Some aspects of traffic control and performance evaluation of ATM networks

Fan, Zhong

Some Aspects of Traffic Control and Performance Evaluation of ATM Networks

Zhong Fan

School of Engineering
University of Durham

September 1997

A thesis submitted for the degree of
Doctor of Philosophy (Ph.D.) of the University of Durham.

Zhong Fan
Some Aspects of Traffic Control and Performance Evaluation of ATM Networks
Ph.D. 1997.

Abstract

The emerging high-speed Asynchronous Transfer Mode (ATM) networks are expected to integrate through statistical multiplexing large numbers of traffic sources having a broad range of statistical characteristics and different Quality of Service (QOS) requirements. To achieve high utilisation of network resources while maintaining the QOS, efficient traffic management strategies have to be developed. This thesis considers the problem of traffic control for ATM networks.

The thesis studies the application of neural networks to various ATM traffic control issues such as feedback congestion control, traffic characterization, bandwidth estimation, and Call Admission Control (CAC). A novel adaptive congestion control approach based on a neural network that uses reinforcement learning is developed. It is shown that the neural controller is very effective in providing general QOS control. A Finite Impulse Response (FIR) neural network is proposed to adaptively predict the traffic arrival process by learning the relationship between the past and future traffic variations. On the basis of this prediction, a feedback flow control scheme at input access nodes of the network is presented. Simulation results demonstrate significant performance improvement over conventional control mechanisms. In addition, an accurate yet computationally efficient approach to effective bandwidth estimation for multiplexed connections is investigated. In this method, a feedforward neural network is employed to model the nonlinear relationship between the effective bandwidth and the traffic situations and a QOS measure. Applications of this approach to admission control, bandwidth allocation and dynamic routing are also discussed.

A detailed investigation has indicated that CAC schemes based on effective bandwidth approximation can be very conservative and prevent optimal use of network resources. A modified effective bandwidth CAC approach is therefore proposed to overcome the drawback of conventional methods. Considering statistical multiplexing between traffic sources, we directly calculate the effective bandwidth of the aggregate traffic which is modelled by a two-state Markov modulated Poisson process via matching four important statistics. We use the theory of large deviations to provide a unified description of effective bandwidths for various traffic sources and the associated ATM multiplexer queueing performance approximations, illustrating their strengths and limitations. In addition, a more accurate estimation method for ATM QOS parameters based on the Bahadur-Rao theorem is proposed, which is a refinement of the original effective bandwidth approximation and can lead to higher link utilisation.

Declaration

I hereby declare that this thesis is a record of work undertaken by myself, that it has not been the subject of any previous application for a degree, and that all sources of information have been duly acknowledged.

Acknowledgements

The invaluable encouragement and guidance of my supervisor, Professor Philip Mars has been greatly appreciated. Most of all, I would like to thank him for giving me the chance to study in UK. I am also very grateful to John Mellor for his constant support throughout my study.

I would like to thank the members of Telecommunication Networks Research Group at the University of Durham, in particular Mark, Martin, Steve, Fred, Philip, and Jian-Guo, for their generous help.

Finally, I would like to express my deepest appreciation to my parents, both my brothers, and my girlfriend for all their love and encouragement. I dedicate this thesis to them.

Contents

List of Figures

List of Tables

List of Abbreviations

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
AI	Artificial Intelligence
ATM	Asynchronous Transfer Mode
BECN	Backward Explicit Congestion Notification
BISDN	Broadband Integrated Services Digital Network
BP	BackPropagation
BT	Burst Tolerance
CAC	Connection Admission Control
CAP	Cell Arrival Pattern
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CPE	Customer Premises Equipment
CS	Convergence Sublayer
CTD	Cell Transfer Delay
DTS	Dynamic Time-Slice
EFCI	Explicit Forward Congestion Indication
EWMA	Exponentially Weighted Moving Average
FCVC	Flow Controlled Virtual Circuit
FECN	Forward Explicit Congestion Notification
FIR	Finite Impulse Response
FRP	Fast Reservation Protocol
GCRA	Generic Cell Rate Algorithm
GFC	Generic Flow Control
HEC	Header Error Control
IDC	Index of Dispersion for Counts
IDI	Index of Dispersion for Intervals

IPP	Interrupted Poisson Process
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LAC	Link Admission Control
LDP	Large Deviations Principle
LLR	Least Loaded Routing
LMS	Least Mean Square
LRD	Long-Range Dependence
MA	Moving Average
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MLP	MultiLayer Perceptron
MMDP	Markov Modulated Deterministic Process
MMPP	Markov Modulated Poisson Process
MSE	Mean Squared Error
NN	Neural Network
NNI	Network Node Interface
NNTEM	Neural Network Traffic Enforcement Mechanism
OSI	Open Systems Interconnection
PC	Priority Control
PCR	Peak Cell Rate
pdf	probability density function
PTI	Payload Type Identifier
QOS	Quality Of Service
RM	Resource Management
SAR	Segmentation And Reassembly
SCR	Sustainable Cell Rate
SMDP	Semi-Markov Decision Problem
SND	Standard Normal Deviate
SONET	Synchronous Optical Network
TDNN	Time Delay Neural Network
TS	Traffic Shaping
TSP	Travelling Salesman Problem
UBR	Unspecified Bit Rate
UNI	User Network Interface
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VC	Virtual Channel
VCI	Virtual Channel Identifier
VP	Virtual Path
VPI	Virtual Path Identifier

Chapter 1

Introduction

In the past few years, Broadband Integrated Services Digital Network(BISDN) has received increased attention as a communication architecture capable of supporting multimedia applications. Asynchronous Transfer Mode(ATM) has been chosen to implement BISDN, in which the information is transmitted using short fixed-size cells consisting of 48 bytes of payload and 5 bytes of header. The fixed size of the cells reduces the variance of delay, making the networks suitable for integrated traffic consisting of voice, video, and data. ATM provides not only a flexible means for supporting a continuum of transport rates, but also a potential efficiency from the statistical sharing of network resources(e.g., bandwidth, buffers, etc.) by multiple users. To be competitive with specialized high-speed private network alternatives, ATM networks will need to be engineered to fully exploit this potential for efficiency.

In an ATM network, most traffic sources are bursty. A bursty source may generate cells at a near-peak rate for a very short period of time and immediately afterwards it may become inactive, generating no cells. Since an ATM network supports a large number of such bursty traffic sources, statistical multiplexing can be used to gain bandwidth efficiency, allowing more traffic sources to share the bandwidth. However, such a gain is achieved at the risk of congestion and consequential cell loss and delay when too many sources transmit at the same time. Therefore, traffic control is essential for ATM networks to allow such efficiency gains while guaranteeing that Quality of Service(QOS) standards are respected for all admitted connections in both normal and exceptional conditions(overloads and breakdowns).

Due to the effects of high-speed channels in ATM networks, the performance bottleneck of the network, which was once the channel transmission speed, is shifted to the processing speed at the network switching nodes and the propagation delay of the channel. Because of the increased ratio of propagation delay to cell transmission time, a large number of cells can be in transit between two ATM switching nodes. Thus, some of the congestion control schemes developed for existing networks may no longer be applicable in future high-speed networks. In addition, the increased ratio of processing time to cell transmission time requires the use of simplified protocols, making it difficult to implement hop-by-hop control schemes. Therefore, traffic control in ATM networks is a challenge, and new network control strategies have to be developed.

The aim of this thesis is to identify the causes of the limitations of current congestion control mechanisms and to propose novel solutions to various problems such as general QOS control, traffic characterization, access flow control, bandwidth estimation and connection admission control. Our approaches are based on two different techniques: artificial neural networks and large deviations (which leads to the notion of effective bandwidth). The objective of both approaches is to provide fast and yet accurate decisions related to traffic control and hence to increase the network utilisation while meeting the QOS requirements.

1.1 Summary of the Thesis

The main body of the thesis is divided into 7 chapters. Chapter 2 provides an overview of some of the basic concepts of BISDN and ATM. Chapter 3 gives a state-of-the-art survey of various traffic control schemes. In Chapter 4 the application of neural networks to adaptive congestion control for ATM networks is discussed in depth, and a novel congestion control approach based on reinforcement learning is proposed. Chapter 5 addresses a neural network method for multimedia traffic prediction and its application to access flow control. The predictor configuration and the network training algorithm are described. The proposed control scheme has a superior performance compared with conventional feedback control methods. Chapter 6 develops the use of feedforward multilayer perceptrons for effective bandwidth estimation so as to overcome the shortcomings of conventional approximations. The potential for application to dynamic bandwidth allocation and routing is also investigated. A modified effective-bandwidth-based admission control algorithm is presented in Chapter 7. It takes into account the statistical multiplexing gain across sources by modelling the aggregate traffic directly and the network utilisation is increased. Chapter 8 investigates the effective bandwidth approximation for ATM multiplexers in more detail, with the aim of finding out its limitations and possible improvement. A more accurate approximation is introduced, which is justified by a mathematical development based on large deviations asymptotics. In Chapter 9 a summary of the thesis is given, and areas for future work are identified.

Chapter 2

Asynchronous Transfer Mode

This chapter provides an overview of ATM. We begin with a brief introduction to BISDN and related issues. Subsequently we discuss the basic concept of ATM, the ATM cell format, and the ATM protocol reference model.

2.1 Broadband Integrated Services Digital Networks

The concept of BISDN has undergone considerable discussion during the past several years. In the evolution from the current telecommunication networks towards the BISDN, some important directions and guidelines have recently been made. It is expected that some new services such as teleconference, video based education, high speed data transfer, video on demand and High Definition TV(HDTV) will be added to existing services. Preferably, all these services should be provided by a single network, rather than a number of different networks(as is the case currently). BISDN is regarded as an all-purpose digital network. Activities currently under way are leading towards the development of a worldwide networking technology based on a common set of user interfaces and universal communications. Once deployed throughout the world, BISDNs will facilitate worldwide information exchange between any two subscribers without any of the limitations that can be imposed by the communication media.

ITU-T(Telecommunication Standardization Sector of International Telecommunication Union) Recommendation I.113 defines "broadband" as "a service or system requiring transmission channels capable of supporting rates that are greater than the primary access rate" [1]. Currently, BISDN interfaces support up to 622 Mb/s, with the possibility of defining higher rates in the future. ITU-T classified possible broadband applications into four categories [2]: conversational services, retrieval services, messaging services and distribution services. BISDNs will support services with both constant and variable bit rates, interactive and distributive services, bursty and continuous traffic, connection-oriented and connectionless services, and point to point and complex communications, all in the same network. Hence, at least conceptually, BISDNs not only support all types of existing communication applications, but also provide the framework to support future applications that are not fully understood, or even known of, today. Accordingly, a BISDN should be capable of allocating usable capacity dynamically on demand while taking the bursty nature of some applications into consideration. Also, BISDN switching fabrics should be capable of switching all types of services.

The introduction of highly reliable fibre systems into the access network provides the necessary high bandwidth required for BISDN. However, there are a number of issues that need to be satisfactorily addressed before BISDN networks become a reality [3]. As technology advances rapidly to meet the need for high-speed communications, the bottlenecks in communication networks are moving from the transmission media to the communications processors. The throughput and end-to-end delay requirements of applications are now limited by the processing power at network nodes, necessitating fast network protocols. It is still not clear whether current network protocols are suitable for BISDN services, or a new protocol needs to be designed. Congestion control is another major area that needs further investigation. BISDNs will support a large number of connections with different traffic characteristics simultaneously in the network. Simple call control schemes used in today's telephone systems or hop-by-hop flow control used in current packet networks can no longer be effective in BISDN networks. The problem is further complicated by the introduction of high-bandwidth links with relatively large propagation delays into the backbone.

2.2 Asynchronous Transfer Mode

Both the need for a flexible network and the progress in technology and system concepts led to the definition of the Asynchronous Transfer Mode. ATM is the transfer mode of choice for BISDN. In ATM, user information is transmitted between communicating entities using fixed-size packets, referred to as the ATM cells. An ATM cell is 53 bytes, consisting of a 48-byte information field and a 5-byte header, as shown in Figure 2.1.

2.2.1 Transfer Modes

ITU-T defines the transfer mode as a technique used for transmission, multiplexing, and switching aspects of communication networks. The most commonly used types of transfer modes can be categorized as follows: circuit switching, message switching and packet switching(consisting of datagram packet switching and virtual circuit packet switching).

In general, it is envisaged that the chosen transfer mode of BISDN should have the following properties [3]:

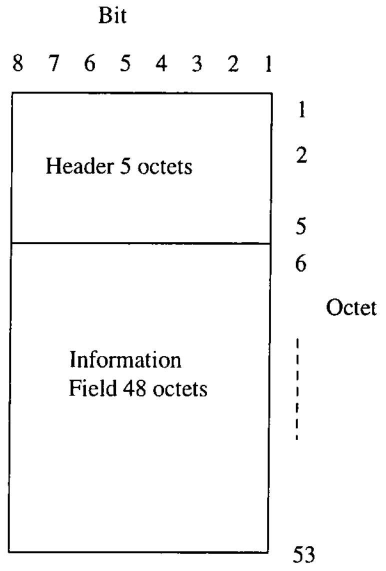

Figure 2.1: ATM cell structure

1. Supports all existing services as well as those with unknown characteristics that will emerge in the future.
2. Utilizes network resources as efficiently as possible.
3. Minimizes the complexity of switching.
4. Minimizes the processing time at the intermediate nodes to be able to support very high transmission speeds.
5. Minimizes the amount of buffers required at the intermediate nodes to bound the delay and the complexity of buffer management.
6. Guarantees performance requirements of existing and expected applications.

ATM is an attempt to meet all these objectives in a unique manner. Compared with the other transfer modes, it is closest to virtual circuit packet switching, in which all packets are of the same size. ATM has various features that extend the capabilities of current packet switching networks by incorporating the most desirable features of circuit switching to support real-time traffic most efficiently.

ATM is a connection-oriented protocol that supports both connection-oriented and connectionless services, with Constant Bit Rate(CBR) and Variable Bit Rate(VBR) traffic characteristics. The short cell size of ATM at high transmission rates is expected to offer full-bandwidth flexibility and high-bandwidth utilisation, and provide a wide range of quality of services required by various applications through statistical multiplexing. The term statistical multiplexing refers to the fact that several connections share a link with a capacity less than the sum of their peak bandwidth requirements, whereas the

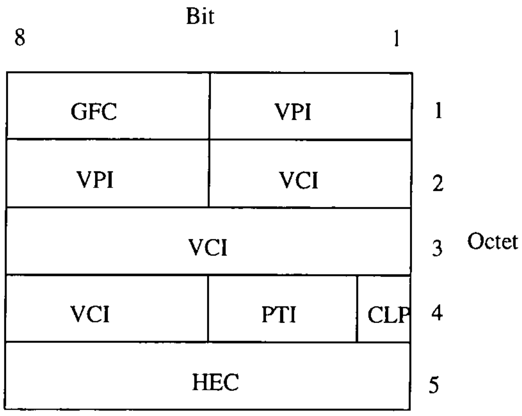

Figure 2.2: ATM cell header format at the UNI

term asynchronous means that cells of an information unit may appear at irregular intervals over the network links.

ATM has been accepted as the ultimate solution for the BISDN by ITU-T, and plans are being made by different organizations to realize experimental ATM pilots. Examples of these experiments are several RACE project trials, the Belgian broadband experiment and the US multigigabit project [4]. A non-profit organization, ATM Forum, has also been founded joining all types of industry (computer and telecommunication) with over 100 members worldwide. There is an "ATM fever", somewhat analogous to the "digital fever" in telecommunication networking of the 1970s and 1980s. On the other hand, ATM is also the subject of a heated debate. For example, Lea argues [5] that the two main features of ATM, statistical multiplexing and continuous bit rate, have made ATM a bundle of contradictions. Moving its protocol's functions to the edge to reduce the processing time as much as possible is ATM's fundamental principle, but the new trend is adding more functions inside the network. ATM is intended to statistical multiplex all sorts of traffic, but to make it work we have to demand that all diverse traffics conform to some prescribed distributions. ATM intends to handle any bit rate, but we may not even know the grade of service of our network. ATM advocates claim a charge-by-usage policy, but the real policy is charge-by-behaviour. Therefore, according to Lea, ATM's current goal: total flexibility in bandwidth allocation and utilisation is questionable.

2.2.2 The ATM Cell Format

ATM employs fixed-size cells with a 5-byte header and a 48-byte information payload. Two different formats for the cell header are adopted, respectively, for the User-Network Interface(UNI) at the edge of the network and the Network Node Interface(NNI) at the network nodes. They are shown in Figure 2.2 and Figure 2.3.

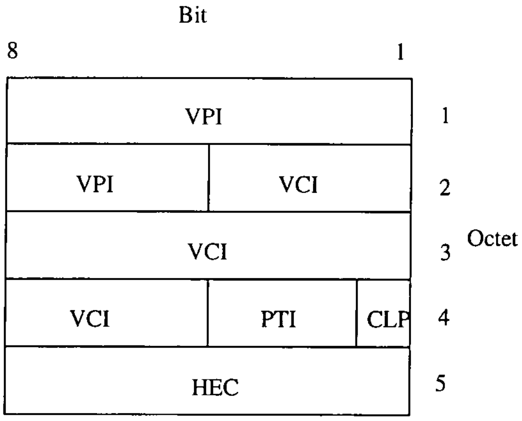

Figure 2.3: ATM cell header format at the NNI

The four-bit GFC(Generic Flow Control) in the UNI header permits multiplexing the transmissions of several terminals on the same interface. The GFC field has no use within the network and is meant to be used by access mechanisms that implement different access levels and priorities. Two modes of operation are defined for the GFC field: uncontrolled access and controlled access.

ATM uses labeled-channel multiplexing in which a label in the cell header, called the "connection identifier", explicitly associates the cell with a given virtual channel on a physical link. The connection identifier, which consists of two sub-fields, the Virtual Channel Identifier(VCI), a 16-bit field, and the Virtual Path Identifier(VPI), an 8 or 12-bit field, is used in multiplexing, demultiplexing, and switching the cells through the network. The two levels of routing hierarchies, Virtual Channels(VC) and Virtual Paths(VP), are defined in ITU-T Recommendation I.113 [1] as follows:

- VC is a concept used to describe unidirectional transport of ATM cells associated by a common unique-identifier value, referred to as the VCI.
- VP is a concept used to describe unidirectional transport of cells belonging to VCs that are associated by a common unique-identifier value, referred to as the VPI.

A VP is a collection of a set of VCs between two nodes in a BISDN. A predefined route is associated with each VP in the physical network. Furthermore, each VP has its own bandwidth, limiting the number of VCs that can be multiplexed on a VP. VPs can be viewed as semi-permanent connections in the network. VPIs are used to route packets between two nodes that originate, remove, or terminate the VPs, whereas VCIs are used at the end nodes to distinguish between different connections.

The 3-bit PTI(Payload Type Identifier) field has recently been redefined to indicate whether the cell contains upper-layer management information or user data. The

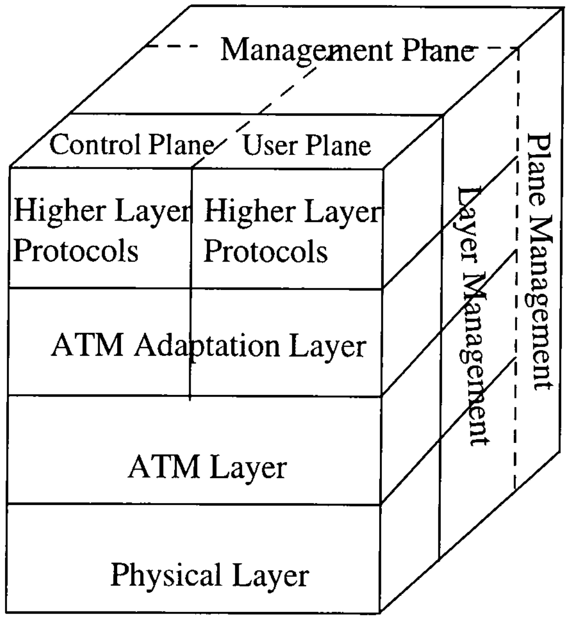

Figure 2.4: The BISDN ATM protocol reference model

CLP(Cell Loss Priority) bit is used for buffer management in conjunction with congestion control(cell discarding process). The HEC(Header Error Control) field allows to either correct single bit errors or detect multiple bit errors.

2.2.3 The ATM Protocol Reference Model

The OSI(Open Systems Interconnection) model of ISO(International Organization for Standardization) is well known and used with great success to model all sorts of communication networks. Using the same model as in OSI, the following ATM BISDN protocol reference model was defined(Figure 2.4) [6]. This model consists of three planes: a user plane to transport user information, a control plane mainly composed of signalling information, and a management plane used to maintain the network and to perform operational functions. In addition, a third dimension is added to this model, called the plane management, which is responsible for the management of the different planes. The user plane and control plane each has a layered structure to describe the functions associated with each layer. The layers are the physical layer, the ATM layer, the ATM Adaptation Layer(AAL), and layers above the AAL.

The physical layer transports ATM cells between two ATM entities and is based on SONET(Synchronous Optical Network) transmission standards. This layer also guarantees within a certain probability the cell header integrity and merges user cells with the transmission overhead to generate a continuous bit stream across the physical medium. The ATM layer is common to all services and provides cell transfer capabilities. In other words, the ATM layer corresponds to the boundary between functions devoted to the header and functions devoted to the information field. It is fully independent of the physical medium used. The ATM layer provides cell multiplexing, demultiplexing, and routing functions using the VPI and VCI fields of the cell header. Furthermore, the ATM layer may supervise cell flow to ensure that connections stay within the limits negotiated at the call establishment phase. The ATM layer is also responsible for cell sequence integrity for each source.

Since ATM supports many kinds of services with different traffic characteristics and system requirements, the AAL adapts the different classes of applications to the ATM layer. The AAL functions can be classified into two categories: continuous bit stream oriented services adaptation functions and bursty data services adaptation functions. The AAL consists of two sublayers: the Segmentation And Reassembly(SAR) sublayer and the Convergence Sublayer(CS). The main purpose of the SAR sublayer is segmentation of the higher layer information into a size suitable for the payload of the consecutive ATM cells of a virtual connection, and the inverse operation, reassembly of contents of the cells of a virtual connection, into data units to be delivered to the higher layer. The convergence sublayer performs functions like message identification, time/clock recovery, etc.

Three sets of requirements of BISDN services used to classify AAL functions are defined by ITU-T:

- Time relation versus no time relation between source and destination;
- Constant versus variable bit rate;
- Connection-oriented versus connectionless services.

Only four types out of the theoretically eight combinations of those three parameters result in valid existing services. These four classes are defined as(see Figure 2.5):

- Class A. This class corresponds to constant bit rate connection-oriented services with a timing relation between source and destination. The two typical examples are 64 kb/s voice and constant bit rate video.
- Class B. This class corresponds to variable bit rate connection-oriented services with a timing relation between source and destination. Typical examples are variable bit rate video and audio.
- Class C. This class corresponds to variable bit rate connection-oriented services with no timing relation between source and destination. An example is connection-oriented data transfer.
- Class D. This class corresponds to variable bit rate connectionless services with no timing relation between source and destination. An example of such a service is connectionless data transport.

	Class A	Class B	Class C	Class D
Timing between source and destination	required		not required	
Bit rate	constant	variable		
Connection mode	connection-oriented			connectionless

Figure 2.5: Service classes for AAL

Corresponding to these four classes, four types of AAL protocols have been recommended up to now by ITU-T, namely AAL 1, AAL 2, AAL 3/4 and AAL 5. Recommendation I.362 [7] states that CBR services will utilise AAL Type 1, but other AAL protocols for CBR are for further study. Connectionless data services will use AAL Type 3/4. Frame Relay services will use AAL 5. The specific association of other services with an AAL type is still for further study. AAL 5 may be recommended for signalling information.

2.3 Summary

The basic concepts and main features of BISDN and ATM have been described. In particular, the ATM cell header fields and the ATM protocol reference model have been discussed in detail. The ATM-based BISDN has the very ambitious goal of eventually becoming the unique means of communication all over the world. However, there are still a number of unsolved research problems in the area of ATM networks. Among them, traffic and congestion control is the main issue conditioning the availability of ATM services. We will address this topic in the next chapter.

Chapter 3

ATM Traffic and Congestion Control

The success or failure of ATM networks depends on the development of an effective congestion control framework. This chapter critically reviews some of the traffic and congestion control approaches for ATM networks. Several control functions are classified and their related issues are discussed separately. Research in this area is advancing very rapidly, so it is impossible to cover the whole range of the relevant work in the literature. The main focus of this chapter is placed on some representative strategies.

3.1 Quality of Service Attributes and Service Categories

We begin with a discussion of various quality of service attributes and service categories.

3.1.1 Quality of Service Attributes

While setting up a connection on ATM networks, users can specify the following parameters related to the input traffic characteristics and the desired quality of service [8]:

1. **Peak Cell Rate(PCR)**: The maximum instantaneous rate at which the user will transmit.

2. **Sustainable Cell Rate(SCR)**: This is the average rate as measured over a long interval.

3. **Cell Loss Ratio(CLR)**: The percentage of cells that are lost in the network due to error or congestion and are not delivered to the destination, i.e.,

$$\text{CLR} = \frac{\text{Number of lost cells}}{\text{Number of transmitted cells}}. \tag{3.1}$$

Recall that each ATM cell has a Cell Loss Priority(CLP) bit in the header. During periods of congestion, the network will first discard cells that have CLP bit set. Since the loss of cells with CLP = 0 is more harmful to the operation of the application, CLR may be specified separately for cells with CLP = 1 and for those with CLP = 0.

4. **Cell Transfer Delay(CTD)**: The delay experienced by a cell between network entry and exit points is called the cell transfer delay. It includes propagation delays, queueing delays at various intermediate switches, and service times at queueing points.

5. **Cell Delay Variation(CDV)**: This is a measure of variance of CTD. High variation implies larger buffering for delay-sensitive traffic such as voice and video.

6. **Cell Delay Variation Tolerance(CDVT) and Burst Tolerance(BT)**: For sources transmitting at any given rate, a slight variation in the inter-cell time is allowed. A leaky bucket(which will be described later) type algorithm called "Generic Cell Rate Algorithm(GCRA)" is used to determine if the variation in the inter-cell times is acceptable. The bucket size parameter of the GCRA used to enforce PCR is called cell delay variation tolerance and of that used to enforce SCR is called burst tolerance.

7. **Maximum Burst Size(MBS)**: This is the maximum number of back to back cells that can be sent at the peak cell rate but without violating the sustainable cell rate. BT and MBS are related as follows:

$$\mathrm{BT} = (\mathrm{MBS} - 1)\left(\frac{1}{\mathrm{SCR}} - \frac{1}{\mathrm{PCR}}\right). \tag{3.2}$$

   Note that PCR, SCR, CDVT, BT, and MBS are input traffic characteristics and are enforced by the network at the network entry. CLR, CTD and CDV are qualities of service provided by the network and are measured at the network exit point.

8. **Minimum Cell Rate(MCR)**: This is the minimum rate desired by a user.

3.1.2 Service Categories

The above QOS attributes help define various classes of service. There are five categories of service. The QOS parameters for these categories are summarized in Table 3.1 and are explained in the following [8]:

- **Constant Bit Rate(CBR)**: This class is used for emulating circuit switching, where the bit rate is constant. Cell loss ratio is specified for cells with CLP = 0 and may or may not be specified for cells with CLP = 1. Examples of applications that can use CBR are telephone, video conferencing, and television.

Attribute	CBR	RT-VBR	NRT-VBR	ABR	UBR
CLR for CLP=0	S	S	S	S	U
CLR for CLP=1	O	O	O	S	U
CTD	S	S	S	U	U
CDV	S	S	U	U	U
SCR and BT	N	S	S	N	N
PCR and CDVT	S	S	S	S	S
MCR	N	N	N	S	N

Table 3.1: ATM layer service categories(S: Specified, U: Unspecified, O: Optional, N: Not applicable)

- **Variable Bit Rate(VBR)**: This class allows users to send at a variable rate. Statistical multiplexing is used and may result in small nonzero random loss. Depending upon whether or not the application is sensitive to cell delay variation, this class is subdivided into two categories: real-time VBR(RT-VBR) and nonreal-time VBR(NRT-VBR). For NRT-VBR, only mean delay is specified, while for RT-VBR, maximum delay and peak-to-peak CDV are specified. An example of RT-VBR is interactive compressed video while that of NRT-VBR is multimedia email.

- **Available Bit Rate(ABR)**: This class is designed for normal data traffic such as file transfer and email. Although the standard does not require the cell transfer delay and cell loss ratio to be guaranteed, it is desirable for switches to minimize the delay and loss as much as possible. Depending upon the congestion state of the network, the source is required to control its rate. The users are allowed to declare a minimum cell rate, which is guaranteed to the VC by the network. Most VCs will ask for an MCR of zero. Those with higher MCR may be denied connection if sufficient bandwidth is not available.

- **Unspecified Bit Rate(UBR)**: This class is designed for those data applications that want to use any left-over capacity and are not sensitive to cell loss or delay. Such connections are not subject to admission control and not policed for their usage behaviour. Examples of UBR applications are email and file transfer.

3.2 Congestion Control Problem

Viewed as a new paradigm for the future BISDN [9], ATM can achieve total service integration, total flexibility and efficiency in bandwidth allocation and utilisation. In ATM, VBR, or bursty, traffic streams are statistically multiplexed. Statistical multiplexing is more bandwidth-efficient and allows more calls to enter the network. However, with the BISDN/ATM goals of supporting diverse service and traffic mixes, and

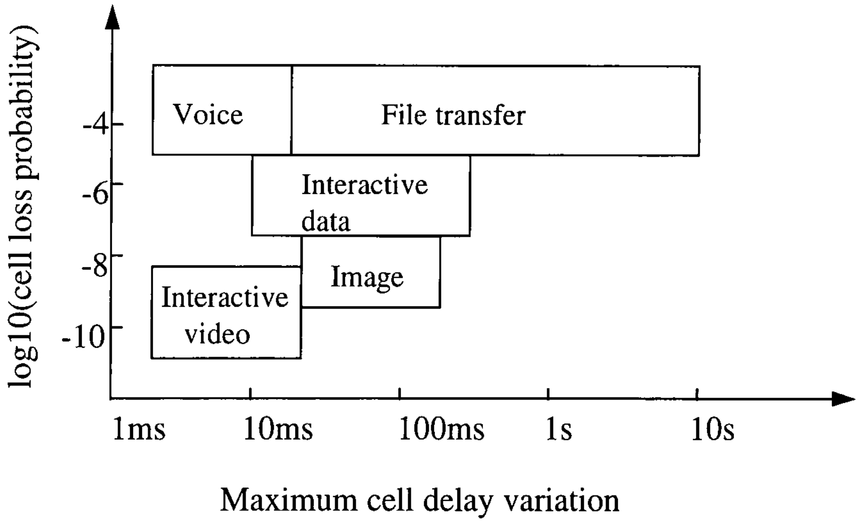

Figure 3.1: Approximate ATM traffic performance requirements

of efficient network resource engineering, the design of a congestion control becomes an important challenge. The response to this challenge will greatly influence the ability of BISDN/ATM to compete, and thus the eventual viability of BISDN/ATM [10] [11].

According to ITU-T Recommendation I.371 [12], in BISDN, congestion is defined as a state of network elements(e.g. switches, concentrators, cross-connects and transmission links) in which the network is not able to meet the negotiated network performance objectives for the already established connections and/or for the new connection requests. So the primary role of traffic control and congestion control for BISDN is to protect the network and the user in order to achieve network performance objectives. An additional role is to optimize the use of network resources. The following are the high-level goals of a BISDN/ATM congestion control architecture [13] [14]:

- Flexibility: it should support a set of ATM layer QOS classes sufficient for all existing and foreseeable services.
- Simplicity: simple control algorithms are more likely to prove implementable.
- Robustness: the requirement of achieving high resource efficiency under any traffic circumstance while maintaining simple control functions.
- Controllability: through this control architecture, congestion can be adequately controlled so that efficient network resource utilisation is achieved without paying a penalty in performance.

Despite the past experience gained from circuit-switched and packet-switched networks, congestion control in ATM networks remains an unresolved issue. Some aspects of ATM networks that complicate the control problem include [3] [15]:

1. Various BISDN VBR sources generate traffic at significantly different rates. The bit generation rates of some applications can often have time-varying nature. Fur-

thermore, a single source may generate multiple types of traffic with different characteristics. To allow statistical multiplexing, bursty calls should only be allocated some bandwidth less than the peak rate. Determining how much bandwidth to allocate to a bursty call must be resolved.

2. In addition to the performance metrics of call blocking and packet loss probabilities in current networks, ATM networks have to deal with cell delay variation, maximum delay, etc.

3. Different services have different QOS requirements at considerably varying levels. A service class is a set of services that have the same QOS requirement. These requirements are usually measured in terms of maximum delays and cell loss rates. Figure 3.1 shows approximate delay and loss requirements for some expected services [16]. The service requirement of the traffic can be delay-sensitive(such as voice) or loss-sensitive(such as image transfer). In ATM, even if a call is admitted to the network, the network delay and loss may not be guaranteed due to ATM's packet switching nature.

4. Traffic characteristics of various types of services are not well understood.

5. Two speeds for BISDN access have been recommended by ITU-T, namely, 155 Mb/s and 622 Mb/s. One effect of a high-speed channel is that at these link speeds, cells must be switched at a rate greater than one cell per $3\mu s$ or $0.7\mu s$. Internal links may operate at the rate of Gb/s. Therefore the cell processing schemes in ATM must be simple enough so as to be performed at speeds comparable to the high switching speeds. Another problem caused by the high link rate is the increased propagation delay-bandwidth product, the amount of traffic that can be in transit during a propagation delay time. This can make some feedback congestion control schemes inefficient.

In general, congestion control procedures can be grouped into two categories: preventive control and reactive control. In preventive control one sets up schemes which prevent the occurance of congestion. Connection Admission Control(CAC), Usage Parameter Control(UPC), Priority Control(PC), Traffic Shaping(TS) and Fast Reservation Protocol(FRP) are examples of preventive control. In reactive control one relies on feedback information for controlling the level of congestion. Explicit Congestion Notification(ECN) is one of the proposed methods for reactive control. The main problem with the reactive scheme is the large propagation delay-bandwidth product in ATM networks, which introduces the unique problem that by the time a source receives a notification it may be too late to react.

A classification of the congestion control functions according to their location within the network is shown in Figure 3.2 [17]. The network level allocates resources to virtual

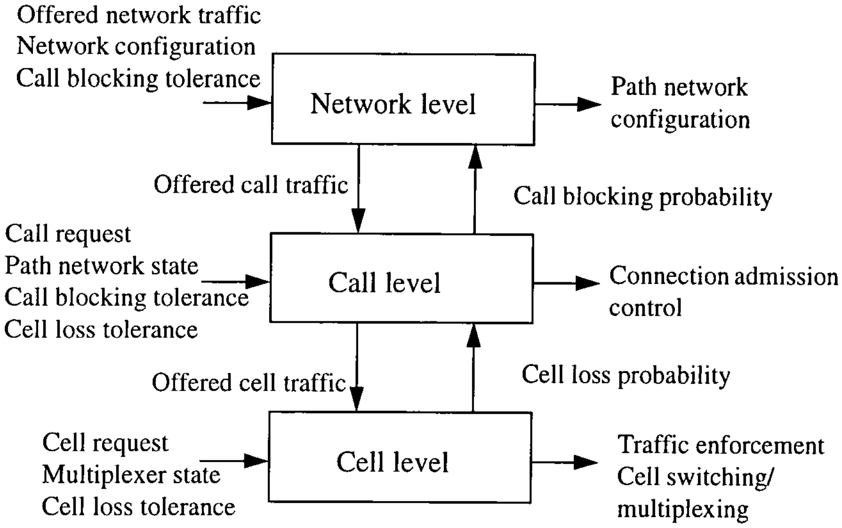

Figure 3.2: ATM network multilevel control model

paths, given the offered call traffic and tolerated call blocking probabilities. The virtual path concept allows several calls to be switched and handled together, which simplifies CAC, but decreases the network utilisation. The call level performs CAC within the path network and allocates bandwidth and switch buffer capacity to individual calls. The cell level allocates resources during the cell transfer phase, and is responsible for cell traffic enforcement(also called "policing") at access switches and arbitration of cells during switch overload. We can also classify control options according to time scales at which they are most effective(see Figure 3.3) [3]. Various control functions will be discussed in the following sections.

3.3 Resource Provisioning

Resource provisioning methods determine the physical quantities of equipment to be placed in the ATM network. The network topology, the number of links and their bandwidths, and the number of switching and access nodes are all determined based on some understanding of traffic requirements. As time evolves, the number of users, the amount of traffic generated, and the types of applications used in networks change. Therefore, with network provisioning, the challenge is to ensure that sufficient resources are available to accept all potential connections, while still maintaining a cost-effective network design. This leads to a tradeoff between the quantities of resources that should be placed in the network and the expected utilisation that they can achieve. For resource provisioning, long-term measurements of switch and trunk utilisation must be collected and forecasted against future predicted subscriber loads and usage characteristics [10].

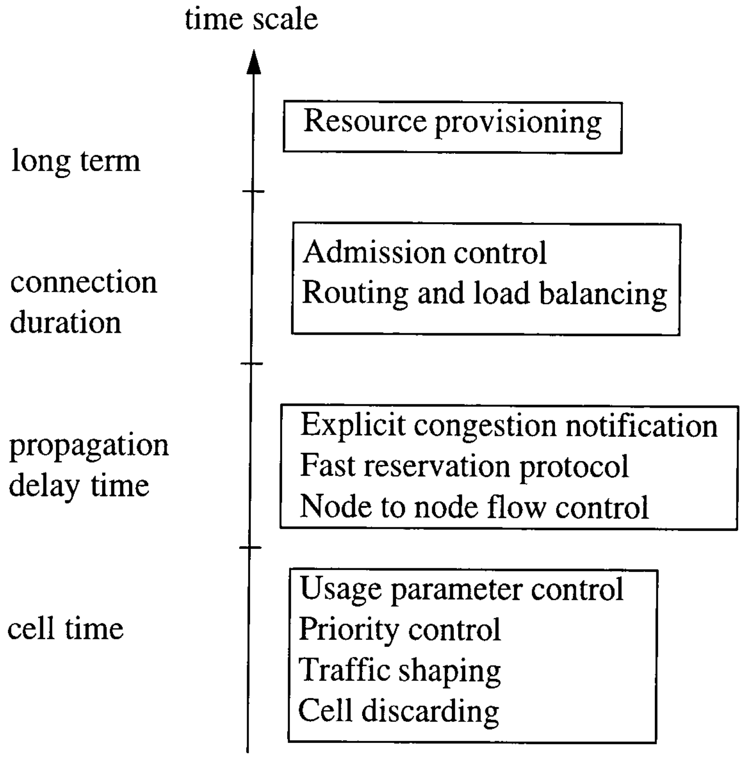

Figure 3.3: ATM traffic control options at different time scales

3.4 Connection Admission Control

Connection admission control represents the set of actions taken by the network at call set-up phase in order to accept or reject an ATM connection [12]. A connection request for a given call is accepted only when sufficient resources are available to carry the new connection through the whole network as its required QOS(e.g., cell loss probability, cell delay) while maintaining the agreed QOS of already established connections in the network. Accordingly, there are two questions that need to be answered [3]. The first question is how to determine the amount of bandwidth required by a new connection, while the second one is how to assure that the QOS required by existing connections are not affected when multiplexed together with this new connection. Any technique designed to answer these two questions should do so in real-time and attempt to maximize the utilisation of network resources.

3.4.1 Bandwidth Allocation

Admission control is based on bandwidth allocation. There are two alternative approaches for bandwidth allocation: deterministic multiplexing and statistical multiplexing. In deterministic multiplexing, each connection is allocated its peak bandwidth. Doing so causes large amount of bandwidth to be wasted for bursty connections, particularly for those with large peak-to-average bit rate ratios. This goes against the philosophy of the ATM framework since it does not take advantage of the multiplexing capability of ATM and restricts the utilisation of network resources. An alternative method is statistical multiplexing. In this scheme, the amount of bandwidth allocated in the network to a VBR source is less than its peak, but necessarily greater than its average bit rate. Hence, statistical multiplexing allows more connections to be multiplexed in the network than deterministic multiplexing, thereby allowing better utilisation of network resources.

In general, efficiency gain due to statistical multiplexing is dependent on several factors. The most important factor is the ratio of the peak bit rate of the call to the link rate. The peak-to-link rate ratio must not increase above 0.1 in order to have an effective statistical multiplexing gain and avoid congestion [15]. If statistical multiplexing is profitable, then burstiness, defined as the peak-to-average bit rate ratio, is the second important factor. The third factor is burst length. There are some other factors, including mean bit rate, cell loss and delay requirements of calls [18]. The above traffic parameters may be included in a set of traffic descriptors specified by the users or monitored by the network.

