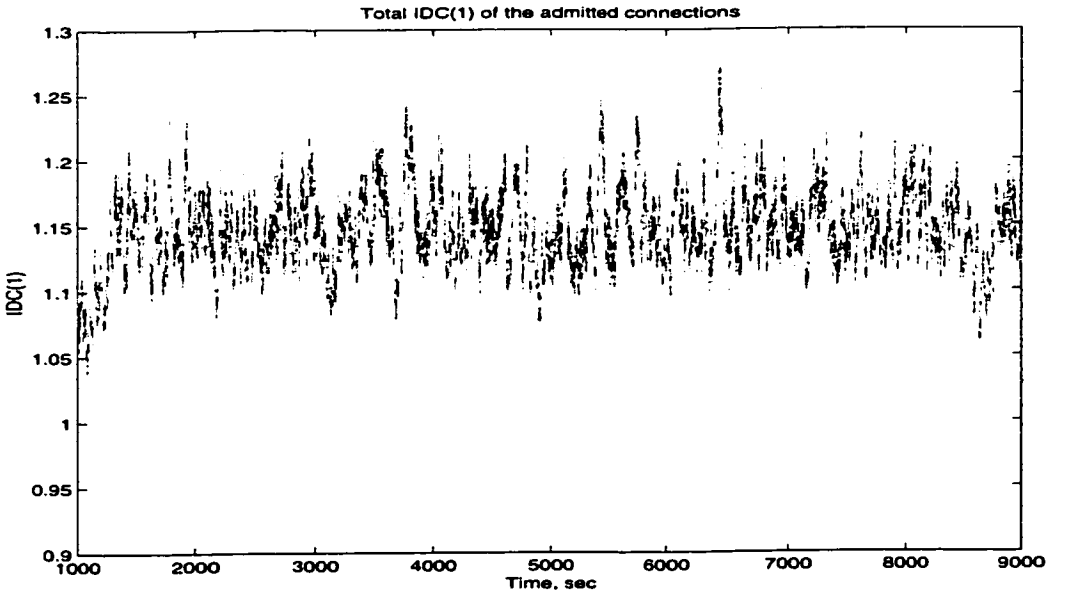

Figure 7.12: Measured IDC(1) versus simulation time.

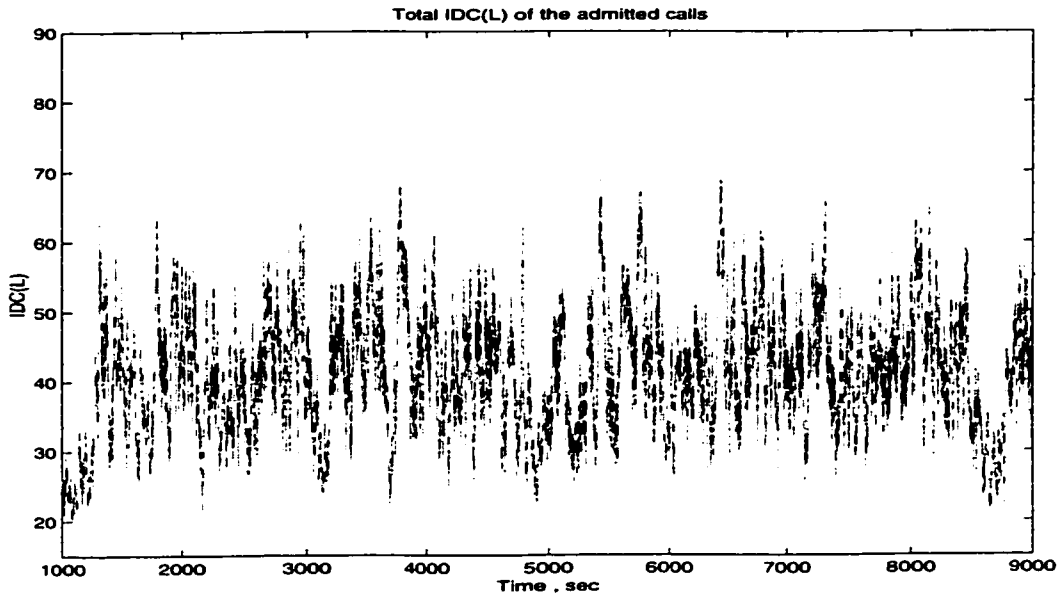

Figure 7.13: Measured IDC(L) versus simulation time.

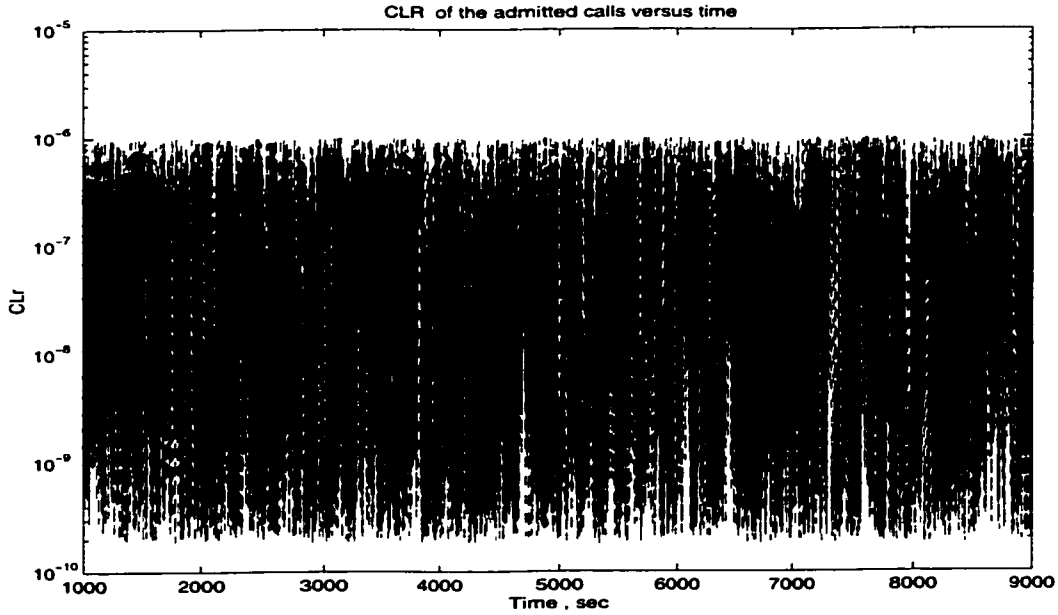

Figure 7.14: CLR versus simulation time.

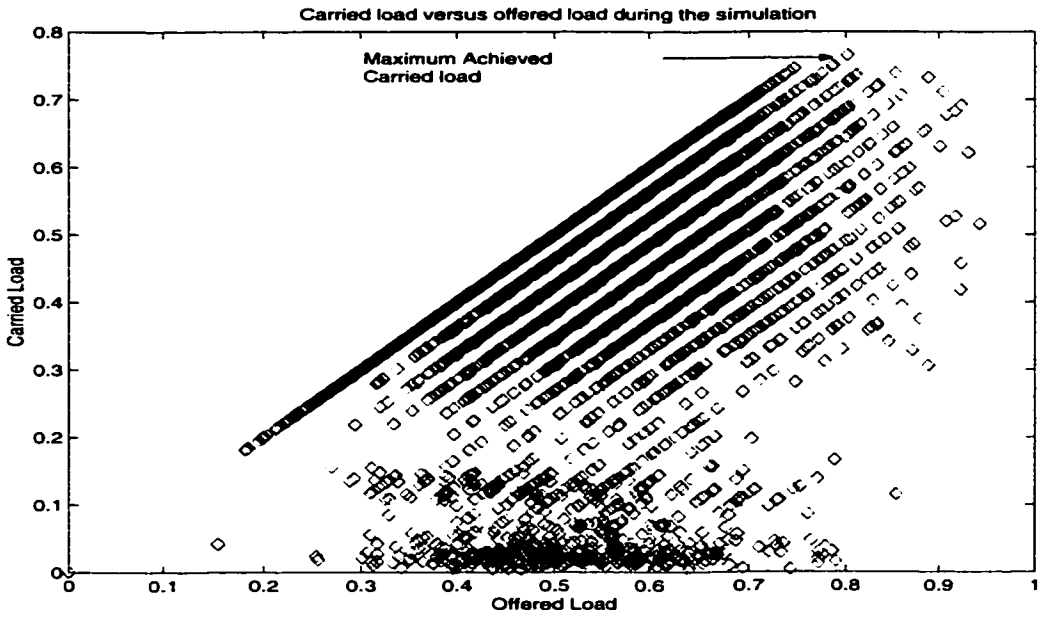

Figure 7.15: Carried load versus Offered Load for 9000 samples over 9000 seconds of simulation.

Chapter 8

Conclusions and Recommendations for Future Research

8.1 Conclusions

In broadband networks, prediction of QoS plays an important role in traffic engineering. The state-of-the art approaches use *a priori* characterization of traffic for the convenience of mathematical tractability. However, it is difficult to provide an accurate and tight statistical model that can adequately describe the traffic patterns of different data sources. In this thesis, we proposed a new approach that closely predicts QoS, e.g. CLR. The QoS predictor is further applied in formulation of traffic control functions, e.g. admission control, that maximize traffic load, hence revenue, of the system.