3.4.2 CAC Algorithms

In this section, we discuss various call admission algorithms proposed in the literature. It should be pointed out that although these techniques may be used in the early deployments of ATM networks, they do have drawbacks and still need to be further improved. For example, some of the algorithms may allocate more bandwidth in the network than that required to provide the QOS requirements for connections, thereby causing underutilisation of network resources.

One of the important research issues in admission control is to investigate the effect of various traffic parameters on the allocated bandwidth. In [19], a method is proposed to calculate the bandwidth required to satisfy a given performance requirement. In the case where homogeneous traffic sources are multiplexed, the bandwidth required to satisfy a given cell loss requirement is given by

$$W = R(b,n,L)nB/b, \tag{3.3}$$

where n is the number of active traffic sources; B is the peak bit rate; b is a measure of burstiness defined as the peak-to-mean bit rate ratio; L is the mean number of cells generated from a burst; and $R(b,n,L)$ is a coefficient called expansion factor, whose value depends on the triplet (b,n,L). Thus, the peak-to-mean bit rate ratio and the mean number of cells generated in a burst are used to determine the required bandwidth. To implement this approach, the values of $R(b,n,L)$ need to be precomputed through the simulation and stored in each node. Therefore, the number of possible combinations of (b,n,L) needs to be tractably small. This may limit the size of the network.

In [20], a call is characterized by three parameters that capture the essential features of the traffic source(e.g., the peak bit rate, utilisation, and average active time). Using this information the bandwidth required for a new connection can be estimated from the combination of two approximations. The first approximation is based on the so-called "fluid flow model" and is to estimate the "equivalent capacity" when the impact of the individual connection characteristics is critical. This method may significantly overestimate the actual value of the required bandwidth for the aggregate traffic since the interaction between individual connections is not taken into consideration. To capture the effect of multiplexing, a second approximation is used. This approximation assumes that the aggregate traffic from a large number of connections is of Gaussian distribution and determines the required bandwidth according to a calculation of the mean and standard deviation of the Gaussian distribution. The Gaussian approximation is also an overestimate since it fails to account for the link buffers. The call admission procedure is as follows. Given the parameters of a new connection and the current values of the existing traffic statistics, calculate the total bandwidth by taking the minimum of the above two approximations. If this bandwidth is less than the provisioned link bandwidth, then accept the connection, otherwise reject it. It should be noted that in some cases admission control based on the equivalent capacity algorithm may be very conservative and we will address this problem in more detail in later chapters.

To perform CAC efficiently, one must provide efficient bandwidth allocation and management schemes. Various methods have been studied, including fast buffer reservation [21], the partial allocation scheme [22] and the virtual path allocation scheme [15]. Saito [23] presents dynamic call admission control using the distribution of the number of cells arriving during the fixed interval. In this scheme, the control unit continues to use traffic parameters specified by users, but improves the estimated distribution by data obtained from measurement. Call acceptance is decided on the basis of on-line evaluation of the upper bound of cell loss probability, derived from the above distribution. This control mechanism is effective when the number of call classes is large.

One of the drawbacks of most of the current CAC approaches is that only the first-moment statistics(peak rate, average rate, average burst length) are used, since a second-moment algorithm will be computation-intensive. This raises a question: Can the first-moment statistics fully and correctly characterize the traffic of ATM networks? Second-moment or even higher-moment statistics may be needed.

Because the CAC mechanisms rely on the traffic parameters negotiated during call establishment, these parameters must be enforced to ensure proper functioning.

3.5 Usage Parameter Control

Usage parameter control(UPC, i.e., policing) is defined as the set of actions taken by the network to monitor and control traffic in terms of traffic offered and validity of the ATM connection at the user access [12]. Its main purpose is to ensure that the traffic generated by a source conforms to that assumed for the bandwidth allocation, i.e., that the source stays within its "contract". If a violation of this contract is detected, the policing mechanism will enforce the original contractual parameters by an appropriate action. This action could be [3]:

- Dropping nonconforming cells;
- Delaying nonconforming cells in a queue so that the departure from the queue conforms to the contract;
- Marking violating cells differently than the cells that stay within the negotiated parameters and transmitting them so that the network can treat them differently when congestion arises;
- Adaptively controlling the traffic by throttling the source bit rate.

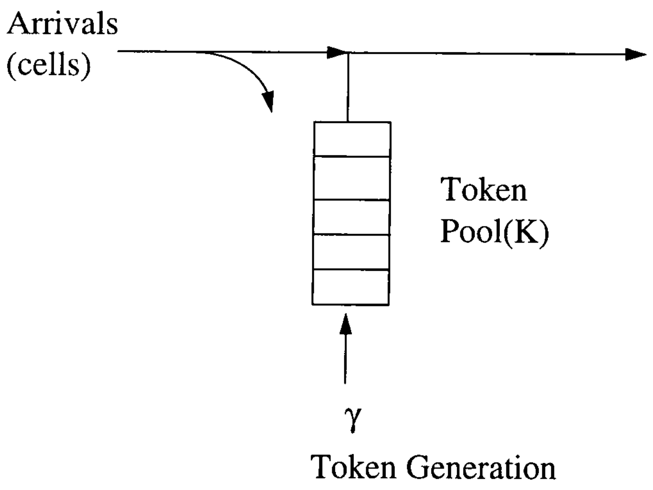

Figure 3.4: Leaky bucket traffic policing scheme

For "well behaving" users or subnetworks, UPC should be transparent.

There are several reasons for exceeding the contract(a malfunctioning terminal or a deliberate attempt by the user). If a source exceeds its contract, this may affect the QOS of other sources. Also, there is the possibility of revenue loss. So policing is very important to the network operator. Various policing methods have been proposed in the literature. In most of these schemes, the controlled source parameters include the peak and average bit rates and the length of active periods.

The leaky bucket [24] is an effective policing scheme and is well received [25]. In a leaky bucket scheme(Figure 3.4), a cell is accepted only when it can draw a token from a token pool, the leaky bucket. If no tokens are available, the cell is lost. Tokens are generated at a fixed rate γ and stored in the token pool. The pool has a finite size of K. If the token pool is full, the generated tokens are lost. The size of the token pool imposes an upper bound on the burst length and determines the number of cells that can be transmitted back to back. As tokens are generated at a constant rate, this scheme can be used to control either the peak or the average cell transmission rate(but not both). A token pool can be implemented using a counter that increases when tokens are generated and decreases when tokens are used.

Policing can be combined with shaping in a system in which cells queue instead of being discarded when the token pool is empty(Figure 3.5). The cell blocking probability, the probability that a cell arrives to an empty token pool, depends on the sum of the capacity of the cell buffer and the token pool. This implies that by increasing the token pool capacity, the cell buffer can be eliminated without affecting the steady state throughput and blocking. This is desirable, if the network can handle larger bursts, since delay due to a cell buffer can be reduced and the implementation cost of a large token pool is smaller than that of a large cell buffer.

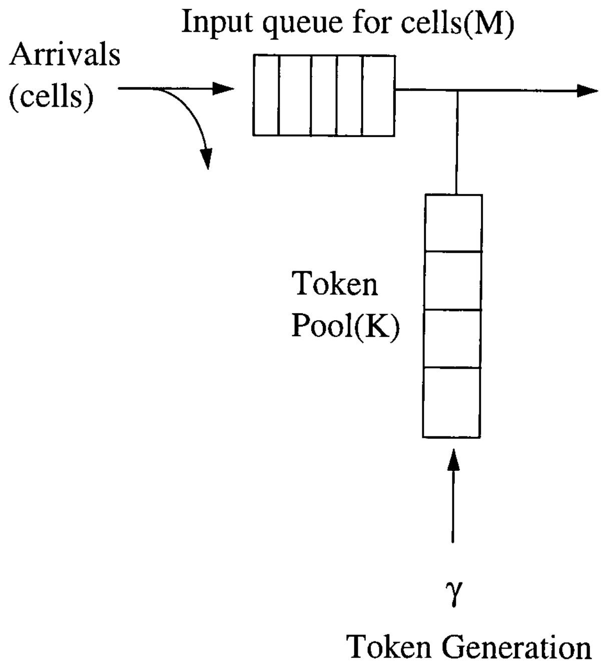

Figure 3.5: Buffered leaky bucket

One disadvantage of the leaky bucket is that the bandwidth enforcement the token pool introduces is operational even when the network load is light. In addition, cells may be lost even though the long term average rate of the source is within the allocated bandwidth. To solve this problem, a virtual leaky bucket has been proposed [19]. In a virtual leaky bucket, cells arriving at an empty token pool are marked and transmitted without a token, while those with a token are unmarked. Marked cells are considered violators of allocated bandwidth since the call must have exceeded the allocated bit rate for some time for the token pool to be empty. Because bandwidth may still be available in the network, marking cells allows the call to exceed its allocated bit rate if it does not adversely affect other calls. If at some point along its path a marked cell reaches a congested link, it may be discarded so the throughput of the unmarked cells is not severely affected. Marking not only allows flexibility for the user to exceed allocated bandwidth, but flexibility for the network in determining allocation as well.

One disadvantage of the virtual leaky bucket is that the marking system has no correlation to user level data priority. With the current ATM cell structure, a conflict may arise when the one-bit CLP field is used to implement both(contradicting) priority assignment. Bemmel and Ilyas [26] propose a solution to this problem based on a new 4-class priority strategy that unifies the two marking approaches, by utilising a 2-bit CLP field. A new variant of the marking leaky bucket UPC mechanism, called the "forgiving leaky bucket" is then positioned at the NNI of interworking ATM-based BISDN subnetworks. The scheme additionally has the power of unmarking(forgiving) previously

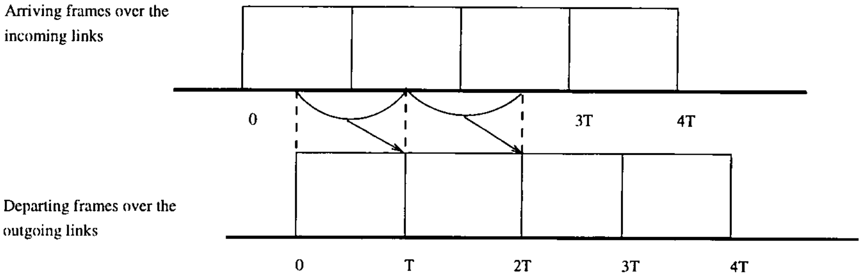
Arriving frames over the incoming links
0
3T
4T
Departing frames over the outgoing links
0
T
2T
3T
4T

Figure 3.6: Stop-and-go queueing

marked cells, wherever the network conditions are appropriate. It is shown that this new strategy provides a significant improvement over the traditional marking leaky bucket UPC mechanism.

In [27], some of the policing mechanisms are compared. They include: the "leaky bucket", the "jumping window", the "triggered jumping window", the "moving window" and the "Exponentially Weighted Moving Average"(EWMA). It is shown that the leaky bucket and EWMA are the most promising methods. The other window mechanisms are not flexible enough to cope with the short-term statistical fluctuations of the source traffic.

3.6 Traffic Shaping

A key element of the traffic contract from the user perspective is the sequence of cells that can be sent to the network and still be compliant with the traffic parameters in the traffic contract. The method specified in standards is called "traffic shaping". In other words, the user equipment can process the source cell stream such that the resultant output toward the network is conforming to the traffic parameters according to the leaky bucket algorithm configuration in the traffic contract. Possible implementations of traffic shaping as proposed in the literature include buffering, spacing [28], peak cell rate reduction, scheduling, etc.

Even if cells enter the network smoothly, they can cluster together to form longer bursts at intermediate nodes in the network. Golestani [29] proposes stop-and-go queueing(also known as framing) as a possible solution to maintain the original smoothness through intermediate nodes in the network. In stop-and-go queueing, cells arriving during some smoothing interval, F, of length T, do not become eligible for transmission until the next smoothing interval, $F + 1$ (Figure 3.6). This method not only maintains the smoothness property of traffic entering the network, but also places an upper bound on a call's total queueing delay and required buffer space.

3.7 Priority Control

To provide multiple grades of services with ATM, we can use priorities between and within service classes. Having determined priority levels for various services, we must handle prioritized cells in an appropriate manner during cell discarding and scheduling. Priority in discarding determines which cells are dropped when network congestion occurs. Scheduling priority determines the order of cell transmission.

Priority schemes can be used as local congestion control schemes to satisfy different cell loss requirements of different classes of traffic. When congestion is detected, priority is given to loss-sensitive traffic over loss-insensitive traffic, and cells from lower priority classes are discarded first. Selective cell discarding is based on the fact that there may be more and less significant cells in voice and video coding. For video in BISDN, layered coding schemes such as subband(wavelet) and discrete cosine transform coding produce data of higher and lower perceptual significance. These can be conveyed in cells of different priorities. Two selective discarding mechanisms have received considerable attention in the literature: push out and threshold [3].

Various priority schemes can be used as a scheduling method at a switching node in an ATM network. The simplest priority scheme is the static priority scheme. In this, priority is always given to the more delay-sensitive class. This scheme frequently causes starvation for the less delay-sensitive traffic. To overcome the drawbacks of static priority scheme, a dynamic priority scheme is needed. In the proposal of [30], each class of service can be guaranteed a minimum of bandwidth, which can prevent the low-priority service class from starvation.

3.8 Reactive Congestion Control Mechanisms

It is generally agreed that preventive control techniques are not sufficient to eliminate congestion in ATM networks and that when congestion occurs it is necessary to react to the problem. When congestion is detected, sources are requested to slow down or stop transmission for a while, until the congestion is cleared. Reactive control mechanisms have been successfully used in low-speed packet-switched networks. However, as the propagation delay-bandwidth product increases significantly in ATM networks, reactive control mechanisms are not as effective as they are in low-speed packet-switched networks. The effectiveness of a reactive control method in the ATM environment mainly depends on the connection duration, the burst length, and the distance involved between the two communication entities [3]. Furthermore, in ATM networks, it may not be easy to identify which source is causing the congestion. Hence, most reactive schemes require a number of sources to throttle their traffic generation rates, which introduces the issue of fairness. The design and implementation of reactive control schemes in ATM networks remains an open issue. Nevertheless, they are required as safeguard mechanisms and may potentially be used to increase resource utilisations beyond what can be achieved by preventive schemes alone.

Once congestion is detected in an intermediate network node, the end nodes need to be notified in order to be able to react. Each node in the network monitors the queue occupancies of its trunks. When the queue size of a trunk at a given node reaches a predefined threshold value, it is thought that congestion happens at that node. There are two ways for congestion notification in ATM networks. One proposed reactive control method is Forward Explicit Congestion Notification(FECN) [10]. This scheme forwards the congestion condition along the path to the destination, implemented via a forward congestion indicator in the header of ATM cells. If this indicator is set upon arrival at the destination, then it signifies the presence of congestion at some point along the path. The destination can then signal back to the source to trigger appropriate actions to rectify the situation. The FECN schemes are effective only when the congestion duration is of some order of magnitude of the propagation delay. It is also possible for the network element to inform the source directly, but to do this the network element needs to generate a special cell to carry the message back to the source. This scheme is called Backward Explicit Congestion Notification(BECN) [31]. The overhead for this scheme may make it impractical. Some other reactive control schemes have also been proposed, such as adaptive rate control, in-call parameter negotiation and dynamic source coding [3].

3.9 Flow Control for Available Bit Rate ATM Service

Over the past two years, the ATM standards community has recognized that data traffic often requires no firm guarantee of bandwidth, but instead can be sent at whatever rate is convenient for the network. This is called Available Bit Rate(ABR) or "best-effort" traffic by the ATM Forum. ABR traffic gives the network the opportunity to offer guarantees to high priority traffic, and divide the remaining bandwidth among ABR connections. To support ABR service, the network requires a feedback mechanism in order to tell each source how much data to send. The two leading mechanisms are called credit-based flow control [32] and rate-based flow control [33].

The credit-based scheme requires link-by-link flow control and a separate buffer for each VC(per-VC queueing). Each link consists of a sender node and a receiver node. The receiver monitors queue lengths of each VC and determines the number of cells that the sender can transmit on that VC. This number is called "credit". The sender transmits only as many cells as allowed by the credit. This scheme as described so far is called "Flow Controlled Virtual Circuit(FCVC)" scheme and is considered by many switch vendors to be too expensive and inflexible.

The rate-based schemes make use of the "Explicit Forward Congestion Indication

(EFCI)", a particular combination of the PTI field in the cell header that can be set by the switches during congestion. The destination monitors these indications for a periodic interval and sends a Resource Management(RM) cell back to the source. The sources use an additive increase and multiplicative decrease algorithm to adjust their rates. These rate-based methods are based on end-to-end control and do not require per-VC queueing nor accounting. Thus, most switch vendors find the rate-based approaches appealing because of their simplicity and implementation flexibility. However, the early rate-based proposals were found to suffer problems of fairness. Many of the current rate-based approaches fix the problems by using the so-called "intelligent marking" technique without the need of per-VC queueing nor per-VC accounting [34]. In late 1994, the ATM Forum voted for rate-based flow control for supporting ABR services, but without committing to the details of any particular algorithm. Actually, both rate and credit solutions have their pros and cons and to a large extent they can be viewed as complementary. Thus, an integrated scheme that combines the advantages of both approaches into a single proposal for ATM flow control has been proposed [35]. It suggests that rate control is the most appropriate for the wide area, while static credit control has distinct advantages in the local area.

3.10 Summary

This chapter has given an overview of traffic management and congestion control in ATM networks. A number of control functions and associated algorithms have been described. As we pointed out earlier, the effectiveness of reactive control schemes is limited by the duration of feedback delays and requires very large buffers. On the other hand, preventive control techniques are often sensitive to the parameters of the source traffic, which itself is an open issue. Even with accurate traffic characterization, the proposed techniques often restrict utilisation of network resources. The problem is further complicated due to the existence of different applications with diverse QOS requirements. Therefore the congestion control framework in ATM networks remains an open issue and has been one of the most active areas of telecommunications research. The dynamic, heterogeneous, time-varying network environment, with different service requirements, drives the designer of congestion control mechanisms to investigate many new concepts and approaches [36] [37] [38] [39] [40].

Recently, some researchers have suggested that developments in neural network technology might provide capabilities that are well suited to the solution of some challenging, outstanding control problems in high-speed communication networks. The following chapter introduces some existing neural-network-based traffic control methods and proposes a novel congestion control approach using reinforcement learning.

Chapter 4

Neural Networks for Adaptive Congestion Control in ATM Networks

As we mentioned in previous chapters, ATM is designed to support a wide variety of services with different traffic characteristics and QOS requirements at the cell and call levels. ATM utilises the bursty nature of the traffic to effectively allocate the network resources via statistical multiplexing. Although statistical multiplexing provides efficient use of the network resources(e.g., bandwidth) and enough flexibility to support multiple connections with different bit rates, it does not come without a price. The price is the need to design elaborate traffic and congestion control mechanisms. The nature of ATM makes traffic control a challenging task. Most of the schemes proposed to date suffer from serious shortcomings [41]. Some are simple but include many approximations and assumptions that are hard to justify. Others include complicated mathematical solutions that are not feasible for real-time implementation.

The application of Neural Networks(NNs) and other Artificial Intelligence(AI) techniques is being recommended by many researchers to provide an alternative to conventional traffic control approaches for ATM networks. Neural networks are thought to have several properties that are valuable when implementing ATM congestion control. Their learning and adaptive capabilities can be utilised to construct adaptive control algorithms for optimal allocation of resources. In the mean time, the parallel structure of NNs can be exploited in hardware implementation, which provides short and predictable response times. In this chapter, we first provide a brief introduction of the basic concept of neural networks, followed by an overview of various applications of NNs to ATM traffic control. From the discussion of previous work, we try to answer the question "why neural networks in ATM traffic control?" and give useful comments on the strengths and limitations of NN-based methods. Then a novel adaptive congestion control approach

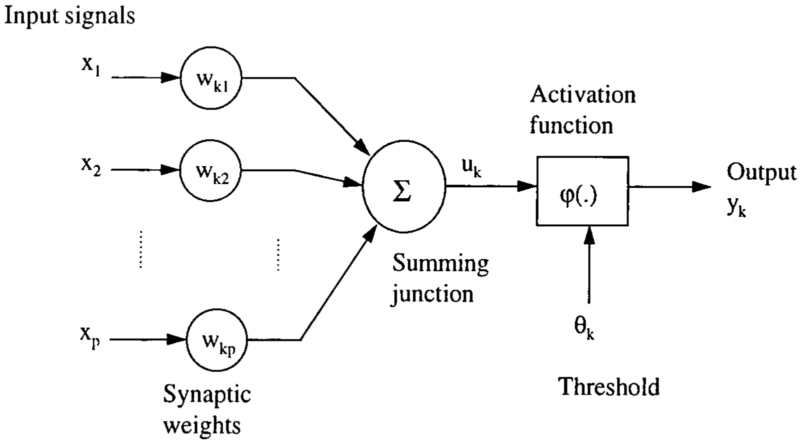

Figure 4.1: Nonlinear model of a neuron

based on a neural network that uses reinforcement learning is presented.

4.1 A Brief Review of Neural Networks

In spite of the fact that modern digital computers have made great progress in both speed and processing power, there are certain tasks that may not be performed satisfactorily by digital computers due to the complexities associated with the problems. Such tasks include optimization, pattern recognition, generalization, and classification. Neural networks can solve these complex problems since they do not require accurate modelling of the system under study. All that is required are examples of the relationship between given input and desired output variables. With proper training, a neural network model can learn such a relationship and produce accurate outputs even when it is fed by new input data. Work on neural networks has been motivated by the way that the human brain processes information. Accordingly, a neural network derives its computing power through its massively parallel distributed structure and its ability to learn and generalize. The use of neural networks offers some useful properties and capabilities [42]: nonlinearity, input-output mapping, adaptivity, fault tolerance, VLSI implementability, among others.

A neural network is composed of large numbers of basic information processing units called neurons that are interconnected in a certain topology. Figure 4.1 shows the model for a neuron. Each neuron accepts a number of input signals (from other neurons) $x_1, ..., x_p$ and has one output signal y_k that can be input to other neurons. There is a set of "synapses", each of which is characterized by a weight of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. An adder sums those weighted inputs and the result is called the linear combiner(u_k in Figure 4.1). An activation function φ defines the output of a neuron in terms of the activity level at its input. There are three basic types of activation functions, namely,

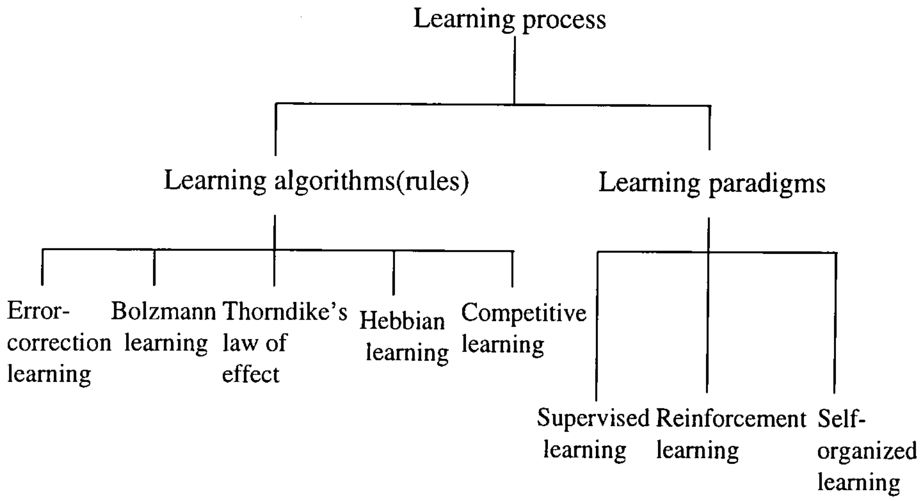

Figure 4.2: A taxonomy of the learning process

threshold function, piecewise-linear function and sigmoid function(or hyperbolic tangent function). The model of a neuron shown in Figure 4.1 also includes an externally applied threshold θ_k that has the effect of lowering the net input of the activation function.

The manner in which the neurons of a neural network are interconnected, i.e., the network structure can be divided into four classes [42]: single-layer feedforward network, multilayer feedforward network, recurrent network and lattice structure. Different structures are closely linked with different learning algorithms used to train the networks. Haykin provides a taxonomy of the learning process in [42], as shown in Figure 4.2. Among the learning algorithms, error-correction learning is rooted in optimum filtering, while both Hebbian learning and competitive learning are inspired by neurobiological theories. Boltzmann learning is different altogether in that it is based on thermodynamics and information theory. Among the learning paradigms, supervised learning is performed with the supervision of an external "teacher", i.e., it is supervised in the sense that one has to supply input-target vector pairs which give explicit instructions as to the desired network response. This is most often done off-line(learning phase), until the network is considered to have learned the task. The network is then put into operation(recall phase), where no learning takes place. The goal of the learning phase is to find a mapping which generalizes well to previously unseen data encountered in the recall phase. Reinforcement learning is the on-line learning of an input-output mapping through a process of trial and error designed to maximize a scalar performance index called a reinforcement signal. Unsupervised learning is also referred to as self-organized learning, where no external teacher or critic exists to oversee the learning process. Many neural network models based on the above architectures and learning algorithms have been proposed. Among them, multilayer feedforward network [43] and Hopfield type feedback network [44] are two common models of neural networks.

The multilayer feedforward network model is often called a MultiLayer Percep-

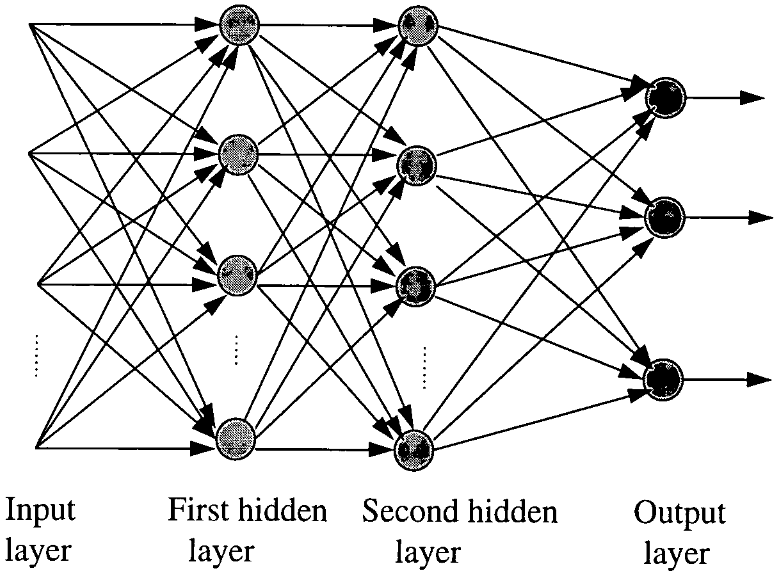

Figure 4.3: Architectural graph of a multilayer perceptron with two hidden layers

tron(MLP). Multilayer perceptrons are the most prevalent neural networks for control applications because they have the ability to learn dynamic system characteristics through nonlinear mappings. Typically, an MLP consists of an input layer of source nodes, one or more hidden layers of computation nodes, and an output layer of computation nodes. The network exhibits a high degree of connectivity in that the neurons between layers are interconnected by variable connections, i.e., weights. Changing these weights will alter the behaviour of the whole network. The model of each neuron in the network includes a nonlinearity at the output end. A commonly used function of smooth nonlinearity is a sigmoid function. Figure 4.3 shows the architecture of a multilayer perceptron with two hidden layers. MLPs are trained in a supervised manner with a highly popular algorithm known as the BackPropagation(BP) algorithm. This algorithm is based on the error-correction learning rule. As such, it may be viewed as a generalization of the famous adaptive filtering algorithm: Least Mean Square(LMS) algorithm. To speed up the convergence of the BP algorithm, one can use an adaptive learning rate algorithm intended to adjust the learning rate automatically as the learning process proceeds. The details of the BP algorithm are given in Appendix A.

In the feedback neural network model, the connection topology and the weights are determined from the problem constraints. One of the most prevalent applications of this type of neural network is to solve constrained optimization problems. Solving an optimization problem requires minimization of a cost function subject to a set of constraints imposed by the problem. This cost function is known as the energy function of the neural network, and it is referred to as the total energy stored in the neural network circuit. By minimizing the energy, the NN converges to a stable state, producing an optimal(or near optimal) solution. The neural optimization approach maps the optimization problem into the form of this energy function that describes the dynamics of a neural system. Due to the massive parallelism and possibly fast convergence to solutions, NNs can be much more effective than conventional algorithms in terms of computation time. This has been proved by many examples, such as the Travelling Salesman Problem(TSP) [44].

So far neural networks have found various practical applications in many areas, such as pattern recognition, signal processing (radar, image, speech, etc.), robotics, and system control and identification. Moreover, neural networks have been implemented on VLSI chips to bring the high speeds and strong processing capabilities into reality. However, there is still a long way to go, in both theory and applications.

4.2 Neural Congestion Control

Congestion control in ATM networks has to satisfy various QOS requirements, while using network resources efficiently. It also must be able to adapt to changes in traffic characteristics, since many new services will be introduced after network design and installation. With high-speed transmission, control algorithms should be effective in terms of taking effect immediately at the onset of congestion, preferably being incorporated into hardware implementation. As we mentioned previously, a variety of congestion control strategies for future ATM networks have been proposed. However, the behaviour of network dynamics in the presence of congestion, and the proper ways of handling traffic to obtain more reliable and predictable performance, are not yet sufficiently understood. The difficulty stems mainly from the uncertainties about traffic patterns and the time-varying nature of network conditions.

Most traditional control methods are based on results obtained from thorough analyses of offered traffic characteristics and service quality. However, it is difficult to analyze all possible situations in ATM networks because of the large variety of services and their combinations. The controller becomes complicated and inflexible with traditional strategies, especially when new services are introduced. Also, a major shortcoming of the currently available queueing models in analytical performance evaluation is that only steady-state results are tractable. Consequently, any control function tailored on the basis of such models can ensure optimal performance only under steady-state conditions. However, performance driven control methods that dynamically regulate traffic flows according to changing network conditions require an understanding of network dynamics. Furthermore, these models often contain simplified assumptions based on mathematical calculations and computer simulations, which can seldom be justified in real life. This has led many researchers to believe that new congestion control algorithms with some form of adaptive and learning capabilities are required to meet such challenges.

Compared to conventional methods, the use of neural networks makes a significant difference in the performance of a system for a real world application. Through the use of neural networks, we can deal with difficult problems in unknown or partially known nonlinear dynamic systems where conventional approaches are proven ineffective, or for which there is no other solution. There is currently a great deal of interest in applications of neural networks to various control problems. This should not be surprising, because, after all, the human brain is a computer, the outputs of which as a whole system are actions. In the context of control, the brain is living proof that it is possible to build a generalized controller that takes full advantage of parallel distributed hardware, that can handle thousands of actuators in parallel, that can handle noise and nonlinearity, and that can optimize over a long-range planning horizon [42]. This is what has been called *neurocontrol.* Narendra and Parthasarathy have shown that it is possible to design an adaptive controller using supervised neural networks so that the overall system is globally asymptotically stable [45]. The potential benefits expected from using neural networks for adaptive control are [46]:

- Adaptivity: Neural networks have a built-in capability to adapt their synaptic weights to the changes in the surrounding environment. Detailed description and mathematical understanding about the underlying network to be controlled are not required as a neural network can learn from observations or examples during the course of network operation. This makes it an ideal tool for adaptive control.
- High computation rate: This is due to the massive parallel structure of the hardware implementation of neural networks. In general, the computation time is independent of neural network dimension and the number of control variables.
- Generalization on learning: A neural network can generalize learning to conditions not specifically involved in the training phase. This is particularly useful for learning in a dynamic environment for congestion control in ATM networks where observations may be incomplete, delayed or partially available.
- Robustness: Owing to the distributed nature of information in the network, a neural network is inherently fault tolerant in the sense that its performance degrades gracefully under adverse operating conditions.

A block diagram of an ATM traffic controller using NNs is shown in Figure 4.4(adapted from [41]). As illustrated in the figure, NNs are applied to the call level control functions such as CAC in order to predict QOS from observed traffic and, hence, make optimal decisions. Neural networks are also applied to the cell level control functions such as traffic measurements, policing and rate-based feedback congestion control at the access to the network. Moreover, NNs can be applied to the network level control functions such as optimal link capacity allocation and dynamic routing.

For applications of NNs in ATM networks, both MLPs and Hopfield type feedback networks have been used. As far as the Hopfield type networks are concerned, they

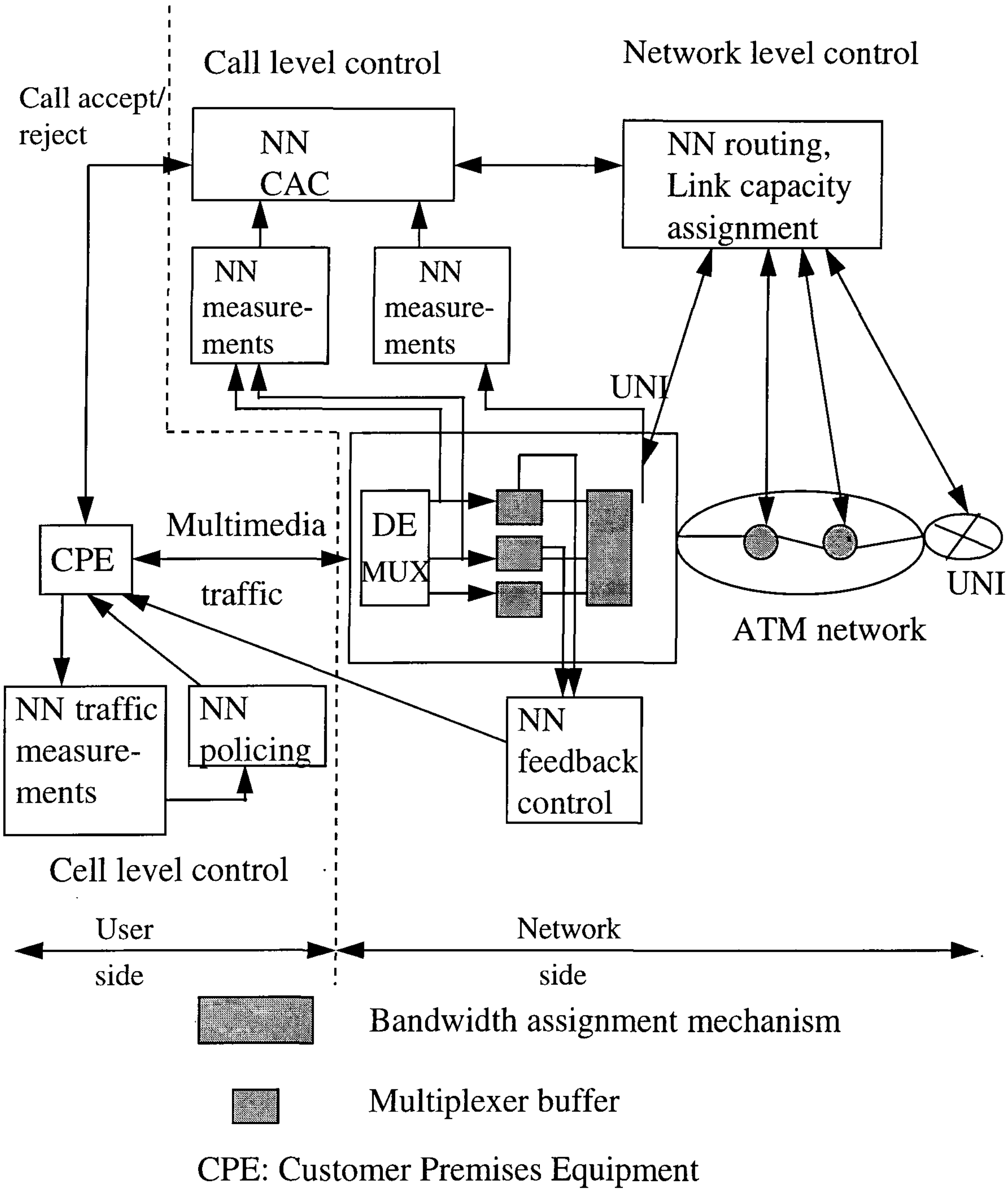

Figure 4.4: Neural networks for ATM traffic control

have been used to solve some combinatorial optimization problems within the networks that are intractable with conventional software control and even custom digital circuitry. They are very popular for shortest path computation and routing [47], switch control [48] [49], optimal packet scheduling [50] and input access control in multicast packet switching [51]. Actually, there are quite a few research groups that have got satisfactory results by using this kind of feedback neural networks to solve some difficult problems in computer networks. But in the following, we will concentrate on the applications of feedforward neural networks to traffic control for ATM networks.

4.2.1 Neural Networks for Adaptive Link Allocation

The link allocation problem arises during routing when a virtual path consists of several physical links. The objective of the link allocation function is to maximize the long-term revenue, while maintaining the call level grade of service. A flexible solution to the ATM link allocation problem must also manage non-Poisson arrivals and general call holding time distributions, which can arise in the BISDN. Two adaptive methods based on NNs and reinforcement learning have been proposed.

In the first method, called BP-HT(BackPropagation on Hypothetical Targets) [52], a single NN is trained on bipolar reward, indicating if the performed link allocation action was a success or failure. For each action, weight changes are computed using the BP algorithm on two hypothetical target vectors, one under the assumption that the action will turn out to be good(optimistic) and one under the opposite assumption(pessimistic). The weight changes are accumulated and discounted over time, and the sign of the reward indicates which one to apply when updating the weights. Preliminary experiments on a small allocation problem show that the proposed method is able to learn this task, reaching a performance comparable to conventional(non-adaptive) methods. Future work on the BP-HT approach includes exploiting the ability to switch between reinforcement and supervised learning, and testing the approach in a non-stationary environment.

The second method [53] adapts the link allocation policy to the offered call traffic such that long-term revenue is maximized. It decomposes the link allocation task into a set of Link Admission Control(LAC) tasks, formulated as Semi-Markov Decision Problems(SMDPs). The LAC policies are directly adapted by reinforcement learning, using the temporal-difference learning scheme. Simulations show that the reinforcement method yields a long-term revenue(throughput) comparable to the model-based dynamic programming method. The advantage of reinforcement learning is that the computational complexity and computer memory requirements can be reduced by using NNs for function approximation. However, in [53], only Poisson call traffic is considered. As the limitations of traditional Poisson model for network arrival processes have been demonstrated in recent studies [54], the performance under non-Poisson traffic should be examined in the future.

4.2.2 Neural Networks for Connection Admission Control

Hiramatsu [55] has applied a three-layer fully connected NN to create a call admission controller in ATM networks. For simplicity, only the cell loss rate is considered as a service quality parameter in [55]. Figure 4.5 shows the block diagram of the call admission controller proposed in [55]. Input signals to the MLP are the observed status of the multiplexer(such as cell arrival rate, cell loss rate, cell generation rate, trunk utilisation rate, number of connected calls). The parameters declared in a call setup request(such as average bit rate and bit rate fluctuation of the call and holding time) are also inputs. A history of the past observed status will be input to the NN in parallel format when the sequence of the data is expected to contain important information. Output signals are the predicted QOS parameters and the decision values for acceptance or rejection of a connection request. The network is trained using backpropagation to learn this mapping and is then used in a multiplexer to carry out blocking of calls for a period of time to achieve call control. Applications to single bit rate, as well as multiple bit rate traffic have been presented. In [55], Hiramatsu uses a so-called "leaky pattern table method" for training data selection. There are two pattern tables, one for low-loss-rate events and the other for high-loss-rate events. He randomly selects an exemplar for training, and randomly replaces an old observation by the latest observation at each backpropagation step. This NN-based CAC is quite similar to the situations where NNs are applied to pattern classification. The boundary between acceptance and rejection is the call admission boundary, which the NN learns from the data observed from the operating network. It has been shown that the neural network can learn call admission boundaries for various link capacities. Moreover, this CAC method does not depend on analytical models of call bit rate variations. Therefore, it can manage many bit rate classes with unknown characteristics and can adapt to changes in the characteristics of each bit rate class. The main problems of ATM CAC using neural networks were the exponentially wide range of QOS values and the real-time training data sampling. In [56], Hiramatsu proposes the concept of training with relative target and virtual output buffer to overcome these problems.

In [49], Morris and Samadi have proposed a similar method of CAC. The approach adopted here is that key network performance parameters(e.g., delay, loss, jitter) are observed while carrying various combinations of calls, and their relationship is learned by a three-layer feedforward neural network. The NN has the ability to interpolate or extrapolate from past experienced results. It also has the ability to adapt to new and changing conditions. Rather than trying to judge the traffic behaviour from the fine structure of its arrival patterns as in [55], the controller in [49] estimates the traffic's entire congestive behaviour (burstiness, peak rate, etc.) from its impact on the output queue via measurements of quantities such as mean delay, loss and jitter. The neural network is trained to adaptively estimate the performance metric of interest as a function

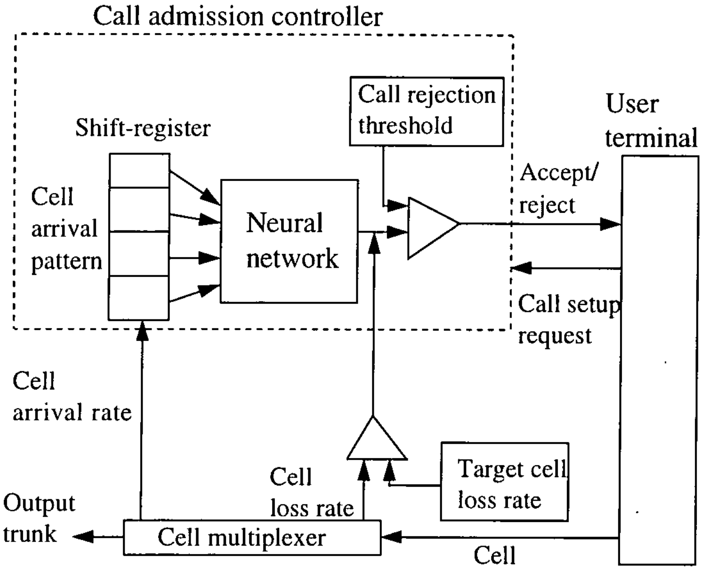

Figure 4.5: Neural networks for call admission control

of the offered traffic. Actually, prediction or parameter estimation is one of the main learning tasks that befit the use of neural networks [42]. In this case the call will be admitted if, and only if, the estimated performance metric is less than or equal to a predefined threshold. It is well known that one of the fundamental shortcomings of backpropagation is that it is prone to getting trapped in some local minima. To overcome this problem, [49] has adopted a systematic initialization of local search(a multistart method). This technique is important because it makes the random multistart search method(with its global convergence properties) practical, and also because a fairly large window of observations is necessary owing to the large statistical variability in queueing behaviour.

A new strategy for CAC has been recently proposed [57], which manages the competitive access of new connections related to different services, and considers QOS objectives established in terms of time and semantic transparency. In this method, neural networks are employed for traffic prediction. The neural network inputs are the allocated bandwidth to each service class, and the outputs can be the expected delay, cell loss, and the maximum and minimum buffer occupation. A "quality of operation" function is defined as a measure of network performance. This function incorporates the allocated bandwidth, the free transmission capacity, the connection rejection rate and some time and semantic transparency variables(cell loss rate, delay and jitter). When a connection requests service, each node control unit asks its neural network about the expected traffic load patterns for the node and adjacent link, with and without the inclusion of the new connection. The NN answers with the expected patterns and then the quality of operation can be evaluated for both cases. Finally, the control entity accepts the call if the expected quality of operation in every BISDN node and link of the call route is higher with the new connection than without it. Simulation results show that with the proposed technique, all the service classes share the available resources more efficiently than other methods [57].

In [58], NN-based CAC is combined with adaptive link capacity control. Neural networks are trained to estimate the call loss rate from observed traffic and link capacities, and link capacity assignment is optimized by a random optimization method according to the estimated call loss rate. Simulation results show that the NN method for call loss rate estimation can have better accuracy than an approach based on a traditional teletraffic method. The integration of adaptive CAC and adaptive link capacity control achieves optimal network throughput and yields an efficient ATM traffic control system suitable for multimedia services with unknown traffic characteristics. The effective distributed implementation of the neural network for link capacity control is an issue for further study, and the various optimization methods need to be evaluated in a large network with multiple bit rate classes.

4.2.3 Neural Networks for Traffic Policing

A number of desirable features of traffic policing can be summarized as follows [59]:

- Capability of detecting any non-compliant traffic situation;
- Ability to determine whether the user's behaviour is within an acceptable region;
- Rapid response time to parameter violations;
- Simplicity of implementation.

Most of the existing policing mechnisms attempt to police the peak and mean bit rates of the traffic. But the peak and mean policing functions check only one parameter of the probability density function(pdf) of the bit rate of the source. This can reduce the effectiveness of the policing algorithm as well as that of the CAC algorithm(as we have mentioned earlier). On the other hand, the policing mechanisms that try to police the pdf face the difficulty of complicated calculations of higher order moments. In [59], a Neural Network Traffic Enforcement Mechanism(NNTEM) is proposed. To police the pdf of the traffic, the NNTEM uses two backpropagation neural networks that implicitly learn the pdf of the traffic count process through many learning trials. One neural network(NN1) captures the actual pdf of the ideal "non-violating" traffic, whereas the other(NN2) is trained to adaptively characterize and predict any type of traffic violations by learning the past and future traffic variations. The error signal between NN1 output and NN2 output can detect the individual contractual parameter violations as well as any combinations of the contractual parameter violations. Hence, the NNTEM does not rely on the policing of simple parameters such as mean bit rate or peak bit rate, but rather uses an elaborate and very accurate function(pdf) which includes all statistical properties of the traffic. Moreover, the reaction time of the NNTEM is small compared with that of window-based mechanisms.

4.2.4 Neural Network Feedback Control

In [60], an explicit congestion notification mechanism for ATM networks using neural networks to estimate the amount by which sources need to reduce their transmission rates is proposed. Three models using NNs have been presented and the obtained results are compared. In the first approach, the current buffer status(queue length) and the Cell Arrival Patterns(CAPs) in the past few cycles are used to predict the possible cell loss rate. Based on this prediction, a feedback cell sends an explicit value to the sources, at which the sources must regulate their transmission rates. In the second method, the CAPs are processed using a Standard Normal Deviate(SND) model before fed into the NN. This enables the NN to have knowledge of the relationships between traffic statistical characteristics in the past cycles and near future. In the third method, the CAPs processed by a Moving Average(MA) model enable the NN to detect traffic inhomogeneities. Simulations show that these novel mechanisms have better cell loss rate improvement than feedback congestion control with static threshold values, while transmission delay introduced by the NN controller is also smaller than the static approach in most cases.

4.3 An ATM Congestion Controller Using Reinforcement Learning

In this section, we present an adaptive congestion control approach based on a neural network that uses reinforcement learning. This is achieved via the formulation of a performance measure function which is used to adaptively tune the weights of the neural network. A control signal is generated to regulate the incoming traffic so as to meet QOS requirements. The results show that the proposed control mechanism is adaptive in the sense that it is applicable to any type of traffic. Also, the control signal is optimal in the sense that it maximizes the performance of the system in terms of its performance measure function. Hence, our approach provides effective control of congestion in ATM networks.

4.3.1 Formulation of a General QOS Control Problem

The general problem of QOS control in ATM networks consists of adaptively regulating access of external traffic into the network to guarantee the desired performance. The schematic diagram representation of the control problem is shown in Figure 4.6, where

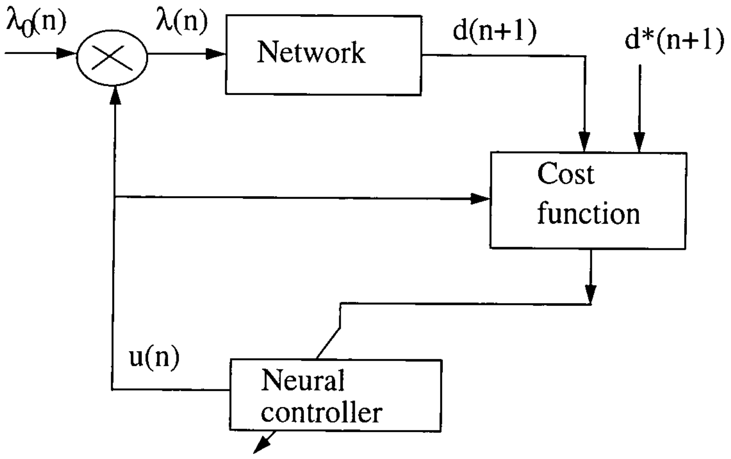

Figure 4.6: Neural congestion controller

the network represents a practical ATM network, such as a switching node, an end-to-end connection, or a simple queue. Let $\lambda_0(n)$ denote the average traffic arrival rate (n is the sampling number) and $d^*(n)$ the required bound of QOS. Let $d(n)$ be the performance observed from the network. $d(n)$ and $d^*(n)$ can be taken as cell loss rate, cell delay, etc. Note that all quantities considered are time-dependent averages to capture the dynamics of both the traffic load and the network condition. Since external traffic arrivals are assumed to be independent of network state, the absence of control on traffic load may cause severe violation of the given performance bound. Therefore the objective of congestion control is to design an adaptive controller at the UNI that will maximize the input traffic to the network while keeping the QOS within the desired bound.

More specifically, the neural controller generates an optimal control signal $u(n)$ which defines the portion of the offered traffic, $\lambda_0(n)$, that can be admitted to the network, i.e.,

$$\lambda(n) = u(n)\lambda_0(n), \qquad 0 < u(n) \leq 1. \tag{4.1}$$

The control signal is optimal in the sense that it not only satisfies the QOS constraint($d(n) \leq d^*(n)$) but also maximizes the network throughput $\lambda(n)$. This control algorithm has several advantages. It can be classified as a preventive type congestion control mechanism since the algorithm is applied at the input access node of the network, and its speed is not limited by the propagation delay. Hence, any control action will be in time to avoid the potential congestion. Also, as we will show later, it is very flexible in establishing various performance objectives (maybe somewhat conflicting requirements) that can be properly incorporated into the learning control process. Although we assume that the network has a single input and single output, the basic control structure and learning algorithms are applicable to multi-input and multi-output networks, where $\lambda(n)$ and $d(n)$ are replaced by variable vectors.

4.3.2 Neural Control Using Reinforcement Learning

It is widely known that the congestion control problem can be treated as an optimization problem. However it is very difficult for the classical control methods to solve this problem because these methods rely on a very accurate mathematical model of the system to be controlled. In general this model does not exist or if it exists it must be adaptive to the time-varying arrival traffic. Such a model would involve very long computation time and would be infeasible for real-time implementation. Here we use a reinforcement learning neural network to overcome the above-mentioned limitations of the classical optimal control methods in this application.

A reinforcement learning system addresses the problem of improving performance and therefore learning on the basis of any measure whose values can be supplied to the system. We may therefore view a reinforcement learning system as an *evaluative* feedback system. In contrast, the performance measure used for a supervised learning system is defined in terms of a set of targets(i.e., desired responses) by means of a known error criterion (e.g., mean square error). A supervised learning system may therefore be viewed as an *instructive* feedback system [42]. Reinforcement learning is more general than supervised learning in that instead of trying to determine target control signals from target environment response, one tries to determine target control signals, or desired changes in the control signals, that would lead to increases in a measure of the environment performance.

The reinforcement learning method evaluates the performance of the system in terms of a defined performance index(cost function) and generates an evaluation signal. This signal is used to adjust the weights of a neural controller in such a manner that the produced control signal results in maximization of the system performance. Hence the reinforcement learning method depends mainly on the defined cost function of the system and always tends to minimize it. Moreover, detailed knowledge of the system under study is not required in this method.

The cost function J is defined in terms of two main objectives: 1) to keep the network performance within the required bound; 2) to maximize the actual input traffic. Thus, one possibility is:

$$J(P) = \sum_{n=1}^{\rho} \alpha S(n+1)[d(n+1) - d^*(n+1)]^2 + \beta[u(n) - 1]^2 \qquad (4.2)$$

where ρ is the sampling number within one trial, P is the trial number, α and β are weight values of the contributions to the cost function made by the QOS value and the arrival rate respectively. $S(n+1)$ is 1 if $d(n+1) \geq d^*(n+1)$ and 0 otherwise. Obviously, the cost function represents the deviation of the system performance from the desired optimal one and is used to change the weights of the neural controller. The NN of the controller is a feedforward network which has three layers and both the hidden and the output layers have a sigmoid function f to perform the nonlinear mapping. The function f is given by

$$f(x) = \frac{1}{1+\exp(-2x)}. \tag{4.3}$$

The hidden layer has four neurons and the output layer has one neuron. The neural network output is $u(n)$, the control signal to alter the input rate, and it is expressed by the following equation using the same notation as given in a previous paper [61]:

$$u(n) = f[w^T(P)f(W(P)I(n))] \tag{4.4}$$

where w is the weight vector from the hidden layer to the output layer, and W is the weight matrix from the input layer to the hidden layer. In our simulations, initial weights are chosen randomly from a uniform distribution. I is the input vector(all vectors here being column vectors) to the neural network and is given by

$$I^T(n) = [d(n), d(n-1), u(n-1), u(n-2)]. \tag{4.5}$$

Both the weight vector w, and the weight matrix W, are tuned using the steepest descent method so as to minimize the cost function J defined in (4.2):

$$w(P+1) = w(P) - \zeta\frac{\partial J(P)}{\partial w(P)} \tag{4.6}$$

$$W(P+1) = W(P) - \zeta\frac{\partial J(P)}{\partial W(P)} \tag{4.7}$$

where ζ is the learning rate. The work reported in [61] has assured the convergence of the cost function using the weight tuning algorithm given in (4.6) and (4.7).

4.3.3 Examples and Simulation Results

In this section we present some numerical examples to test the performance of the suggested control scheme. Three queueing models for the network are taken from [46] [62] and their dynamics are known so that the performance of the neural controller can be easily evaluated. The models are simplified on the basis of first-order difference approximation, so we can focus on the control mechanism itself. To improve the accuracy of the queueing models, high-order difference equations may be used.

Example 1: In the first example, we consider an M/M/1 dynamic queueing model as a real network. With first-order approximation, the model is described by the difference equation $d(n+1) = g(d(n)) + \lambda(n)$, where $d(n)$ and $\lambda(n)$ represent the time-dependent average delay and arrival rate, respectively.[1] The function $g(n)$ is unknown

[1] Note that time units are normalized in such a way that the service capacity is equal to unity.

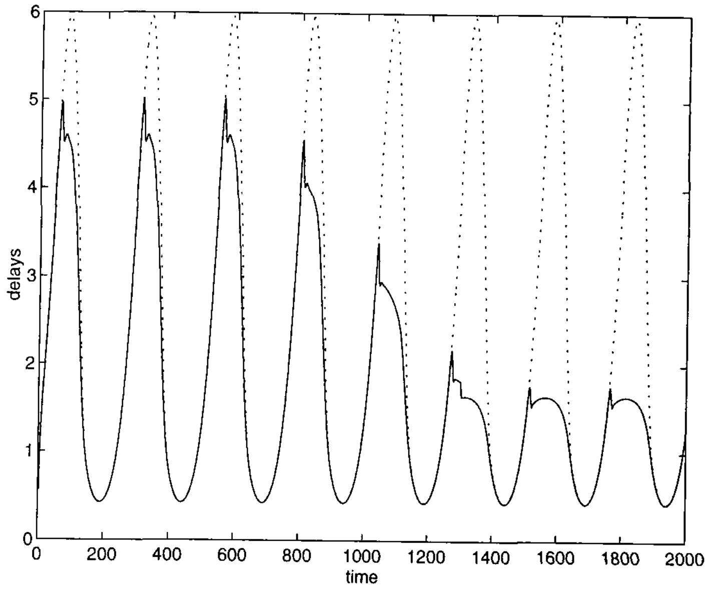

Figure 4.7: Controlled (solid line) and uncontrolled (dotted line) delays for example 1

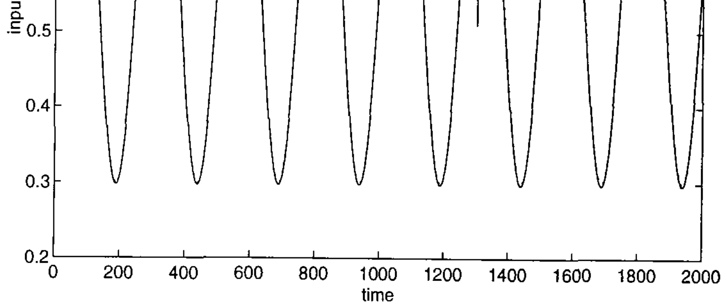

Figure 4.8: Controlled (solid line) and uncontrolled (dotted line) input rates for example 1

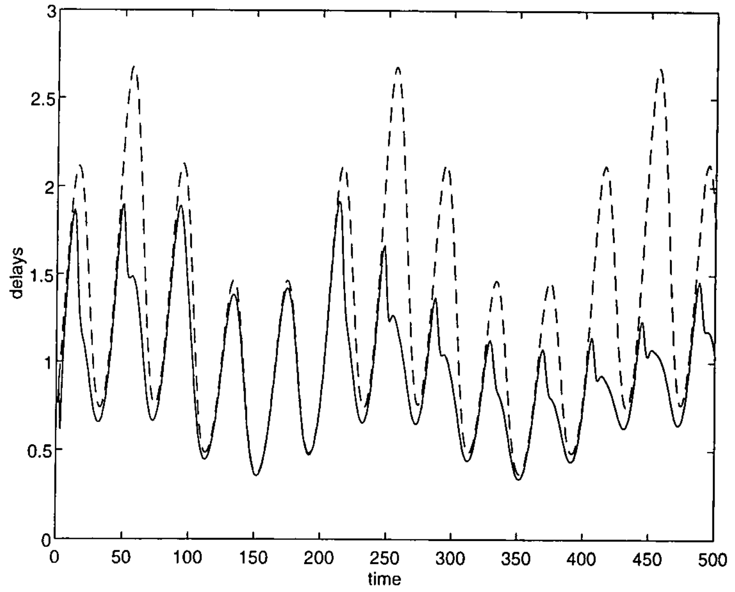

Figure 4.9: Controlled (solid line) and uncontrolled (dashed line) delays for example 2

a priori and has the form $g(d(n)) = d^2(n)/(1 + d(n))$. The objective of control is to regulate the arrival rate $\lambda(n)$ subject to the specified delay bound $d^*(n)$. The external input rate is given by $\lambda_0(n) = 0.6 + 0.3\sin(\pi n/125)$, and the desired delay bound is specified by

$$d^*(n) = \begin{cases} 4.5, & \text{if } 0 < n < 700 \\ 8 - n/200, & \text{if } 700 < n < 1300 \\ 1.5, & \text{if } n > 1300. \end{cases}$$

The control parameters are: $\alpha = \beta = 0.5, \zeta = 0.2$. In Figure 4.7, the delay of the network with control is compared with that without control. It is clear that the delay performance has been successfully controlled and kept below the given delay bound. Figure 4.8 shows the controlled and uncontrolled input rates to the network.

Example 2: Here we take an M/D/1 dynamic queueing model as the real network. This model is governed by the difference equation $d(n+1) = \sqrt{d^2(n) + 1} - 1 + \lambda(n)$, where $d(n)$ and $\lambda(n)$ represent the time-dependent average delay and arrival rate, respectively. The external input rate is $\lambda_0(n) = 0.6 + 0.2\sin(\pi n/20) + 0.1\sin(\pi n/100)$ and the desired delay bound is $d^*(n) = 1.5 + 0.5\sin(\pi n/250)$. The control parameters are given by: $\alpha = 0.7, \beta = 0.5, \zeta = 0.1$. The responses of the network with and without control are shown in Figure 4.9, which illustrates the effectiveness of the neural control method.

Example 3: In this example, the real network considered is composed of an M/M/k dynamic queueing model without a quque. This system is often used as a loss system in performance evaluation. The system is described by the following difference

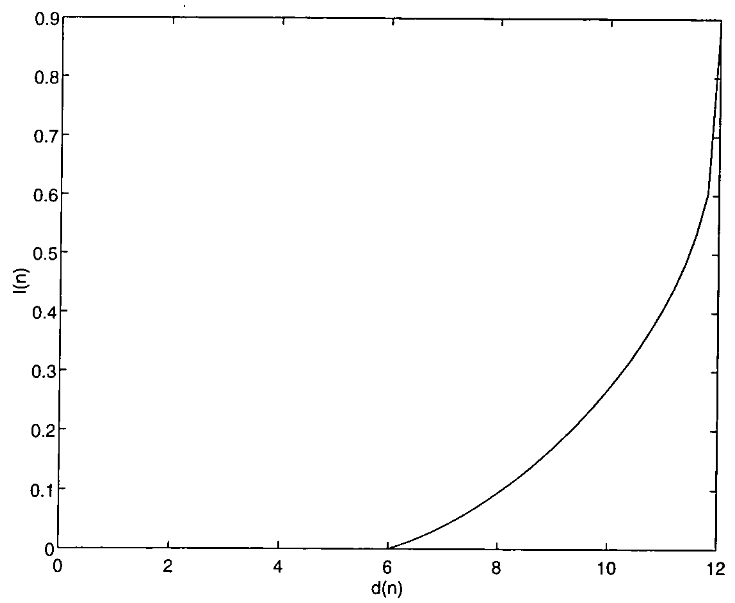

Figure 4.10: The relationship between $l(n)$ and $d(n)$ in example 3

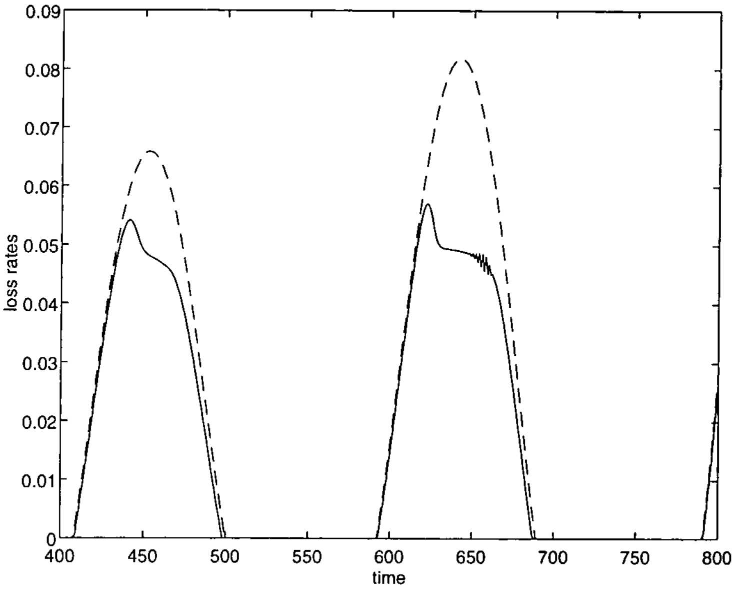

Figure 4.11: Controlled (solid line) and uncontrolled (dashed line) loss rates for example 3

equation:

$$d(n+1) = d(n) - G(d(n)) + \lambda(n) \tag{4.8}$$

together with

$$l(n) = 1 - \frac{d(n)}{G(d(n))} \tag{4.9}$$

where $d(n)$ and $l(n)$ represent the average number of packets and loss rate in the system, respectively. The function G is given by

$$G(x) = \begin{cases} x, & \text{if } x < k/2 \\ k/2 - 7\log(2 - 2x/k), & \text{if } k/2 \leq x < k. \end{cases}$$

In this case the performance we are concerned with is the time-dependent average loss rate $l(n)$. The parameter k is chosen to be 12. Figure 4.10 shows the relationship between $l(n)$ and $d(n)$. We assume the external input rate is given by $\lambda_0(n) = 6 + 3\sin(\pi n/100) + \sin(\pi n/125)$ and the desired loss rate bound $l^*(n) = 0.05$. If we want to use (4.8) as the system equation, we have to convert $l^*(n)$ into $d^*(n)$. It is easy to see that the desired average number of packets $d^*(n)$ is 7.241. The control parameters are chosen as: $\alpha = 0.1, \beta = 0.5, \zeta = 0.1$. The outputs of the network with and without control are plotted in Figure 4.11. Again, as expected, the loss rate has been satisfactorily controlled.

4.3.4 Discussion

In ATM traffic control, conventional mathematical calculations and computer simulations do not work effectively in the controller design process because of the diversity of traffic characteristics of the users and services. As an alternative, we present an adaptive congestion control scheme based on neural networks. The neural network employs reinforcement learning to tune its weights so as to produce an optimal control signal. Simulation results show that the proposed method is adaptive to the changing network environment and optimal control is achieved by minimizing a cost function which contains two important performance measures. Although we have confined our attention to the QOS control in which the input traffic to the network is required to be controlled, we believe that the proposed scheme is general in that it can apply to many other applications of traffic control in ATM networks that fit within this control framework.

4.4 Summary

Neural networks provide an attractive alternative to traditional strategies in dealing with congestion control in ATM networks, namely adaptivity to changing environment, hardware implementation(high speed), a high degree of robustness and capability of handling different performance objectives. This approach makes no assumption about the detailed knowledge of the underlying network, nor about the nature of the traffic sources. It relies only on learning and observations of performance to adapt to changes in traffic loads and network conditions. Most of the ATM traffic management problems can be formulated in the form of a nonlinear function that relates many variables to some outputs. The problem is that, in most situations, this function is too complicated to formulate or solve in real-time using conventional algorithmic approaches such as queueing or simulation techniques. For example, consider the problem of regulating the flow of traffic in an ATM node such that the QOS is maintained. It can be described as a function of the traffic patterns, link utilisation, QOS, etc. Obviously, this function is time-variant, and it is difficult to design an adaptive control system for such a function. However, an NN system can approximate this function with great accuracy, since it only requires examples of the input-output relationship.

The work that has been done on neural congestion control is still tentative. It is expected that NNs could be applied to global network management and to the integration of multiple levels of control, although large neural networks presents some engineering difficulties, such as determining the number of neurons in each hidden layer as well as the number of hidden layers in an MLP, and ensuring the quality of near optimal solutions in feedback neural networks. Nevertheless, significant progress has been achieved and more and more encouraging results are being obtained. This chapter only covers some aspects of applications of neural networks to adaptive link allocation, CAC, traffic enforcement and feedback congestion control. It should be noted that other AI techniques have also been applied to this area, such as fuzzy set theory [63] [64] and genetic algorithms [65]. Actually, with the ultimate goal of achieving *intelligent control*, the combination of neural network and fuzzy logic seems to be a promising solution, since these two techniques can work in a complementary manner [42].

In this chapter, we present an NN-based approach to general QOS control, in which only some simple network models are considered. In the next chapter, we consider more realistic ATM traffic scenarios and propose a novel ATM traffic prediction method using NNs, with applications to adaptive access flow control.

Chapter 5

ATM Traffic Prediction Using FIR Neural Networks with Applications to Access Flow Control

5.1 Introduction

As we mentioned in previous chapters, ATM networks are expected to support a diverse set of applications, such as data, voice and video, each having different traffic characteristics. Accurate characterization of the multimedia traffic is essential in order to develop a robust set of traffic descriptors. Such a set is needed by various traffic management algorithms to guarantee QOS requirements and provide efficient utilisation of network resources. However, for the time being, there are no comprehensive measurements that permit designers to satisfactorily address the characteristics of various communication services in a realistically accurate manner. This is especially true for VBR traffic.

During the duration of a connection, the period at which a source generates traffic is referred to as an *active* period, whereas a *silent* period corresponds to the time between the active periods during which no traffic is generated. Traffic generated by a VBR source either alternates between the active and silent periods, or is a continuous bit stream with varying rates. This traffic is highly bursty and correlated(in comparison to a Poisson process). Burstiness can be defined by the ratio of the peak bit rate to average bit rate or the squared coefficient of variation of the interarrival times of cells, c_I^2(variance divided by the square of the mean). For example, c_I^2 for the packet arrival process from a single voice source is 18.1, while c_I^2 for a Poisson process is 1 [66] [67]. Although the aggregate packet arrival process with many components does behave like a Poisson process over relatively short time intervals, under heavy loads the congestion in the multiplexer is determined by the behaviour of the arrival over much longer time intervals, where it does not behave like a Poisson process. Accordingly, characterization of traffic from VBR sources is very difficult.

Congestion control schemes(e.g., CAC and UPC) in ATM networks require specific knowledge of the statistical behaviour of the input traffic declared via its traffic descriptors. Parameters such as peak bit rate, average bit rate, and burst length are often used as a simple set of parameters characterizing the traffic. More complicated second-order time domain parameters(e.g., Index of Dispersion for Intervals(IDI) and Index of Dispersion for Counts(IDC)) are also used to capture the burstiness and correlation properties of the arrival stochastic process especially those of VBR video and voice sources [68]. In [67], the aggregate arrival process from N voice sources is approximated by a non-renewal process, i.e., a two-state Markov Modulated Poisson Process(MMPP). In [69], very complex mathematical models such as semi-Markov process and continuous-time Markov chain are used to characterize the voice traffic. Traffic descriptors using simple parameters will not accurately characterize very rapid changes in the bit rate time variations of the traffic over short intervals and often ignore the bursty nature of the traffic. On the other hand, those mechanisms using more sophisticated parameters are computationally expensive and impractical.

To solve this problem, in this chapter, we present a novel neural network approach to adaptively characterize and predict the traffic arrival process. The Finite Impulse Response(FIR) multilayer perceptron model and its training algorithm are discussed. It is shown that the FIR neural network can adaptively predict the complex stochastic process by learning the relationship between the past and future traffic variations and hence has an excellent potential for use in some congestion control schemes. On the basis of this prediction, an access flow control approach at the UNI is then proposed. This control mechanism operates on the principle of feedback control. The prediction of traffic arrival patterns in conjunction with the current queue information of the buffer can be used as a measure of congestion. When the congestion level is reached, a control signal is generated to throttle the input arrival rate. Simulation results suggest that the scheme is able to significantly reduce cell loss rate and provides a simple and efficient traffic management for ATM networks.

5.2 ATM Traffic Prediction Using FIR Neural Networks

5.2.1 FIR Neural Network

It has been proved that neural networks are capable of performing nonlinear mappings between real-valued inputs and outputs. A three-layer feedforward neural network(MLP), with sigmoidal units in the hidden layer, is able to approximate an arbitrary

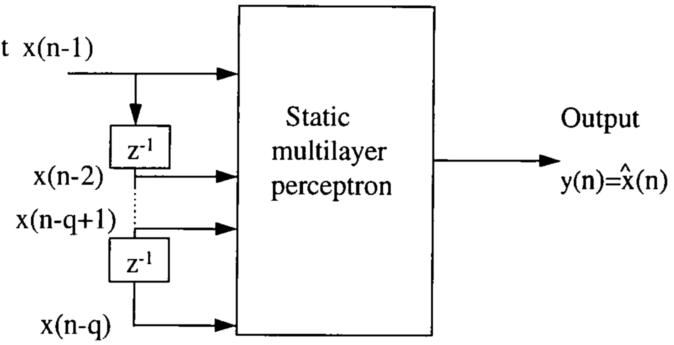

Figure 5.1: Static multilayer perceptron used as a nonlinear predictor

nonlinear function to any desired degree of accuracy [70] [71]. This kind of NN is trained with the backpropagation algorithm. One limitation of the standard BP algorithm is that it can only learn an input-output mapping that is *static*. This form of static input-output mapping is well suited for pattern recognition applications, where both the input and output vectors represent *spatial* patterns that are independent of time [42].

The standard BP algorithm may also be used to perform nonlinear prediction on a stationary time series [72]. We may use a static multilayer perceptron, as depicted in Figure 5.1, where the input elements labeled z^{-1} represent unit delays. The input vector $\mathbf{x}$ is defined in terms of the past samples $x(n-1), x(n-2), ..., x(n-q)$ as follows:

$$\mathbf{x} = [x(n-1), x(n-2), ..., x(n-q)]^T \tag{5.1}$$

where q is the prediction order. Thus the scalar output $y(n)$ of the multilayer perceptron equals the one-step prediction $\hat{x}(n)$, as shown by

$$y(n) = \hat{x}(n) \tag{5.2}$$

The actual value $x(n)$ of the input signal represents the desired response.

However, if we want to capture the *dynamic* properties of time-varying signals, we have to extend the design of a multilayer perceptron so as to represent *time*. One of the methods is the so-called Time Delay Neural Network(TDNN), which was first used in [73] to perform speech recognition. The TDNN is a multilayer feedforward network in which the outputs of a layer are buffered several time steps and then fed fully connected to the next layer. It was devised to capture explicitly the concept of time symmetry as encountered in the recognition of an isolated phoneme using a spectrogram.

The TDNN topology is in fact embodied in a multilayer perceptron in which each synapse is represented by a finite impulse response filter. This latter neural network is referred to as an FIR multilayer perceptron, which can be trained with an efficient algorithm called *temporal backpropagation* [74]. It can be shown that the TDNN and the FIR network are functionally equivalent. However, the FIR network is more easily related to a standard multilayer network as a simple temporal or vector extension. The FIR representation also leads to a more desirable adaptation scheme. So we adopt this kind of FIR network as our traffic predictor.

FIR Network Model

As mentioned above, the traditional model of a multilayer perceptron forms a static mapping; there are no internal dynamics. To extend the usefulness of this model for temporal processing, we need to modify it so as to account for the temporal nature of the input data. A modification of the basic neuron in an MLP is accomplished by replacing each synaptic weight by an FIR linear filter [74]. By FIR we mean that for an input excitation of finite duration, the output of the filter will also be of finite duration. For this filter, the output $y(k)$ equals a weighted sum of past delayed values of the input $x(n)$:

$$y(k) = \sum_{n=0}^{M} w(n)x(k-n) \qquad (5.3)$$

On the basis of (5.3), we may formulate the model of an FIR neuron as follows. Let $w_{ji}(l)$ denote the weight connected to the lth tap of the FIR filter modeling the synapse that connects the output of neuron i to neuron $j(i = 1, 2, ..., p)$. The index l ranges from 0 to M, where M is the total number of delay units built into the design of the FIR filter. Let $y_j(n)$ denote the output signal of neuron j and $x_i(n)$ the input signal. Hence, we have

$$v_j(n) = \sum_{i=1}^{p} s_{ji}(n) - \theta_j = \sum_{i=1}^{p}\sum_{l=0}^{M} w_{ji}(l)x_i(n-l) - \theta_j \qquad (5.4)$$

$$y_j(n) = \varphi(v_j(n)) \qquad (5.5)$$

where $v_j(n)$ is the net activation potential of neuron j, θ_j is the externally applied threshold and $\varphi(\cdot)$ is the nonlinear activation function of the neuron.

We may rewrite (5.4) and (5.5) in matrix form by introducing the following definitions for the state vector and weight vector for synapse i, respectively:

$$\mathbf{x}_i(n) = [x_i(n), x_i(n-1), ..., x_i(n-M)]^T \qquad (5.6)$$

$$\mathbf{w}_{ji} = [w_{ji}(0), w_{ji}(1), ..., w_{ji}(M)]^T \qquad (5.7)$$

We may thus express the output $y_j(n)$ of neuron j by the following equation:

$$y_j(n) = \varphi(\sum_{i=1}^{p} \mathbf{w}_{ji}^T \mathbf{x}_i(n) - \theta_j) \qquad (5.8)$$

This FIR model of a single artificial neuron is shown in Figure 5.2, where the weight w_{j0}

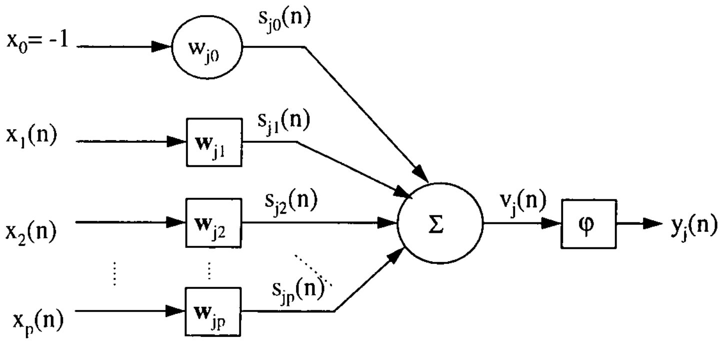

Figure 5.2: Dynamic model of a neuron, incorporating synaptic FIR filters

connected to the fixed input $x_0 = -1$ represents the threshold θ_j. The signal-flow graph representation of an FIR filter is shown in Figure 5.3.

We may construct a multilayer perceptron whose hidden and output neurons are all based on the above FIR model. Such a neural network structure can be referred to as an FIR multilayer perceptron. The difference between the FIR multilayer perceptron and the standard one is that the static forms of the synaptic connections between the neurons in the various layers of the network are replaced by their dynamic versions (i.e., scalars are replaced by vectors and multiplications by vector products).

Temporal Backpropagation Learning

Assume that neuron j lies in the output layer with its actual response denoted by $y_j(n)$ and that the desired response for this neuron is denoted by $d_j(n)$, both of which are measured at time n. Define an instantaneous value for the sum of squared errors produced by the network as follows:

$$E(n) = \frac{1}{2} \sum_j e_j^2(n) \qquad (5.9)$$

where the index j refers to the neurons in the output layer only, and $e_j(n)$ is the error signal, i.e.,

$$e_j(n) = d_j(n) - y_j(n) \qquad (5.10)$$

Therefore the objective of training corresponds to minimizing the cost function:

$$C = \sum_n E(n) \qquad (5.11)$$

where the sum is taken over all time.

In [74], an algorithm called temporal backpropagation is proposed to minimize C.

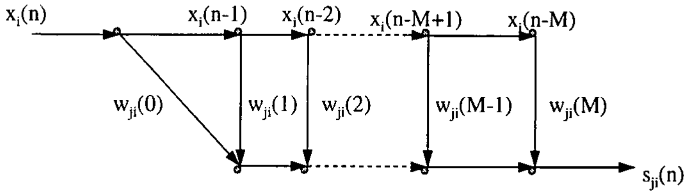

Figure 5.3: Signal-flow graph of a synaptic FIR filter

The weight-update operation is shown by the following pair of equations:

$$\mathbf{w}_{ji}(k+1) = \mathbf{w}_{ji}(k) - \eta \frac{\partial C}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial \mathbf{w}_{ji}(k)} = \mathbf{w}_{ji}(k) + \eta \delta_j(k) \mathbf{x}_i(k) \qquad (5.12)$$

$$\delta_j(k) = \begin{cases} e_j(k)\varphi'(v_j(k)), & \text{neuron } j \text{ in the output layer} \\ \varphi'(v_j(k)) \sum_{m \epsilon \mathcal{A}} \mathbf{\Delta}_m^T(k) \mathbf{w}_{mj}, & \text{neuron } j \text{ in a hidden layer} \end{cases} \qquad (5.13)$$

where η is the learning rate parameter, $\mathcal{A}$ is defined as the set of all neurons whose inputs are fed by neuron j in a forward manner and $\mathbf{\Delta}_m(k)$ is defined as follows:

$$\mathbf{\Delta}_m(k) = [\delta_m(k), \delta_m(k+1), ..., \delta_m(k+M)]^T \qquad (5.14)$$

It is obvious that the above equations represent a *vector generalization* of the standard backpropagation algorithm. In fact, if we replace the input vector $\mathbf{x}_i(n)$, the weight vector $\mathbf{w}_{mj}$, and the local gradient vector $\mathbf{\Delta}_m$ by their scalar counterparts, the temporal backpropagation algorithm reduces to the standard backpropagation for static networks. To calculate $\delta_j(k)$ for a neuron j located in a hidden layer, we filter the δ's from the next layer backwards through the FIR synapses for which the given neuron feeds(see Figure 5.4). Thus δ's are formed not by simply taking weighted sums, but by backward filtering. For each new set of input and desired response vectors, the forward filters are incremented one time step and the backward filters one time step. The weights are then adapted on-line at each time increment.

Temporal backpropagation preserves the symmetry between the forward propagation of states and the backward propagation of error terms. The sense of parallel distributed processing is thereby maintained. Furthermore, each unique weight of synaptic filter is used only once in the computation of the δ's; there is no redundant use of terms experienced in the instantaneous gradient model.

However, careful inspection of the above equations reveals that the calculations for the $\delta_j(k)$'s are noncausal. We may formulate the causal form of the temporal backpropagation algorithm by a simple reindexing [74]:

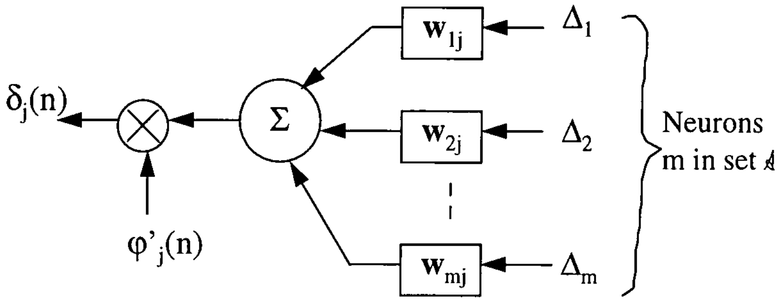

Figure 5.4: Backpropagation of local gradients through an FIR multilayer perceptron

- For neuron j in the output layer, compute

$$\mathbf{w}_{ji}(k+1) = \mathbf{w}_{ji}(k) + \eta\delta_j(k)\mathbf{x}_i(k) \tag{5.15}$$

$$\delta_j(k) = e_j(k)\varphi'_j(k) \tag{5.16}$$

- For neuron j in a hidden layer, compute

$$\mathbf{w}_{ji}(k+1) = \mathbf{w}_{ji}(k) + \eta\delta_j(k-lM)\mathbf{x}_i(k-lM) \tag{5.17}$$

$$\delta_j(k-lM) = \varphi'(v_j(k-lM)) \sum_{m \epsilon A} \Delta_m^T(k-lM)\mathbf{w}_{mj} \tag{5.18}$$

where M is the total synaptic filter length, and the index l identifies the hidden layer in question. Specifically, $l = 1$ corresponds to one layer back from the output layer; $l = 2$, two layers back from the output layer; and so on.