From this thesis work we conclude that:

1. For a given QoS, traffic can be represented by a universal set of parameters.

Assuming that the QoS of interest is CLR, we considered various data traffic profiles that are SRD, LRD, self-similar or mixed. By analyzing both the statistical property and the measured CLR when fed fed into a FIFO queue, we obtain the following set parameters that impact the CLR:

- Traffic load, which is related to the first moment of traffic,
- initial value of IDC, which is related to the second moment (or variance) of traffic,
- the degree of self-similarity, K, which is related to the slope of the IDC curve and
- the burstiness value, or calculated IDC, at a large time interval.

We label this set of parameters as the traffic indicator that is used as a CLR prediction variable.

2. The aggregation of two traffic profiles, each described by a traffic indicator, can be represented by a new traffic indicator using our derived equations.

3. The CLR function in terms of the traffic indicator and buffer size can be captured by a linear combination of a number of sigmoid functions. This is because a linear combination of sigmoid functions can approximate any continuous function as long as there is no limit on the number of sigmoid terms.

4. The results show that the introduced CLR approximator closely follows the measured CLR of various traffic profiles and, therefore, it can be used for traffic engineering of broadband networks.

5. The proposed traffic indicator and the CLR approximator can be used to formulate a new CAC that maximizes the link utilization while the CLR of connections is kept at an acceptable levels. If the predicted CLR of the aggregated two traffic indicators (one for all existing connections and the other for the new connection) is less than a set threshold, the new connection is admitted, otherwise, it is rejected.

8.2 Recommendations for Future Research

We suggest the following research work:

- The proposed CLR predictor can be applied to formulate efficient congestion control schemes in the core of the network in conjunction with flow control schemes at the access in order to maximize the system goodput and minimize re-transmissions while maintaining the QoS of all connections.

- In this research work, we identified the indicator of traffic as a CLR prediction variable. It is possible to identify a set of parameters that impact the delay performance.

Appendix A

Congestion Control Schemes

Traffic control functions are classified into CAC and congestion control schemes. In chapter 2, we discuss CAC schemes in ATM networks. In this appendix, we provide a high level overview of congestion control schemes in both ATM and IP networks. There are two main categories for congestion control: *open-loop* and *closed-loop* as they will be discussed in the following sections.

A.1 Open-Loop Congestion Control

Open-loop congestion control functions, as shown in Figure A.1, are performed at the source and any other node. These functions are described briefly in the following subsections.

A.1.1 Policing and Shaping

Both IP and ATM network elements perform policing and shaping functions. The policing function checks to see whether incoming traffic conforms to a specific traffic

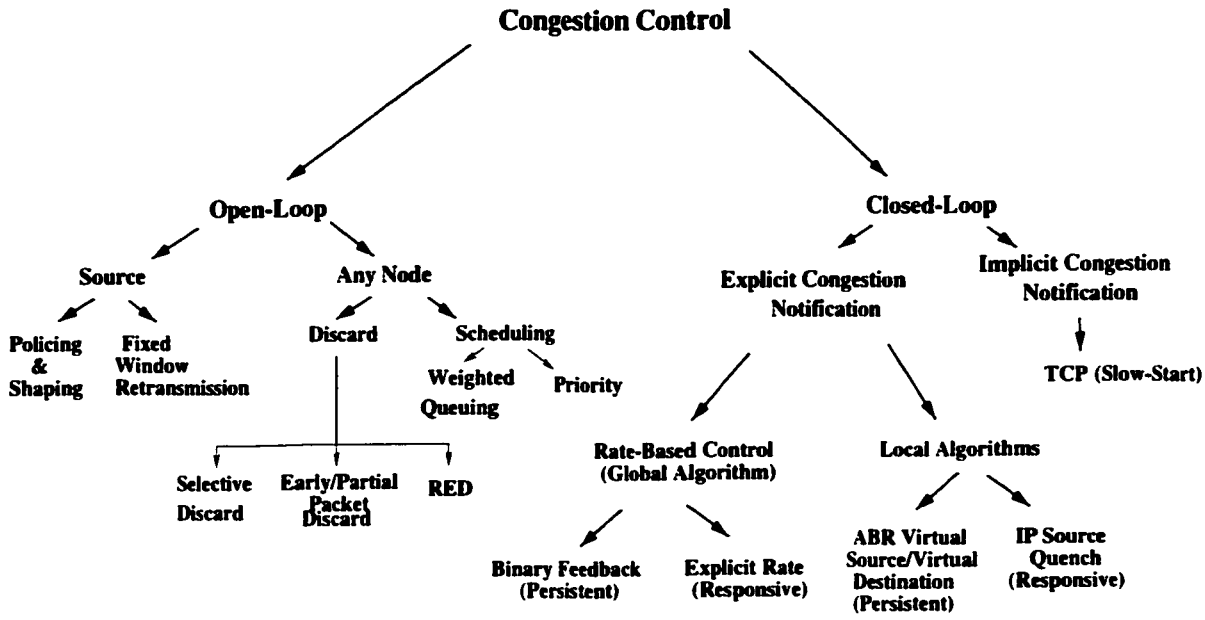

Figure A.1: Classification of congestion control functions.

contract. On the other hand, the shaping function delays outgoing packets or cells to ensure conformance to a traffic contract. Properly shaped traffic never fails a policing check when both functions employ the same traffic contract. Figure A.2 shows the generic placement of policing and shaping functions. Normally, the first network node performs policing of end users as shown by solid box in Figure A.2. Optionally, one network node may police the traffic received from another node. Frequently, end-users shape their traffic outputs to ensure conformance by the policing function at the first node.

a) ATM's Usage Parameter Control (UPC)

ATM networks employ Usage Parameter Control (UPC) and Network Parameter Control (NPC) to check conformance of cell flows from a user, or another network, against negotiated traffic parameters, respectively. In ITU documents, this function is referred to as the *policing* function. The function makes sure

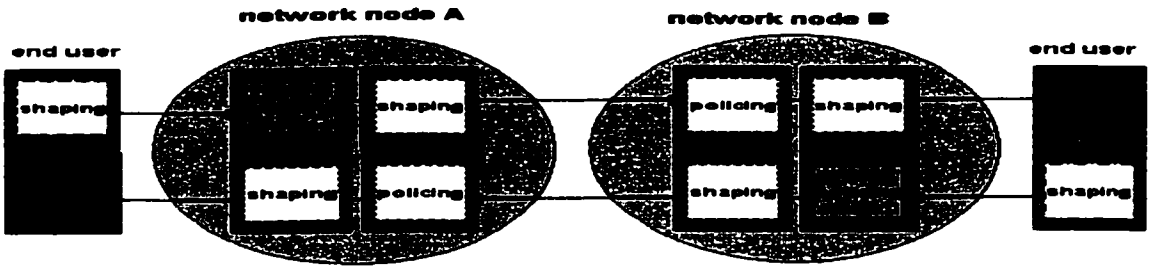

Figure A.2: Generic placement of policing and shaping functions.

that the users obey the negotiated traffic contract by detecting non-conforming sources and takes appropriate action to minimize the potentially harmful effect of the excess traffic. The set of traffic parameters that are controlled are the same set of parameters used to characterize a source. Since the latter remains an open issue, the former is also unresolved. In various proposed mechanisms, the controlled parameters are the peak rate, the average rate and the burst length. Standards do not specify the precise implementation of UPC and NPC functions. Instead, they bound the performance of any UPC/NPC implementation in relation to a Generic Cell Rate Algorithm (GCRA). One representation of GCRA is leaky bucket which is the most commonly used regulator.

Leaky bucket scheme was first introduced in [68]. The idea is that the arriving cells are transmitted when there are tokens in the bucket, otherwise they are dropped or tagged (for selective cell discarding when cell level congestion occurs [69]). Tokens are generated at constant rate[1] with an upper bound for the

[1]The rate at which tokens are generated represent either the average rate or the peak rate of the

number of waiting tokens. This upper bound, which is the size of bucket, controls the burst length. There are two types of enforcement action that can be taken and whether or not there is a user buffer gives rise to four different versions of leaky bucket schemes. In the first scheme, an arriving cell is dropped if there is no token awaiting in the bucket. Since tokens are generated at a constant rate, this scheme can be used to control either peak rate or the average rate but not both. Second scheme tags the violating cells and lets them enter the network and, hence, possibly discards them within the network at the congested nodes. The third and the fourth schemes are similar to schemes one and two, respectively, with the difference that arriving cells with no token waiting are queued. The cells will be dropped when queue is full. In the buffered schemes the operation is not transparent to the user due to the delay introduced by the buffer. Leaky Bucket scheme has been widely analyzed in the literature. The simplest analysis of the leaky bucket uses closed queuing networks in which both cells and tokens are generated by a separate Poisson process. Other complex analyses have also been proposed (e.g., [70] and [71]). In [70], a Markov chain discrete-time analysis has been proposed when there is no buffer for the arrived cells. Tokens are generated periodically in every interval. The approach calculates the steady-state data throughput by first calculating the steady-state probability of the number of tokens in the bucket just before new tokens are generated. The approach in [71] considers buffering of the arrived cells. It uses the fluid flow approximation generalized to a finite buffer case.