5.2.2 Traffic Prediction

Neural networks have adaptation capability that can accommodate nonstationarity. Their generalization capability makes them flexible and robust when facing new and noisy data patterns. Once the training is completed, a neural network can be computationally inexpensive even if it continues to adapt on-line. Here we use the FIR neural network as a multimedia traffic predictor in ATM networks. The role of the neural network is to capture the unknown complex relationship between the past and future values of the traffic.

The training scheme for the FIR network is illustrated in Figure 5.5. Consider a scalar time series denoted by $x(n)$, which is described by a nonlinear regressive model

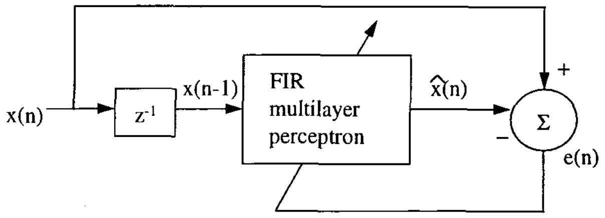

Figure 5.5: Training scheme of the FIR network

of order q as follows [42]:

$$x(n) = f(x(n-1), x(n-2), ..., x(n-q)) + \varepsilon(n) \tag{5.19}$$

where f is a nonlinear function of its arguments and $\varepsilon(n)$ is a residual. It is assumed that $\varepsilon(n)$ is drawn from a white Gaussian noise process. The nonlinear function f is unknown, and the only thing that we have available to us is a set of observables: $x(1), x(2), ..., x(N)$, where N is the total length of the time series. We may use an FIR multilayer perceptron to make a prediction of the sample $x(n)$, given the past q samples $x(n-1), x(n-2), ..., x(n-q)$, as shown by

$$\hat{x}(n) = F(x(n-1), x(n-2), ..., x(n-q)) + e(n) \tag{5.20}$$

where the nonlinear function F is the approximation of the unknown function f, which is computed by the FIR multilayer perceptron. The actual sample value $x(n)$ acts as the desired response. Hence the FIR multilayer perceptron is trained so as to minimize the squared value of the prediction error:

$$e(n) = x(n) - \hat{x}(n), \qquad q+1 \leq n \leq N \tag{5.21}$$

In the neural network literature the above training scheme is referred to as *teacher forcing*, while in the control and signal processing literature, it is referred to as *equation-error adaptation* [42].

In our application, the three-layer FIR MLP has one input neuron, five hidden neurons and one output neuron(denoted by 1-5-1) with 3-tap synaptic filters at both hidden layer and output layer(denoted by 3:3). Selection of these dimensions is based mostly on trial and error. In general, selection of dimensions for neural networks remains an open question in need of further research. The FIR network is trained with the causal form of temporal backpropagation and the Mean Squared Error(MSE) is used as a performance measure. To increase the rate of learning and yet avoid the danger of instability, a momentum term is added to the weight-update equation, i.e.,

$$\mathbf{w}_{ji}(k+1) = \mathbf{w}_{ji}(k) + \zeta[\mathbf{w}_{ji}(k) - \mathbf{w}_{ji}(k-1)] + \eta\delta_j(k)\mathbf{x}_i(k) \tag{5.22}$$

where ζ is a positive number called the momentum constant. The learning rate η and momentum constant ζ are set at 0.1 initially. It has been found that the BP learning algorithm may learn faster when the sigmoidal activation function built into the neuron model of the network is asymmetric than when it is nonsymmetric. So we adopt the hyperbolic tangent activation function in thé hidden layer, which is defined by

$$\varphi(v) = c\tanh(dv)$$

where $c = 1.716$ and $d = 2/3$. In some of our experiments, we have also used some heuristics to accelerate the convergence of backpropagation learning through learning rate adaptation [42]. Since we use the logistic function $\varphi(v) = 1/(1 + \exp(-v))$ for the output neuron, we have to normalize the traffic data so that all the values fall between 0 and 1. The data set used for training should be a uniform representation of the different traffic patterns or situations. Examples of these patterns include ones with sudden changes in the bit rate process, from low values to very high ones occurring over very short periods, and slow time-varying ones.

5.2.3 Traffic Models

In this section, we briefly describe the models for video arrival process and voice arrival process used in our experiments.

Video Arrival Process Model

Video is presented to users as a series of frames in which the motion of the scene is reflected in small changes in sequentially displayed frames. Video frames are generated at a constant rate defined by the playout rate. As the amount of data transmitted per frame varies due to intraframe and interframe coding, video applications generate traffic in a continuous manner at varying rates. Video is a relatively new service in communication networks and its traffic characteristics are not well understood. It is also quite different from voice or data in that its bit streams exhibit various types of correlations between consecutive frames.

The characteristics of the video signal depend primarily on two factors: 1) the nature of the video scene, and 2) the type of VBR coding technique employed(e.g., motion-compensated discrete cosine transform, interframe DPCM, etc.). For the purpose of simplicity, we focus on video services with uniform activity level scenes, i.e., the change in the information content of consecutive frames is not significant [3]. A typical application of this type is video telephone where the screen shows a person talking. In general, correlations in video services with uniform activity levels last for a short duration and decay exponentially with respect to the time. The simulation model used to generate

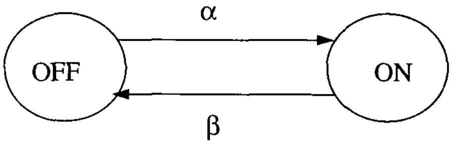

Figure 5.6: IPP model

this kind of video coded traffic is a continuous-state discrete-time stochastic process. A first-order autoregressive(AR) Markov model is proposed in [75], which estimates the bit rate at the nth frame from the bit rate at the $(n-1)$st frame to be

$$\lambda(n) = a\lambda(n-1) + bw(n) \tag{5.23}$$

where $\lambda(n)$ denotes the bit rate of the nth frame in bits/pixel, a and b are constants and $w(n)$ is a Gaussian random variable with mean m and variance 1. There are about 250000 pixels per frame and 30 frames/s, thus 1 bit/pixel corresponds to 7.5 Mbits/s. The parameters a, b and m are given by:

$$a = 0.8781, \qquad b = 0.1108, \qquad m = 0.572 \tag{5.24}$$

The model is found to be quite accurate compared with the actual measurements and is suitable for simulation studies.

Voice Arrival Process Model

A voice source alternates between talkspurts(active) and silent periods. To achieve higher resource utilisation, a speech activity detection may be used at the VBR voice source so that voice packets are generated only when the source is active, thereby, increasing the transmission efficiency. The correlated generation of voice packets within a call can be modeled by an Interrupted Poisson Process(IPP) [76]. In an IPP model, each voice source is characterized by ON (corresponding to talkspurt) and OFF (corresponding to silence duration) periods, which appear in turn. During the ON period, the interarrival times of packets are exponentially distributed(i.e., in a Poisson manner), while no packets are generated during the OFF period. The transition from ON to OFF occurs with the rate β, and the transition from OFF to ON occurs with the rate α(see Figure 5.6). Hence the ON and OFF periods are exponentially distributed with means $1/\beta$ and $1/\alpha$, respectively. To specify this model completely, we assume that the packet generation rate during the active period is 32 kbps, the mean talkspurt is $1/\beta = 352$ ms and the mean silence period is $1/\alpha = 650$ ms.

FIR network	MSE for the training set	MSE for the test set
1-5-1(3:3)	0.00414	0.00423
1-5-1(5:5)	0.00390	0.00402
1-10-1(3:3)	0.00410	0.00415
1-10-1(5:5)	0.00391	0.00390

Table 5.1: MSE of the experiments for video traffic

5.2.4 Numerical Results

In this section, we demonstrate the effectiveness of the neural network used as a traffic predictor. Extensive simulations have been performed. The packet arrival process is generated from packetized video sources or/and packetized voice sources according to the models discussed in the previous section. In the initial stage of our experimental study, we also used a conventional MLP-based TDNN (like that proposed in [59]) for the prediction. In contrast with FIR network, TDNN suffers from much longer training period with no significant performance advantage. For the training and testing of the FIR network, we used different data sets by choosing different initial values of the arrival process or different seeds of the random number generator. We have also tried more complicated network models such as a three-layer network with 1-10-1 nodes or/and 5:5 taps per layer for the same data sets, but no significant performance improvement was observed. It can be interpreted in that, in statistical estimation, increasing complexity of the model over some optimal point may degrade performance due to the *bias/variance dilemma* [77]. The values of MSE of the above experiments are summarized in Table 5.1.

Experiment 1: In this experiment, we use three video sources. The FIR network is used to predict the bit rate of the superposition video arrival process over the next frame. Therefore the lag time is set to 1/30 sec, which is the frame generation rate. This choice is due to the fact that the temporal correlations among successive frames are more dominant than those within a single frame. Figure 5.7 shows the neural network prediction compared with the actual traffic(generated by simulation). The autocorrelation functions of the above two processes are shown in Figure 5.8, illustrating that the predicted traffic has almost the same statistical characteristics as those of the actual traffic.

Experiment 2: In this experiment, we use three voice sources. Here the time series $x(n)$ is used to represent the count process $N(0,t)$ which measures the number of packet arrivals in time $(0,t)$. The arrival process is sampled at every sampling period T_s. The choice of the parameter T_s is influenced by the type of the traffic and should guarantee that the used sampled version of the arrival process captures all correlations contained in the actual process. In [59], in order to select the best sampling interval T_s, a number

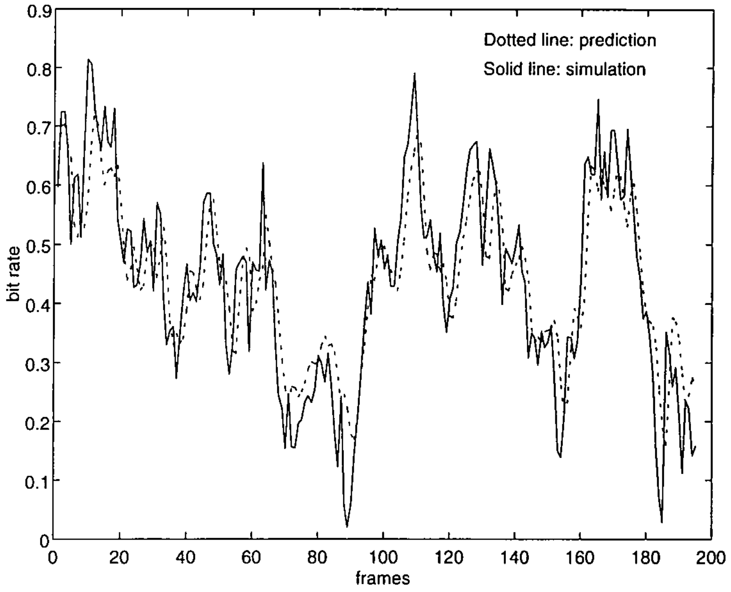

Figure 5.7: Prediction results for the bit rate of the video traffic

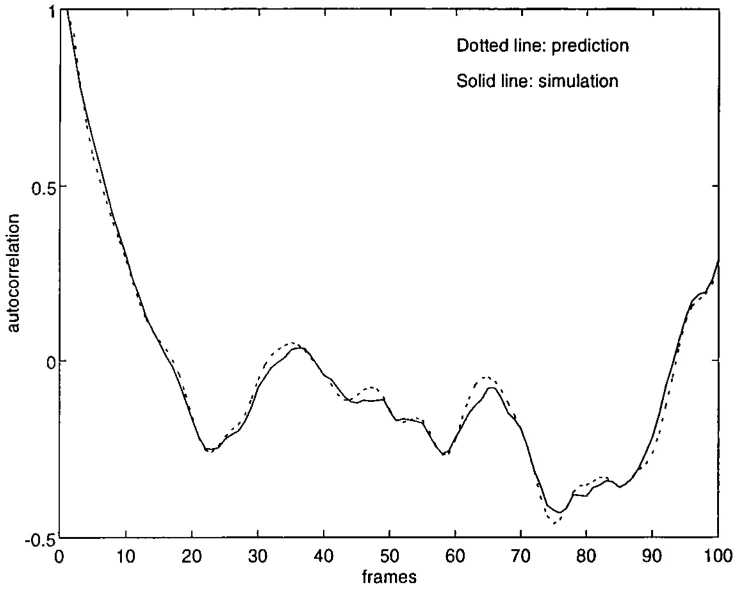

Figure 5.8: Comparison of the autocorrelation function of the predicted video traffic with that of the actual traffic

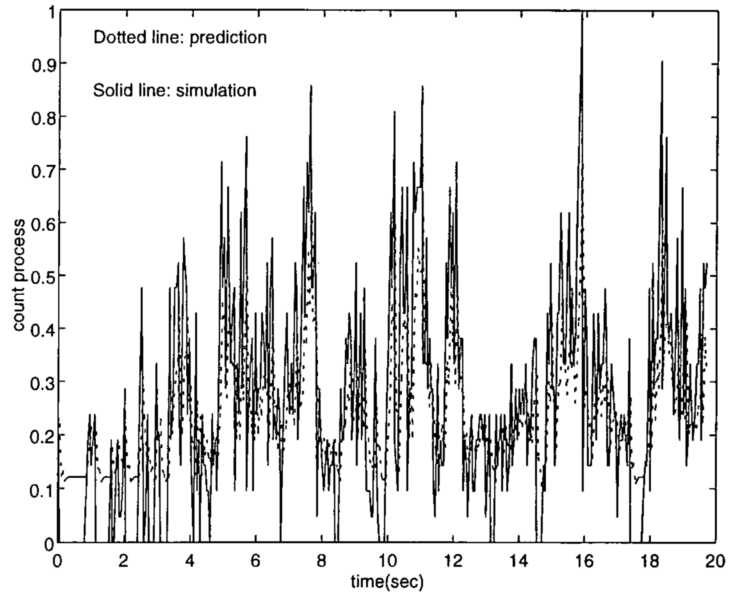

Figure 5.9: Prediction results for the arrival process of voice sources

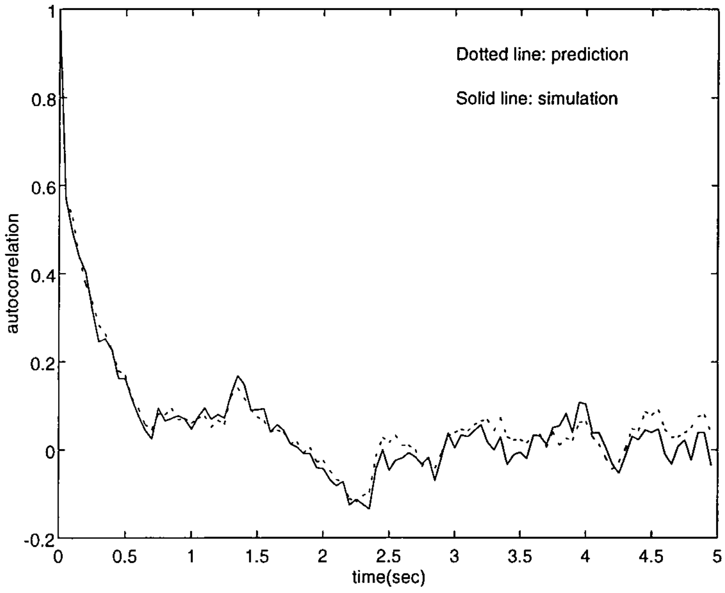

Figure 5.10: Comparison of the autocorrelation function of the count process of the predicted voice traffic with that of the actual traffic

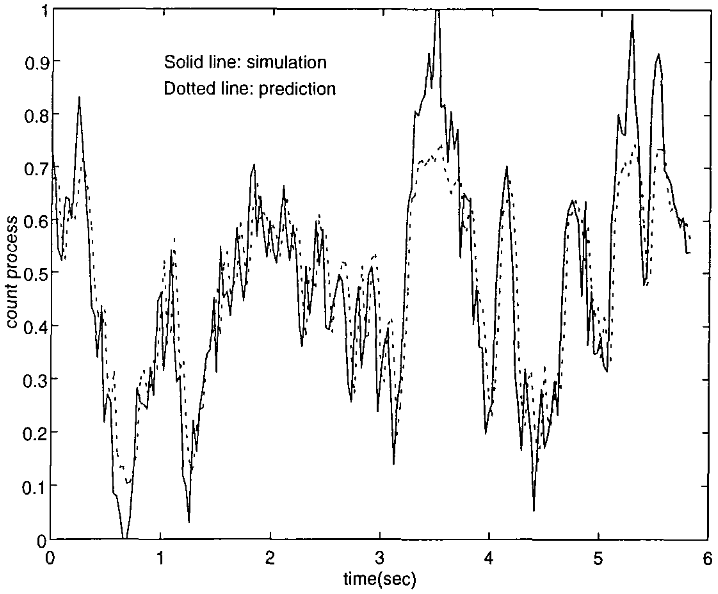

Figure 5.11: Prediction results for the heterogeneous traffic

of voice sources were simulated and their aggregate arrival process was observed at every T_s. Several experiments were performed using different values for T_s. According to the power spectrum analysis of the traffic in each experiment, T_s has been found to be 10 ms. Figure 5.9 and Figure 5.10 show that the neural network prediction is very close to the actual traffic values.

Experiment 3: In this experiment, one video source and three voice sources are used to generate a heterogeneous superposition arrival process. The sampling interval T_s is selected as 10 ms to capture the instantaneous variations for both video and voice traffic. Prediction results for the count process are shown in Figure 5.11 and Figure 5.12. As we can see from the figures, the NN can characterize and predict the multimedia traffic quite accurately.

Experiment 4: Recently, Leland et al. demonstrated that Ethernet local area network traffic is statistically *self-similar* [54]. To capture this fractal behaviour, they proposed to model the traffic using deterministic *chaotic* maps. Chaos is a dynamical system phenomenon in which simple, low order, nonlinear deterministic equations can produce behaviour that mimics random processes. To illustrate the underlying idea, consider a nonlinear map $f(\cdot)$ that describes the evolution of a state variable $x(n) \in (0,1)$ over discrete time as $x(n+1) = f(x(n))$. The packet generation process for an individual source can now be modeled by stipulating that the source generates one or no packet at time n depending on whether $x(n)$ is above or below an appropriately chosen threshold. If f is a chaotic map, the resulting packet process can mimic complex packet traffic phenomena. Once an appropriate chaotic map has been derived from a set of traffic

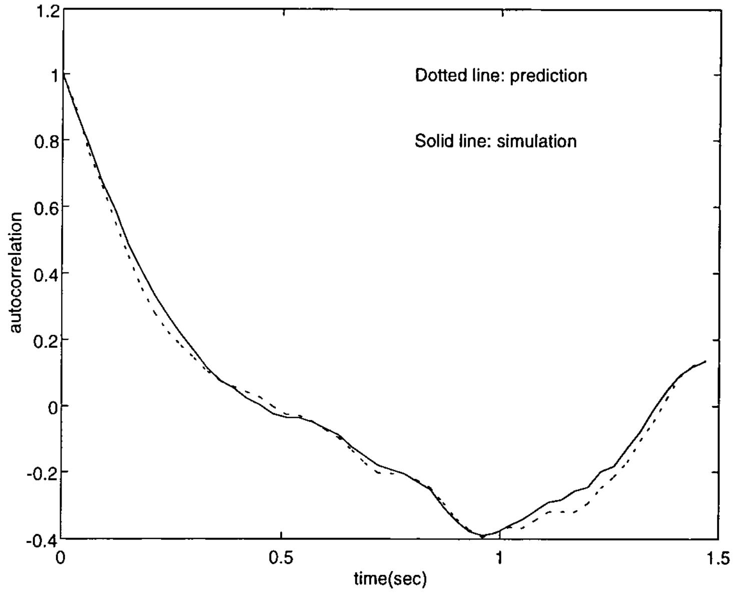

Figure 5.12: Comparison of the autocorrelation function of the predicted traffic with that of the actual traffic in experiment 3

measurements, generating a packet stream for an individual source is generally quick and easy. On the other hand, deriving an appropriate nonlinear chaotic map based on a set of actual traffic measurements currently requires considerable guessing and experimenting. Nevertheless, studying arrival streams to queues that are generated by nonlinear chaotic maps may well provide new insight into the performance of queueing systems where the arrival processes exhibit fractal properties.

Here as another experiment, we train the FIR network to perform one-step prediction of a chaotic time series. A chaotic time series generated by the so-called logistic map is defined as [78]

$$x(n+1) = 4x(n)(1 - x(n)) \tag{5.25}$$

where the values of $x(n)$ are all in the range $(0, 1)$. The prediction results for the training and test data sets are encouraging, as shown in Figure 5.13 and Figure 5.14 respectively.

In this section, we have shown that an FIR network constitutes a powerful tool for use in ATM traffic prediction. The theoretical justification of this approach is that neural networks are capable of approximating any continuous function and perform non-parametric regression. Furthermore, an FIR neural network extends the standard multilayer perceptron to a temporal processing version which is more suitable for modeling of time series. After completing the training phase of the neural network, it can successfully learn the actual probability density function of the offered traffic(instead of the approximated simple parameters, such as the peak and mean bit rates).

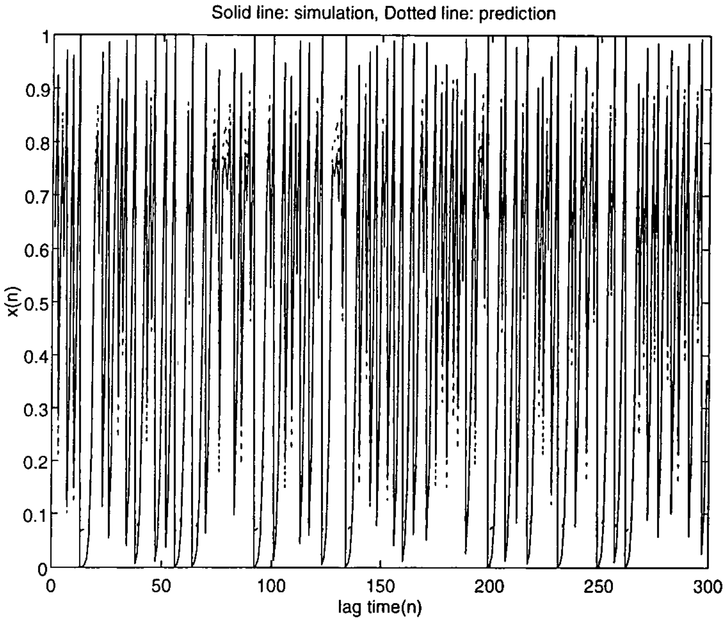

Figure 5.13: Prediction results for the training set of the chaotic time series

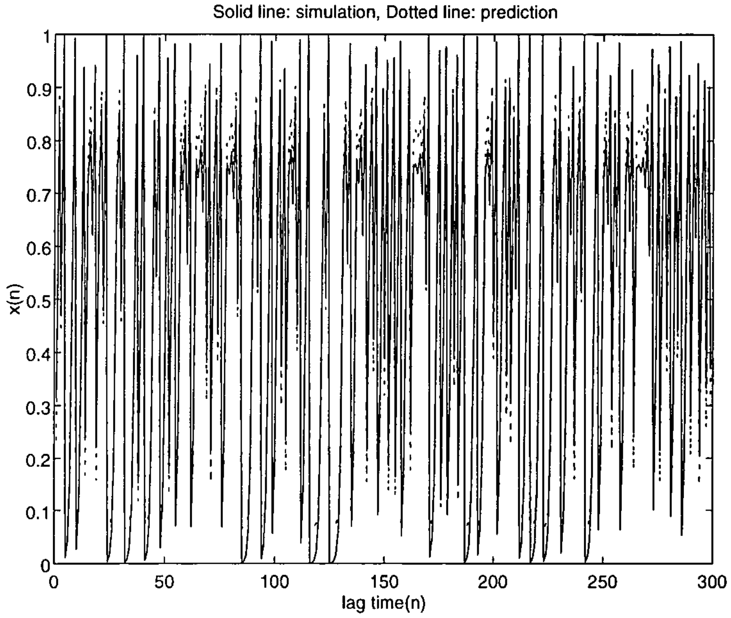

Figure 5.14: Prediction results for the test set of the chaotic time series

In ATM networks, traffic management techniques require traffic parameters that can capture the various traffic characteristics and adapt to the changing network environment. It is shown that the neural prediction is accurate enough to characterize the actual traffic and therefore it can be incorporated into traffic control functions in order to achieve better network performance. Neural traffic prediction has been used in traffic policing [59] and dynamic bandwidth allocation [79]. Recently, Amenyo et al. [80] have proposed a new congestion control scheme called proactive control. Underlying its feasibility and effectiveness are traffic predictions of correlated input traffic streams into network nodes. These predictions are used to obviate the problem of propagation delays. So we can apply our neural prediction method to this framework as well. However, in the next section, we will discuss an application of neural traffic prediction to ATM access flow control.

5.3 Access Flow Control Using a Neural Network Traffic Predictor

The main function of flow control is to regulate the flow of the traffic into the network such that it approximately matches the capacity of the limited resource in the network. One of the problems that makes congestion control in ATM networks difficult is the uncertainty and highly time-varying nature of the diverse mix of traffic sources. Furthermore, due to the very small cell transmission time and the small buffer sizes, it is imperative that any effective congestion control algorithm must be simple with minimal reaction time.

Feedback schemes have been widely studied for congestion control purposes, e.g., [81] [31] [82]. Here we propose a feedback flow control algorithm which is based on traffic prediction by neural networks. The main control action in our scheme is to reduce the peak rate of traffic sources when the neural network predicts possible congestion in the multiplexer. Simulation results presented later in this section suggest that the proposed scheme provides a simple and efficient traffic management for ATM networks.

5.3.1 Rate-based Feedback Control

It is rather difficult to design an access flow control mechanism which can control the superimposed stream of traffic with widely different correlations and burstiness coefficients. Very little control can be done along the transit nodes, so it is essential to smooth down the burstiness of the input traffic to avoid congestion caused by the formation of long bursts inside the network. The most effective method to decrease the burstiness, and hence avoid congestion, is by throttling the peak arrival rate [83]. In this section, we consider a flow control method based on feedback throttling of the arrival process to

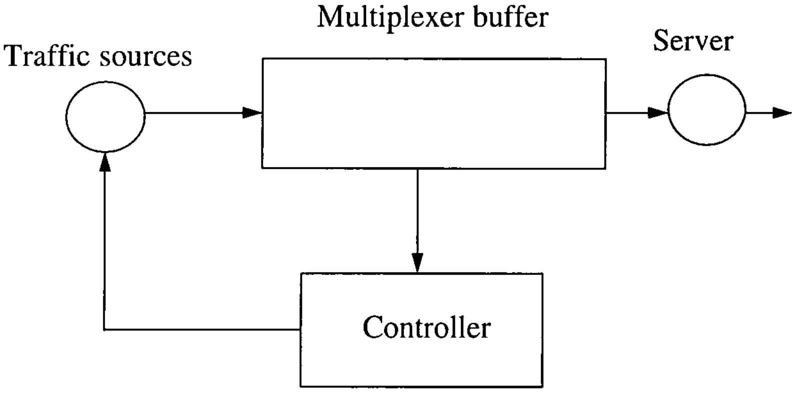

Figure 5.15: Feedback controller

the input statistical multiplexer(see Figure 5.15). During the periods of buffer overload, a feedback control signal is generated to change the source rate by decreasing the coding rate(number of bits per sample). Obviously, this approach is a closed-loop control system.

There are several advantages associated with an algorithm based on feedback throttling [83]. First, it is not of the reactive control type, since it is applied at the input access node to the network, and its speed is not limited by the propagation delay. Any control action taken will be in time to alleviate the potential congestion. Second, it is applicable regardless of the type of encoder used. Third, it provides the means for the maximum possible shaping of the input arrival process through decreasing the peak bit rate. Consequently, the bandwidth allocated to the input call can be reduced, and yet the same required QOS could be achieved. Also, the statistical multiplexing gain is enhanced, since more sources can be supported for each multiplexer. The price to be paid is a slight degradation of the quality of the video/voice delivered.

Consider a multiplexer buffer with size of Q. We divide time into small units, which are time intervals with the same length of T. A_i is the number of cells that arrive between time T_{i-1} and time T_i, where $T_i - T_{i-1} = T$, for all i. The queue has a constant service rate c cells per period of T. Denoting by q_i the queue length at time T_i, then we have the following Lindley's equation:

$$q_{i+1} = \min(\max(q_i + A_{i+1} - c, 0), Q) \qquad (5.26)$$

Cells are lost if the buffer has overflowed.

In our mechanism, at each time interval T_i, the buffer queue length of the next time cycle q_{i+1} is *predicted.* If q_{i+1} reaches a threshold limit Q_{th}, a control signal is sent back and each source reduces its rate to 75% of the current rate. From (5.26) it is easy to see that the measure of congestion q_{i+1} depends on the current queue length q_i and the number of arriving cells in the next lag time, A_{i+1}. Therefore, a key question is how

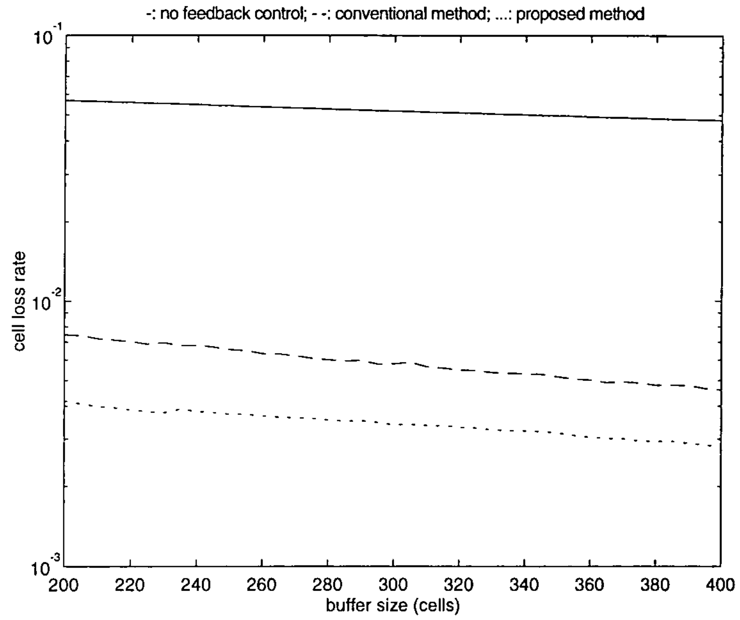

Figure 5.16: Cell loss rate vs. buffer size, single source

to effectively predict the incoming traffic A_{i+1} using on-line traffic measurement. This is achieved by the FIR neural network as discussed previously. From those numerical experiments, we can see that after it has been well trained, the FIR neural network can accurately predict the traffic with various statistical characteristics. This information can then be used by the flow controller to detect possible congestion in the multiplexer buffer.

5.3.2 Simulations

In this section, we explore the efficiency of the proposed feedback control mechanism by running several simulations. Here for simplicity, we only consider the video arrival process. We assume that the buffer threshold Q_{th} is $0.5Q$ and the sampling period T is 1/30 sec. We compare the results of our method with those of a conventional technique, in which the current queue length q_i is monitored to detect congestion. In that case, when $q_i \geq Q_{th}$, a control signal is sent back to all traffic sources and each source reduces its coding rate to 75% of its original value. We also consider the case in which no feedback control is applied. The cell loss rate is used as a performance measure.

In Figure 5.16 and Figure 5.17, we show the simulation results for a single source case. In Figure 5.16, a single video process is fed to the multiplexer and cells are removed from the buffer at a constant rate of 4.875 Mb/s, which yields a utilisation of 0.8. Figure 5.17 shows the performance of the multiplexer versus different traffic loads. The mean bit rate of the source is 3.9 Mb/s; the channel speed is adjusted to provide a specific

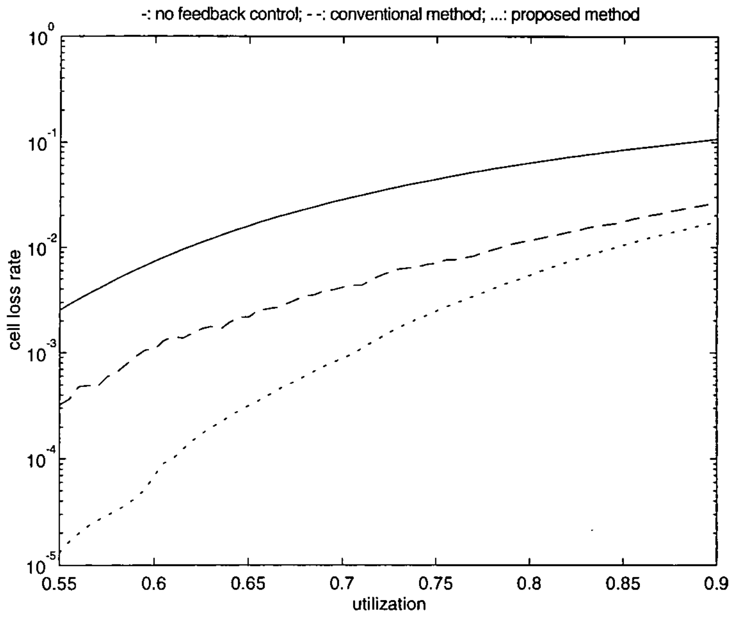

Figure 5.17: Cell loss rate vs. utilisation, single source, buffer size = 100 cells

utilisation. It is clear that the cell loss rate of the proposed flow control mechanism is lower than that of the other two methods. This is because even if the current queue length doesn't exceed the threshold, there is still the possibility of buffer overflow in the next cycle due to the bursty nature of the traffic. In our scheme, the neural network can capture the changing traffic characteristics and hence its prediction enables the controller to respond rapidly and precisely to the onset of situations that could lead to congestion in the network.

In Figure 5.18, three video traffic sources are fed into the network. The queue output rate is set as 14.625 Mb/s. Again, as expected, the performance of the neural-network-based flow control approach is the best. The results reported here prove the importance of applying this type of flow control technique to accommodate more sources with high peak rates without sacrificing the efficiency. As a result, the multiplexing gain is significantly improved.

5.4 Summary

In this chapter, a neural network approach is adopted in the context of ATM traffic modelling and characterization. The FIR NN model and the associated temporal BP training algorithm have been discussed in detail. It is shown that the FIR network can accurately predict the traffic arrival patterns in the near future. A feedback flow control mechanism based on neural traffic prediction has been presented for efficient rate regulation in ATM networks. The predicted output in conjunction with the current

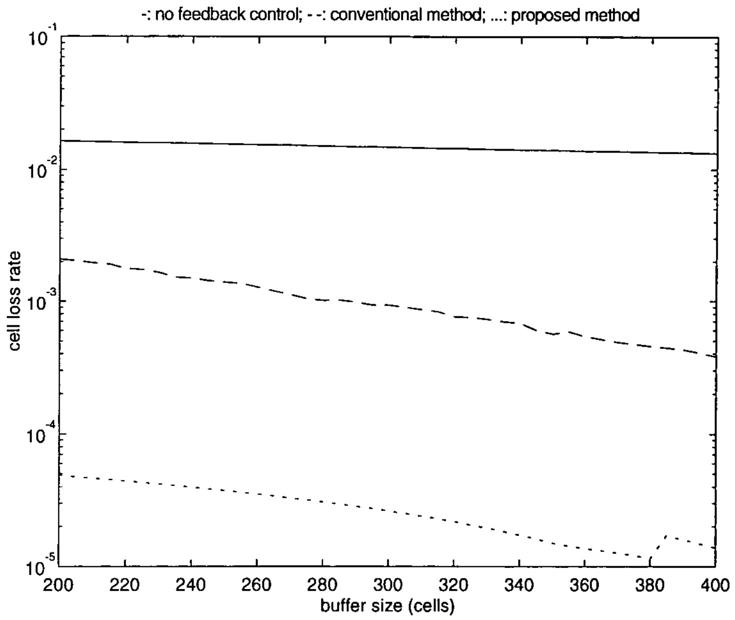

Figure 5.18: Cell loss rate vs. buffer size, 3 sources

queue information of the buffer can be used as a measure of congestion. When the congestion level is reached, the arrival rate is decreased to 75%. Because the feedback control is implemented at the input access node, it is not limited by the propagation delay which is dominant in high-speed networks. Simulation results show that the proposed method outperforms conventional control schemes in terms of cell loss rate.

Traffic prediction and feedback control described in this chapter belong to cell level control. In the next chapter, we will discuss another important application of neural networks to call level traffic control, i.e., bandwidth estimation and admission control.

Chapter 6

Neural Networks for Effective Bandwidth Estimation

6.1 Introduction

One of the important issues in ATM networks is bandwidth management and allocation. Bandwidth allocation deals with determining the amount of bandwidth required by a connection for the network to provide the desired QOS. The problem of bandwidth allocation in ATM networks has been addressed in a number of papers [18] [19] [84]. The major drawback of the methods in these papers is their limited flexibility since they essentially rely on sets of curves or tables, obtained by analysis or simulation, which are to be used as guidelines to determine the required bandwidth of connections. In addition, the static nature of the information may not accurately reflect the dynamic and changing nature of real-time network traffic conditions and connections characteristics. In this chapter, we consider the problem of characterizing the *effective bandwidth* assigned to traffic sources as a function of the desired QOS. Effective bandwidth is needed by the admission control algorithm to decide if and how to accept incoming connection requests. By preventing admission to an excessive number of calls or sources to the multiplexer, call admission policies strive to make a balance between QOS(e.g., delay and cell loss probability) and efficient use of network resources. Effective bandwidth is determined by the source characteristics in conjunction with the admission criteria. Moreover, the corresponding procedure must be computationally simple enough so as to be consistent with *real-time* requirements.

The effective bandwidth of a set of connections multiplexed on a link is defined as the amount of bandwidth required to achieve a desired QOS, e.g., buffer overflow probability, given the offered aggregate bit rate generated by the connections. CAC is performed by comparing the total effective bandwidth of the incoming connection and existing ones with the available link capacity to make call acceptance/rejection decisions. In this chapter, loss or buffer overflow probability is selected as the sole performance criterion for QOS measurements. Other constraints, e.g., delay or jitter, can typically be handled through other link-level mechanisms.

There are two important issues concerning bandwidth estimation. One is efficiency(in terms of computation time) of the adopted technique and the other is its accuracy. In connection admission control and dynamic bandwidth allocation, efficiency is a necessity due to the real-time nature of the applications. Accuracy is also a significant factor since a more accurate technique usually leads to a better utilisation of network resources. Unfortunately, efficiency and accuracy often conflict with each other. Thus we have to find a proper trade-off between them.

The fluid flow model [85] is a sufficiently accurate link performance model which has been widely used in voice and video traffic modelling. However, to obtain the exact solution of the effective bandwidth it is necessary to solve a set of differential equations and this is time-consuming. The main problem is that the number of states of the aggregate traffic process increases exponentially with the number of calls. Hence as the connection setup time is constrained, this rather complex model is not practical and approximations must be made at the price of inaccurate results. To overcome this efficiency-accuracy dilemma, in this chapter, a multilayer perceptron is proposed to model the nonlinear relationship between the effective bandwidth and the traffic situations and a QOS measure. This is based on the fact that the function approximation capability of NNs can be employed to implement accurate performance functions, computed off-line. Furthermore, additional performance data can be collected and applied during on-line operation to compensate for possible errors in the traffic and/or performance model. The neural network is trained and tested via a large number of patterns generated by the accurate fluid flow model. Due to the neural network's adaptive learning, high computation rate and generalization features, this method is feasible for real-time network traffic control applications. On the other hand, since the admission control is based on an accurate performance model, a higher number of accepted connections is possible, resulting in a more economical ATM network utilisation. Figure 6.1 shows a schematic diagram of the proposed CAC method using neural bandwidth estimation.

The case of multiple superimposed ON/OFF sources (homogeneous or heterogeneous) is considered. The accuracy of our method is examined by comparing it to both exact computations and conventional approximation results, and is found to be acceptable across the range of possible connection characteristics. We also discuss some applications of this NN-based bandwidth estimation approach to dynamic time-slice schemes and dynamic routing. First, in the next section, we briefly describe the underlying fluid flow queueing model.