Leaky bucket approach in conjunction with buffering has been used for traffic

cells.

shaping. This has been performed by transforming a non-conforming cell flow into a conforming cell flow by delaying the transmission of non-conforming cells.

b) **Policing and Shaping in IP networks**

In the Internet, a bucket collects tokens that control the average transmission rate and burst duration. The network periodically adds tokens to the bucket(s) corresponding to each flow at a rate determined by the traffic parameters. If a token bucket is full nothing adverse happens. It means that the source has not transmitted data in a while and has a full set of token permissions to transmit once it becomes active again. On the other hand in ATM, arriving cells fill a logical set of buckets which leak at a rate as specified by the traffic parameters. If an arriving cell overflows any of the buckets, the network considers the cell non-compliant to the traffic parameters. The Internet Resource reSerVation Protocol (RSVP) [72] uses the token bucket algorithm to describe traffic parameters corresponding to a specific flow of IP packets. Two parameters completely specify the token bucket: an average rate r and a bucket depth b. RFC 2215 [72] defines the token-controlled average rate r as the number of bytes of IP datagrams permitted by the token bucket per second.

A.1.2 Scheduling

A traffic policer acts as a traffic cop at the node connected to the user at the edge of the network and optionally at points where networks interconnect. Although IP and ATM employ different algorithms to define traffic policing parameters, they have similar semantics. The two popular scheduling schemes are Weighted Fair Queuing [73] (WFQ) and Priority Queuing (PQ). In WFQ, as shown in Figure A.3, N flows

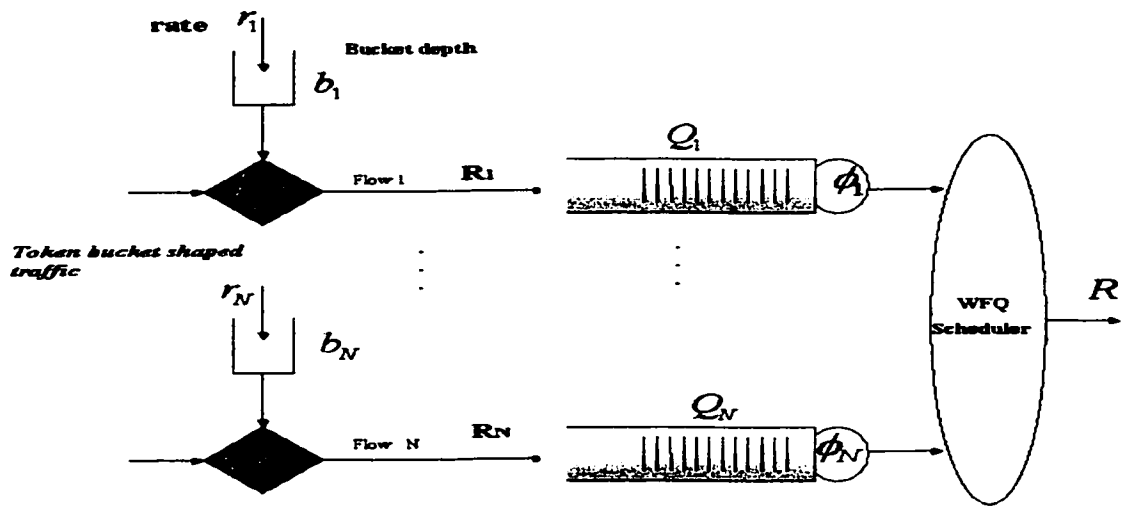

Figure A.3: Weighted Fair Queuing (WFQ) scheduling scheme.

characterized by r_i, the token rate, and b_i, the token bucket depth arrive to an individual queue Q_i via input links operating at a rate of R_i bps. A weighted scheduler services each queue at a proportion ϕ_i of the overall output link rate R of the node. For the case where the input link speeds are infinite (i.e., $R_i = \infty$), the worst case nodal delay $D_{i,\ Max}$ for flow i is bounded by

$$D^{WFQ}_{i,\ Max} \leq \frac{b_i}{r_i} + \frac{L_{max}}{R}$$

where L_{max} is the maximum packet length across all flows. It is interesting to compare the performance of a lossless WFQ system with that of a PQ system that has been engineered to achieve a low loss. For a PQ system with buffer capacity B and link rate R, the maximum delay for a high priority flow is:

$$D^{PQ}_{i,\ Max} \leq \frac{B}{R} + \frac{L_{max}}{R}.$$

The required buffer capacity is a function of offered load ρ, loss probability P_{loss} and characteristics of traffic. For Markovian traffic patterns, the ratio of the maximum

delay for WFQ and PQ is approximately:

$$\frac{D_{i,\,Max}^{WFQ}}{D_{i,\,Max}^{PQ}} \approx \frac{R}{r_i} \cdot \frac{(1-\rho)}{-\ln P_{loss}}$$

The first term is the ratio of the multiplexer link rate to the regulated source rate. The second term is a constant on the order of a fraction $\frac{(1-\rho)}{-\ln P_{loss}}$. As a result, sources with an average rate r_i on the order of a fraction of the link rate achieve comparable performance with WFQ and PQ. Sources that have a rate on the order of the link rate achieve better performance with WFQ than PQ. Finally, sources with low average rates achieve better performance with PQ than with WFQ but at the expense of poorer performance of low priority traffic.

A scheduling algorithm that deliver optimal delay with minimal buffer space exist if the user shapes the traffic to a level less than or equal to that enforced by the policing function [74]. The analysis in [74] compares three schemes: Fixed Allocation (allocates a worst-case buffer to each flow), Semi-Flexible Allocation (allocates a buffer to each flow based on knowledge of r_i and b_i) and Flexible Allocation (implements a completely shared memory buffer). The result of the analysis indicates that the buffer requirement decreases as the flexibility of the scheduling policy increases. The paid price for better buffer utilization is the increased complexity. Examples includes Non-Preemptive Earliest Deadline First (NPEDF) and a Tracking policy based upon Preemptive Earliest Deadline First (TPEDF) [74].

A.1.3 Discard Mechanisms

IP routers and ATM switches can recover from congestion by discarding traffic. There are three discard mechanisms:

- Selective Discard of Lower Priority traffic (ATM and IP)
- Early / Partial Packet Discard, EPD/PPD (ATM)
- Random early detection (IP)

Selective Discard: Policing allows the ingress traffic node to either discard the traffic that fails to conform to the traffic parameters, or mark non-conforming traffic at a lower priority. In ATM, the Cell Loss Priority (CLP) bit is used to indicate whether the cell is of high priority (CLP=0) or low priority (CLP=1). The IP diffserv standard allows implementations to selectively mark non-conforming packets. The experimental field in the Multi-Protocol Lable Switching (MPLS) header may also support selective tagging of non-conforming packets. Additionally, the IP integrated services architecture document [75] recommends tagging non-conforming packets if such a means is available. Selective discard gives preferential treatment to higher priority cells or packets over lower priority cells or packets during periods of congestion in order to maintain a guaranteed QoS for higher priority traffic. In ATM, selective discard is an important standardized network function for recovering from congestion. If network is not congested then the network may provide higher throughput by also transferring non-compliant traffic. Selective discard is effective when $\omega\rho < 1$, where ω is the percentage of high priority traffic and ρ is the total load. In extreme overload situations ($\omega\rho > 1$), other congestion recovery mechanisms need to be employed.

Early/Partial Packet Discard: Loss of a single cell in the ATM Adaptation Layer 5 (AAL5) Segmentation and Reassembly (SAR) process means losing an entire packet. A number of studies and tests (for example [76]) show that a more

effective reaction to congestion is to discard at the frame level rather than at the cell level. As a result, ATM Forum [77] specifies an intelligent frame discard function as an optional congestion recovery procedure. Such intelligence is used at the heavily congested points (such as a highly utilized link of an ATM switch) to maximize the number of complete packets transferred. Early Packet Discard (EPD) occurs when a device in a congested state discards every cell from an AAL5 PDU. EPD prevents cells from entering the buffer, reserving the remaining buffer capacity for the cells from packets already admitted to the buffer. Partial Packet Discard (PPD) occurs when a device discards all remaining cells except the last one. PPD acts when some cells from a packet have already been admitted to the buffer.

Random Early Detection (RED): The objective of RED algorithm (described in [78]) is to fairly distribute the effects of congestion across multiple user flows competing for a congested resource. The operation of the RED algorithm is as follows. A buffer queue that is served by a transmission link has two thresholds: min and max. The queue keeps a low-pass filtered average using the current queue length q with memory factor of x, i.e., $avg_{new} = avg_{old}(1-x)+xq$. If the low-pass average is less than the minimum threshold, then the node admits the packet. If the average exceeds the maximum threshold, then the node discards every incoming packet to avoid congestion. If the average is between minimum and maximum threshold, then the node drops packets with the following probability:

$$Pr(drop) = p_{max}\frac{avg - min}{max - min}$$

where p_{max} is the maximum drop probability. Other refinements and enhancements of the RED algorithm include dropping all packets from a randomly iden-

tified flow and implementing thresholds on a per-flow basis upon prioritization across different flows [79].

A.1.4 Fixed Window Retransmission

Fixed window retransmission protocols are invoked at the source to combat the packet (or cell) loss. Packet loss or excessive delay results in a time-out or negative acknowledgment in a higher layer protocol (e.g., TCP) which then retransmits one or more packets. Higher layer protocols recover from detected errors, or time-outs, by one of the two basic methods called *Go-Back-N* and *Selective Reject.*

The *Go-back-N* strategy, which is the default protocol in TCP, retransmits the packets that are erroneous or have been time-out followed by all other packets sent afterwards. If the transmitter has sent N packets upon detecting a lost packet or time-out, then it must retransmit the same N packets again. The *selective reject* strategy retransmits only those packets that were actually lost or timed out. It is superior to Go-Back-N strategy as the transmitter resends only the selectively rejected packets. On the other hand, this method requires greater complexity in the receiver and sender protocols. Standards call this a selective reject, selective retransmission or selective acknowledgement strategy. The ATM Service Specific Connection Oriented Protocol (SSCOP) details a sophisticated selective reject algorithm [80]. The IETF defines optional selective acknowledgement strategy as extensions to TCP (e.g., [81]) in support of communications over long delay paths (e.g., Satellite Communications); however, few implementations support it.

A.2 Closed Loop Congestion Control

When a large number of sources start generating traffic simultaneously, momentary periods of cell loss occur within the network. Admission control and policing schemes reduce the buffer overflow probabilities, however, they can not prevent it. On the other hand, the optimum selection of window size for the adapted retransmission protocol depends upon a number of parameters that real applications cannot statically configure. As a result, IP and ATM-based applications utilize adaptive flow control techniques to set the window size or adjust the transmission rate. Since these techniques utilize feedback, they are called closed loop congestion control schemes. Depending on how the status of congestion is notified to the source, gives rise to two classes of closed loop congestion control: Explicit Congestion Notification and Implicit Congestion Notification. We briefly describe them in the following subsections.

A.2.1 Implicit Congestion Notification

TCP slow-start congestion control [82] [83] is an example of implicit congestion notification. It is an enhancement to the initial TCP implementation designed to dynamically maximize throughput and prevent congestion collapse by changing the window size. A TCP sender keeps track of a congestion window which is never greater than the window size reported by the receiver in the TCP packet. TCP geometrically increases the window size (i.e., 1, 2, 4, 8, and so forth) during the initial start-up phase. However, TCP congestion control has another function that limits the interval of geometric increase until the window size reaches a threshold of one-half of the congestion window size achieved before the previous unacknowledged segment. After this point, TCP increases the congestion window by one segment for each round-trip

time instead of doubling it during the geometric growth phase. The linear phase of TCP window growth is called *congestion avoidance* while the geometric growth phase is called *slow start.*

A.2.2 Explicit Congestion Notification

As shown in Figure A.1, explicit congestion control functions can be performed locally or globally. Rate-based control are global congestion control algorithms whereas ATM Available Bit Rate (ABR) Virtual Source/ Destination and IP source quench are local congestion control algorithms. For either local and global algorithms, there are two sub-categories: persistent and responsive. A persistent algorithm continuously generates feedback, while a responsive algorithm only generates feedback when congestion actually occurs. The suite of ATM Available Bit Rate closed-loop flow control algorithms all operate using explicit feedback. The following subsections briefly describe these algorithms.

Rate-based Congestion Control

In a rate-based scheme, the ABR source adjusts its allowed cell rate according to a feedback messages carried by Resource Management (RM) cells coming from the network. The source changes its actual cell rate based on the information in the Congestion Indication (CI) and Explicit Rate (ER) fields of the RM cell. Researchers have proposed various forms of rate-based congestion control algorithms for the ABR service category[84]. These algorithms can be grouped into two main classes: Binary Feedback (BF) and explicit rate(ER).

Binary mode is like green/red lights at the entrance to congested freeways. This mode

involves ATM switching nodes setting Explicit Forward Congestion Indicator (EFCI) in the forward direction so that the destination end station can set the CI field in a returned RM cell to control the flow of the sending end-station. The binary mode ensures interoperability with older ATM switches which can only set EFCI bit in the forward direction in response to congestion. Although the binary mode is simple, it experiences higher loss rates in certain situations such as those where congestion occurs at multiple points in the network. Furthermore, unless the network elements perform per connection queuing, unfairness may result when sharing a single queue.

In Explicit rate mode, each network element explicitly sets the maximum allowed rate in the RM cells looped back by the destination as they progress backward along the path to the source. The goal is that each user receives a fair allocation of available capacity and buffer resources in proportion to their traffic contract in a responsive manner. Simultaneously, the ABR service should operate at higher utilization with negligible loss. This mode requires tuning of far fewer parameters and hence is the preferred method in networks that are capable of supporting explicit rate ABR.

In contrast to the explicit rate algorithms, binary mode algorithms are known to be less responsive to changes in network congestion and less fair in allocating bandwidth among active sources [85]. On the other hand, binary algorithms are much simpler and require less hardware complexity.

Given the performance superiority [86], several ER algorithms have been proposed in the literature, for example [87] and [88]. In general, the proposed ER algorithms can be classified into two main classes: *approximate* and *exact*. As the name implies, approximate algorithms estimate the bandwidth fairness for different connections using

exponential running averages rather than computing fair-share values. As a result, approximate algorithms are relatively less responsive to network changes and can result in high oscillations in the generated bandwidth allocations. At the expense of higher implementation complexity, exact algorithms are more accurate in computing a fair-share bandwidth and, in most cases, offer an oscillation-free bandwidth allocation [85].

ABR Virtual Source/Destination (VS/DS)

The ABR VS/DS approach incorporates segmentation using the concept of mated pairs of virtual sources and destinations. These sources and destinations form closed flow control loops across the sequence of network elements involved in an ABR connection. The situation in which every node in the network is a virtual source/destination is called *hop-by-hop* flow control.

IP Source Quench

Source quench messages are send to hosts to throttle back their traffic to avoid congestion [89]. A source quench may be sent by destination host or by some gateway. The host that receives a Source Quench should throttle itself back for a period of time, then gradually increase the transmission rate again. The mechanism to respond to Source Quench may be in the transport layer (for connection-oriented protocols such as TCP) or in the application layer (for protocols that are built on top of UDP).

Appendix B

Procedure to Derive Parameters of a 2-State MMPP Source

In this appendix, which is an extended discussion to Section 5.1, we discuss the mapping of many (N) On-Off sources into a 2-state MMPP process. A 2-state MMPP process has been used in OPNET to generate variable bit rate (VBR) bursty traffic with short-range dependency.

The procedure uses underload and overload approach. Given a set of N independent and homogeneous On-Off sources, each with average On-time, α^{-1}, average Off-time, β^{-1}, and arrival rate in the On-state, A, we would like to map N sources into a single 2-state MMPP source with four parameters: arrival rate in state 1, λ_1, arrival rate in state 2, λ_2, average sojourn time in state 1, σ_1^{-1} and average sojourn time in state 2, σ_2^{-1}. Before we start deriving parameters of 2-state MMPP from On-Off parameters, let us first consider an On-Off source shown in Figure B.1. Since the On-Off source forms a Markov chain, we can solve the stationary probability of being in On state or

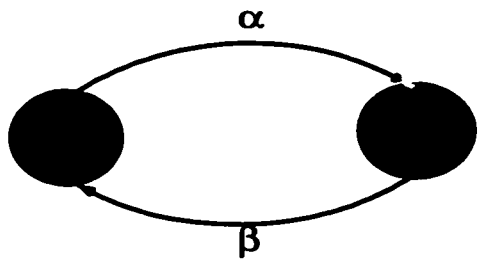

Figure B.1: A single On-Off source that models bursty traffic sources.