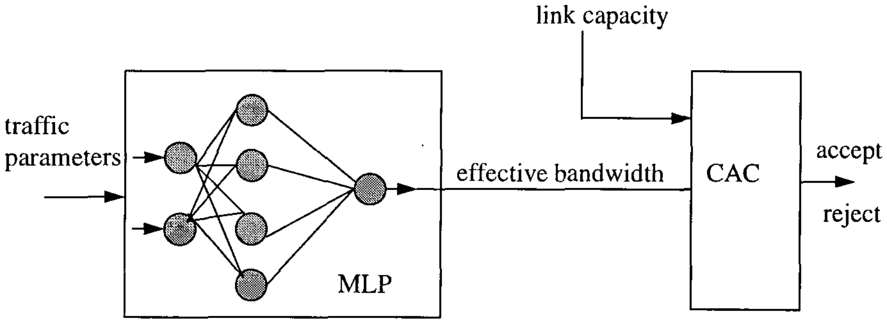

Figure 6.1: CAC based on effective bandwidth estimation by a neural network

6.2 The Fluid Flow Model

There are quite a few analytical models for ATM multiplexers, among which the fluid flow model is one of the most accurate. The model treats traffic sources as sources of fluid flow generating continuous streams of cells and is inherently characterized by long-term continuous-time statistics of the system. A thorough analysis of the model is presented in [85]. Anick et al. [85] show that when the Markov-modulated rate process consists of the superposition of a finite number of independent identical continuous flow sources each modulated by a two-state Markov process, and the service capacity is constant, it is possible to derive closed-form solutions for the buffer-occupancy distributions, as well as very simple asymptotic approximations. In [86], Mitra considers multiple sources and multiple servers coupled by a finite buffer.

The fluid flow model has been used to produce reasonably accurate predictions of queue length distributions in packet voice multiplexers [69] [87]. Nagarajan et al. [88] have shown that the results of the fluid approximation for computing packet loss probabilities in a voice multiplexer are as accurate as the other methods based on MMPP models. Maglaris et al. [75] have used this model to analyze the multiplexer performance in packet video communications successfully. However, the model lacks the concept of packetization, and therefore cannot model the packet arrival process in any detail. Actually, it is pointed out in [89] that there are two distinct components to congestion phenomena in ATM networks, i.e., cell scale congestion and burst scale congestion. Cell scale congestion occurs due to the simultaneous arrival of cells from independent sources when the overall arrival rate due to active sources is less than link capacity. Burst scale congestion occurs when the overall arrival rate is momentarily greater than link capacity. Here we only consider burst scale performance in terms of the buffer overflow probability, assuming that the buffer is dimensioned to resolve the cell scale conflicts.

In a two-state fluid flow model, the packet generation process is taken to be a collection of N identical sources alternating between active(ON) and inactive(OFF) states. The ON periods as well as the OFF periods are exponentially distributed for each source. Without loss of generality, the unit of time is selected to be the average ON period; with this unit of time, the average OFF period is denoted by $1/\lambda$. Again, without loss of generality, the unit of information is chosen to be the amount generated by a source in an average ON period. In these units an ON source transmits at the uniform rate of 1 unit of information per unit of time and the server removes information from the buffer at a uniform rate of c units per unit of time. We assume that the buffer is infinite [1] and that the following stability condition is satisfied [85]:

$$\frac{N\lambda}{c(1+\lambda)} < 1 \tag{6.1}$$

The queueing system can be expressed by a set of differential equations. The probability of overflow is defined as $G(x) = \text{Pr[buffer content} > x]$ and is given by [85]

$$G(x) = -\sum_{i=0}^{N-[c]-1} e^{z_i x} d_i (\mathbf{1}^T \phi_i) \tag{6.2}$$

where z_i is some eigenvalue of the queueing system, ϕ_i is the associated right eigenvector, $\mathbf{1}$ denotes the identity vector and $(.)^T$ denotes transpose. The coefficients $\{d_i\}$ in (6.2) must be obtained via boundary conditions.

Next, we extend the fluid flow queueing model for identical ON/OFF traffic sources to the case with heterogeneous ON/OFF sources. The equilibrium buffer distribution is again found as solution to a set of first order differential equations. When the number of different sources is large, the number of states needed to represent the input stream is very large. A so-called "decomposition method" has been developed in [90] to deal with this problem. It is shown that the separability property can permit a decomposition of the equations for the equilibrium probabilities of the system. The decomposition technique leads to a solution of the equilibrium equations expressed as a sum of terms in Kronecker product form, hence reducing the computational complexity for large systems. For more details regarding the analytical solution of the fluid flow model, please refer to Appendix B.

Recently, an efficient bandwidth allocation method based on asymptotic approximations of the fluid flow model is proposed [20] [91] [92]. In the model of statistical multiplexing considered, for given buffer size B and target overflow probability ϵ, let the QOS be $\{G(B) \leq \epsilon\}$, which is also taken to be the admission criterion. c is the link service rate. First consider the situation where there is only a single-source. In [20], Guerin

[1] In the case of infinite buffers, the probability that the queue length exceeds the buffer size x is an upper bound for the overflow probability.

et al. adopted a two-state ON/OFF model, while in [91], Elwalid and Mitra considered more general sources of higher-dimensions. It is shown that in the asymptotic regime where $\epsilon \to 0$ and $B \to \infty$ in such a manner that $\log \epsilon / B \to \zeta \in (-\infty, 0]$, the admission criterion is satisfied if $e < c$ and violated if $e > c$. e is called the effective bandwidth and is found to be the maximal real eigenvalue of the inverse queueing system. Suppose that the above single source is, in fact, the aggregate of K arbitrary sources. Then the effective bandwidth $e = \sum_{k=1}^{K} e_k$, where e_k is the effective bandwidth of the source k computed as if it is a single source in the system.

The above mentioned method is computationally simple and quite accurate for large buffer size B and small overflow probability ϵ(of the order of 10^{-9}). But it must be noted that the statistical multiplexing effect of a large number of sources is ignored in this approach and hence it may significantly overestimate the required bandwidth for the aggregate traffic. This is due to the following approximation:

$$G(x) \sim A \exp(z_0 x) \sim \exp(z_0 x) \tag{6.3}$$

where z_0 is the *dominant eigenvalue* of the queueing system described previously. (6.3) means that not only do we approximate the tail with a single exponential term, but we take the leading constant $A = 1$. In fact A can be very different from 1, say, of the order of 10^{-5}, reflecting important characteristics of multiplexing [93].

6.3 The Neural Network Approach

In this section, we discuss a neural network approach for effective bandwidth estimation.

6.3.1 Effective Bandwidth for Identical Sources

Here the traffic can be described by the following parameters: the number of sources N, the constant cell generation rate or peak rate R, and the average duration times of the ON and OFF periods $1/b$ and $1/a$, which characterize the exponential distributions. The buffer capacity B, and the overflow probability of ϵ(the desired QOS), are also included in the model. Suppose that the aggregate bit rate is offered to a buffer which is emptied at a constant rate of C. For the sake of simplicity, we express this model in the unit of time and information notations discussed in Section 6.2. The unit of time is taken to be $1/b$. One unit of information would thus correspond to R/b and the average time units that the source is in an OFF state is $\lambda^{-1} = b/a$. The buffer capacity in units of information equals $x = Bb/R$. Furthermore, let c denote the ratio of C to an ON source's transmission rate R.

We are interested in determining the smallest value $\hat{c}$ of c that, for a given buffer size x, ensures a buffer overflow probability smaller than ϵ. The value $\hat{c}$ is called the normalized(with respect to R) effective bandwidth of the multiplexed connections, while the value $\hat{C} = R\hat{c}$ is the actual effective bandwidth. The determination of the effective bandwidth requires that we first obtain an expression giving the distribution of the buffer content as the function of the connections characteristics and the service rate. This expression must be *inverted* to determine the value of the service rate, which ensures an overflow probability smaller than ϵ for the available buffer size x. This value is the effective bandwidth that should be allocated to the connections. The distribution of the buffer content can be derived using the method described in Section 6.2. However, the associated computational complexity is often not compatible with the real-time requirements. This is because, even when the buffer content distribution can be explicitly derived, the resulting expression cannot be easily inverted to yield the effective bandwidth as a function of other parameters. To solve this problem, approximations are used to get the effective bandwidth in [91] [20].

Here a different approach based on neural networks is proposed in the following. It has been shown in [71] [70] that multilayer feedforward networks are capable of approximating any measurable function to any desired degree of accuracy. To be specific, let p denote the number of input(source) nodes of an MLP, and let q denote the number of neurons in the output layer of the network. The input-output relationship of the network defines a mapping from a p-dimensional Euclidean input space to a q-dimensional Euclidean output space, which is infinitely continuously differentiable. The universal approximation theorem states that a single hidden layer is sufficient for a multilayer perceptron to compute a uniform ε approximation to a given training set represented by the set of inputs $x_1, ..., x_p$ to a desired (target) output $y(x_1, ..., x_p)$. This theorem provides the mathematical justification for the approximation of an arbitrary continuous function as opposed to the exact representation.

We can represent the unknown complex function of the effective bandwidth as

$$\hat{c} = f(\lambda, N, x, \epsilon) \tag{6.4}$$

where the parameters are as those defined previously. A three-layer feedforward neural network is used to model this relationship. We assume that ϵ and x are fixed values. The inputs to this network are λ and N, while the output is the effective bandwidth $\hat{c}$. In this application, the neural network consists of 2 input neurons, 5 hidden neurons and 1 output neuron. The hidden and output neurons have nonlinear sigmoid activation functions. This multilayer perceptron is trained with the backpropagation algorithm. The desired outputs of the training samples are obtained as follows.

To evaluate the effective bandwidth $\hat{c}$, we solve the following equation:

$$g(\hat{c}) - \epsilon = 0, \qquad N\lambda/(1+\lambda) < \hat{c} < N \tag{6.5}$$

where g is a function that estimates the overflow probability derived from the analytical method. Actually, the function f in (6.4) is some kind of inverse of g. We can use some numerical methods (e.g., bisection method) to get the solution $\hat{c}$ of (6.5). The different N's and λ's, together with the corresponding different $\hat{c}$'s are used as a training set. When the off-line training is complete, the normalized effective bandwidth can be given as

$$\hat{c} = \mathcal{N}(\lambda, N) \tag{6.6}$$

where $\mathcal{N}$ is the output function of the neural network. The actual effective bandwidth $\hat{C}$ is $R\hat{c}$. From the above discussion, we can see that the multilayer perceptron is used as a powerful inverse modeling tool.

6.3.2 Effective Bandwidth for Heterogeneous Sources

In the case of heterogeneous sources, the distribution of the content of a buffer, fed by the aggregate bit rate and served by a constant rate server, can also be determined. The procedure is very complex and essentially numerical in spite of the existence of the powerful simplification technique, i.e., the decomposition method. The distribution of the buffer content is completely determined from the values of the associated eigenvalues, eigenvectors, and corresponding coefficients. There are typically no explicit expressions for these quantities, which must then be determined numerically. Accordingly, the effective bandwidth corresponding to a set of multiplexed heterogeneous sources can be obtained using iterative numerical techniques. Such a procedure, although exact, is unfortunately not compatible with a dynamic and real-time environment. Alternative methods must be developed. In the following, we try to extend the result of Section 6.3.1 to the case of heterogeneous sources.

Now consider the case of heterogeneous sources of k classes. The identical sources in class j are characterized by $(N_j, 1/b_j, 1/a_j, R_j)$, where N_j denotes the number of sources in class j, $1/b_j(1/a_j)$ is the average duration in the ON(OFF) state for sources in class j and R_j is the constant transmission rate of each source in class j in the ON state. In the OFF state no data are transmitted. The buffer capacity is B and the desired overflow probability is ϵ. From b_j and a_j, we can get the corresponding $\lambda_j (j = 1, 2, ..., k)$.

To obtain the effective bandwidth for heterogeneous sources, we make the following approximation. The statistical multiplexing effect is taken into account among the identical sources in the same class, while it is ignored between different classes. The exact effective bandwidth for the sources in class j can be obtained via the neural network approach proposed in Section 6.3.1:

$$\hat{C}_j = R_j \mathcal{N}(\lambda_j, N_j) \tag{6.7}$$

Then the total effective bandwidth for all the sources is given by

$$\hat{C} = \sum_{j=1}^{k} \hat{C}_j \tag{6.8}$$

The major advantages of (6.8) are its computational simplicity and its flexibility. However, the linearity of (6.8) certainly simplifies the accounting of how bandwidth is allocated to connections and hence to some extent overestimates the effective bandwidth. This can be seen from the simulation results reported below.

6.4 Numerical Results

In this section, we provide a number of numerical examples that check the accuracy of the neural network approach to estimate the effective bandwidth. Extensive simulations have been performed to obtain the neural network's data set, for both training and operation phases. The traffic situations are chosen randomly, with the restrictions $N \in \{1, 2, ..., 150\}$ and $\lambda \in \{0.05, 0.1, ..., 1.0\}$. The multilayer perceptron is trained with an adaptive learning rate backpropagation algorithm. The learning rate is nominally set at 0.4(this is selected heuristically and then varied during the course of training). The momentum constant is chosen to be 0.1. The data are fed to the neural network until the convergence of the mean squared error is achieved. The rate of convergence is illustrated in Figure 6.2. It shows a plot of the MSE vs the number of times a data set has been presented to the network. It takes about 2400 iterations to reach an acceptable MSE level of 10^{-5}. After training, the neural network can be used as an effective bandwidth estimator. Figure 6.3 shows the neural network's output $\hat{c}$ as a function of the inputs N and λ. It can be seen from the figure that the effective bandwidth is nonlinearly increasing as N or λ increases.

In the following figures we compare the results of exact analysis, the neural network approach and Guerin et al.'s approximation method [20]. We assume the source peak rate $R = 1$ Mb/s and the mean of the burst period $1/b = 1$ s. The mean of the inactive period is denoted by λ^{-1} time units. The normalized buffer capacity is 10 and the buffer overflow probability is set to 10^{-5}. According to the approximation method of Guerin et al. [20], the effective bandwidth for N sources is given by

$$\hat{c} = \frac{N}{2\alpha(1-\rho)}(\alpha(1-\rho) - x + \sqrt{[\alpha(1-\rho) - x]^2 + 4x\alpha\rho(1-\rho)}) \tag{6.9}$$

where $\alpha = \ln(1/\epsilon)$ and $\rho = \lambda/(1+\lambda)$. Figure 6.4 shows, as a function of λ, the effective bandwidth that needs to be allocated to 45 sources. The results illustrate that the flow approximation overestimates the required bandwidth especially at low loads(when λ is small), while the output of the NN agrees remarkably well with the

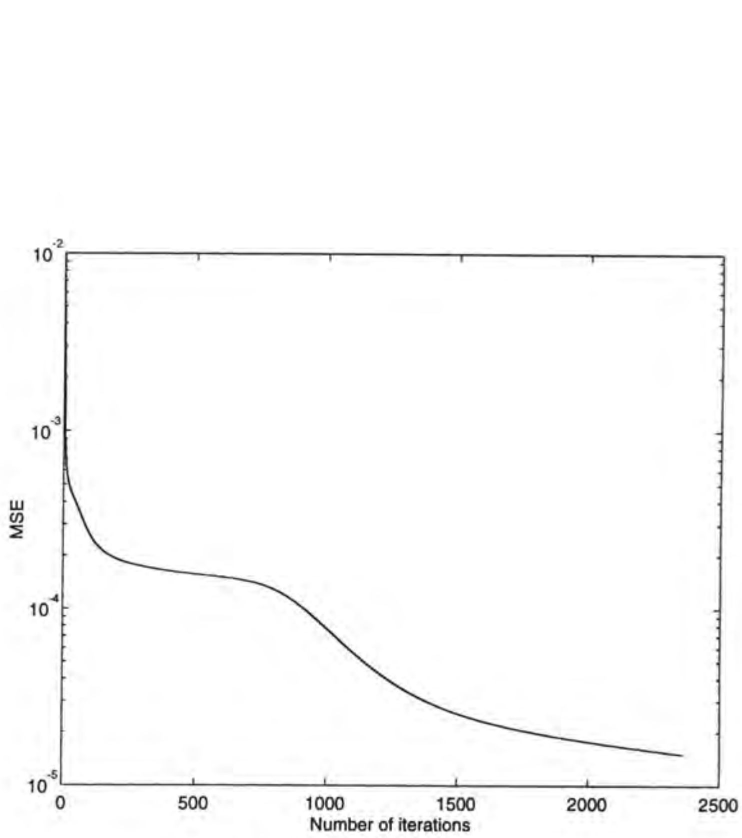

Figure 6.2: Rate of convergence

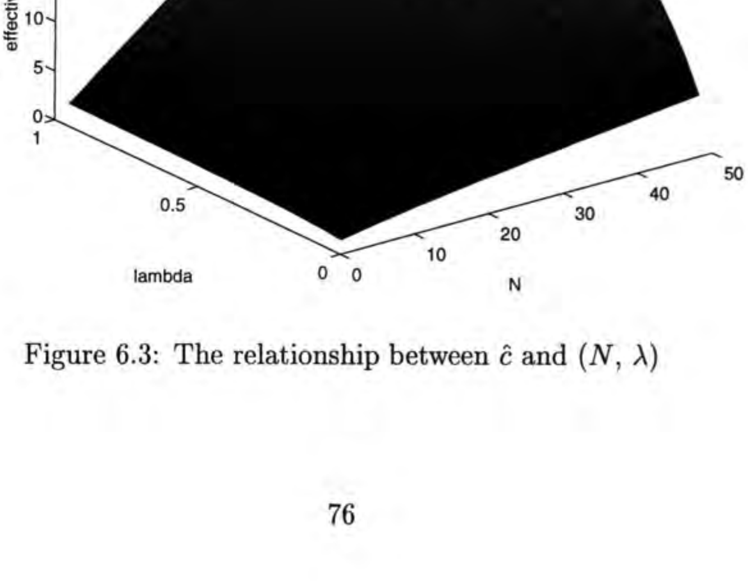

Figure 6.3: The relationship between $\hat{c}$ and $(N,\, \lambda)$

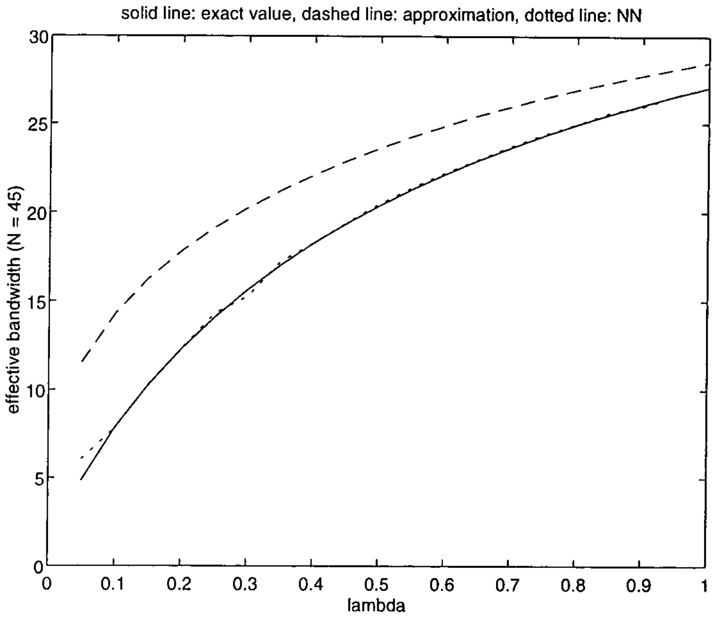

Figure 6.4: Effective bandwidth for $N = 45$

exact value. We then fix λ at 0.2 and let the number of sources N vary. This case is shown in Figure 6.5. It is obvious that as N increases, the approximation method becomes more and more conservative since it ignores the interaction between sources. On the other hand, the neural network produces accurate results over the whole range. Similar conclusions can be drawn from the results shown in Figure 6.6 and Figure 6.7. In our simulations, we have also found that during the on-line operation phase, the time required by the NN to compute the effective bandwidth is significantly less than that of the exact solution(several orders of magnitude improvement). The software simulations have verified the soundness of the use of NN approach for bandwidth estimation for ATM traffic and suggest that the real advantages of NNs can be fully utilised by future special-purpose hardware or neurocomputer implementations.

As an example for admission control, we consider the following scenario. Let $R = 1, b = 1, \lambda = 0.1$. The channel capacity is 7.273 and the buffer size is 10. Then we choose the maximal number of sources so that the buffer overflow probability $\epsilon \leq 10^{-5}$. The exact analysis indicates that the above QOS is attained with $N = 41$ sources, while our neural estimation yields the same result. If, instead, we use the approximation (6.9), it turns out that the queue can accommodate only 22 sources. Hence, the neural network approach leads to much higher resource utilisation than the approximation one.

Now consider the case of superposition of nonidentical sources. Here a simple example is given to illustrate the effectiveness of the NN approach. There are two source classes. The sources of both classes are ON/OFF with exponentially distributed ON and OFF

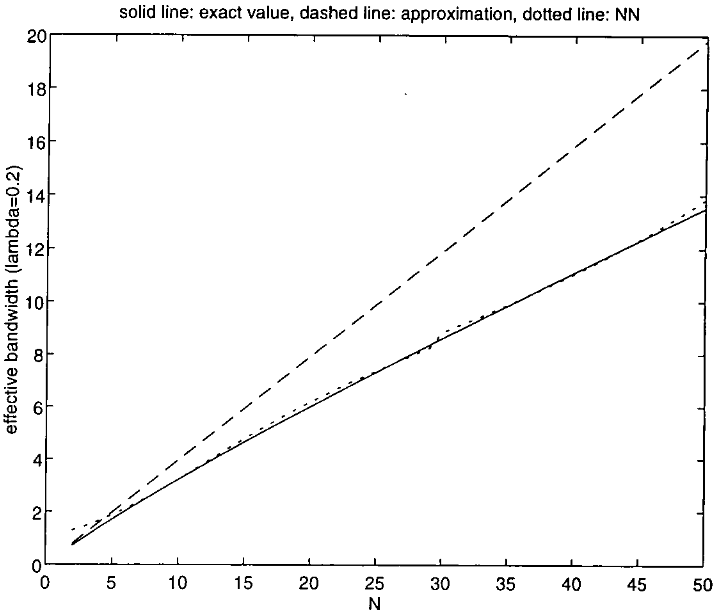

Figure 6.5: Effective bandwidth for $\lambda = 0.2$

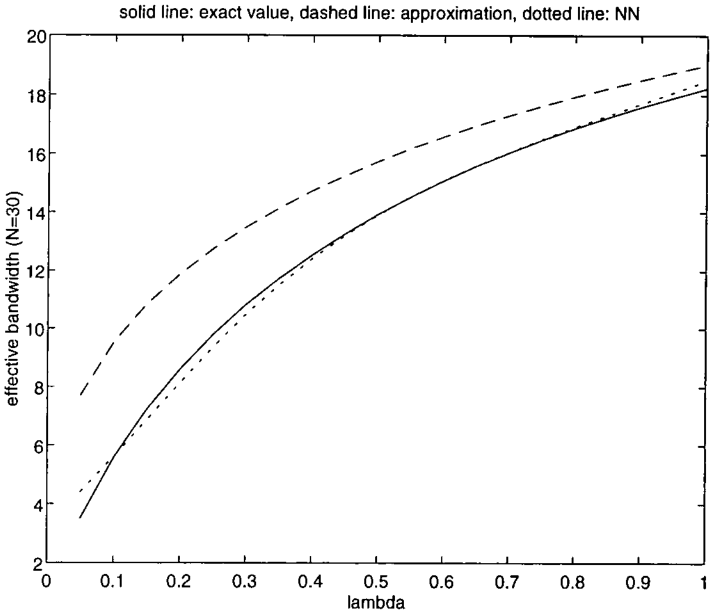

Figure 6.6: Effective bandwidth for $N = 30$

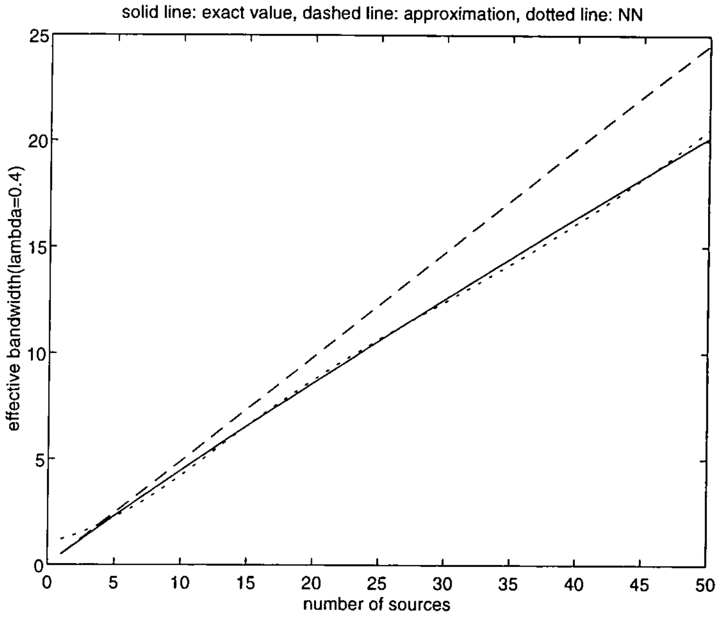

Figure 6.7: Effective bandwidth for $\lambda = 0.4$

periods. The source parameters are as follows:

$$R_1 = R_2 = 1, b_1 = b_2 = 1, \lambda_1 = 0.1, \lambda_2 = 0.2, N_1 = 2, N_2 = 3$$

The buffer size and the desired buffer overflow probability are the same as those in the previous examples. Thus the effective bandwith obtained by the neural network approach and the flow approximation are $\hat{c}_N = 1.6003$ and $\hat{c}_A = 1.8242$, respectively. Taking the above two bandwidth values as the channel capacity, we compare the resulting buffer overflow probabilties ϵ_N and ϵ_A with the buffer size $x = 10$. This is done by using the decomposition method mentioned in Section 6.2. Then we have $\epsilon_N = 4.013 \times 10^{-6}, \epsilon_A = 1.411 \times 10^{-6}$. As expected, the result of the neural approximation is closer to the desired QOS (10^{-5}). Since the effective bandwidth obtained from the flow approximation is simply the sum of all the individual sources, it is a more conservative estimate, hence reducing the throughput of the network. However, as shown in the numerical example, this drawback can be improved by the NN estimation method in which the interaction between identical sources is taken into consideration. In the next two sections, we discuss some applications of the NN-based bandwidth estimation.

6.5 A Dynamic Time-Slice Scheme with Bandwidth Estimation by NNs

In this section, we apply the NN-based bandwidth estimation method to a bandwidth allocation scheme called dynamic time-slice. In [94], Sriram proposed ATM cell multiplexing using a Dynamic Time-Slice (DTS) scheme which allocates and guarantees a required bandwidth for each traffic class and/or virtual circuit. The scheme is dynamic in that it allows the different traffic classes or VCs to share the bandwidth with a soft boundary. Any bandwidth momentarily unused by a class or a VC is made available to the other traffic present in the multiplexer. The scheme guarantees a desired bandwidth to connections which require a fixed wide bandwidth. Although the DTS scheme has the above good features, it still has one major drawback: It uses analytical and hence *static* traffic tables to provide the amount of bandwidth that is needed to multiplex a given number of calls in the queue for a particular call class. The traffic tables are updated only when new coding and compression methods are developed for voice, image and video. Therefore, this technique is not able to adapt gracefully to the time-varying nature of traffic behaviour and network conditions, especially when new services are being continually introduced after network design and installation.

Here we propose a neural-network-based DTS(NN-DTS) method which uses neural networks to compute the required effective bandwidths of various traffic classes. Simulation results reported above show that the neural estimation is accurate and fast and can lead to optimal use of network resources. This is especially useful for CBR conference video, VBR video and other real-time high-bandwidth CBR services.

6.5.1 The NN-DTS Scheme

The NN-DTS scheme is shown in Figure 6.8. Like the DTS scheme, NN-DTS also assigns a separate queue to each call type and services all the queues by cyclically visiting each queue and allocating a slice of time to it. The time slice allocation to a queue would be proportional to the bandwidth required by that queue, which is estimated by the NN. For example, in Figure 6.8, 30 cells in a cycle of 450 cells are allocated to the voice queue to give 10 Mb/s (1/15-th of the link bandwidth) to all the voice calls collectively. Let M_c denote the maximum number of cells that can be transmitted in a DTS cycle. For the example in Figure 6.8, $M_c = 450$. Let k denote the number of queues(call types) in the NN-DTS configuration at a particular instance. As k varies, the time-slice allocations, $T_1, T_2, ..., T_k$, for the k queues, are also reassigned values to reflect the proportionate bandwidth requirements for all traffic classes. The units of the time-slices are in terms of the number of ATM cells. Though not illustrated in Figure 6.8, we assume that there is one queue at each link that is dedicated to the signalling traffic; we call it queue 0.

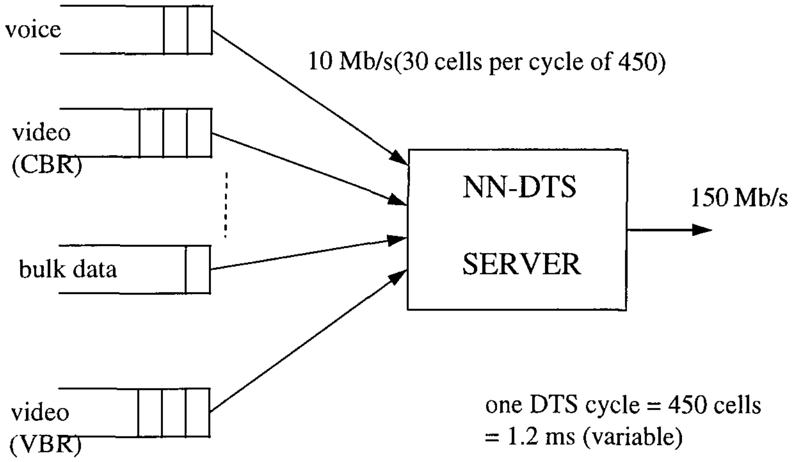

Figure 6.8: An illustration of operation of the NN-DTS scheme

Suppose that $k+1$ queues are to be set up with bandwidth requirements in the fractions of $f_0, f_1, ..., f_k$ of the link bandwidth. If the link service rate is C, then the fraction of bandwidth required by type i call, f_i, is given by $f_i = \hat{C}_i / C$, where $\hat{C}_i$ is the actual bandwidth for type i call estimated by the NN. The time-slice parameters $(T_0, T_1, ..., T_k)$ are chosen such that the following relationships hold [94]:

$$f_i = \frac{T_i}{\sum_{i=0}^{k} T_i}, \qquad 0 \leq i \leq k, \tag{6.10}$$

$$\sum_{i=1}^{k} f_i \leq 1 - f_0, \tag{6.11}$$

$$\sum_{i=0}^{k} T_i \leq D_c, \qquad (D_c \approx 1 \text{ to } 2 \text{ ms}) \tag{6.12}$$

where $D_c = M_c \tau$ is the DTS service cycle time, M_c is the number of cells in a DTS service cycle as defined before, and τ is the cell transmission time on the link. Note that (6.12) guarantees that the cell delays never exceed 1 or 2 ms for all CBR and VBR video calls. Since D_c is crucial in terms of maximum delay, the massive computational capability of NN makes it an ideal choice to provide faster response in reallocation of bandwidth. For more details of DTS servicing strategy and implementation considerations, see [94].

In the original DTS scheme, traffic tables are maintained for each call class to provide the amount of required bandwidth. This static method does not work well as the number of traffic classes grows or the statistical characteristics of traffic sources change dynamically. To cope with this problem, we can use the neural network approach described previously to adaptively estimate the effective bandwidths of various call types. This method reflects the time-varying nature of traffic conditions and is fast enough to satisfy the stringent delay requirement.

6.6 Dynamic Routing with Effective Bandwidth Estimation by NNs

Consider the problem of routing in VP-based ATM networks. The network selects a path for the new connection, i.e., find a set of VPs to connect the source to the destination. Routing is closely related to call admission control, since the network accepts a call only if it can find a suitable path. The bandwidth requirement for calls, is first calculated at call setup with knowledge of the traffic characteristics, such as peak rate, mean silence duration and mean burst length. As we mentioned earlier, computation of exact values for effective bandwidth is extremely difficult and it is necessary to use asymptotic approximation techniques. However, because they ignore the statistical multiplexing effect of a large number of sources, such approximations can be very conservative and overestimate the real bandwidth by a large amount. Thus routing algorithms based on this bandwidth estimation may result in under-utilisation of network resources. Here we apply the NN-based bandwidth estimation method to the Least Loaded Routing(LLR) scheme. Numerical results show that the neural estimation is accurate and fast and can achieve much lower call blocking probabilities than the original LLR method.

6.6.1 The Proposed Routing Algorithm

Let's consider a network consisting of a set of nodes M, a set of links(or VPs) L, and a set of possible Origin-Destination(O-D) pairs W. Associated with each O-D pair $w, w \in W$, is a set of possible paths. We assume that for any link $l(l = 1, ..., L)$, its capacity is a fixed value C_l. The traffic is modelled by homogeneous ON/OFF sources as described previously. Individual call requests arrive according to a Poisson process and the call holding times are exponentially distributed.

The LLR algorithm is one of the most popular real-time adaptive routing schemes [95]. Here we modify the algorithm by using NNs to estimate the required bandwidths on any link. Since it has been well recognized that it wastes too much network resources to route calls over a path with more than two VPs [96], we restrict the choice of path to single-link(direct) and two-link(alternative) routes. Note that this restriction imposes no penalty on network cost in ATM networks. Suppose we want to establish a VC between an O-D pair with a required end-to-end cell loss probability ϵ. If we route the call in a direct path, the cell loss probability should be less than or equal to ϵ. However, if a two-link alternative route is used, cell loss can occur at the either two VP input buffers along the route and thus the sum of the cell loss probabilities of the two VPs should not exceed ϵ. Therefore, the individual cell loss probability in each VP could be taken as $\epsilon/2$. To this end, we employ two neural networks, $\mathcal{N}_D$ and $\mathcal{N}_A$, for computing the effective bandwidths under the constraints of ϵ and $\epsilon/2$ respectively.

The modified LLR algorithm is outlined in the following. Assume at the call arrival instant there are already N_l^D direct connections and N_l^A alternative connections on link $l(l = 1, ..., L)$. Taking the new connection into consideration, we define the residual capacity C_P of a path P by

$$C_P = \min_{l \in P}\{C_l - R(\mathcal{N}_D(N_l^D, \lambda) + \mathcal{N}_A(N_l^A + 1, \lambda))\}. \tag{6.13}$$

Let S be the set of two-link alternative paths whose residual capacity is larger than or equal to 0, i.e., $S = \{P : C_P \geq 0\}$. Thus, for an incoming connection,

1. first attempt to set up the connection along the direct link if $C_l \geq R(\mathcal{N}_D(N_l^D + 1, \lambda) + \mathcal{N}_A(N_l^A, \lambda))$.

2. If there is not enough bandwidth to establish connection in the direct link, select a two-link alternative path with the largest residual bandwidth in S.

3. If there are more than one such path, pick one randomly.

4. If the candidate set S is empty, block the call.

6.6.2 Simulation Results

We consider a fully connected and symmetric network which has $M = 6$ nodes. For each simulation the source peak rate R is 10 Mb/s, the mean burst duration $1/b$ is 0.05 s, the mean silence duration $1/\lambda$ is 4 time units and the required overflow probability ϵ is set to 10^{-6}. Each VP has capacity $C_l = 150$ Mb/s. The mean call holding time is 10 s. In order to investigate the performance of the routing algorithms under a wide range of possible loading scenarios, the arrival rate of new connections to the network is varied. For comparison, we also consider the original LLR method in which effective bandwidths are computed using Guerin et al.'s approximation [20]. For each set of parameters the simulation is run for about 10^5 VC arrivals to the system and the initial 10 % results are discarded to account for transient effects.

In Figure 6.9 and Figure 6.10, we plot the call blocking probability as a function of the arrival rate for buffer sizes of 1.5 Mb and 5 Mb, respectively. It is clear that the modified LLR (denoted by NN-LLR) performs better than the conventional LLR(denoted by LLR). We can compare the performance gain of the NN-LLR method over the LLR method by averaging over all the arrival rates we considered. The NN-LLR method accepts about 50% and 17% more calls than the LLR method for the buffer sizes of 1.5 Mb and 5 Mb, respectively. We also note that as the buffer size increases, the gain of NN-LLR over the LLR method decreases. This is not surprising since the effective bandwidth approximation employed in the LLR method approaches optimality as the buffer size tends toward infinity.

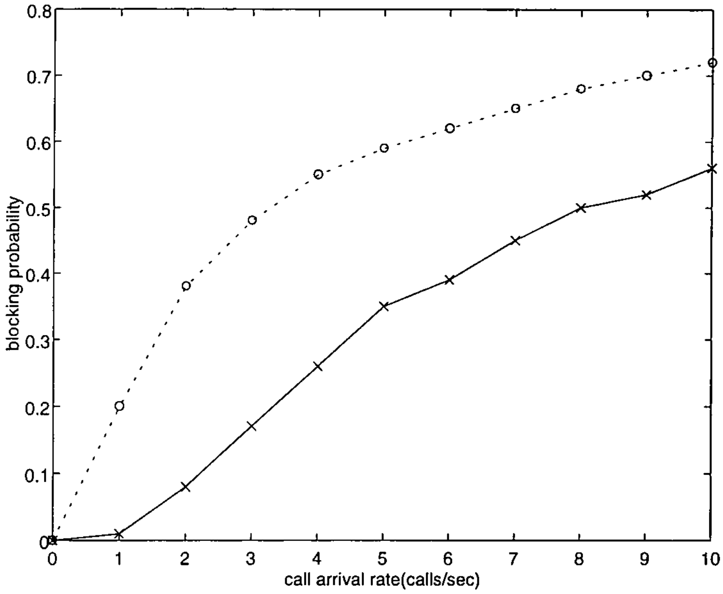

Figure 6.9: Call blocking probability vs. call arrival rate, 1.5 Mb buffer. Solid line: NN-LLR; dotted line: LLR

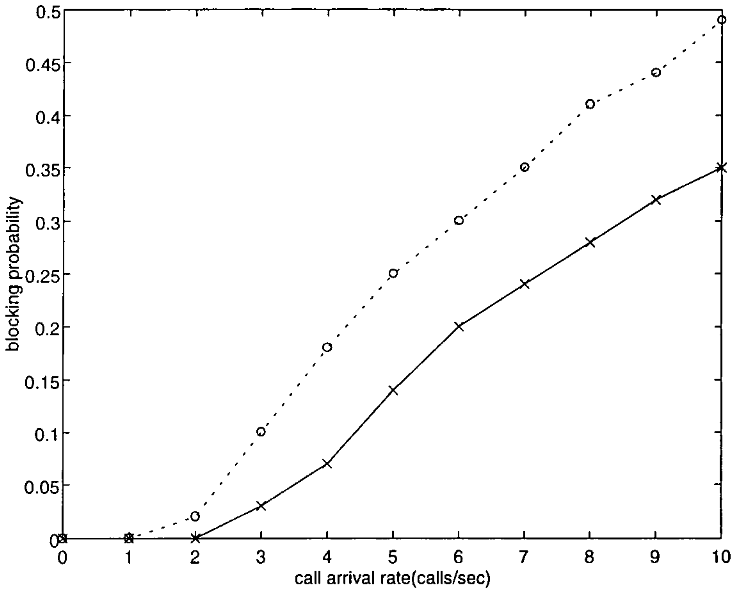

Figure 6.10: Call blocking probability vs. call arrival rate, 5 Mb buffer. Solid line: NN-LLR; dotted line: LLR

6.7 Summary

In this chapter a novel neural network approach for effective bandwidth estimation is presented. It is accurate enough for applications to connection admission control, bandwidth management and dynamic routing schemes. This is justified by the fact that multilayer feedforward networks are a class of universal approximators and suitable for nonlinear regression. In the mean time, the neural network approach can meet the real-time requirements of ATM networks due to its high computation rate. Simulations show that the results of our method are very close to those of the exact stochastic fluid model.

As mentioned previously, it is crucial that QOS or bandwidth estimation mechanisms should provide *accurate* and *timely* information to the CAC function. One of the main limitations of the NN approach to bandwidth estimation lies in obtaining the training data. The data can be collected from the solution of an existing analytical model or by running extensive simulations. This either involves very long computation time(off-line) or may be restricted to a specific traffic model, which needs further improvement. On the other hand, when we consider more realistic traffic instead of traffic generated by a parametric model, the architecture of the neural network becomes very complicated and its training is not a trivial task. Due to the above limitations, in the following chapters, we try to seek other options for performing call admission control to satisfy both accuracy and efficiency requirements. These methods make some modifications to the original effective bandwidth approximation and are shown to be more accurate for use in CAC.