Off state at any time t,

$$\begin{aligned} P_{on}(t) + P_{off}(t) &= 1 \\ P_{on}(t+\Delta t) &= P_{on}(t)(1-\alpha\Delta t)\beta\Delta t \\ \frac{P_{on}(t+\Delta t) - P_{on}(t)}{\Delta t} &= -\alpha P_{on}(t) + \beta P_{off}(t). \end{aligned} \tag{B.1}$$

In steady-state, the left hand side of (B.1) approaches zero, $P_{on}(t) = P_{on}$ and $P_{off}(t) = P_{off}$, thus

$$\begin{aligned} P_{on} &= \frac{\beta}{\alpha} P_{off} \\ P_{on} + P_{off} &= 1 \end{aligned}$$

and

$$P_{on} = \frac{\beta}{\alpha+\beta}.$$

Now, given N On-Off sources, the probability that k of them are in On state is given by a binomial distribution, i.e.,

$$Pr[k \ \ source\ in\ On\ state] = \pi_k = \left(\begin{array}{c} N \\ k \end{array} \right) \left(\frac{\beta}{\alpha+\beta}\right)^k \left(\frac{\alpha}{\alpha+\beta}\right)^{N-k} \tag{B.2}$$

and the average arrival rate is

$$\overline{\lambda}_{N\ On-Off} = N\frac{\beta}{\alpha+\beta}A. \tag{B.3}$$

Since the peak arrival rate, $\lambda_{peak\ \ On-Off}$, is equal to NA, we can find the peak to average ration, PAR, as

$$PAR = \frac{\lambda_{peak\ On-Off}}{\bar{\lambda}_{N\ On-Off}} = \frac{\alpha+\beta}{\beta} = \frac{1}{P_{on}}. \tag{B.4}$$

The Index of Dispersion for Counts (IDC) for each one of the On-Off sources is the same as the IDC for aggregate N source and it has been derived by Heffes and Lucantoni [7] as

$$IDC_{N\ On-Off} = \frac{1-(1-\alpha/A)^2}{(\alpha/A+\beta/A)^2}. \tag{B.5}$$

The basis of the matching technique using *overload* and *underload* approach is due to the fact that there are two regions in a CLR versus buffer size: cell level and burst level. A 2-state MMPP attempts to capture these two regions through the two states. Of the N On-Off sources, if more than L of them are active, we say the MMPP source is in the overload region, whereas if less than L sources are active we say MMPP is in the underload region. As the average rate is given by equation (B.3), the average number of sources which are active is given by $\frac{N}{PAR}$. As a result, a good choice for L is

$$L = \left\lceil \frac{N}{(PAR)\zeta} \right\rceil^{+} \tag{B.6}$$

where ζ is a parameter between 0.9 and 1.1. Setting it to 1 is a good choice.

Figure B.2 shows a 2-state MMPP source in which subscript 1 refers to the underload state and the subscript 2 refers to overload state. For a 2-state MMPP, the average arrival rate and the IDC is given by

$$\bar{\lambda}_{MMPP} = \frac{\lambda_1\sigma_2+\lambda_2\sigma_1}{\sigma_1+\sigma_2} \tag{B.7}$$

$$IDC_{MMPP}(\infty) = 1+\frac{2\sigma_1\sigma_2(\lambda_1-\lambda_2)^2}{(\sigma_1+\sigma_2)^2(\lambda_1\sigma_2+\lambda_2\sigma_1)} \tag{B.8}$$

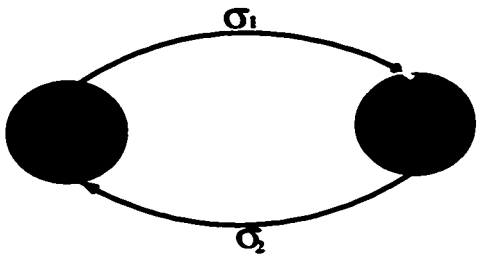

Figure B.2: A 2-state MMPP that models aggregation of many On-Off sources.

The matching procedure matches the followings: a) the average arrival rate in the underload and the overload region to the average arrival rate in states 1 and 2 of the MMPP source, respectively, b) the total average arrival rate between the two states and c) the asymptotic value of IDC. As a result,

$$\lambda_1 = \frac{\sum_{i=0}^{L} iA\pi_i}{\sum_{i=0}^{L} \pi_i} \tag{B.9}$$

and

$$\lambda_1 = \frac{\sum_{L+1}^{N} iA\pi_i}{\sum_{L+1}^{N} \pi_i} \tag{B.10}$$

where π_k is given by (B.2) and L by (B.6). The remaining parameters σ_1 and σ_2 are obtained through

$$\begin{aligned} IDC_{N\ On-Off}(\infty) &= IDC_{MMPP}(\infty) = IDC = 1 + \frac{2\sigma_1\sigma_2(\lambda_1-\lambda_2)^2}{(\sigma_1+\sigma_2)^2(\lambda_1\sigma_2+\lambda_2\sigma_1)} \\ \overline{\lambda}_{N\ On-Off} &= \overline{\lambda}_{MMPP} = \overline{\lambda} = \frac{\lambda_1\sigma_2+\lambda_2\sigma_1}{\sigma_1+\sigma_2}. \end{aligned}$$

Note that

$$\lambda_1 - \overline{\lambda} = \frac{(\lambda_1-\lambda_2)\sigma_1}{\sigma_1+\sigma_2} \tag{B.11}$$

$$\overline{\lambda} - \lambda_2 = \frac{(\lambda_1-\lambda_2)\sigma_2}{\sigma_1+\sigma_2}. \tag{B.12}$$

Using the above equations with some manipulations we obtain

$$\sigma_1 = \frac{2(\lambda_1-\overline{\lambda})^2(\overline{\lambda}-\lambda_2)}{(\lambda_1-\lambda_2)\overline{\lambda}(IDC-1)} \tag{B.13}$$

and

$$\sigma_2 = \frac{2(\lambda_1 - \overline{\lambda})(\overline{\lambda} - \lambda_2)^2}{(\lambda_1 - \lambda_2)\overline{\lambda}(IDC - 1)}. \tag{B.14}$$

Now, the mapping procedure can be summarized. Given the number of On-Off sources, N, peak-to-average ratio, PAR, peak-rate, A, and $IDC_{N\ On-Off}(\infty)$ (or α and β) the following steps are required to obtain the parameters of a 2-state MMPP source:

1. Find $\overline{\lambda} = \overline{\lambda_{NOn-Off}}$ using (B.3)

2. Find IDC using (B.5).

3. Find π_k, $k = 0,\ 1,\ \cdots,\ N$ using (B.2).

4. Find L using (B.6).

5. Find λ_1 and λ_2 using (B.9) and (B.10), respectively.

6. Find σ_1 and σ_2 using (B.13) and (B.14), respectively.

Appendix C

Fractional Brownian Motion Traffic (FBM) Generator

In this appendix, which is an extended discussion to Section 5.1, we discuss fast generation of FBM patterns. We deploy both MMPP and FBM generators, with various parameters and different mixture ratio, to obtain aggregate multimedia traffic.

The FBM generator uses Markov-Gauss processes to generate low frequency and high frequency of traffic arrivals. Low frequency shows long-range dependency and high frequency shows short-range dependency. Low frequency component is generated by weighting the sum of N Markov-Gauss processes with an appropriate weighting function and a high frequency component is generated by one Markov-Gauss process. The covariance function of standardized discrete time FBM is given by

$$C(s,\ H) = \frac{1}{2}\{(s+1)^{2H} - 2s^{2H} + (s-1)^{2H}\} \quad \text{for } s = 0,\ 1, \cdots \tag{C.1}$$

The interest is in the high lag of (C.1). Thus by expanding on powers of s^{-1} we have

the following approximation:

$$C_l(s,\ H) = H(2H-1)s^{2H-2} \qquad \text{for} \quad s >> 0. \tag{C.2}$$

To construct Fast Fractional Gaussian Noise (FFGN), which is a continuous version of the Fast FBM, with no loss of generality we consider standardized $X(t,\ H)$ which has mean of zero and variance of 1. It is assumed that $X(t,\ H)$ can be decomposed into two parts:

$$X(t,\ H) = X_l(t) + X_h(t)$$

where $X_l(t)$ denotes low-frequency (or long-range dependent) term and $X_h(t)$ denotes high-frequency (or short-range dependent) component. The separation between these two terms is arbitrary, but Mandelbrot [57] has proposed it to be $\frac{1}{e} = 0.368$. The low frequency component $X_l(t)$ is implemented by a weighted sum of N+1 standardized Markov-Gauss processes,

$$X_l(t) = \sum_{n=0}^{N} W_n^{\frac{1}{2}} M^{(n)}(t).$$

Markov-Gauss process are assumed to be pairwise uncorrelated. The autocorrelation function for the kth standardized Markov-Gauss process is

$$R_k(s) = E\left[M^{(k)}(t+s)M^k(t)\right] = r_k^{|s|}$$

The auto-correlation function $C_l(s,\ H)$ of $X_l(t)$ is given by

$$\begin{aligned} C_l(s,\ H) &= E\left[X_l(t+s)X_l(t)\right] \\ &= E\left[\sum_{n=0}^{N}(W_n)^{\frac{1}{2}} M^{(n)}(t+s) \cdot \sum_{n=0}^{N} W_n^{\frac{1}{2}} M^{(n)}(t)\right] \\ &= \sum_{n=0}^{N} W_n r_n^{|s|}. \end{aligned}$$

By setting $r_n = e^{-u}$ the analog version of the above formula is obtained as

$$C_l(s,\ H) = \int_0^{\infty} e^{-su} W(u)\ du. \tag{C.3}$$

It is noticed that $C_l(s,\ H)$ and $W(u)$ are a Laplace transform pair. Using the equation (C.2), we can get

$$W(u) = L^{-1}\left[C_l(s,\ H)\right] = \frac{2H(1-H)(2H-1)}{\Gamma(3-2H)}\ u^{1-2H}.$$

In FFBM the values of r and u are not incremented uniformly. The reason is that uniform increments put too much emphasis on high and mid-frequency components. Uneven increments are obtained by substituting $B^{-\nu}$ for u where B is called *Mandelbrot base.*

Substituting $u = B^{-\nu}$ into equation (C.3), we get

$$\begin{aligned} C_l(s,\ H) &= \int_{-\infty}^{+\infty} e^{-sB^{-\nu}}\ \log B\ \frac{2H(1-H)(2H-1)}{\Gamma(3-2H)}\ B^{2(H-1)\nu} d\nu, \\ &= \frac{2H(1-H)(2H-1)}{\Gamma(3-2H)}\ \log B \sum_{n=-\infty}^{+\infty} \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} e^{-sB^{-\nu}} B^{2(H-1)\nu} d\nu. \end{aligned}$$

Approximating the exponential inside the integral by its mid-span value $e^{-sB^{-n}}$ we have

$$\begin{aligned} C_l(s,\ H) &= \frac{2H(1-H)(2H-1)}{\Gamma(3-2H)}\ \log B \sum_{n=-\infty}^{+\infty} e^{-sB^{-n}} \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} B^{2(H-1)\nu} d\nu \\ &= \frac{H(2H-1)}{\Gamma(3-2H)}(B^{1-H} - B^{H-1}) \sum_{n=-\infty}^{+\infty} e^{-sB^{-n}} B^{2(H-1)n}. \end{aligned}$$

In practice the limits of the summation can not be $-\infty$ or $+\infty$ which corresponds to $r = 0$ and $r = 1$, respectively. The lower limit is replaced by $n = 0$ corresponding to $u = 1$ and the upper limit is replaced by N. Mandelbrot has employed a procedure to establish a precise value of N as a function of the length of record to be simulated.

However, Chi *et al* [93] recommended a value of $N = 20$ which gives $e^{-B^{-20}} \approx 1$. Also, they provided graphs of B versus N for different values of H. From these graphs if we select $N = 20$, then

$$B = \max(2, 10H - 5)$$

can be chosen for different values of H. Putting finite limits on N gives

$$C_l(s,\ H) = \sum_{n=0}^{N} W_n e^{-sB^{-n}}$$

where

$$W_n = \frac{H(2H-1)}{\Gamma(3-2H)}(B^{1-H} - B^{H-1})B^{2(H-1)n}.$$

To produce one sample of $X_l(t,\ H)$, W_n from the above, and r_n and $M_n(t)$ given by

$$\begin{aligned} M_n(t) &= r_n M_n(t-1) + \sqrt{1-r_n^2}\, G(t) \\ r_n &= e^{-B^{-n}} \end{aligned}$$

are calculated for all the values of $n = 0$ to N and are substituted into equation

$$X_l(t,\ H) = \sum_{n=0}^{N} (W_n)^{\frac{1}{2}} M^{(n)}(t).$$

For $X_h(t)$, the high frequency component, a single Markov-Gauss process $M(t) = r_n M(t-1) + \sqrt{1-r_n^2}G(t)$ is used rather than $N+1$ Markov-Gauss processes as in the case of $X_l(t)$. After setting the two values of N and B, the only input to the generator is Hurst parameter and the output is the corresponding $X(t,\ H)$.

Appendix D

Function Approximation Ability of Sigmoid Networks

In this appendix, first we mention the general representation of functions of an N-dimensional real variable, $\mathbf{x} \in \mathcal{R}^N$, by finite linear combinations of the form

$$\sum_{j=1}^{n} \alpha_j \sigma(y_j^T x + \theta_j) \tag{D.1}$$

where $\mathbf{y} \in \mathcal{R}^N$ and $\alpha_j,\ \theta_j \in \mathcal{R}$, σ is the sigmoid function with the following property:

$$\sigma(t) \rightarrow \begin{cases} 1 & as \quad t \rightarrow \infty \\ 0 & as \quad t \rightarrow -\infty \end{cases}$$

It is noted that the above function is indeed the activation function in the neural network. Now, we investigate conditions under which the above summation is dense in $C(D)$, where C is the set of continuous functions on D and D is the N-dimensional unit cube ($D = [0,\ 1]^N$). We present one Theorem and one Lemma. By combining the Theorem and the Lemma, it can be shown that the networks with one hidden layer and an arbitrary number of continuous sigmoid functions can approximate any

continuous function with arbitrary precision provided that no constraints are placed on the number of nodes or the size of the weights[94].

Let D denote the N-dimensional unit cube $[0,\ 1]^N$. The space of continuous functions on D is denoted by $C(D)$ and $||f||$ is used to denote the supremum (or uniform) norm of a function $f \in C(D)$. The space of finite, signed regular Borel measures on D is denoted by $M(D)$[96]. The objective is to investigate the conditions under which sums of the form

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma(y_j^T x + \theta_j) \tag{D.2}$$

are dense in $C(D)$ with respect to the supremum norm.

Definition 1: We say σ is *discriminatory* if for a measure $\mu \in M(D)$

$$\int_D \sigma(y^T x + \theta) d\mu(x) = 0$$

for all $y \in \mathcal{R}^N$ and $\theta \in \mathcal{R}$, implies that $\mu = 0$.

Definition 2: we say σ is *sigmoid* if

$$\sigma(\tau) \to \begin{cases} 1 & for\ \ \tau \to \infty \\ 0 & for\ \ \tau \to -\infty \end{cases}$$

Theorem 1: Let σ be any continuous discriminatory function. Then the finite sums of the form

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma(y_i^T x + \theta_j) \tag{D.3}$$

with $\alpha_j \in \mathcal{R}$ are dense in $C(D)$. In other words, given any $f \in C(D)$ and $\epsilon > 0$, there is a sum $G(x)$ for which

$$|G(x) - f(x)| < \epsilon \quad for\ \ all\ \ x \in D$$

Proof: Assume $S \subset C(D)$ is the set of functions of the form $G(x)$ as in D.3. Obviously, S is a linear subspace of $C(D)$. We argue that the closure of S is all of $C(D)$. Assume that the closure of S is not all of $C(D)$ and it is T, which is a closed proper subspace of $C(D)$. By the Hahn-Banach theorem, there is a bounded linear functional L on $C(D)$, with the property $L \neq 0$ but $L(T) = L(S) = 0$. By the Rieze representation theorem, this bounded linear functional L is of the form:

$$L(h) = \int_D h(x) d\mu(x)$$

for some $\mu \in M(D)$ and for all $h \in C(D)$. In particular, since our discriminatory function $\sigma(y^T x + \theta)$ is in T for all y and θ, we should have

$$\int_D \sigma(y^T x + \theta) d\mu(x) = 0 \quad for \;\; all \;\; y \; and \; \theta$$

Since we assumed that σ is discriminatory, this condition implies that $\mu = 0$, contradicting the assumption. Hence, the subset S must be dense in $C(D)$.

This proof shows that the sums of the form D.3 are dense in $C(D)$ provided that σ is continuous and discriminatory. Now we show that the sigmoid functions are discriminatory.

Lemma 1 [94]: Any bounded, measurable sigmoid function, σ, is discriminatory. In particular, any continuous sigmoid function is discriminatory.

Proof: To demonstrate this, note that

$$\sigma(\lambda(y^T x + \theta) + \phi) \begin{cases} \to 1 & for \;\; (y^T x + \theta) > 0 \;\; as \;\; \lambda \to +\infty \\ \to 0 & for \;\; (y^T x + \theta) < 0 \;\; as \;\; \lambda \to +\infty \\ = \sigma(\phi) & for \;\; (y^T x + \theta) = 0 \;\; for \; all \; \lambda \end{cases}$$

for any x, y, θ and ϕ.

Therefore, the function $\sigma_\lambda(x) = \sigma(\lambda(y^T x + \theta) + \phi)$ converge pointwise and boundedly to the function

$$\gamma = \begin{cases} 1 & for \ \ (y^T x + \theta) > 0 \\ 0 & for \ \ (y^T x + \theta) < 0 \\ \sigma(\phi) & for \ \ (y^T x + \theta) = 0 \end{cases}$$

as $\lambda \to \infty$.

Let $\prod_{y,\,\theta}$ be the hyper-plane defined by $\{x \mid y^T x + \theta = 0\}$ and let $H_{y,\,\theta}$ be the open half-space defined by $\{x \mid y^T x + \theta < 0\}$. By the Lesbegue Bounded convergence theorem, we have

$$\begin{aligned} 0 &= \int_D \sigma_\lambda(x) d\mu(x) \\ &= \int_D \gamma(x) d\mu(x) \\ &= \sigma(\phi)\mu(\prod_{y,\,\theta}) + \mu(H_{y,\,\theta}) \end{aligned}$$

for all ϕ, θ and y.