Chapter 7

A Novel Effective Bandwidth Approach to CAC

7.1 Introduction

In ATM networks, connection admission control can be regarded as the most important of the preventive control functions which aim at restraining congestion in the network nodes. CAC is defined as a set of actions taken by the network during the call setup or a call renegotiation phase to establish a connection without degrading network performance. At connection setup, a route through the network is selected. Then, the QOS of each affected link is estimated, taking the effect of the new connection into consideration. The connection request is accepted if each link can offer sufficient QOS to all connections. According to [76], admission control methods must address three major issues:

- What parameters are needed to accurately describe the traffic generated by a connection? The parameters that are used to describe a connection must be complete enough to allow the admission control method to accurately predict the effect of the newly admitted connection on network performance. However, the set of parameters should be as small as possible in order to limit the computational complexity, latency and other resources needed for CAC. It is also crucial to provide an accurate characterization of a connection's burstiness.
- How does CAC decide whether or not admit a new connection? The network must guarantee some QOS level to each connection that it admits. The two most important QOS measures are cell delay and cell loss probability. Both are closely related to the level of network congestion. In this chapter we focus only on the cell loss probability assuming that delay requirements can be met by restricting the buffer size.

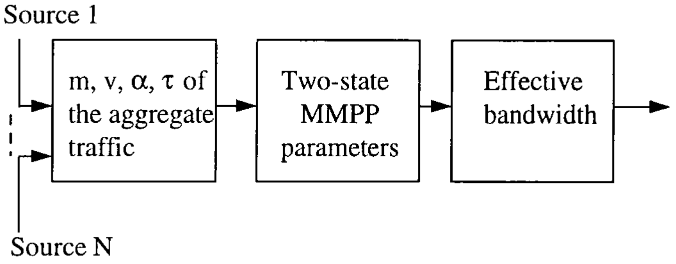
Source 1
m, v, α, τ of the aggregate traffic
Two-state MMPP parameters
Effective bandwidth
Source N

Figure 7.1: The proposed method for estimating effective bandwidth of the aggregate traffic

- How does the network performance depend on the traffic parameters? The network performance should be a function of the traffic parameters that are chosen to characterize each connection.

Many CAC schemes have been proposed, as reviewed in previous chapters. They vary widely in terms of computational complexity, computational accuracy, traffic models and parameters, memory usage, and in other ways.

Typical CAC methods employs the concept of effective bandwidth. The use of effective bandwidth simplifies the CAC procedure by estimating the total bandwidth of the aggregate traffic(including the new connection). The new call is accepted only if the updated total effective bandwidth is less than the link capacity. In the last chapter, we describe a CAC approach which uses NNs for effective bandwidth estimation. However, in this chapter, we attack the problem from a different angle. Considering statistical multiplexing between traffic sources, we directly calculate the effective bandwidth of the aggregate traffic rather than summing up individual bandwidths, hence overcoming the drawback of conventional methods. The aggregate arrival traffic is characterized by four appropriately selected parameters and then accurately modelled by a two-state Markov Modulated Poisson Process(MMPP) via matching four important statistics. If the buffer size is large, admission control can be achieved by computing the effective bandwidth of the two-state MMPP. Figure 7.1 shows a block diagram of the proposed method. Simulation tests show that our approach is simple and results in higher utilisation compared with conventional methods.

7.2 The Model

For traffic control mechanisms, we must define a unique set of parameters which can describe a wide range of service characteristics. Here we adopt the four parameters given in [97] because they can be easily evaluated and monitored by traffic control

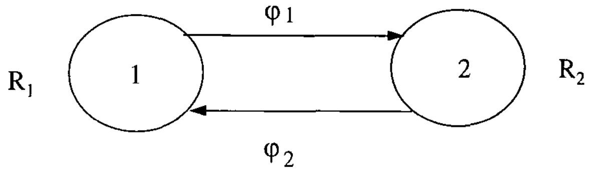

Figure 7.2: A two-state MMPP model

units. They are: the mean arrival rate m, the peak-to-mean ratio α, the autocovariance time coefficient τ, and the variance v. Parameters m, α, τ, v are all connected with the burstiness of the source, especially α describing the peak rate, and τ being related to the burst length. Parameter v governs the probability distribution of a source rate with fixed m and α. As an example, we consider the case for an ON/OFF source. As described previously, an ON/OFF source model turns ON with an exponential rate λ and OFF with a rate μ. In the ON state, the model generates cells at a constant rate r, and in the OFF state no cells are generated. Then we have

$$m = r\lambda/(\lambda+\mu), \qquad v = m(r-m), \qquad \tau = (\lambda+\mu)^{-1}, \qquad \alpha = (\lambda+\mu)/\lambda \tag{7.1}$$

The aggregate traffic from N independent sources can also be represented by the four parameters m, α, τ, v. The relationship between the aggregate traffic parameters and the individual ones are:

$$m_a = \sum_{i=1}^{N} m_i, \qquad v_a = \sum_{i=1}^{N} v_i, \qquad \tau_a = \sum_{i=1}^{N} v_i\tau_i/v_a, \qquad \alpha_a = \sum_{i=1}^{N} m_i\alpha_i/m_a \tag{7.2}$$

From the above equation, we observe that all these statistics are of direct incremental or additive nature, which is important since connection requests arrive sequentially.

Next we approximate the aggregate traffic by a two-state MMPP because of its versatility and simplicity. The MMPP is a doubly stochastic Poisson process where the two states of a continuous-time Markov chain correspond to two Poisson processes. An MMPP model can be completely characterized by four parameters, R_1, R_2, φ_1 and φ_2, as shown in Figure 7.2. Parameter $R_j, j = 1, 2$, is the mean rate of a Poisson process in state j, and the state duration time at each state has an exponential distribution with mean φ_j^{-1}. The MMPP models can accurately characterize the aggregate arrival process because a large number of statistics can be matched and the correlation among the arrival rates can be captured over large time intervals. In [67], Heffes and Lucantoni use an MMPP to successfully model average delay of voice packets through an infinite buffer multiplexer. In [88], different sets of MMPP parameters are used to model the superposition of ON/OFF sources and packet loss in finite-buffered multiplexers. It has

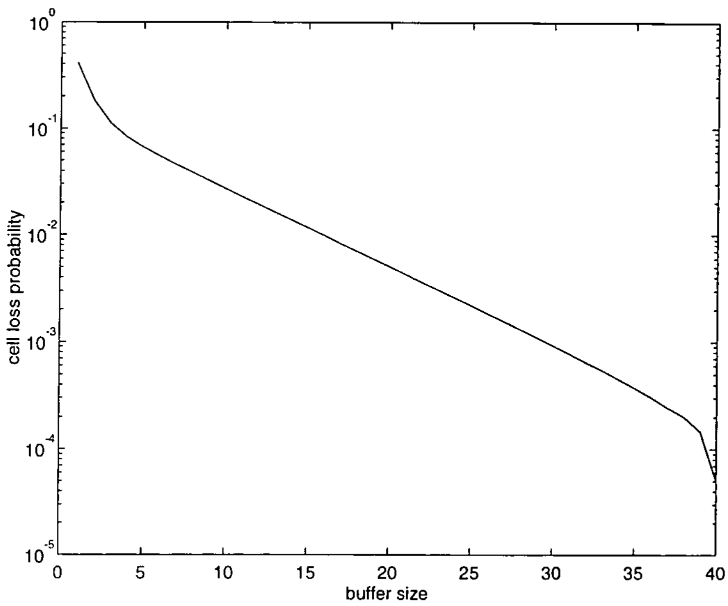

Figure 7.3: Cell loss probability vs. buffer size with traffic intensity $= 0.6, v = 0.1, \tau = 10.0, \alpha = 5.0$

been shown that the approaches based on MMPPs are many orders of magnitude better than modeling the superimposed sources simply as a Poisson process.

In [67], Heffes and Lucantoni introduce a scheme to determine the MMPP parameters by matching them to the statistical moments of real traffic. However, the resulting MMPP model turns out to underestimate the queueing performance when the arrival traffic consists of bursty sources with high peak rate such as compressed video. So in [97], a new scheme is proposed to evaluate the MMPP parameters based on the four traffic parameters of the aggregate traffic, i.e., m_a, v_a, τ_a and α_a. The MMPP parameters $R_j, \varphi_j (j = 1, 2)$ are given by:

$$R_1 = m_a + \sqrt{\alpha_a v_a}, \qquad R_2 = m_a - \sqrt{v_a/\alpha_a}, \qquad \varphi_1 = q/\tau_a, \qquad \varphi_2 = (1-q)/\tau_a \quad (7.3)$$

where $q = \alpha_a/(1+\alpha_a)$. The simulation results in [97] show that the performance of the proposed model appears as good as the Heffes and Lucantoni's one for the multiplexed voice sources, and is much better for the integrated sources including video signals. Therefore, the MMPP model derived by this approach is a more suitable model for the integrated traffic of BISDN.

Given the above MMPP model as the input process, let's consider an ATM statistical multiplexer with deterministic service rate and finite capacity. Hence, this multiplexer can be modeled as an MMPP/D/1/K queue. The details of the analysis of the MMPP/D/1/K queue are given in Appendix C, which leads to the evaluation of the cell

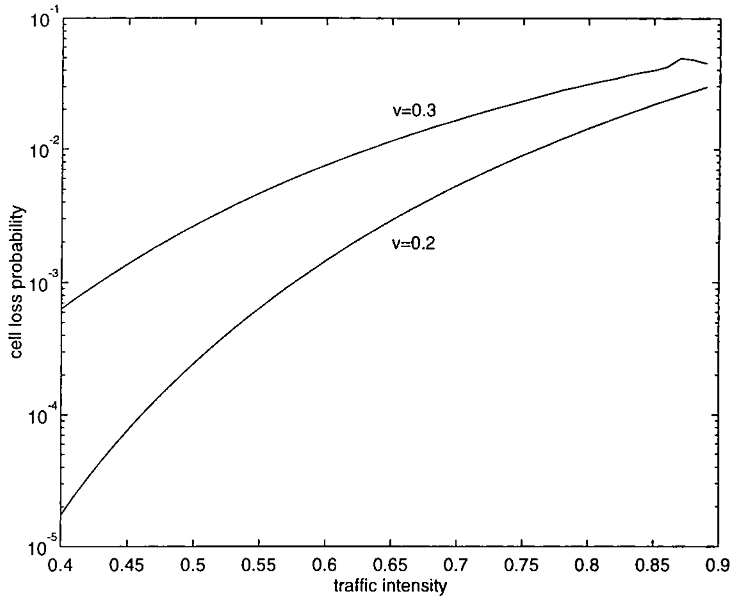

Figure 7.4: Cell loss probability vs. traffic intensity for different values of variance, with $\alpha = 2.0, \tau = 10.0$

loss probability. The effect of buffer size on the cell loss probability is shown in Figure 7.3. It is observed, as expected, that the cell loss probability is inversely proportional to buffer size. In Figure 7.4, Figure 7.5 and Figure 7.6, we show the cell loss probabilities as a function of traffic intensity. A buffer with finite capacity K of 32 cells is selected. We can observe that the traffic parameters v, α and τ have a strong influence on the queueing behaviour.

7.3 The Proposed CAC Scheme

Although the above model gives elegant formulas for computing cell loss probability in the MMPP/D/1/K queue, the applicability of this method seems limited to off-line buffer dimensioning(due to its computational complexity) rather than on-line performance evaluation(e.g., in CAC for determining if a specified QOS can be satisfied). Actually, in order to use the analytical result of the MMPP/D/1/K queue, the authors of [97] employ the effective bandwidth concept along with table-lookup procedure to perform CAC. The problem with this approach is that a huge table has to be maintained and it is not flexible enough to accommodate new services and possible network changes. It also becomes computationally infeasible for large buffer sizes. Therefore, in this section, we propose a novel effective bandwidth approach to perform real-time CAC.

The effective bandwidth of a time-varying source is the minimum amount of band-

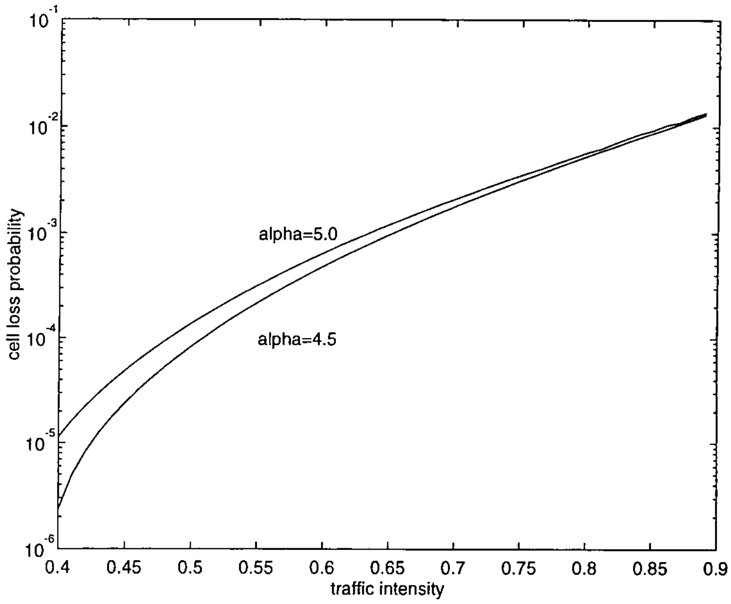

Figure 7.5: Cell loss probability vs. traffic intensity for different values of peak-to-mean ratio, with $v = 0.1, \tau = 10.0$

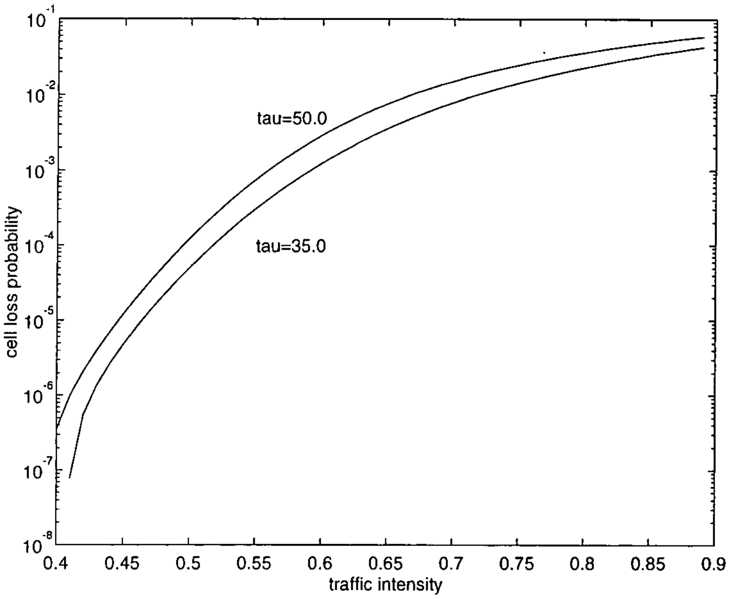

Figure 7.6: Cell loss probability vs. traffic intensity for different values of autocovariance time coefficient, with $v = 0.1, \alpha = 2.0$

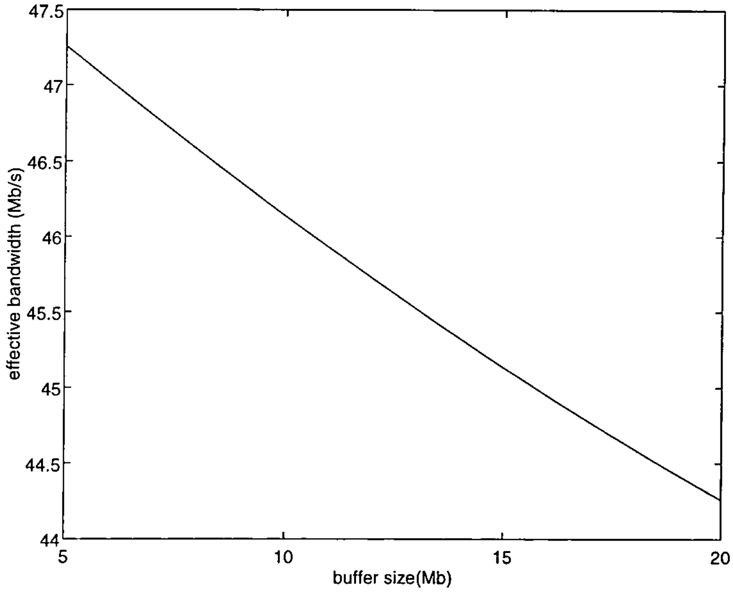

Figure 7.7: Effective bandwidth vs. buffer size

width required to satisfy its QOS. The call admission problem is then solved by checking whether the effective bandwidth of the aggregate user population, including the new user, exceeds the service capacity. This value should be easily computable so that online computations may be carried out. Note that the notion of effective bandwidth is based on large deviations asymptotics for the tail probabilities of large buffer queues, which will be discussed in detail in the next chapter. Here we take the buffer overflow probability in an infinite buffer as an approximation to the cell loss ratio(QOS) in a finite buffer. For a very small overflow probability of ϵ and a large buffer size of B, we define θ as $\theta = -\log(\epsilon)/B$. Then the effective bandwidth e of the two-state MMPP can be calculated as [98]

$$e = \frac{1}{2\theta}(-a(\theta) + \sqrt{a^2(\theta) - 4b(\theta)}) \tag{7.4}$$

where $a(\theta) = \varphi_1 + \varphi_2 - (e^\theta - 1)(R_1 + R_2)$ and $b(\theta) = (e^\theta - 1)^2 R_1 R_2 - (e^\theta - 1)(\varphi_1 R_2 + \varphi_2 R_1)$.

Figure 7.7 shows the effective bandwidth for 10 video sources(whose model will be described below) versus buffer size, with $\epsilon = 10^{-5}$. It is clear that effective bandwidth is a monotonically decreasing function of buffer size. Figure 7.8 shows the effective bandwidth per video source versus the number of multiplexed sources. In this case, $\epsilon = 10^{-5}$ and the buffer size is 5 Mb. In the figure we include both our novel effective bandwidth calculation method and a conventional one proposed by Guerin et al. [20], which simply sums up individual effective bandwidths to estimate the total bandwidth required by the aggregate traffic. Obviously, statistical multiplexing gain is achieved in

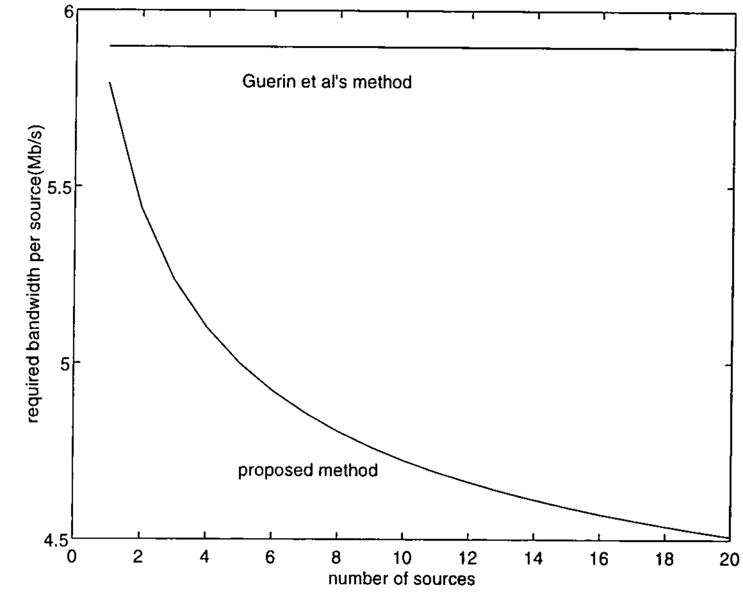

Figure 7.8: Effective bandwidth per source vs. number of multiplexed sources

the proposed method.

The proposed CAC scheme is summarized as follows. We represent the aggregate traffic(including the new call request) using four parameters, m, v, α, and τ. Then we obtain a two-state MMPP via matching the four parameters. Finally the effective bandwidth of the two-state MMPP is calculated according to (7.4). The new call is admitted if and only if the total effective bandwidth of all calls is less than the link capacity.

7.4 Simulation Results

We have compared the efficiency of our approach with three other methods, i.e., peak rate allocation, Gaussian approximation and sum of individual effective bandwidths. The peak rate allocation scheme is the simplest one and it accepts or rejects calls on the basis of their peak bit rates. In this scheme, the QOS is always guaranteed because the aggregate bit rate will never exceed the link rate of the system. However, it leads to low utilisation of network resources. The Gaussian approximation of the required bandwidth is based on the assumption that the distribution of the stationary bit rate is Gaussian and is given by [20]:

$$c_g = m_a + \beta\sigma_a \tag{7.5}$$

where m_a and σ_a represent the mean and standard deviation of the total arrival rate, respectively, and $\beta = \sqrt{-2\log(\epsilon) - \log(2\pi)}$. For N fluid flow ON/OFF sources, the sum

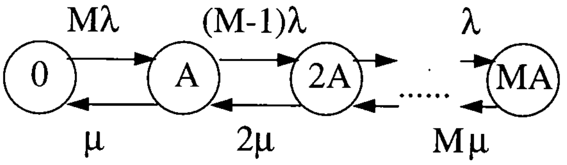

Figure 7.9: State transition diagram of the birth-death process

of individual effective bandwidths is defined as [20]:

$$c_s = \sum_{i=1}^{N} c_i = \frac{1}{2\theta} \sum_{i=1}^{N} (-\lambda_i - \mu_i + \theta r_i + \sqrt{(\lambda_i + \mu_i - \theta r_i)^2 + 4\theta \lambda_i r_i}) \tag{7.6}$$

where $1/\lambda_i$ and $1/\mu_i$ are the mean OFF and ON periods of source i, respectively, and r_i is the corresponding peak rate.

We consider a single server queue(equivalent to an ATM link) with a deterministic service rate of 150 Mb/s. The buffer size is taken to be 5 Mb and the desired overflow probability is set at 10^{-5}. Two types of traffic sources are used. The first type represents a model for voice calls and the second for video telephony. A voice source can be characterized by the ON/OFF model we mentioned earlier [88]. In [75], Maglaris et al. consider the problem of modelling videotelephone scenes with a uniform activity level, e.g., showing a person talking. In their model, the arrival rate $r(t)$ is quantized into finite discrete levels. Transitions between levels are assumed to occur with exponential transition rates that may depend on the current level. Thus, the rate variations over time are approximated by a continuous-time process with discrete jumps at random Poisson times. This model is a discrete finite-state, continuous-time Markov process. Its state space is the set of the quantized levels up to a maximum level. This model is used in [75] to analyze the statistical multiplexer queue as a fluid flow reservoir that is filled from N variable rate sources each with rate $r(t)$.

It has been shown a birth-death Markov model as shown in Figure 7.9 will accurately describe the aggregate video source bit rate. The rate $r_N(t)$ of the process in Figure 7.9 represents the quantized level of the aggregate bit rate of N sources. We assume uniform quantization step A bits/pixel, and $M+1$ possible levels, $\{0, A, ..., MA\}$. It is easy to see that the rate $r_N(t)$ can be thought of as the aggregate rate from M independent minisources, each alternating between transmitting 0 bits/pixel (the OFF state) and A bits/pixel (the ON state) according to a Bernoulli distribution. A minisource turns ON with exponential rate λ and OFF with rate μ. Thus this model is almost the same as that used in analyzing statistical multiplexing of M voice sources. The quantization step, the number of states, and the transition rates can be tuned to fit the mean, variance and

Traffic source	M	$\lambda(1/s)$	$\mu(1/s)$	r(Mb/s)
voice	1	1/0.65	1/0.352	0.064
video	10	1.3078	2.5922	1.163

Table 7.1: Model parameters of ON/OFF sources

Proposed scheme	Gaussian approximation	Sum of individual effective bandwidths	Peak rate allocation
0.90	0.81	0.82	0.34

Table 7.2: Comparison of utilisation

autocovariance function of the measured data. The results are:

$$A = \frac{C_N(0)}{E(r_N)} + \frac{E(r_N)}{M} \tag{7.7}$$

$$\mu = 3.9/(1 + \frac{E^2(r_N)}{MC_N(0)}) \tag{7.8}$$

$$\lambda = 3.9 - \mu \tag{7.9}$$

where $E(r_N)$ and $C_N(0)$ are the average and the variance of the aggregate arrival process from N identical and independent sources. They are given by $E(r_N) = 0.52N$ bits/pixel and $C_N(0) = 0.0536N(\text{bits/pixel})^2$, respectively. Since there are about 250000 pixels per frame and 30 frames/s, 1 bit/pixel corresponds to 7.5 Mbits/s. The number of states M is set to be $10N$. It is found that this value of M yields reasonable results that are close to the measured data.

The parameter values used for simulation are summarized in Table 7.1. In the table, we denote by M the number of ON/OFF sources required for characterizing one traffic source.

In Figure 7.10, we plot the acceptable numbers of voice sources and video sources. It is clear that the proposed approach can accept more calls than other methods, while all cases meet the QOS requirement. This is because the Gaussian approximation assumes zero buffer and the conventional effective bandwidth approach ignores the statistical multiplexing effect between multiple sources. As expected, the peak rate allocation yields the poorest performance. Table 7.2 compares different utilisations obtained by four CAC schemes. We see that our CAC method can increase revenues by at least 10% over the other schemes.

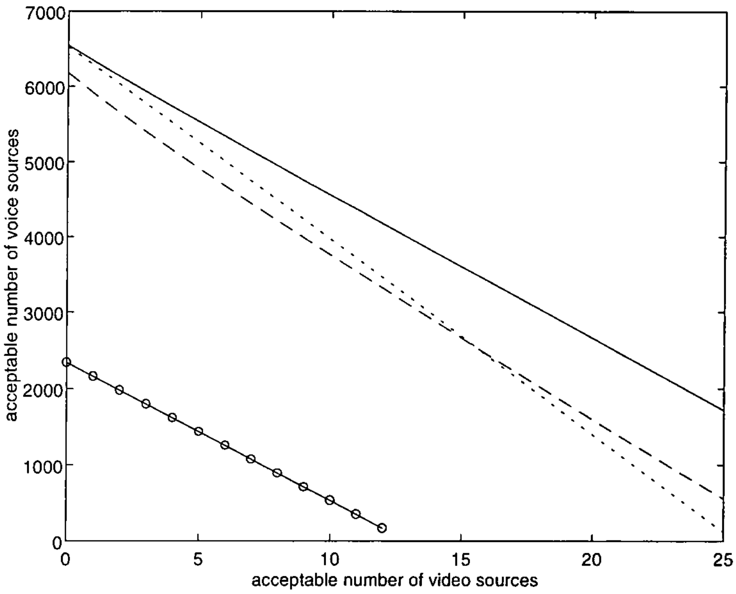

Figure 7.10: Comparison of acceptable number of sources by different CAC approaches. Solid line: proposed scheme; dashed line: Gaussian approximation; dotted line: sum of individual effective bandwidths; circled line: peak rate allocation

7.5 Summary

The CAC scheme proposed in this chapter is based on the concept of effective bandwidth. In contrast to conventional methods, we model the aggregate traffic from heterogeneous sources as a two-state MMPP by a novel matching technique and then estimate the required bandwidth. Since this approach takes the statistical multiplexing effect into account, it can lead to higher resource utilisation.

The theory of effective bandwidth is an active area of research and is being applied to the design, simulation and analysis of high-speed ATM networks. In the next chapter, we address this topic and related issues in more detail. We use the theory of large deviations to provide a unified description of effective bandwidths for various traffic sources and the associated ATM multiplexer queueing performance approximations, illustrating their strengths and limitations. On the basis of this discussion, we propose a more accurate estimation method for ATM QOS parameters, which is a refinement of the original effective bandwidth approximation.

Chapter 8

Fast and Accurate Estimation of ATM QOS Parameters

8.1 Introduction

The success of the future BISDN depends heavily on the ability to perform statistical multiplexing of VBR video, voice and data traffic. To make such a multiplexing scheme feasible, it is necessary to study traffic control procedures enabling the network to meet various quality of service constraints defined for different services. The design of such traffic controls relies on a sound understanding of the way network performance is related to the characteristics of the offered traffic streams and network capacities.

Except for very simple source models(e.g., Poisson or Markov modulated process with few states), it is difficult to analyze the QOS parameters in an ATM multiplexer exactly. The difficulty is due to the complexity of the large state space when the number of sources or the buffer size is large. On the other hand, using simulation modelling techniques to obtain QOS values often involves unacceptably long computer run time. Therefore people turn to asymptotic analyses. Recently there has been a flood of literature on the tail behaviour of the queue length distribution using the theory of large deviations. One of the most important results is that $G(B) = \Pr(Q > B) \approx e^{-\theta^* B}$, where Q represents the stationary buffer occupancy and $G(B)$ is the overflow probability of a buffer of size B. This result leads to the notion of effective bandwidth, which is a prevailing technique in ATM traffic modeling and control. In [99], it is argued that for the purposes of estimating QOS parameters(e.g., cell loss ratio and mean cell delay), it is enough to know the large deviation rate function of the ATM traffic stream; the modeling procedure can be by-passed if we can estimate the rate function directly. Actually, the large deviation rate function is the same kind of mathematical object as the well known entropy function in equilibrium thermodynamics. Some preliminary experiments have also been conducted on a real network to verify the above statement [100]. In [101], a general logarithmic equivalent is given for the stationary complementary distribution function of a fluid queue fed by a large number of ON/OFF sources. Another approach based on large deviations for empirical distributions of Markov chains is devised by Courcoubetis et al. [102] to estimate the loss probability in switch buffers. In [103], Duffield and O'Connell obtain more general results which can be applied to traffic streams with long range dependence, or self-similar structure.

Although the CAC algorithm based on effective bandwidths is simple and fast, it ignores the effect of statistical multiplexing of large numbers of sources and hence it is a too conservative approach resulting in under-utilisation of network resources. In the previous chapter we consider a modified effective bandwidth method for CAC, which is based on the mapping of heterogeneous traffic to a two-state MMPP. In this chapter, firstly, we investigate the effective bandwidth method from a more general point of view using large deviations techniques. Secondly, we propose a more accurate approximation for ATM multiplexer queueing performance. We demonstrate how fast and accurate estimates of QOS parameters can be obtained for an ATM multiplexer queueing model fed by heterogeneous Markovian traffic sources. We achieve this by constructing a simple approximation of the buffer overflow probability: $\Pr(Q > B) \approx De^{-\theta^* B}$, where θ^* is the asymptotic decay rate of the tail of the distribution, and D is a prefactor obtained from the Bahadur-Rao theorem. Upper bounds on other QOS parameters, such as cell loss ratio, mean cell delay and cell delay variance, can all be derived from this estimate. It has been shown that this approximation is much more accurate than the pure effective bandwidth one $\Pr(Q > B) \approx e^{-\theta^* B}$. One attraction of the proposed method is that the speed of computation of D and θ^* is independent of the size of the system. Even for large scale systems the time required to compute D and θ^* is trivial compared to the conventional simulations or theoretical analyses. Thus our analytical techniques can be implemented fast enough for real-time administration of admission control in ATM networks. Some numerical results are presented to verify the accuracy of the proposed approach. A brief introduction to the theory of large deviations is given in Appendix D.

8.2 The Effective Bandwidth Approximation

In ATM networks, since different classes of traffic usually require different grade of service, a challenging problem is to design schemes that integrate these different classes of traffic efficiently. In order for streams to share resources one must guard against traffic fluctuations by inserting buffers. To ease the task of managing such a network it is desirable to obtain a circuit-switched model for which relatively simple call admission, routing, and network planning algorithms are available. One of the most interesting approaches in dealing with this problem is the recently developed theory of effective bandwidth. Briefly, it has been noted that it is possible to associate an easily calculated quantity with each source of data, referred to as the effective bandwidth of that source, that captures the behaviour of the tail of the response time at a multiplexer. The call admission problem is then solved by checking whether the effective bandwidth of the aggregate user population, including the new user, exceeds the service capacity. In [104], Kelly discusses effective bandwidths for both M/GI/1 and D/GI/1 queues subject to either mean delay or tail constraints. In [92], Gibbens and Hunt consider heterogeneous ON/OFF fluid sources which alternate between exponentially distributed periods of transmission at the peak rate and quiescence. Guerin et al. [20] independently obtain similar results through insightful heuristics. The general framework of the theory, including the computation(or approximation) of the effective bandwidth for Markov processes and other general processes and the associated calculus, is carried out in [105] [98] [91] [106] (For a historical review, see [106]). Tse et al. [107] study effective bandwidths for multiple time-scale Markov streams using large deviations theory. Further development of the theory for traffic filtering, resource management, fast simulation of ATM intree networks and other applications can be found in [108] [109] [110], among many others.

8.2.1 General Effective Bandwidths

Now we use the theory of large deviations to provide a brief overview of effective bandwidths for ATM networks. The large deviations approach used is a unified framework to handle buffer sources modeled in different ways. Also we will show that there is an intimate connection between the behaviour of probabilites of queue lengths and effective bandwidth. Consider an ATM multiplexer with constant service rate C and infinite buffer capacity [1]. Let $a(t)$ and $q(t)$ be the number of cells arriving at time t and the number of cells in the queue at time t respectively. Under a work-conserving policy, we have the following Lindley's equation:

$$q(t+1) = (q(t) + a(t+1) - C)^+ \tag{8.1}$$

where $(x)^+ = \max(0, x)$. For a high-speed ATM network, the unit of time is small, e.g., in the order of microseconds or smaller. Thus, if we view the arrival process as a continuous fluid flow with a rate process $a(t)$, we can approximate the discrete-time equation (8.1) by a continuous-time version:

$$dq(t)/dt = \begin{cases} a(t) - C, & \text{if } a(t) - C > 0 \text{ or } q(t) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8.2}$$

[1] The overflow probability $\Pr(Q > B)$ in an infinite buffer queueing system is often used to approximate the loss probability in the corresponding finite buffer system with buffer size B.

We can then assume that the source behaves as a constant rate fluid with rate α for a period of time t with probability density function $f(\alpha; t)$. Thus,

$$\Pr(q(t) \geq x) = \int_{(\alpha-C)^{+}t \geq x} f(\alpha; t) d\alpha \tag{8.3}$$

Let $A(t_1, t_2) = \sum_{t=t_1}^{t_2-1} a(t)$ be the number of arrivals in $[t_1, t_2)$. Assume that $A(0, t)$ satisfies the conditions of the Gartner-Ellis theorem [111]. That is, assume that the asymptotic log moment generating function of $A(0, t)$,

$$\Lambda(\theta) = \lim_{t\to\infty} \frac{1}{t} \log \mathrm{E}e^{\theta A(0,t)} \tag{8.4}$$

exists for all real θ, and that $\Lambda(\theta)$ is differentiable and convex. The Legendre transform of $\Lambda(\theta)$ is given by

$$\Lambda^*(\alpha) = \sup_{\theta}(\theta\alpha - \Lambda(\theta)) \tag{8.5}$$

Since Λ and Λ^* are convex conjugates or Legendre transform pair, $\Lambda(\theta) = \sup_{\alpha}(\theta\alpha - \Lambda^*(\alpha))$. It then follows from the differentiability of Λ and Λ^* that

$$\Lambda^*(\Lambda'(\theta)) = \theta\Lambda'(\theta) - \Lambda(\theta) \tag{8.6}$$

$$\Lambda(\Lambda^{*'}(\alpha)) = \alpha\Lambda^{*'}(\alpha) - \Lambda^*(\alpha) \tag{8.7}$$

In thermodynamics, $\Lambda(\theta)$ and $\Lambda^*(\alpha)$ are called the "energy" function and the "entropy" function respectively. In view of the Legendre transform and the fact that $\Lambda(0) = 0$, $\Lambda^*(\alpha)$ is nonnegative, and strictly convex, and it has a global minimum at $\alpha_0 = \Lambda'(0)$ such that $\Lambda^*(\alpha_0) = 0$. Actually α_0 is the mean rate of the source.

It can be shown that $f(\alpha; t)$ has the form of Gibb's distribution [112]:

$$f(\alpha; t) \approx e^{-t\Lambda^*(\alpha)} \tag{8.8}$$

So

$$\Pr(q(t) \geq x) \approx \int_{(\alpha-C)^{+}t \geq x} e^{-t\Lambda^*(\alpha)} d\alpha \tag{8.9}$$

If we further assume that $\{q(t), t \geq 0\}$ converges in distribution to a finite random variable $Q = q(\infty)$, then $\sup_t \Pr(q(t) \geq x) = \Pr(Q \geq x)$. After some manipulations, we have

$$\Pr(Q \geq x) \approx \exp(-x \inf_{\alpha} \frac{\Lambda^*(\alpha)}{(\alpha - C)^{+}}) \tag{8.10}$$

From the definition of $\Lambda^*(\alpha)$ (8.5), $\Lambda^*(\alpha)/(\alpha - C) \geq (\theta\alpha - \Lambda(\theta))/(\alpha - C)$. If θ^* is a solution of $\Lambda(\theta)/\theta = C$, then

$$\inf_{\alpha > C} \frac{\Lambda^*(\alpha)}{\alpha - C} \geq \theta^* \tag{8.11}$$

Also, if the solution of $\Lambda(\theta)/\theta = C$ is unique, then it follows from (8.6) that

$$\Lambda^*(\Lambda'(\theta^*)) = \theta^*\Lambda'(\theta^*) - \Lambda(\theta^*) = \theta^*\Lambda'(\theta^*) - C\theta^* \tag{8.12}$$

Notice that $\Lambda'(\theta^*) > C$. So the bound in (8.11) is achieved when $\alpha = \Lambda'(\theta^*)$ and we have

$$\Pr(Q \geq x) \approx \exp(-\theta^* x) \tag{8.13}$$

Since C is the capacity of a link or a switch, the function

$$e(\theta) = \Lambda(\theta)/\theta \tag{8.14}$$

is called the effective bandwidth function of the arrival process subject to the condition that the tail distribution of the queue length has the decay rate θ. The effective bandwidth is a nondecreasing function in θ, with the mean rate of the source $e(0)$ and the peak rate being $e(\infty)$.

The above discussion can be extended to the case of multiclass traffic sources, in which the arrival stream consists of J types of traffic, with N_j sources of type j, all having a Markovian statistical nature. Let $A_j(0,t)$ denote the number of arrivals from each type j stream over the time interval $[0,t)$. Assume that the asymptotic log moment generating function of $A_j(0,t)$,

$$\Lambda_j(\theta) = \lim_{t\to\infty} \frac{1}{t} \log \mathrm{E}e^{\theta A_j(0,t)} \tag{8.15}$$

exists for all real θ, and that $\Lambda_j(\theta)$ is differentiable and convex. Assuming the steady state queue length is Q, then, by large deviations theory, we have (For full proof, see [105] [98]):

$$\Pr(Q > B) \approx \exp(-\theta^* B), \tag{8.16}$$

where θ^* is the unique positive solution of $\sum_{j\in J} N_j\Lambda_j(\theta)/\theta = C$. In a more rigorous way, the following result holds:

$$\sum_{j\in J} N_j\Lambda_j(\theta)/\theta \leq C, \Longleftrightarrow \lim_{B\to\infty} \frac{1}{B} \log \Pr(Q > B) \leq -\theta.$$

8.2.2 Effective Bandwidths for Markovian Sources

We now give expressions for the effective bandwidths for some popular Markov sources used to characterize bursty ATM traffic. These formulas will be used later in this chapter.

Discrete-Time Markov Sources

Let Z_t be a discrete-time, finite state, irreducible, stationary Markov chain with state space $\mathcal{S} = \{1, 2, ..., m\}$, and let $\mathbf{P}$ be its transition matrix, i.e., $p_{i,j}$ is the transition probability from state i to state j. The arrival stream X_t is modulated by the Markov chain Z_t, such that the distribution of X_t at time t depends only on the source state Z_t at time t, and given a realization of the chain Z_t, the X_t's are independent. Clearly, X_t is stationary since Z_t is stationary. The source state Z_t can be thought of as modeling the burstiness of the stream at time t; the Markov structure models the correlation in the cell arrival statistics over time. For stability, we assume that the average number of cells arriving per time slot is less than the channel capacity.