Now we show that the measure of all half-planes being 0 implies that the measure μ itself must be zero. This would be trivial if μ were a positive measure, but in our case μ is not necessarily a positive measure. It will be shown that $\mu = 0$ in what follows.

Let us fix y. For a bounded measurable function f, let's define the linear function F according to

$$F(h) = \int_D h(y^T x) d\mu(x).$$

Since μ is a finite single measure, F is a bounded functional on $L^{\infty}(\mathcal{R})$. Let

$$h(u) = \begin{cases} 1 & if \ \ u \geq \theta \\ 0 & if \ \ u \leq \theta \end{cases}$$

be the indicator function of the interval $[\theta, \infty)$. Then

$$F(h) = \int_D h(y^T x) d\mu(x) = \mu(\prod_{y, -\theta}) + \mu(H_{y, -\theta}) = 0.$$

Similarly, $F(h) = 0$ if h is the indicator of the open interval (θ, ∞). By linearity, $F(h) = 0$ for the indicator function of any interval and hence for any simple function since simple functions are the sum of indicator functions of intervals. Since simple functions are dense in $L^{\infty}(\mathcal{R})$[95], then $F = 0$.

In particular, the bounded measurable functions $s(u) = \sin mu$ and $c(u) = \cos mu$ give

$$\begin{aligned} F(s(u) + jc(u)) &= \int_D \cos(m^T x) + j \sin(m^T x) d\mu(x) \\ &= \int_D e^{jm^T x} d\mu(x) = 0 \end{aligned}$$

for all m.

Thus Fourier transform of μ is 0 and so μ must be 0. Therefore, it can be concluded that σ is discriminatory.

Having addressed Theorem 1 and Lemma 1, the results can be applied to the neural networks with sigmoid functions as neurons. A straightforward combination of the above mentioned theorem and lemma shows that networks with one internal layer (or hidden layer) and any arbitrary continuous function as neurons, can approximate continuous functions with arbitrary precision provided that no constraints are put on the number of neurons or the size of the weights. As a consequence we mention the Theorem 2 for function approximation of a single hidden layer neural network.

Theorem 2: Let σ be any continuous sigmoid function. Then the finite sums of the form

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma(y_j^T x + \theta_j)$$

are dense in $C(D)$. In other words, given any $f \in C(D)$ and $\epsilon > 0$, there is a sum $G(x)$ for which

$$|G(x) - f(x)| < \epsilon \quad for \ \ all \ \ x \in D$$

Proof: Combining Theorem 1 and Lemma 1 by considering the fact that continuous sigmoid functions satisfy the conditions of the lemma, the theorem is proven.

Bibliography

[1] K. G. Coffman and A. M. Odlyzko, "Internet Traffic: Is there a 'Moore's Law' for data traffic?" *AT&T Labs-Research*, www.att.com, June 2001.

[2] G. Kesidis, J. Walrand, and C.-S. Chang, "Effective bandwidths for multiclass Markov uids and other ATM sources," *IEEE/ACM Trans. Networking*, vol. 1, pp. 424-428, Aug. 1993.

[3] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 968-991, Sept. 1991.

[4] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Mar-kovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 329-343, June 1993.

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

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

[7] H. Heffes and D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance", *IEEE JSAC*, vol. SAC-4, no. 6, pp 856-868, 1986.

[8] G. Choudhary, D. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.*, vol. 44, pp. 203-217, Feb. 1996;

[9] D. Anick, D. Mitra, and M. Sondhi, "Stochastic theory of a data handling system with multiple sources,*Int. Conf. Communications*, Seattle, WA, 1980, pp. 13.1.1-13.1.5.

[10] B. Tsybakov and N. D. Geoganas, "On Self-Similar in ATM Queues: definitions, overflow probability and cell delay distribution", *IEEE/ACM Trans. Networking*, vol. 53, June 1997, pp. 397-409.

[11] I. Norros, "A Storage Model with Self-Similar Input", *Queuing Systems, 16*, pp. 387-396, 1994.

[12] A. Erramilli, O. Narayan and W. Willinger, "Experimental queuing Analysis with long-range dependent traffic," *IEEE/ACM Trans. Networking*, Vol. 4, pp. 209-223, 1996.

[13] Onuttom Narayan, "Exact Asymptotic Queue Length Distribution for Fractional Brownian Traffic", *Advances in performance Analysis*, Vol. 1 (1), Notable Publications, 1998, pp. 39-63.

[14] K. Elsayed and H. G. Perros, "Analysis of An ATM Multiplexer with Heterogeneous Markovian On-Off Sources and Application to Call Admission Control", *J of High Speed Networks*, Aug. 1997.

[15] D. D. Botvich and N. G. Duffield, "Large deviations, the shape of the loss curve, and economies of scale in large multiplexers," *Queueing Syst.*, vol. 20, pp. 293-320, 1995.

[16] B. Bensaou, J. Guibert, J. W. Roberts, and A. Simonian, "Performance of an ATM multiplexer queue in the fluid approximation using the Benes approach," in *Ann. Oper. Res.*, vol. 49, pp. 137-160, 1994.

[17] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic(extended version), *IEEE/ACM Trans. Networking,* Vol. 2, no. 1, pp. 1-15, Feb. 1994.

[18] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, "Long Range Dependency in Variable Bit Rate Video Traffic", *IEEE Trans. On Communications,* Vol 43, no. 24, pp 1566-1579, Feb/March/April 1995,

[19] ATM Forum Technical Committee, Traffic Management Working Group. *ATM Forum Traffic Management Specification Version 4.0,* October 1995.

[20] R. Braden, D. Clark and S. Shenker, *Integrated Service in the Internet Architecture: an Overview*, RFC 1633, IETF, June 1994.

[21] S. Shenker, C. Partridge and R. Guerin, *Specification of Guaranteed Quality of Service*, RFC 2212, IETF, September 1997.

[22] ITU-T Study Group 13. Recommendation I.371, Geneva, May 1996.

[23] G. Gallasi *et al*, "ATM: Bandwidth Assignment and Bandwidth Enforcement Policies", *GLOBCOM'89*, pp. 1788-1793, 1989.

[24] Raif O. Onvural, "Asynchronous Transfer Mode Networks, Performance Issues", 1995 ARTECH House Inc.

[25] S. Yasuda *et al*, "A study of Admission Control Scheme and Circuit Capacity Management for ATM", *IEICE National Conference Record, Autumn,* vol. B-381, 1990.

[26] H. Saito, *Teletraffic Technologies in ATM Networks*, Artech House Inc., MA, pp. 70-122, 1994.

[27] R. Bolla *et al*, "Adaptive Bandwidth Allocation by Hierarchical Control of Multiple ATM Traffic Classes", *INFOCOM'92*, pp. 30-38, 1992.

[28] P. W. Glynn and W. Whitt, "Logarithmic asymptotics for steady-state tail probabilities in a single-server queue," *Adv. Appl. Prob.*, vol. 26, pp. 131-156, 1994.

[29] D. Mitra, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer", Adv. Appl. Prob., 1988, pp. 646-676.

[30] G. H. Choudhury, D. M. Lucantoni and W. Whitt, "On the Effective Bandwidths for Admission Control in ATM Networks". *Proc. 14th Int'l Teletraffic Congress*, ITC'94, pp 411-20.

[31] I. Norros, "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks", *IEEE JSAC*, Vol.13, No. 6, August 1995.

[32] H. Saito, "Call Admission Control in ATM Networks using Upper-bound of Cell Loss Probability", *IEEE Trans. Commun.*, vol. 40, 1992, pp. 1512-21.

[33] A. Simonian and J. Guibert, "Large deviations approximation for fluid queues fed by a large number of on/off sources," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1017-1027, Aug. 1995.

[34] H. Yang, D. Towsley and W. Gong, "Efficient Calculation of Cell Loss in ATM Multiplexer", *Proc. of Globecom'95*, pp.1226-1230.

[35] S. Crosby, "In-Call Renegotiation of Traffic Parameters", *INFOCOM'93*, 1993.

[36] H. Saito and K. Shiomoto, "Dynamic Call Admission in ATM Networks", *IEEE JSAC*, vol. 9, no. 7 pp. 982-989, September 1991.

[37] J. Qiu and E. Knightly, "QoS Control via Robust Envelope-Based Measurement Based Admission Control", *The Proceedings of IWQoS'98*, NY, May 1998.

[38] S. Lee and J. Song, "A Measurement-based Admission Control Algorithm Using Variable-sized Window in ATM Networks", *ICICS'97*, NY, Sept. 1997.

[39] R. Braden *et al*, Resource ReSerVation Protocol (RSVP) - Version 1, Functional Specifications, RFC 2205, IETF, Sept. 1997.

[40] S. Jamin *et al.*, "A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks", *IEEE/ACM Trans. on Networking*, vol. 5, no. 1, pp. 56-70, Feb. 1997.

[41] Rajab Faraj, *Modeling and Analysis of Self-similar Traffic in ATM Networks*, PhD. Dissertation, Spring 2000.