Let $g_i(\theta) = \mathrm{E}(\exp(\theta X_t)|Z_t = i)$ be the moment generating function (which we assume to exist and be differentiable for all θ) of the conditional distribution of X_t given the source state $Z_t = i$. Consider the matrix $\mathbf{A}(\theta)$ whose entries are $a_{i,j} = p_{i,j} g_i(\theta), i, j \in \mathcal{S}$. Since the given chain is irreducible, the matrix $\mathbf{A}(\theta)$ is also irreducible for any θ. By the Perron-Frobenius theorem [113], the matrix $\mathbf{A}(\theta)$ has a largest positive simple eigenvalue $\rho(\mathbf{A})$ (the spectral radius of $\mathbf{A}(\theta)$). Chang [105] has shown by using a backward equation approach that the asymptotic log moment generating function of the arrival process is equal to the log spectral radius function of $\mathbf{A}(\theta)$, i.e., $\Lambda(\theta) = \log \rho(\mathbf{A})$. $\Lambda(\theta)$ is convex and differentiable for all real θ, and $\Lambda(0) = 0, \Lambda'(0) = \mathrm{E}(X_1)$. Then the effective bandwidth is given by

$$e(\theta) = \Lambda(\theta)/\theta = \frac{1}{\theta}\log(\rho(\mathbf{A})) \tag{8.17}$$

Consider the following special case that when $Z_t = i$ at a particular time unit, a constant number(R_i) of cells are produced in that time unit. Then

$$e(\theta) = \frac{1}{\theta}\log(\rho(e^{\theta \mathbf{R}}\mathbf{P})) \tag{8.18}$$

where $\mathbf{R} = \mathrm{diag}(R_1, R_2, ..., R_m)$. For a Binary Markov Source(BMS) used in [114] as the voice model, the Markov chain is of the two-state type(m=2) and $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} 1-a & a \\ d & 1-d \end{bmatrix} \tag{8.19}$$

where a is the probability of transition from state 1 to state 2 and d is the probability of the inverse transition. Thus the probability that the source is in state 2 is $p = a/(a+d)$. For bursty(positively correlated) sources, $a + d < 1$. Then by direct calculation, we have

$$e(\theta) = \frac{1}{\theta}\log(\frac{1}{2}(a(\theta) + \sqrt{a^2(\theta) - 4b(\theta)})), \tag{8.20}$$

where $a(\theta) = (1-a)e^{\theta R_1} + (1-d)e^{\theta R_2}$ and $b(\theta) = e^{\theta(R_1+R_2)}(1-a-d)$. Note that (8.20) is the same as equation 13 in [114], which was obtained by a spectral decomposition method.

Markov Modulated Fluids

Among a large number of modelling approaches discussed in the literature, the fluid approximation provides a useful tool for investigating ATM multiplexer congestion occuring at the so-called burst time scale. The activity of each source can then be represented by its instantaneous bit rate and the queue is seen as a reservoir fed by the fluid input of the sources and emptied with constant output rate. Consider a single Markov fluid process with finite state space $\mathcal{S} = \{1, 2, ..., m\}$ and irreducible infinitesimal generator $\mathbf{Q}$. Let $\mathbf{R} = \text{diag}(R_1, R_2, ..., R_m)$, where R_i denotes the rate at which traffic is generated when the source is in state $i \in \mathcal{S}$. By an argument similar to that for discrete-time Markov sources,

$$e(\theta) = \frac{1}{\theta} r(\mathbf{Q} + \theta \mathbf{R}) \tag{8.21}$$

where $r(\mathbf{F})$ is the largest real eigenvalue of the matrix $\mathbf{F}$. This result is the same as that in [91].

If the Markov fluid considered is of the two-state type($m = 2$), then it can be characterized by four parameters λ, μ, R_1 and R_2, i.e.,

$$\mathbf{Q} = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}, \mathbf{R} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \tag{8.22}$$

Obviously, $p = \lambda/(\lambda + \mu)$ is the stationary probability that the source is in state 2.

By direct calculation, we have

$$e(\theta) = \frac{1}{2\theta}(-a(\theta) + \sqrt{a^2(\theta) - 4b(\theta)}) \tag{8.23}$$

where $a(\theta) = \lambda + \mu - \theta(R_1 + R_2)$ and $b(\theta) = \theta^2 R_1 R_2 - \theta(\lambda R_2 + \mu R_1)$. Moreover, $\Lambda^*(\alpha)$ can be derived through the Legendre transform as follows:

$$\Lambda^*(\alpha) = \frac{1}{R_2 - R_1}(\sqrt{\lambda(R_2 - \alpha)} - \sqrt{\mu(\alpha - R_1)})^2 \tag{8.24}$$

The solution of $\Lambda(\theta)/\theta = C, \theta^*(C)$, is given by

$$\theta^*(C) = \frac{(\lambda + \mu)(C - R_{av})}{(C - R_1)(R_2 - C)} \tag{8.25}$$

where $R_{av} = \frac{R_1\mu + R_2\lambda}{\lambda + \mu}$. Note that (8.25) is consistent with the well known result in [85].

In Figure 8.1, Figure 8.2 and Figure 8.3, we plot the functions $\Lambda(\theta) - C\theta, e(\theta)$ and

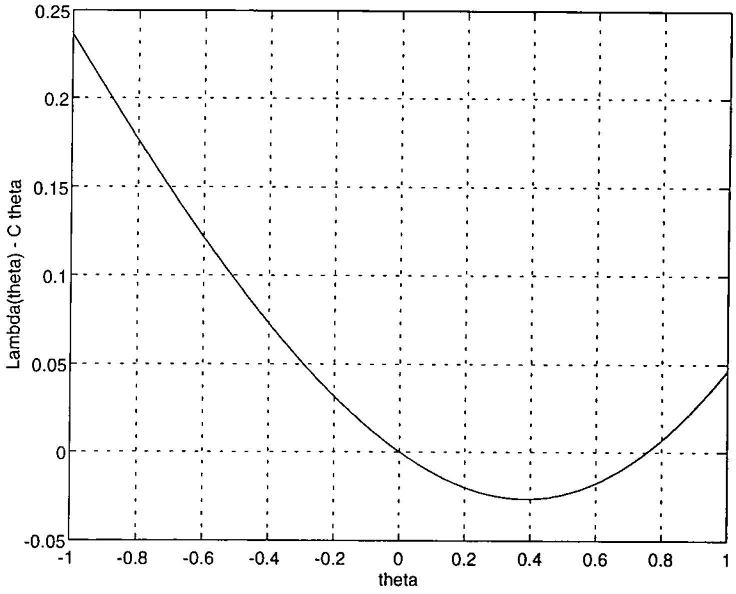

Figure 8.1: The function $\Lambda(\theta) - C\theta$ vs. θ

$\Lambda^*(\alpha)$ respectively for a two-state Markov fluid with $\lambda = 0.4, \mu = 1, R_1 = 0, R_2 = 1.0$ served by an output channel with rate $C = 0.41665$. For this Markov fluid, the average rate $R_{av} = 0.2857$ and the asymptotic decay rate $\theta^* = 0.754$. One can see that $e(\theta), 0 \leq \theta < \infty$, is increasing in θ and ranges between its average rate and its peak rate($R_2 = 1.0$). The function $f(\theta) = \Lambda(\theta) - C\theta$ is convex and its derivative at $\theta = 0$ is negative(which means that the average rate is less than the link capacity). Its derivative as $\theta \to \infty$ is $(R_2 - C)$ and is positive. These properties ensure that the equation $f(\theta) = 0$ has a unique positive solution θ^*. The functions $\Lambda(\theta)$ and $\Lambda^*(\alpha)$ are convex conjugates. The minimum of $\Lambda^*(\alpha)$ is achieved at the average rate. $\Lambda(\alpha)$ is increasing convex in $R_{av} \leq \alpha \leq R_2$ and decreasing convex in $R_1 \leq \alpha \leq R_{av}$.

Markov Modulated Poisson Process(MMPP)

A source to a buffer is called a Markov modulated Poisson process if the cell arrivals are Poisson with intensity $\mathcal{R}$, where $\mathcal{R}$ is a function of a continuous-time Markov chain. Consider an MMPP with arrival rate $\mathcal{R}_{Z(t)}$ at time t, where $Z(t), t \geq 0$ is an irreducible Markov process on $\mathcal{S} = \{1, 2, ..., m\}$ with infinitesimal generator $\mathbf{Q}$. Let $\mathbf{R} = \text{diag}(R_1, ..., R_m)$ denote the rate matrix. By an argument similar to that for discrete-time Markov sources,

$$e(\theta) = \frac{1}{\theta} r(\mathbf{Q} + (e^\theta - 1)\mathbf{R}) \tag{8.26}$$

Now consider a link with time-varying capacity. Denote by $C(t)$ the capacity at time

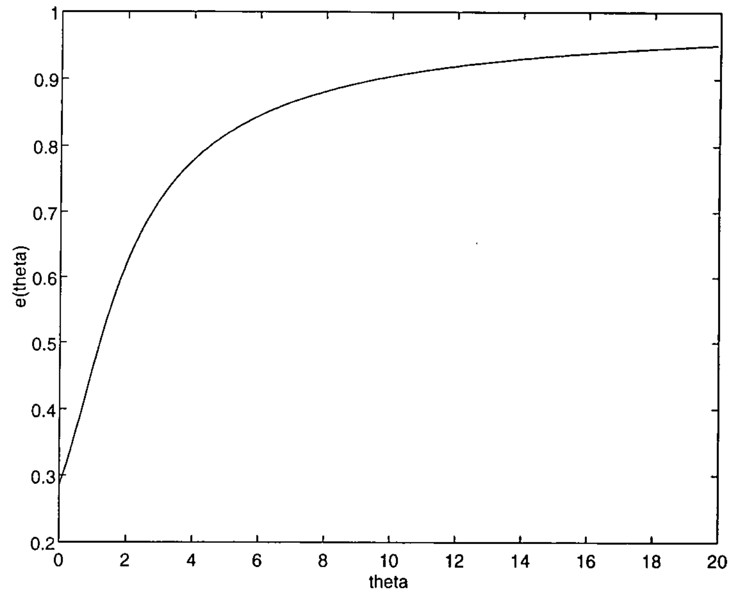

Figure 8.2: Effective bandwidth function $e(\theta)$ vs. θ

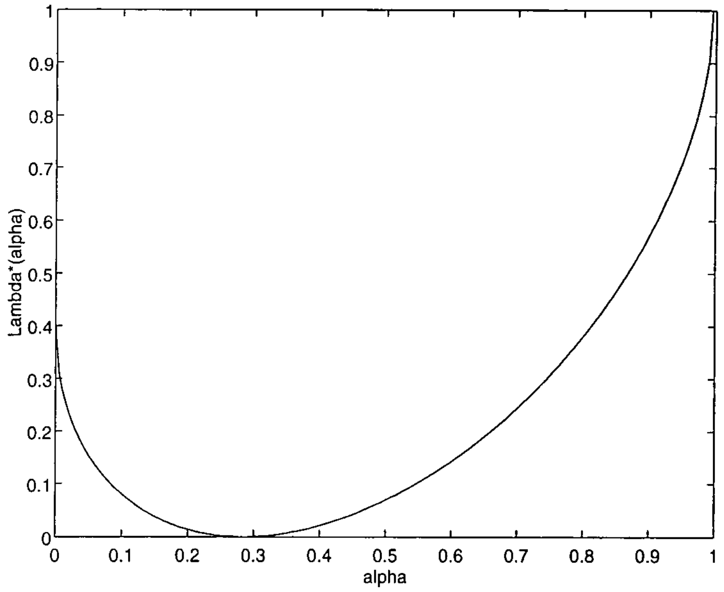

Figure 8.3: The function $\Lambda^*(\alpha)$ vs. α

t(the service rate) and let $\Lambda_C(\theta)$ be the corresponding asymptotic log moment generating function. Now (8.2) can be modified as follows

$$dq(t)/dt = \begin{cases} a(t) - C(t), & \text{if } a(t) - C(t) > 0 \text{ or } q(t) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8.27}$$

In view of (8.27), this is equivalent to multiplexing $a(t)$ and $-C(t)$ and passing it to a server with capacity 0. It is easy to show that the asymptotic log moment generating function of $-C(t)$ is $\Lambda_C(-\theta)$. Denote by $\Lambda_a(\theta)$ the asymptotic log moment generating function of $a(t)$. Then we have

$$\Pr(Q \geq x) \approx \exp(-\theta^* x) \tag{8.28}$$

if θ^* is the unique solution of

$$\Lambda_a(\theta) + \Lambda_C(-\theta) = 0. \tag{8.29}$$

As a special case, for the MMPP arrival process, assume that the service time of each packet is an exponentially distributed independent random variable with mean $1/C$. Then $\Lambda_C(\theta) = C(e^\theta - 1)$. From (8.26), one can rewrite (8.29) as

$$r(\mathbf{Q} + (e^\theta - 1)\mathbf{R}) + C(e^{-\theta} - 1) = 0 \tag{8.30}$$

Again, this coincides with the formula in [91].

8.2.3 The Limitations of the Effective Bandwidth Approximation

The effective bandwidth approximation is appealing because the asymptotic decay rate θ^* is relatively easy to determine and under (8.16) the bandwidth requirement of sources is additive. However, in [93], Choudhury *et al.* argued that for many models, it is possible to show that as $B \to \infty$

$$G(B) \approx De^{-\theta^* B} \tag{8.31}$$

and under some situations D can be very different from 1. For bursty sources, the asymptotic constant D in (8.31) is typically less than 1, and the effective bandwidth approximation in (8.16) is conservative. For multiple bursty sources, it has been shown that the effective bandwidth approximation can be much too conservative since D can be 10^{-5} or less. Namely, there are regions in which the effective bandwidth approximation performs poorly. In some cases, the network can actually support twice as many sources as predicted by the effective bandwidth approximation.

To explain why the asymptotic constant D can be so different from 1 with many sources, we consider the case of N identical sources with fixed total rate. As N in-

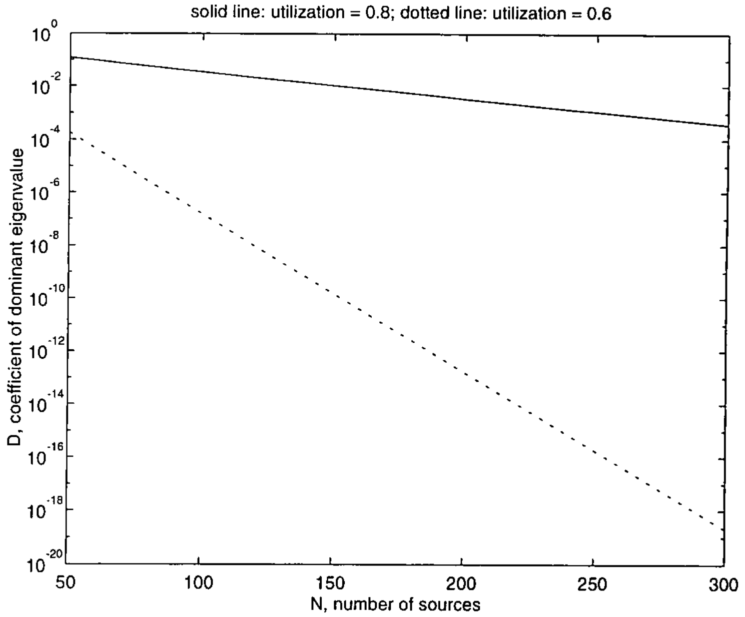

Figure 8.4: Prefactor D vs. number of sources N

creases, the total rate is kept fixed by properly scaling the individual streams. With this structure, it is known that the asymptotic decay rate θ^* in (8.31) is independent of N. Numerical experiments in [93] show that the asymptotic constant with N sources (and this scaling), $D(N)$, is itself asymptotically exponential in N, i.e., $D(N) \approx He^{\delta N}$, where $\delta \leq 0$ for sources more bursty than Poisson and $\delta \geq 0$ for sources less bursty than Poisson. Moreover, $|\delta|$ tends to increase as the source gets burstier or smoother compared to Poisson. As an example, consider the following case of Markov fluid sources. Let $\lambda = 0.9519, \mu = 2.9481, R_1 = 0$. In Figure 8.4, the values of D are shown for various values of N under different channel utilisations. Here the channel utilisation is defined as $\frac{NR_2\lambda}{(\mu+\lambda)C}$. In general, the effective bandwidth approximation tends to get worse as the number of sources increases, the buffer size decreases, the channel utilisation decreases, and the source gets further from Poisson [93]. Although (8.31) is a more accurate approximation by adding a prefactor D to the original effective bandwidth one (8.16), the exact values of D for heterogeneous Markovian sources are not easy to calculate.

8.3 A More Accurate Asymptotic Approximation

Here we bring together two recent strands of work applying the theory of large deviations to queue length asymptotics: namely, large buffer asymptotic, and large N asymptotic for superposed streams. The large buffer asymptotic has been addressed in the previous section. In this section, we focus on the large superposition asymptotic. Our main concern is how to compute the prefactor D in (8.31), which adds significantly to the accuracy of the effective bandwidth approximation. In fact, D can be viewed as the loss in bufferless multiplexing as estimated from Cramer's theorem.

First consider a zero buffer. Let V_j denote the stationary rate of traffic generation by source j, for $j = 1, 2, ..., N$ and let $\{V_j\}$ be a collection of i.i.d. random variables. Loss occurs when the total traffic generation $V_1 + \cdots + V_N$ exceeds the link capacity $C = Nc$, where c is the service rate per source. By using Cramer's theorem, we have

$$\Pr(V_1 + \cdots + V_N > Nc) \approx e^{-N\varphi^*(c)} \tag{8.32}$$

where $\varphi^*(c) = \sup_\theta(\theta c - \varphi(\theta))$, and $\varphi(\theta) = \log M(\theta), M(\theta) = \mathrm{E}e^{\theta V_1}$. This kind of approximation is the same approach as taken by Hui in examining bufferless resources [115].

We can make a refinement to the estimate of $\Pr(\sum_{j=1}^N V_j > Nc)$ given by (8.32) using the Bahadur and Rao theorem [111]:

$$\Pr(V_1 + \cdots + V_N > Nc) \approx \frac{1}{\sqrt{2\pi N}\sigma\theta_0} e^{-N\varphi^*(c)} \tag{8.33}$$

where θ_0 achieves the maximum in $\varphi^*(c) = \sup_\theta(\theta c - \varphi(\theta))$ and $\sigma^2 = \varphi''(\theta_0)$.

Next we use a simple example to show how (8.33) results in a much improved estimate. Let $x_1, x_2, ...$ be standard normal random variables. Then

$$M(\theta) = \frac{1}{\sqrt{2\pi}} \int e^{\theta y} e^{-y^2/2} dy = e^{\theta^2/2},$$

so that $\varphi^*(c) = \sup_\theta(\theta c - \theta^2/2) = c^2/2$. Then Cramer's theorem states that, for any $c > 0$,

$$\Pr(x_1 + \cdots + x_N \geq Nc) \approx e^{-Nc^2/2}.$$

For the Bahadur and Rao theorem, we have $\theta_0 = c, \sigma = 1$. So

$$\Pr(x_1 + \cdots + x_N \geq Nc) \approx \frac{1}{\sqrt{2\pi N}c} e^{-Nc^2/2} \tag{8.34}$$

In this case, we can also perform a direct calculation: $x_1 + \cdots + x_N$ is a normal random variable distributed as $\sqrt{N} x_1$, so

$$\Pr(x_1 + \cdots + x_N \geq Nc) = \Pr(x_1 \geq \sqrt{N}c) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{N}c}^\infty e^{-t^2/2} dt.$$

Using an estimate of this integral,

$$\frac{1}{y + y^{-1}} e^{-y^2/2} \leq \int_y^\infty e^{-t^2/2} dt \leq \frac{1}{y} e^{-y^2/2},$$

we obtain

$$\Pr(x_1 + \cdots + x_N \geq Nc) \approx \frac{1}{\sqrt{2\pi N}c} e^{-Nc^2/2},$$

which is in agreement with (8.34). The simulation results in [116] also show that the Bahadur-Rao approximation (8.33) is much more accurate than the one using Cramer's theorem (8.32).

For a two-state source which alternates between rate R_2 with probability p and $R_1 < R_2$ with probability $1-p$, the parameters in (8.33) can be solved as

$$\theta_0 = \frac{1}{R_2 - R_1} \log \frac{(c - R_1)(1-p)}{p(R_2 - c)}, \tag{8.35}$$

$$\varphi^*(c) = \frac{c - R_1}{R_2 - R_1} \log(\frac{(c - R_1)(1-p)}{p(R_2 - c)}) - \log(\frac{(R_2 - R_1)(1-p)}{R_2 - c}), \tag{8.36}$$

and

$$\sigma^2 = (c - R_1)(R_2 - c). \tag{8.37}$$

The above result can be extended to the case of multiplexing heterogeneous sources. Suppose there are N independent virtual circuits of J types. $N\rho_j$ of them are of type $j, \sum_{j=1}^{J} \rho_j = 1$. Define $\varphi_j(\theta)$ as the log moment generating function of a source of type j. Then

$$\log \mathrm{E}[\exp(\theta(V_1 + \cdots + V_N))] = \sum_{j=1}^{J} N\rho_j\varphi_j(\theta).$$

This is the same as the log moment generating function of the sum of N i.i.d. random variables $Y_1, ..., Y_N$ with $\log \mathrm{E}[\exp(\theta Y_1)] = \varphi(\theta) = \sum_{j=1}^{J} \rho_j\varphi_j(\theta)$. Therefore (8.33) still holds.

Now combining the above approximation with the result of effective bandwidths, we conclude that the stationary overflow probability of a buffer of size B is approximately given by

$$G(B) \approx \frac{1}{\sqrt{2\pi N}\sigma\theta_0} e^{-N\varphi^*(c)} e^{-\theta^* B} \tag{8.38}$$

where θ^* is calculated using the method discussed in the previous section.

8.4 Numerical Results

In this section, we conduct some numerical experiments to study the effectiveness of the proposed approximation.

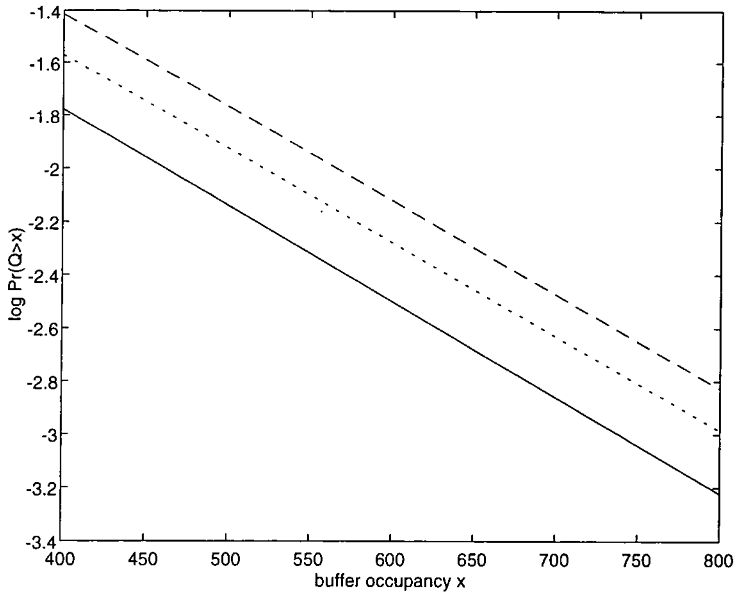

Figure 8.5: Comparison of simulation (solid line), our approximation (dotted line) and effective bandwidth method (dashed line) for discrete-time Markov sources (utilisation = 0.94)

8.4.1 Discrete-Time Markov Sources

We use a two-state discrete-time Markov model as our first example. The queueing behaviour of this model has also been investigated by Duffield (e.g., [117]). There are N independent sources, each of which is represented by a two-state ON/OFF discrete-time Markov model as described previously. In state 1, a source does not generate a cell, and in state 2, it generates only one cell. So $R_1 = 0, R_2 = 1$. To apply the tail approximation in (8.38), the asymptotic decay rate θ^* can be determined by solving the following equation:

$$Ne(\theta) = C \tag{8.39}$$

where $e(\theta)$ is given by (8.20). In the mean time, it is easy to calculate the parameters in (8.33), so the prefactor D can be obtained without difficulty.

Figure 8.5 compares the effective bandwidth approximation with our method for $\Pr(Q > x)$ vs. x in a homogeneous multiplexer with rate $C = 40$. There are $N = 94$ sources. Each source is characterized by $d = 0.045, a = 0.03$. Hence the utilisation of the sources, u, is 0.94. The "actual" overflow probabilities were obtained by simulations. Figure 8.6 shows the corresponding results for another single-server queue(C = 1) fed by $N = 32$ sources. In this case, $d = 0.16667, a = 0.0026455, u = 0.5$. From both examples, we observe that the tail decay rate is well estimated and our approximation is closer to the actual overflow probability. We also note that the difference between our

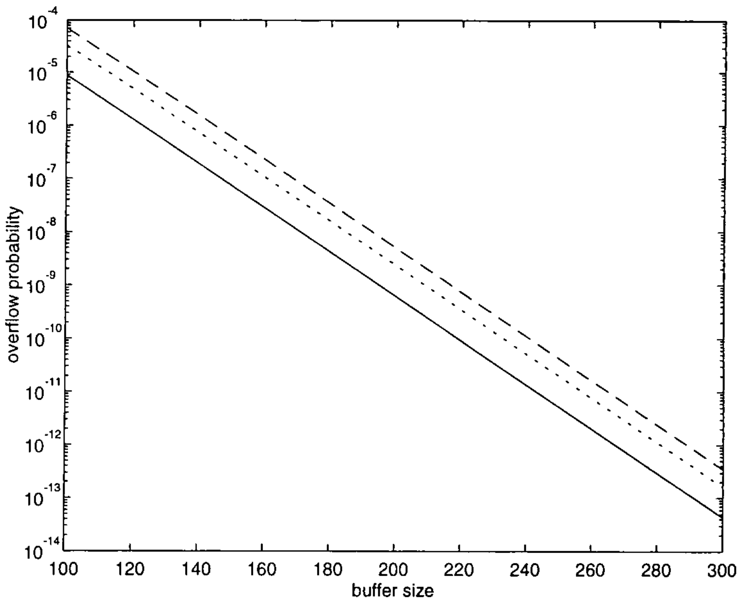

Figure 8.6: Comparison of simulation (solid line), our approximation (dotted line) and effective bandwidth method (dashed line) for discrete-time Markov sources (utilisation = 0.5)

approximation and the effective bandwidth method is small compared to the difference between the approximations and the simulation results.

8.4.2 Markov Modulated Fluids

We consider a system of $N = 100$ sources modelling voice channels. The voice sources can be modeled as Markov modulated fluid ON/OFF sources($R_1 = 0$), with mean silent duration $1/\lambda = 650$ ms and mean talkspurt duration $1/\mu = 352$ ms [88]. The data rate during the talkspurt is $R_2 = 32$ kbps. Figure 8.7 compares the logarithms of overflow probability computed by four different approaches: the effective bandwidth approximation, our approximation, the exact distribution $\Pr(Q > B)$, and an approximation for small buffers due to Hsu and Walrand [116]. In Figure 8.7, the load of the system is 0.82. The buffer size is represented by the corresponding maximum queueing delay. The exact value $\Pr(Q > B)$ was computed by the method proposed by Anick *et al.* [85]. The approximation found in [116] is of the form $D\exp(-NC_2\sqrt{b})$, where $b = B/N$ is the buffer capacity per source and C_2 is derived for small buffer asymptotics. We observe that Hsu-Walrand approximation is only accurate for very small buffers and cannot capture the exponential tail decay rate for large buffers. On the other hand, the effective bandwidth approximation overestimates the small overflow probabilties by several orders of magnitude, while our method yields a much tighter upper bound over a wide range of buffer sizes. We have got similar results for a system with load of 0.66(Figure 8.8).

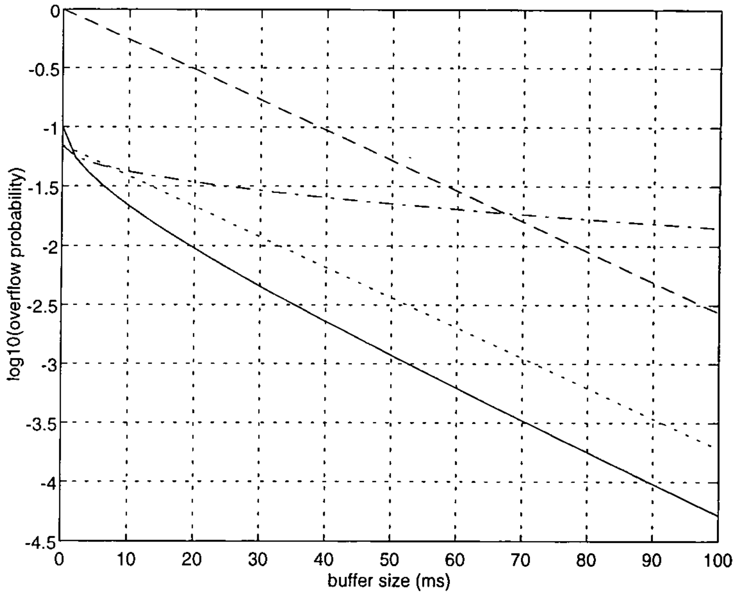

Figure 8.7: Comparison of exact analysis (solid line), our approximation (dotted line), Hsu-Walrand approximation (dash-dotted line) and effective bandwidth method (dashed line) for voice sources (load = 0.82)

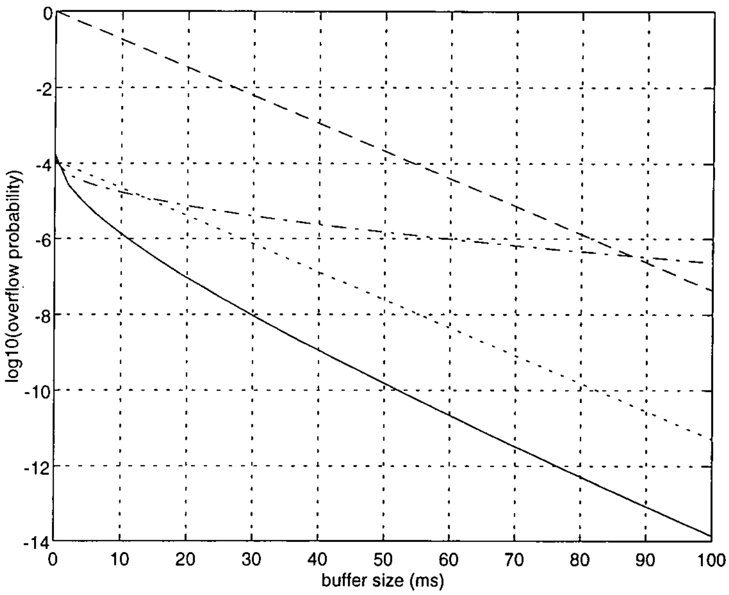

Figure 8.8: Comparison of exact analysis (solid line), our approximation (dotted line), Hsu-Walrand approximation (dashed-dotted) and effective bandwidth method (dashed line) for voice sources (load = 0.66)

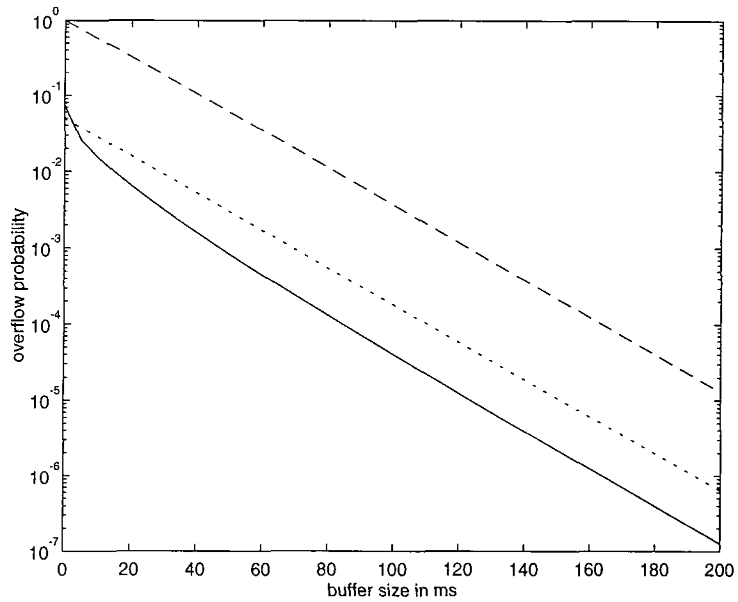

Figure 8.9: Comparison of exact analysis (solid line), our approximation (dotted line) and effective bandwidth method (dashed line) for videotelephone sources

Figure 8.9 compares the results of overflow probability obtained by our method with that by the effective bandwidth approach for 10 videotelephone sources. The traffic parameters are as given in [75]. The service rate is set to 48.75 Mb/s to provide a utilisation of 0.8. The exact loss probabilities were calculated using the standard fluid flow method [85]. Again, we see that the tail decay rate is well estimated and our method yields much more accurate results than the effective bandwidth approximation.

8.4.3 Markov Modulated Poisson Process

Here we consider a scenario from [93] in which the input process to the buffer queueing system is multiple two-state MMPP sources. For this experiment 24 identical sources are served by an ATM multiplexer. The parameters of each source are given by

$$\mathbf{Q} = \begin{bmatrix} -\frac{1}{4363.63} & \frac{1}{4363.63} \\ \frac{1}{436.36} & -\frac{1}{436.36} \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0 & 0 \\ 0 & 0.1375 \end{bmatrix} \tag{8.40}$$

Actually this kind of ON/OFF MMPP model is also known as Interrupted Poisson Process(IPP). The service rate is 1 so that the link utilisation is 0.3.

Figure 8.10 displays the exact tail probabilities $\Pr(Q > B)$(obtained by the numerical method given in Appendix C), the proposed approximation and the effective bandwidth approximation. From Figure 8.10, we see that when the link utilisation is relatively low(0.3), the effective bandwidth approximation is extremely conservative, in error by

Method of computation	Number of sources allowed for buffer size of 600	Buffer size required to support 24 sources
Exact	24	600
Effective bandwidth	12	1146
Proposed method	20	790
Average rate engineering	80	not applicable
Peak rate engineering	7	not applicable

Table 8.1: A comparison of different methods for determining (i) the number of sources for the fixed buffer size of 600 and (ii) the buffer size required to support 24 sources

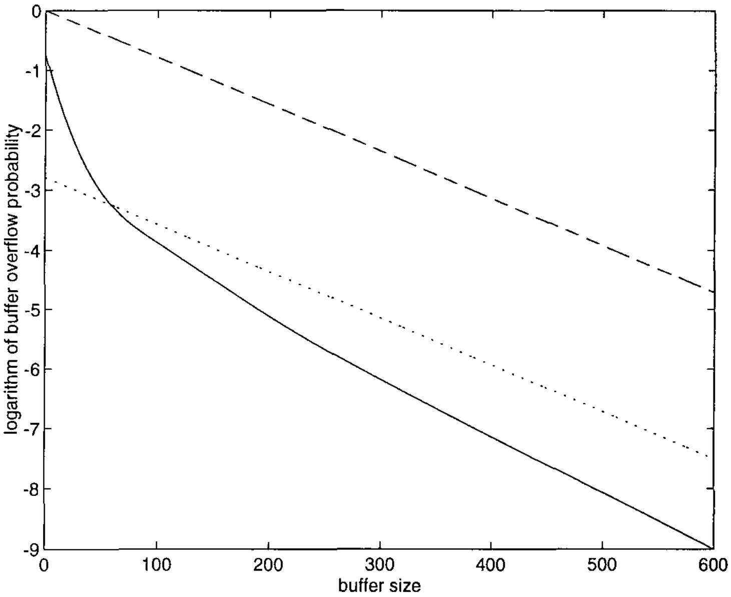

Figure 8.10: A comparison of exact analysis (solid line), our approximation (dotted line) and effective bandwidth method (dashed line) for MMPP sources

a factor of 5 orders of magnitude, while our approximation produces a better estimate. This is because replacing D in (8.31) by 1 introduces a large error. Table 8.1 compares five different procedures for determining the number of sources that can be supported when the target blocking probability is 10^{-9} and the buffer size is 600. We then fix the number of sources at 24 and determine the required buffer size. The results are also presented in Table 8.1. It is clear that the effective bandwidth approximation significantly underestimates the capacity, while our approximation is reasonably good.

8.5 Fast Bounds for Other QOS Parameters

The above simulations show that the Bahadur-Rao approximation (8.38) seems to provide tight upper bounds for infinite buffer queues, although mathematical proof is quite difficult. Therefore these bounds are also useful for the finite buffer case. If we denote the queue in an infinite buffer by Q_∞, and the queue in a finite buffer of size B by Q_B, then $\Pr(Q_B > b) \leq \Pr(Q_\infty > b)$. So we can use any upper bounds on Q_∞ for Q_B too:

$$\Pr(Q_B > b) \leq De^{-\theta^* b} \tag{8.41}$$

For large buffer size B, these bounds will be as good as for the infinite buffer case.

Once we have established bounds on the queue length distribution, we may use them to give estimates for the QOS parameters that are of interest in ATM. Three parameters have been used by ATM Forum to measure the QOS experienced by traffic as it passes through a queueing system. They are: the cell loss ratio, the mean cell delay and the cell delay variance(jitter). It is easy to show that, in a finite buffer of size B cells, the Cell Loss Ratio(CLR) is given approximately by

$$\text{CLR} = \frac{\text{E[number of cells lost]}}{\text{mean activity}} \approx \Pr(Q_B > B) \approx De^{-\theta^* B} \tag{8.42}$$

Letting d denote the delay experienced by a cell, we have the relation $d \approx Q/C$, where Q is the current queue length [117]. Thus, for the mean cell delay, we have

$$\text{E}d \approx \frac{1}{C}\text{E}Q = \frac{1}{C}\sum_{l=0}^{B-1}\Pr(Q > l) \approx \frac{1}{C}\sum_{l=0}^{B-1} De^{-\theta^* l} \tag{8.43}$$

Similarly, the cell delay variance is given by

$$\begin{aligned} \text{Var}d &\approx \frac{1}{C^2}\sum_{l=0}^{B-1}(2l+1)\Pr(Q > l) - [\frac{1}{C}\sum_{l=0}^{B-1}\Pr(Q > l)]^2 & (8.44)\\ &\approx \frac{1}{C^2}\sum_{l=0}^{B-1}(2l+1)De^{-\theta^* l} - (\frac{1}{C}\sum_{l=0}^{B-1} De^{-\theta^* l})^2 & (8.45) \end{aligned}$$

Figure 8.11 and Figure 8.12 show the mean cell delay and delay variance estimates for the videotelephone traffic, respectively. The time unit used is slot, which is equal to per-cell transmission time(53 bytes/cell). It is clear that increasing the buffer capacity increases the cell delay and delay jitter. This is in contrast with the cell loss ratio, which is a nonincreasing and convex function of buffer size. From the above discussion, we note that we only need to compute two numbers, D and θ^*, to get the estimates for the QOS parameters. One great advantage of the proposed method is that the calculation is independent of the system size(the number of sources present), while traditional analyt-

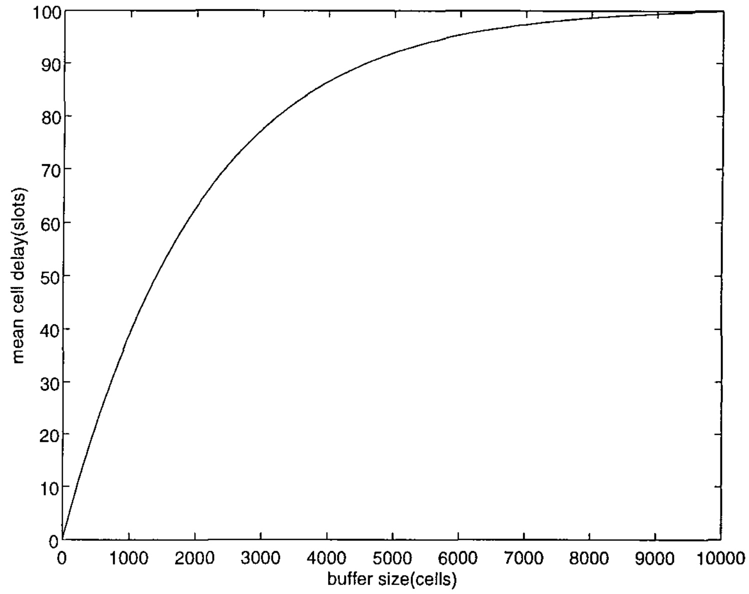

Figure 8.11: Estimate for mean cell delay vs. buffer size

ical methods generally require the solution of matrices whose dimension is proportional to the number of sources. Thus our approach is more suitable for *real-time* admission control.

In practice, cell loss and cell delay are interrelated and sometimes may have conflicting requirements in terms of buffer dimensioning. This is particularly true with video services since these are sensitive to both cell loss and jitter [118]. Hence when allocating network resources for video services, in terms of transmission bandwidth and buffer capacity, it is necessary for both parameters to be considered jointly. However, most research reported to date has treated each parameter separately. Here, on the basis of the above QOS estimation approach, we propose a methodology of admission control to support video-related services based on the joint consideration of cell loss and cell jitter.