[42] V. Paxson and S. Floyd, "Wide-Area Traffic: The failure of Poisson Modelling", *IEEE/ACM Trans. on Networking*, 3(3), pp. 226-244, June 1995.

[43] R. Jain and S. Routhier, "Packet Trains Measurements and a New Model for Computer Network Traffic", *IEEE JSAC*, 4(6), pp. 986-995, Sept. 1986.

[44] R. Gusella, "A Measurement Study of Diskless Workstation Traffic on an Ethernet", *IEEE Trans. Comm.*, 38(9), pp. 1557-1568, September 1990.

[45] H. J. Fowler, W.E. Leland, "Local Area Network Traffic Characteristics, with implications for broadband network Congestion Management", *IEEE JSAC*, no. 9, pp 1139-1149, Sept. 1991.

[46] M. Garret, W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Traffic", *Proceeding of SIGCOMM'94*, Sept. 1994.

[47] B. Maglaris *et al*, "Performance Model of Statistical Multiplexing in Packet Video Communications", *IEEE Trans. Comm.*, vol. 36, pp 834-843, July 1988.

[48] D. P. Heyman and T. V. Lakshman, "Long-Range Dependent and Queuing Effect for VBR Video", *Self-Similar Network Traffic and Performance Evaluation*, Edited by Kihong Park and Walter Willinger, John Wiley & Sons, Inc., 2000.

[49] P. Skelly, P. Schwartz and M. Dixit, "A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer", *IEEE/ACM Trans. Networking*, vol. 1, no. 4, August 1993.

[50] W. Verbiest, L. Pinnoo and B. Vosten, "A Variable Bit Rate Codec for Asynchronous Transfer Mode Networks", *IEEE JSAC*, vol. 7, pp 761-770, 1989.

[51] G. Ramamurthy and B. Sengupta, "Modeling and Analysis of Variable Bit Rate Video Multiplexer", *Int. Teletraffic Conference*, Morristown, NJ, 1990.

[52] L. Kleinrock, *Queuing Systems, Volume 1: Theory*, John Wiley & Sons, 1975.

[53] L. Trajkovic, A. Neidhardt, "Effect of Traffic Knowledge on the Efficiency of Admission Control Policies", *Computer Communication Review*, vol. 29, no. 1, pp. 5-33, Jan. 1999.

[54] R. Gusella, "Characterizing the Variability of Arrival Process with Index of Dispersion", *IEEE JSAC*, Vol. 9, no. 2, Feb. 1991.

[55] David H. Evans, *Probability and Its Applications for Engineers*, Dekker, 1992.

[56] K. Hornik, *et al*, "Multilayer Feedforward Networks are Universal Approximator", *Neural Networks*, vol. 2, pp 359-366, 1989.

[57] B. B. Mandelbrot. "A Fast Fractional Gaussian Noise Generator", *Water Resource Researches 7*, pp 543-553.

[58] J. M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing Company, 1992.

[59] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Marcel Dekker Inc., New York & Basel, 1989.

[60] V. Ramaswami, "Non-linear Matrix Equations in Applied Probability - Solution Techniques and Open problems", *SIAM Review*, vol. 30, No. 2, pp. 256-263, 1988.

[61] J. Huang, T. Zhu and J. F. Hayes, "An Efficient Computational Method for Solving Non-linear Matrix Equation and Its Application in Queuing Analysis", *Journal of Computer Science and Technology*, Allerton Press Inc., 1996.

[62] H. R. Mehrvar, T. Le-Ngoc, "Performance Evaluation of Bursty Traffic Stream Using Neural Networks", CCECE'96, Calgary, May 26-29.

[63] H. R. Mehrvar, T. Le-Ngoc. "ANN Approach for Congestion Control in Packet Switch OBP satellite" , *Proc. IEEE ICC 95*, Seattle, June 1995.

[64] H. R. Mehrvar, T. Le-Ngoc. "Estimation of Degree of Self-Similarity for Traffic Control in Broadband Satellite Communications", *Proc. of CCECE 95*, Montreal, Sept 1995.

[65] M. Roughan, D. Veitch and P. Abry, "Real-Time Estimation of the Parameters of Long-Range Dependence", *IEEE/ACM Trans. on Networking*, vol. 8, no. 4, August 2000.

[66] Chang Bum Lee *et al*,"Hurst parameter Estimation of VBR MPEG Video Traffic with Long-Range Dependence in ATM Networks", *The Proceedings of ICT'98*, Greece, 1998.

[67] K. S. Narendra, K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. Neural Networks*, Vol. 1, No. 1, March 1990, pp. 4-27.

[68] Akhtar, S. , "Congestion Control in a Fast Packet Switching Network", M.Sc Thesis, Washington University.

[69] A. Eckberg, D. Luan and D. Lucantoni, "Meeting the Challenge: Congestion and Flow control strategies for broadband information transport", Globcom'90.

[70] M. Sidi *et al.* "Congestion Control through input rate regulation", *Globcom'89*, Dallas, TX, Nov. 1989, pp 1764-1768.

[71] A. Elwalid and D. Mitra, "Stochastic Fluid Models in the Analysis of Access Regulation in High Speed Networks", *Globcom'91*, Phoenix, 1991, pp 1626-1632.

[72] S. Shenker and J. Wroclawski, *General Characterization of Parameters for the Integrated Service Network Elements*, RFC 2215, IETF, Sept. 1997.

[73] A. Parekh, R. Gallager, "A generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case, " *IEEE/ACM Transaction on Networking*, June 1993.

[74] L. Georgiadis *et al*, "Optimal Multiplexing on a Single Link: Delay and Buffer Requirements," *IEEE Trans. On Information Theory*, Sept 1997.

[75] R. Braden et al, *Integrated Services in the Internet Architecture: An Overview*, RFC 1633, IETF, June 1994.

[76] N.E.T., "Advanced Traffic Management for Multiservice ATM Networks," http://www.net.com/techtop/adtmatm_wp/white.html.

[77] The ATM Forum, "The ATM Traffic Management Specification", v. 4.0, 1996.

[78] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transaction on Networking, August 1993.

[79] S. Floyd, "References on RED (Random Early Detection) Queue Management," http://www.aciri.org/floyd/red.html.

[80] ITU-T, *B-ISDN ATM Adaptation Layer- Service Specific Connection Oriented Protocol (SSCOP)*, Recommendation Q.2110, July 1994.

[81] V. Jacobson and R. Braden, "TCP Extensions for Long-Delay Paths," RFC 1072, October 1988.

[82] W. Stevens, *TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms*, RFC 2001, IETF, January 1997.

[83] M. Allman, V. Paxson and W. Stevens, *TCP Congestion Control*, RFC 2581, IETF, April 1999.

[84] F. Bonomi and K. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service", *IEEE Network*, Vol. 9, No. 2, pp. 25-39, March/April 1995.

[85] A. Arulambalam, X. Chen and N. Ansari, "Allocating Fair rates for Available Bit Rate Service in ATM Networks", *IEEE Comm. Magazine*, pp. 92-100, Nov. 1996.

[86] A. Barnhart, "Explicit Rate performance Evaluations", *ATM Forum Contribution 94-0983*, Sept. 1994.

[87] R. Jain *et al*, "ERICA Switch Algorithm: A Complete Description", *ATM Forum Contribution 96-1172*, Aug. 1996.

[88] F. Chiussi, Y. Xia and V. P. Kumar, "Dynamic Max Rate Control Algorithm for Available Bit Rate Service in ATM Networks", *GLOBCOM'96*, 1996.

[89] J. Postel, *Internet Control Message Protocol*, RFC 792, IETF, Sept. 1981.

[90] H. R. Mehrvar, T. Le-Ngoc, "On Prediction of Bursty Traffic in Broadband Networks", *IEEE* Conf. on Comm., ICC'96, Dallas, TX, June 1996..

[91] H. R. Mehrvar, T. Le-Ngoc, Traffic Control in Broadband Satcom Systems, Technical Report, Concordia Unversity, Nov. 1997

[92] H.R. Mehrvar, T. Le-Ngoc,"Fuzzy Logic in Estimation of Burstiness Level for Admission Control in Broadband Networks", *Proc. IEEE ICC 97*, June 1997, Montreal, Canada

[93] Chi, M. E. Neal and G. K. Yang. "Practical application of Fractional Brownian Motion and Noise to Synthetic Hydrology", *Water Resource Researches 9, 1973*, pp 1569-1582.

[94] G. Cybenko, "Approximation by Superposition of Sigmoidal Function", *Mathematics of Control, Signals and Systems*, Vol. 2, 1989, pp 303-314, Springer-Verlag, New York.

[95] R. B. Ash, *Real Analysis and Probability*, Academic Press, New York, 1972.

[96] W. Rudin, *Functional Analysis*, McGraw Hill, New York, 1973.

[97] H. R. Mehrvar and M. R. Soleymani, "Packet Loss Prediction Using a Universal Indicator of Traffic", *to appear in the Proceeding of ICC'2001*, Helsinki, June 2001.