We impose upper limits ϵ and δ on the cell loss ratio and the cell delay variance respectively, i.e.,

$$\text{CLR} \leq \epsilon, \qquad \text{Var}d \leq \delta. \tag{8.46}$$

During the call setup phase, the control function has to make decisions whether a connection request can be accepted or not by checking both constraints should be satisfied. The aforementioned fast and accurate estimation technique is used to get the QOS values. Let $N_{max,L}$ be the maximum number of connections on the link capable of maintaining the cell loss ratio below ϵ, and $N_{max,D}$ the maximum number of connections under the delay constraint δ. Thus, the admission rule satisfying both requirements is simply the

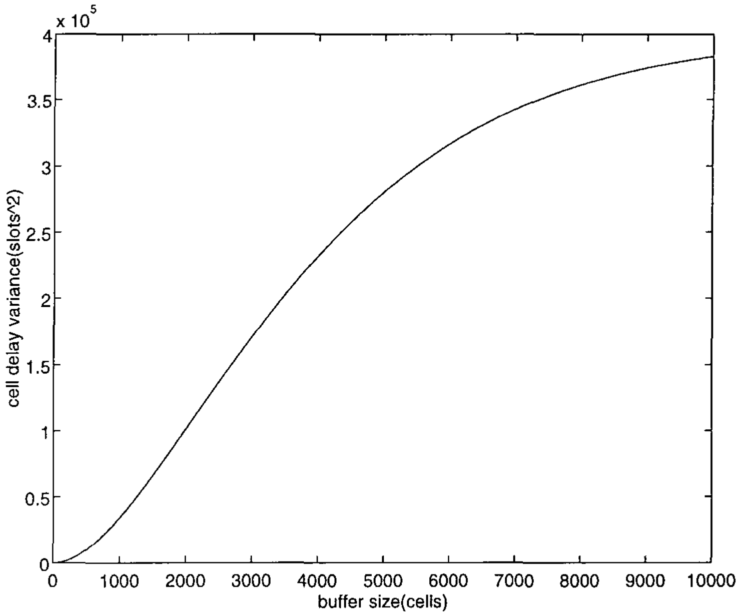

Figure 8.12: Estimate for cell delay variance vs. buffer size

following: a new connection can be accepted if

$$N_0 + 1 \leq \min(N_{max,L}, N_{max,D}), \tag{8.47}$$

where N_0 is the number of connections in progress. It is hoped that future work will lead to the development of an efficient resource allocation algorithm which meets multiple QOS requirements.

8.6 Applications to Call Admission Control

In this section we present some applications of our results to the problems of call admission control in ATM networks. Here the QOS criterion is taken to be the cell loss probability due to buffer overflow.

8.6.1 Voice Sources

Consider a single T1 channel(1.536 Mbps) serving a population of voice sessions. The voice sources are modelled by Markov modulated fluids with parameters given previously. We would like to decide the maximum number of voice sessions that can be supported by the channel such that $\Pr(Q > B) \leq \epsilon$. Let N_m denote this number and ϵ be 10^{-5}. In Figure 8.13, we compare the approximate values of N_m obtained from our method with the exact ones for various buffer sizes(maximum delay at buffer served at the channel

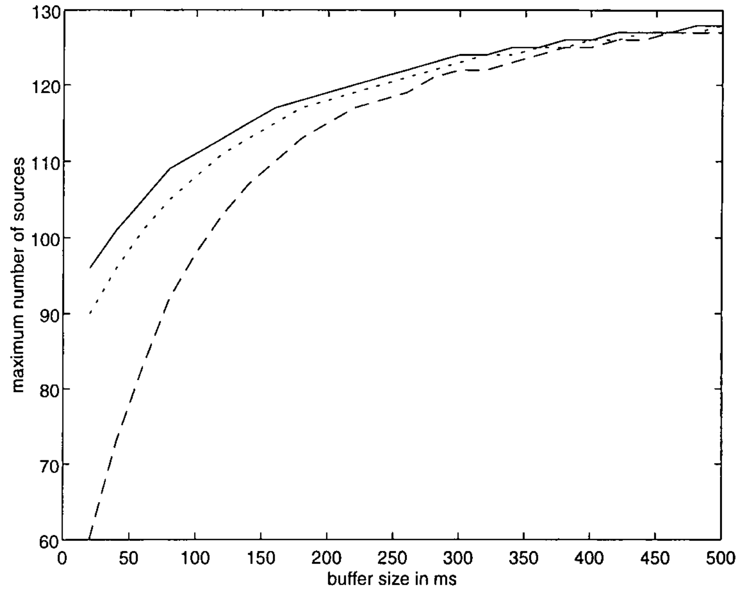

Figure 8.13: Supportable number of voice sources. Solid line: exact analysis, dotted line: our approximation, dashed line: effective bandwidth approach

output rate). Also included are the number of sources that can be supported based on the effective bandwidth approach. It can be seen that the effective bandwidth approach yields poor results, especially for small buffer sizes, while our approximation is much more accurate and increases resource utilisation.

8.6.2 Videotelephone Traffic

Consider the videotelephone source model as discussed in [75]. Let a continuous-state queue (multiplexer buffer) be fed by a number of input sources. In Figure 8.14, we compare the possible maximum utilisations of a 25 Mb/s multiplexer under the loss probability constraint of 10^{-5}. It is clear that our approximation is much closer to the exact result and hence it results in a higher link utilisation.

8.6.3 Video Teleconference Traffic

An important class of video services is video teleconferencing. Consider the real video teleconference sequence in [119] obtained by using a DPCM/DCT coding scheme without motion compensation. The sequence shows head-and-shoulders scenes with moderate motion and scene changes, and with very little camera zoom or pan. This particular set of data consists of a sequence of 38137 numbers(frames), each of which represents the number of (64-byte) cells in the corresponding frame. The frame period is 1/30 s.

Denote a VBR video source by a sequence $\{X_n, n \geq 0\}$, where X_n is the number of

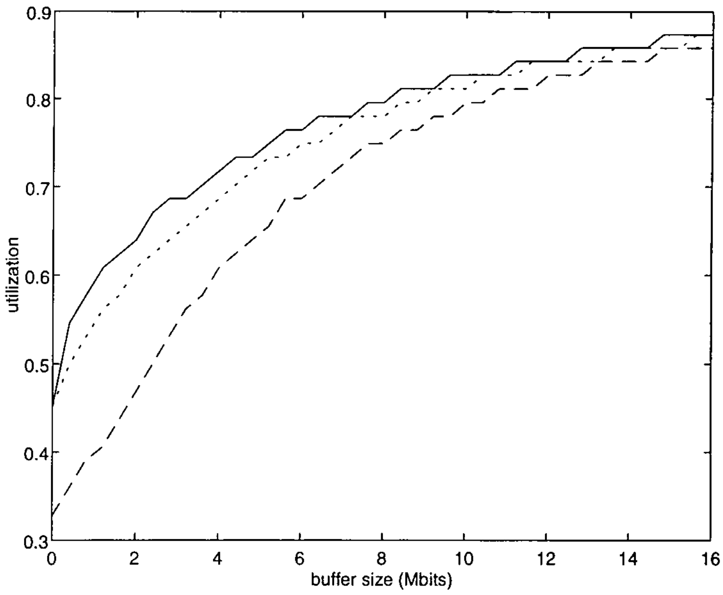

Figure 8.14: Comparison of maximum utilisation for the videotelephone model. Solid line: exact analysis, dotted line: our approximation, dashed line: effective bandwidth approach

R_p (cells/frame)	R_b (cells/frame)	R_m (cells/frame)	Σ (cells/frame)	α
4818	776	1506	513	0.02

Table 8.2: Statistics of the real video data

cells in the nth frame, $n = 0, 1, 2, ...$ It is reasonable to assume that the video traffic considered here possesses the stationary property. Based on this assumption, sequence $\{X_n, n \geq 0\}$ can be sufficiently characterized by its peak rate R_p, bottom rate R_b, mean rate R_m, standard deviation Σ and the coefficient of the autocorrelation function α [2]. The statistics of the real video data are given in Table 8.2.

Let $\{X_n^N, n \geq 0\}$ denote the superposition of N independent, homogeneous, stationary VBR video sources $\{X_n, n \geq 0\}$. Then, the characteristics of the aggregate traffic $\{X_n^N, n \geq 0\}$ will be $(NR_p, NR_b, NR_m, \sqrt{N}\Sigma, \alpha)$.

In [120], the aggregate traffic of VBR video sources is modeled accurately by an $(M+1)$-state Discrete-time Markov Modulated Deterministic Process(D-MMDP) which is the superposition of M identical and independent two-active-state minisources. A

[2] It is well known that the autocorrelation function $A(m)$ has a general shape of the exponential function $(1-\alpha)^m$.

a	d	M	R_2	R_1
0.003612	0.016388	9	522	84

Table 8.3: Matching parameters of the D-MMDP model

minisource is just the same as the two-state discrete-time Markov model that we discussed previously. In order to match $\{X_n^N, n \geq 0\}$ with the D-MMDP, we choose the values of the parameters (M, R_1, R_2, a, d) so that the D-MMDP has the same traffic characteristics as those of $\{X_n^N, n \geq 0\}$ with parameters $(NR_p, NR_b, NR_m, \sqrt{N}\Sigma, \alpha)$. After some manipulations, we have [120]

$$a = \frac{\alpha(R_m - R_b)}{R_p - R_b} \tag{8.48}$$

$$d = \frac{\alpha(R_p - R_m)}{R_p - R_b} \tag{8.49}$$

$$M = \frac{N(R_p - R_m)(R_m - R_b)}{\Sigma^2} \tag{8.50}$$

$$R_2 = \frac{R_p\Sigma^2}{(R_m - R_b)(R_p - Rm)} \tag{8.51}$$

$$R_1 = \frac{R_b\Sigma^2}{(R_m - R_b)(R_p - Rm)} \tag{8.52}$$

where R_1, R_2, M should be rounded to the nearest integer. The matching result for $N = 1$ is summarized in Table 8.3.

On the basis of the above model, we consider the problem of call admission for the teleconference VBR video connections. The network has to estimate amount of bandwidth required by a connection based on its traffic characteristics and the QOS requirement. One popular approach is the effective bandwidth, which can be computed using (8.20). We compare the performance of the proposed approximation with the effective bandwidth approach through the following numerical example. The multiplexer has a buffer space of 100 cells and the QOS(cell loss ratio) is $\epsilon = 10^{-5}$. In Figure 8.15 we plot the estimated bandwidth required by the aggregate traffic against the number of sources. The exact result is obtained from the exact queueing analysis of the D-MMDP/D/1/K queue [120], which is very complicated due to the complexity in calculating the stationary state probabilities. It can be seen that the effective bandwidth approach is very conservative and overestimates the required bandwidth by a large amount(near peak rate allocation). On the other hand, our approach performs much better and is very close to the exact bandwidth. This is because a VBR video source is modelled by the superposition of a number of independent and identical minisources and the statistical multiplexing gain is significant. In [121] the minisource-based D-MMDP model is used to

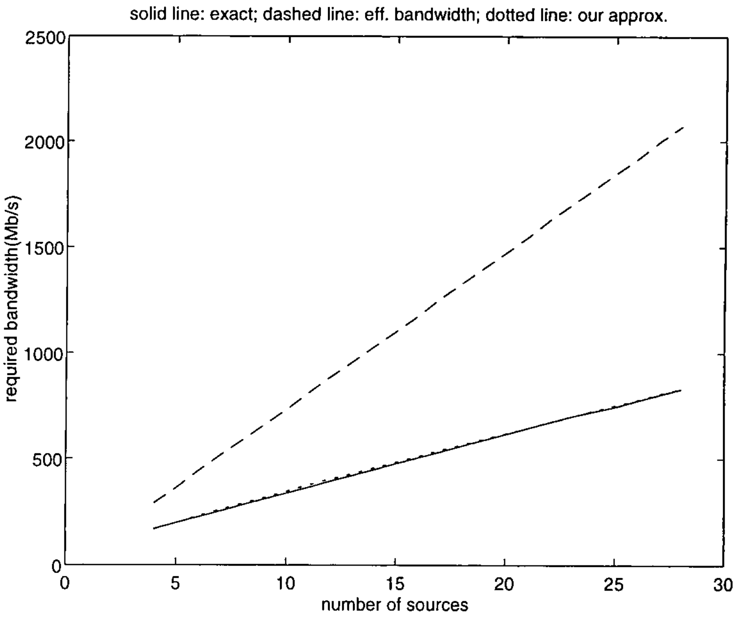

Figure 8.15: Bandwidth requirement comparison

model the macro-frame smoothed VBR MPEG-2 traffic. So our approximation method can be applied successfully to that scenario as well.

8.7 Summary

In this chapter, we have discussed an approximation technique for QOS parameters in an ATM multiplexer fed by Markovian sources. The approximation for buffer overflow probability has been derived using the theory of large deviations, and in particular, the Bahadur-Rao theorem, and gives more accurate estimate than the effective bandwidth approach does. The multiplexing gain is therefore enhanced. It also provides useful qualitative insight as to the statistical behavior of the multiplexer queue with respect to the nature of the Markovian traffic. From this, we establish bounds on other QOS parameters such as mean cell delay and cell delay variance. It has been shown that our approach can be applied efficiently to call admission control for ATM multimedia traffic.

Chapter 9

Conclusions and Further Work

9.1 Conclusions

The emerging high-speed ATM networks are expected to integrate through statistical multiplexing large numbers of traffic sources having a broad range of statistical characteristics and different QOS requirements. To achieve high utilisation of network resources while maintaining the QOS, efficient traffic management strategies have to be developed. This thesis has investigated the design of efficient congestion control schemes for ATM networks.

The basic concepts and main features of BISDN and ATM have been described. In particular, the ATM cell header fields and the ATM protocol reference model have been discussed in detail. A critical review of traffic management and congestion control for ATM networks is given. Various control functions are classified according to their roles within the network and associated algorithms have been described. The effectiveness of reactive control schemes is limited by the duration of feedback delays and requires very large buffers. On the other hand, preventive control techniques are often sensitive to the parameters of the source traffic, which itself is an open issue. Even with accurate traffic characterization, the existing techniques often restrict utilisation of network resources. The problem is further complicated due to the existence of different applications with diverse QOS requirements. Therefore the congestion control framework in ATM networks remains an open issue and has been the subject of significant research effort.

The identification of current limiting mechanisms has prompted investigation and development of novel techniques in an effort to ovecome the performance restrictions. The first technique studied is the use of artificial neural networks. Neural networks not only lead to high flexibility, allowing the satisfaction of specific service needs in the user and network operator perspectives, but are also very efficient and thus optimize the resource allocation under high network load conditions. A critical analysis of the use of neural networks to provide improved congestion control for ATM networks has been carried out. From the discussion of previous work, we try to answer the question "why neural networks in ATM traffic control?" and give useful comments on the strengths and limitations of NN-based methods. Then a novel adaptive congestion control approach using reinforcement learning is presented. The proposed neural controller employs reinforcement learning to tune its weights so as to produce an optimal control signal. Simulation results show that the proposed method is adaptive to the changing network environment and optimal control is achieved by minimizing a cost function which contains two important performance measures.

At the cell level, a neural network approach is adopted in the context of ATM traffic prediction and characterization. The FIR neural network model and the associated temporal backpropagation training algorithm have been discussed in detail. It is shown that the FIR network can accurately predict the traffic arrival patterns in the near future. A feedback flow control mechanism based on neural traffic prediction has been presented for efficient rate regulation in ATM networks. The predicted output in conjunction with the current queue information of the buffer can be used as a measure of congestion. When the congestion level is reached, the arrival rate is decreased appropriately. Simulation results show that the proposed method outperforms conventional control schemes in terms of cell loss rate. At the call level, an accurate yet computationally efficient approach to effective bandwidth estimation for multiplexed connections is investigated. In this method, a multilayer perceptron is employed to model the nonlinear relationship between the effective bandwidth and the traffic situations and a QOS measure. This is justified by the fact that multilayer feedforward networks are a class of universal approximators and suitable for nonlinear regression. In the mean time, the neural network approach can meet the real-time requirements of ATM networks due to its high computation rate. Simulations show that the results of our method are very close to those of the exact stochastic fluid model. Applications of this approach to admission control, bandwidth management and dynamic routing are also discussed.

The second technique investigated is the use of the theory of large deviations applied to effective-bandwidth-based QOS estimation and admission control. Although the CAC algorithm based on effective bandwidths is simple and fast, it ignores the effect of statistical multiplexing of large numbers of sources and hence it is a too conservative approach resulting in under-utilisation of network resources. In contrast to conventional effective bandwidth methods, we directly calculate the effective bandwidth of the aggregate traffic rather than summing up individual bandwidths. The aggregate arrival traffic is characterized by four appropriately selected parameters and then accurately modelled by a two-state MMPP via matching four important statistics. If the buffer size is large, admission control can be achieved by computing the effective bandwidth of the two-state MMPP. We confirm through computer simulations that the proposed CAC scheme outperforms conventional methods with respect to link utilisation.

We use the theory of large deviations to provide a unified description of effective bandwidths for various traffic sources and the associated ATM multiplexer queueing performance approximations, illustrating their strengths and limitations. On the basis of this discussion, we propose a more accurate estimation method for ATM QOS parameters, which is a refinement of the original effective bandwidth approximation. We achieve this by constructing a simple approximation of the buffer overflow probability: $\Pr(Q > B) \approx De^{-\theta^* B}$, where θ^* is the asymptotic decay rate of the tail of the distribution, and D is a prefactor obtained from the Bahadur-Rao theorem. Upper bounds on other QOS parameters, such as cell loss ratio, mean cell delay and cell delay variance, can all be derived from this estimate. Both theoretical studies and simulations have demonstrated that this approximation is much more accurate than one based on the pure effective bandwidth. One attraction of the proposed method is that the speed of computation of D and θ^* is independent of the size of the system. Thus our analytical techniques can be implemented fast enough for real-time administration of admission control in ATM networks.

9.2 Further Work

Traffic predictions can be used for improving ATM network efficiency by incorporating the predictions in schemes for multiplexing, routing, smoothing and bandwidth allocation. In our work reported in Chapter 5, we used off-line learning to perform single-step prediction for relatively simple voice and video traffic models. The advantage of this approach is that it is reasonably accurate and extremely fast(after the initial training phase), which makes it applicable to real-time forecasting. The drawback is that the method assumes a fixed distribution signal, which may not be true for complex video sequences such as entertainment video. Therefore, for video traffic with a number of sudden scene changes, zooms, frame cuts, and rapid movement of objects, further investigations into the continuous on-line learning are required. In the on-line learning algorithm, the weights are adjusted continuously during the whole training process to cope with changes in the distribution for complex sequences. Another challenging issue regarding video traffic prediction is multiple frame prediction [122]. For example, how many previous frames are needed for p-frame($p > 1$) prediction and how many frames ahead(maximum value of p) neural networks can predict without a significant decay in accuracy? This question is not amenable to theoretical analysis and would need to be investigated experimentally.

A possible extension to the work on flow control could be the application of fuzzy logic to traffic rate regulation. The incoming traffic predicted by neural networks and the queue length can be used as input variables of a Fuzzy Congestion Controller(FCC). Based on this data and the linguistic information stored in the fuzzy rule base, the FCC computes the regulated(fractional) flow rate for the sources feeding the ATM switch. Simulations conducted in [123] for ABR traffic have shown that fuzzy controller exhibits a robust behaviour even under extreme network loading conditions, offers fast transient response, and ensures fair share of the bandwidth for all virtual channels regardless of the number of hops they traverse.

With regard to bandwidth estimation by neural networks, further work should base the NN estimate on the way in which the traffic is actually behaving, not on an existing inaccurate parametric model. This could be achieved by either training the NN using traffic measurements collected from testbed networks instead of simulated data, or employing an on-line measurement-based NN admission controller to compute the *real* occupied bandwidth in a more realistic network scenario. For the latter case, the design of such an NN architecture is non-trivial and the parallel multilevel NN configuration proposed in [124] seems to be a promising choice. In Chapter 6, we only consider homogeneous traffic under LLR routing policy. It should be noted that the same idea can be easily extended to heterogeneous traffic arrival processes and other routing algorithms.

In [99], Duffield et al. suggest that since all that is required for the estimation of QOS parameters is a knowledge of the large deviation rate function of the arrival process, it is possible to measure the rate function directly from empirical traffic data using some statistical methods. The advantage of this approach is that it provides a basis for characterizing traffic on the fly so that resources can be allocated dynamically. However, the performance of the estimator depends critically on the choice of block size and sample size, which is not yet theoretically well understood. To this end, a possible direction for future study is to employ neural networks to improve the accuracy of the large deviation rate function estimator. This is due to the fact that several different NN models have been successfully developed for cumulative distribution estimation [125]. It is believed that the combination of neural networks and large deviations techniques can be a promising way to cope with changing traffic patterns and link loads in ATM networks.

Recent studies of real traffic data, mainly at Bellcore [54], have shown that Ethernet traffic cannot be sufficiently represented by traditional Markovian models, but instead can be more accurately matched by self-similar (fractal) models. More recently, VBR video traffic was also found to exhibit self-similar characteristics [126]. The self-similar behaviour of traffic has serious implications for the design, control and analysis of high-speed networks. An important feature of self-similar processes is their Long-Range Dependence(LRD), that is, their autocorrelation function decays less than exponentially fast. This property of persistent correlation can be characterized by the Hurst parameter H, with $H = 0.5$ for traditional Markovian streams and $H > 0.5$ for streams with LRD. Studies by Norros [127], Erramilli et al. [128] suggest that queues with long-range dependent input have subexponential tails. This is in sharp contrast to the exponential tail distribution in queues with renewal or Markov arrival processes, which is predicted by the well-known technique of effective bandwidth. Therefore the work carried out on ATM switch performance analysis and QOS parameter estimation should be extended to multiplexing systems with self-similar input traffic and it is hoped that large deviations theory can provide a powerful tool for obtaining satisfactory explanations of complex queueing behaviour under long-range dependent traffic.

Most of the work reported in this thesis has been limited to one switching node or very simple network scenarios. A more practical control strategy which is suitable for the large scale complexity of communications networks should be investigated. Performance under real traffic profiles should also be examined. There is consensus among many researchers that such a strategy will have to include multiple flow and congestion control algorithms that are organized in a hierarchical multilevel/multilayer structure and operate at different time scales. It could allow us to represent dynamic behaviour, and also allow us to decompose large system problems into smaller subproblems, with some suitable treatment of localized and global objectives.

Appendix A

The Backpropagation Algorithm

This appendix presents a summary of the backpropagation learning algorithm for MLPs. The BP algorithm uses a gradient search technique to minimize a cost function equal to the mean square difference between the desired and the actual network outputs. The NN is trained by initially selecting small random weights and thresholds and then presenting all training data repeatedly. Weights are adjusted after every trial until weights converge and the cost function is reduced to an acceptable value. An essential component of the algorithm is the iterative method described below that propagates error terms required to adapt weights back from nodes in the output layer to nodes in previous layers.

An architectural graph for BP learning, incorporating both the forward and backward phases of the computations involved in the learning process, is presented in Figure A.1(adapted from [42]). The multilayer network shown in the upper part of the figure accounts for the forward phase. The notations used in this part of the figure are as follows [42]:

- $\mathbf{w}^l$ = synaptic weight vector of a neuron in layer l
- θ^l = threshold of a neuron in layer l
- $\mathbf{v}^l$ = vector of net internal activity levels of neurons in layer l
- $\mathbf{y}^l$ = vector of function signals of neurons in layer l

The layer index l extends from the input layer($l = 0$) to the output layer($l = L$); in this figure we have $L = 2$; we refer to L as the depth of the NN. The lower part of the figure accounts for the backward phase, which is referred to as a *sensitivity network* for computing the local gradients in the BP algorithm. The notations used in this second part of the figure are as follows [42]:

- δ^l = vector of local gradients of neurons in layer l
- $\mathbf{e}$ = error vector represented by $e_1, e_2, ..., e_q$ as elements

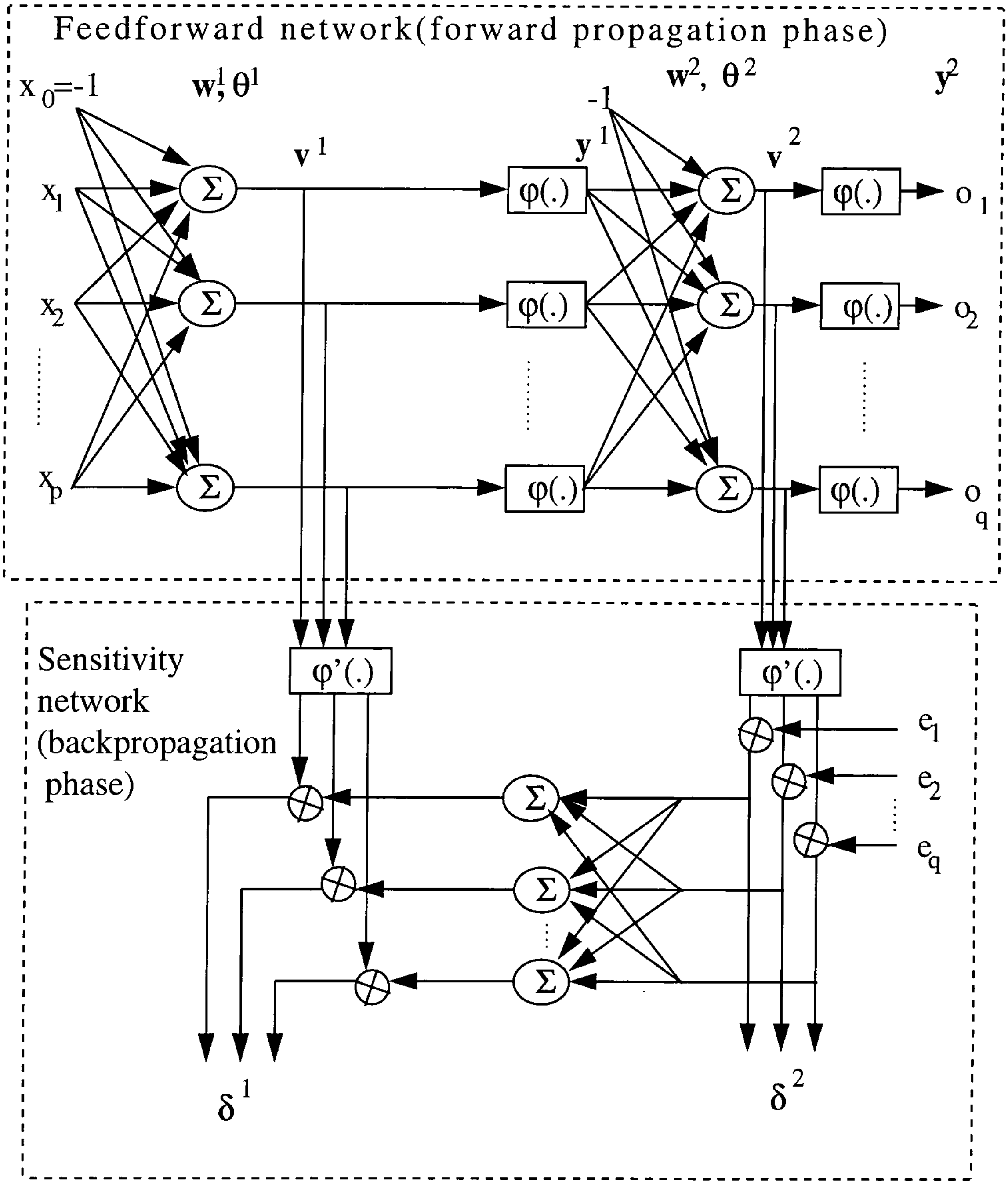
Feedforward network(forward propagation phase)
x0=-1
w1, θ1
v1
-1
y1
w2, θ2
v2
y2
x1
x2
xp
Σ
φ(.)
o1
o2
oq
Sensitivity network (backpropagation phase)
φ'(.)
e1
e2
eq
δ1
δ2

Figure A.1: Architectual graph of the backpropagation algorithm

The pattern mode BP algorithm for the training data $\{[\mathbf{x}(n), \mathbf{d}(n)]; n = 1, 2, ..., N\}$ is summarized as follows:

1. *Initialization.* Set all weights and threshold levels of the network to small random numbers that are uniformly distributed.

2. *Presentation of training examples.* Present the network with an epoch of training examples. For each example in the set ordered in some fashion, perform the following sequence of forward and backward computations in steps 3 and 4, respectively.

3. *Forward computation.* Let a training example in the epoch be denoted by $[\mathbf{x}(n), \mathbf{d}(n)]$, where $\mathbf{x}(n)$ is the input and $\mathbf{d}(n)$ is the desired output. Compute the activation potentials and function signals of the network by proceeding forward through the network, layer by layer. The net internal activity level $v_j^l(n)$ for neuron j in layer l is

$$v_j^l(n) = \sum_{i=0}^{p} w_{ji}^l(n) y_i^{l-1}(n) \tag{A.1}$$

where $y_i^{l-1}(n)$ is the function signal of neuron i in the previous layer $l-1$ at iteration n and $w_{ji}^l(n)$ is the weight of neuron j in layer l that is fed from neuron i in layer $l-1$. For $i = 0$, we have $y_0^{l-1}(n) = -1$ and $w_{j0}^l(n) = \theta_j^l(n)$, where $\theta_j^l(n)$ is the threshold applied to neuron j in layer l. Assume that the sigmoidal nonlinearity of each neuron is defined by a logistic function, i.e., $\varphi(x) = \frac{1}{1+\exp(-x)}$. Then, the function(output) signal of neuron j in layer l is

$$y_j^l(n) = \frac{1}{1 + \exp(-v_j^l(n))}. \tag{A.2}$$

If neuron j is in the first hidden layer(i.e., $l = 1$), set $y_j^0(n) = x_j(n)$, where $x_j(n)$ is the jth element of the input vector $\mathbf{x}(n)$. If neuron j is in the output layer(i.e., $l = L$), set $y_j^L(n) = o_j(n)$. Hence, compute the error signal

$$e_j(n) = d_j(n) - o_j(n) \tag{A.3}$$

where $d_j(n)$ is the jth element of the desired output vector $\mathbf{d}(n)$.

4. *Backward computation.* Compute the δ's(i.e., the local gradients) of the network by proceeding backward, layer by layer:

$$\delta_j^L(n) = e_j^L(n) o_j(n)[1 - o_j(n)] \qquad \text{for neuron } j \text{ in output layer } L \tag{A.4}$$

$$\delta_j^l(n) = y_j^l(n)[1 - y_j^l(n)] \sum_k \delta_k^{l+1}(n) w_{kj}^{l+1}(n) \qquad \text{for neuron } j \text{ in hidden layer } l \tag{A.5}$$

Hence, adjust the weights of the network in layer l by

$$w_{ji}^{l}(n+1) = w_{ji}^{l}(n) + \eta\delta_{j}^{l}(n)y_{i}^{l-1}(n) \tag{A.6}$$

where η is the learning rate parameter. Convergence is sometimes faster if a momentum term is added and weight changes are smoothed by

$$w_{ji}^{l}(n+1) = w_{ji}^{l}(n) + \eta\delta_{j}^{l}(n)y_{i}^{l-1}(n) + \alpha[w_{ji}^{l}(n) - w_{ji}^{l}(n-1)] \tag{A.7}$$

where α is the momentum constant.

5. *Iteration.* Repeat by going to step 2, i.e., iterate the computation by presenting new epochs of training examples to the network until the free parameters of the network stabilize their values and the average squared error computed over the entire training set reaches a minimum or acceptably small value. The order of presentation of training examples should be randomized from epoch to epoch.

Appendix B

Mathematical Overview of the Fluid Flow Model

This appendix addresses the ON/OFF fluid flow model, and in particular, the mathematical solution to the equilibrium buffer distribution, which is described by a set of differential equations. Both the case of homogeneous sources and the more complicated case of heterogeneous sources are considered.

B.1 Homogeneous Sources

In a fluid flow model, the packet generation process is taken to be a collection of N sources alternating between active(ON) and inactive(OFF) states. The ON periods as well as the OFF periods are exponentially distributed for each source. Without loss of generality, the unit of time is selected to be the average ON period; with this unit of time, the average OFF period is denoted by $1/\lambda$. Again, without loss of generality, the unit of information is chosen to be the amount generated by a source in an average ON period. In these units an ON source transmits at the uniform rate of 1 unit of information per unit of time and the server removes information from the buffer at a uniform rate of c units per unit of time. When r sources are ON simultaneously, the instantaneous receiving rate at the server is r. Thus, as long as the buffer is not empty, the instantaneous rate of change of the buffer content is $r - c$. Once the buffer is empty, it remains so as long as $r \leq c$. For simplicity, we assume that the following condition is satisfied:

$$\frac{N\lambda}{(1+\lambda)} < c < N \tag{B.1}$$

Let $F_i(x)$ be the steady state probability of i sources being active and the buffer content being less than or equal to x units. Then we have, for $i \in [0, N]$,

$$\frac{(i-c)dF_i}{dx} = (N-i+1)\lambda F_{i-1} - ((N-i)\lambda + i)F_i + (i+1)F_{i+1} \tag{B.2}$$

In matrix notation,

$$\mathbf{D}\frac{d}{dx}\mathbf{F}(x) = \mathbf{MF}(x) \tag{B.3}$$

where $\mathbf{F}(x) = [F_0(x), F_1(x), ..., F_N(x)]^T$, $\mathbf{D}$ =diag$\{-c, 1-c, 2-c, ..., N-c\}$ and the element of $\mathbf{M}$, $m_{i,j}$, is given by

$$m_{i,j} = \begin{cases} 0, & |i-j| \geq 2 \\ -[(N-i)\lambda + i], & j = i, i = 0, ..., N \\ i+1, & j = i+1, i = 0, ..., N-1 \\ (N-i+1)\lambda, & j = i-1, i = 1, ..., N \end{cases} \tag{B.4}$$

Due to inherent instabilities in the system, any numerical solution technique for differential equations cannot be used without exercising care that the solution does not grow(unbounded) exponentially. Instead, the problem is formulated in [85] as an eigenvalue problem:

$$z\mathbf{D}\phi = \mathbf{M}\phi \tag{B.5}$$

where z is some eigenvalue of $\mathbf{D}^{-1}\mathbf{M}$ and ϕ is the associated right eigenvector.

The eigenvalues for this system of equations are then obtained as the solution of a set of N quadratics

$$A(k)z^2 + B(k)z + C(k) = 0 \qquad k = 0, 1, ..., N \tag{B.6}$$

where $A(k), B(k), C(k)$ are defined as follows [85]:

$$A(k) = (N/2-k)^2 - (N/2-c)^2 \tag{B.7}$$

$$B(k) = 2(1-\lambda)(N/2-k)^2 - N(1+\lambda)(N/2-c) \tag{B.8}$$

$$C(k) = -(1+\lambda)^2[(N/2)^2 - (N/2-k)^2] \tag{B.9}$$

It should be noted that there are $N - [c]$ negative eigenvalues[1], which can be denoted as $z_{N-[c]-1} < ... < z_1 < z_0$. The largest negative eigenvalue z_0 is found to be

$$z_0 = -\frac{1+\lambda-N\lambda/c}{1-c/N} \tag{B.10}$$

[1] We let $[c]$ denote the integer part of c.

The ith component of the eigenvector corresponding to the eigenvalue z_k is

$$\phi_{ki} = (-1)^{N-i} \sum_{j=0}^{k} \binom{k}{j} \binom{N-k}{i-j} r_1^{k-j} r_2^{N-k-i+j} \qquad 0 \leq i \leq N \tag{B.11}$$

where

$$r_1 = (-(z_k + 1 - \lambda) + \sqrt{(z_k + 1 - \lambda)^2 + 4\lambda})/2\lambda \tag{B.12}$$

$$r_2 = (-(z_k + 1 - \lambda) - \sqrt{(z_k + 1 - \lambda)^2 + 4\lambda})/2\lambda \tag{B.13}$$

Finally, solutions to the differential equations in (B.3) can be written as

$$\mathbf{F}(x) = \mathbf{F}(\infty) + \sum_{i=0}^{N-[c]-1} e^{z_i x} \alpha_i \phi_i \tag{B.14}$$

where $\mathbf{F}(\infty)$ is the vector of equilibrium probabilities, which is given by

$$F_i(\infty) = \frac{1}{(1+\lambda)^N} \binom{N}{i} \lambda^i, \qquad 0 \leq i \leq N \tag{B.15}$$

The coefficients $\{\alpha_i\}$ in (B.14) must be obtained via boundary conditions. In the infinite buffer case with identical sources, it can be formulated such that the corresponding coefficient matrix is a Vandermonde matrix, and an analytical solution exists:

$$\alpha_j = -(\frac{\lambda}{1+\lambda})^N \prod_{i=0, i \neq j}^{N-[c]-1} \frac{z_i}{z_i - z_j}, \qquad 0 \leq j \leq N - [c] - 1 \tag{B.16}$$

The probability of overflow is defined as $G(x)$ =Pr[buffer content $> x$] and is thus given by

$$G(x) = - \sum_{i=0}^{N-[c]-1} e^{z_i x} \alpha_i (\mathbf{1}^T \phi_i) \tag{B.17}$$

where $\mathbf{1}$ denotes the identity vector.

B.2 Heterogeneous Sources

In this section, we extend the fluid flow queueing model for identical ON/OFF traffic sources to a system with heterogeneous ON/OFF sources. The equilibrium buffer distribution is again found as solution to a set of first order differential equations. However, that kind of closed-form solution described previously is now unavailable and the equations must be solved numerically. On the other hand, the size of the state space increases rapidly(geometrically) with the number of different sources. This "state explosion" problem poses various computational difficulties: enormous memory requirements, very long computation time, and complete breakdown of numerical algorithms. A so-called "decomposition method" has been developed in [90] to circumvent the above problem. It is shown that the separability property can permit a decomposition of the equations for the equilibrium probabilities of the system. The decomposition technique leads to a solution of the buffer distribution expressed as a sum of terms in Kronecker product form, hence reducing the computational complexity for large systems. We only give a very brief review of this approach here. Further details may be found in [90].

Consider a finite number of exponential ON/OFF sources transmitting data to a buffer. The sources are partitioned into m classes, all sources in a given class being statistically identical. Let N_j denote the number of sources in class j, $1/b_j(1/a_j)$ the average duration in the ON(OFF) state for sources in class j and R_j is the constant transmission rate of each source in class j in the ON state. The data transmitted from the sources are received by a buffer with service rate of C. The average input rate is $\sum_i N_i R_i a_i/(a_i + b_i)$, which is assumed smaller than C. Let k_i denote the number of active sources in class i, and $\mathbf{k} = (k_1, ..., k_m)$ the state vector which the sources are in. Let

$$\mathbf{S} = \{\mathbf{k} = (k_1, ..., k_m), 0 \leq k_i \leq N_i, i = 1, .., m\} \tag{B.18}$$

denote the state space for the sources. $\mathbf{S}$ is of cardinality $N = (N_1 + 1) \cdots (N_m + 1)$.

Again, the system is described as a $(N_1 + 1) \cdots (N_m + 1)$ dimentional matrix differential equation:

$$\mathbf{D}\frac{d}{dx}\mathbf{F}(x) = \mathbf{MF}(x) \tag{B.19}$$

where the element of $\mathbf{F}(x)$, $F_{\mathbf{k}}(x)$, is the probability that the sources are in state $\mathbf{k}$ and that the buffer content is less than or equal to x, $\mathbf{D}$ is a diagonal matrix with entry $(\mathbf{k}, \mathbf{k})$ equal to

$$d_{\mathbf{k}} = (\sum_{i=1}^{m} R_i k_i - C)$$

and entry $(\mathbf{k}, \mathbf{l})$ in $\mathbf{M}$ is as follows:

$$\begin{cases} m_{(\mathbf{k},\mathbf{k})} = -\sum_{i=1}^{m}((N_i - k_i)a_i + k_i b_i), & \text{for } \mathbf{k} \text{ in } \mathbf{S}, \\ m_{(\mathbf{k},k_1,\ldots,k_i-1,\ldots,k_m)} = (N_i - k_i + 1)a_i, & \text{for } \mathbf{k} \text{ in } \mathbf{S}, \\ m_{(\mathbf{k},k_1,\ldots,k_i+1,\ldots,k_m)} = (k_i + 1)b_i, & \text{for } \mathbf{k} \text{ in } \mathbf{S}, \\ m_{(\mathbf{k},\mathbf{l})} = 0, & \text{else.} \end{cases} \tag{B.20}$$

Suppose that the matrix $\mathbf{D}^{-1}\mathbf{M}$ has a basis of eigenvectors $\{\phi_{\mathbf{k}}\}$, and let $z_{\mathbf{k}}$ be the eigenvalue associated with $\phi_{\mathbf{k}}$. With this notation, the solution to (B.19) is given by

$$\mathbf{F}(x) = \sum_{\mathbf{k} \in \mathbf{S}} e^{z_{\mathbf{k}} x} \alpha_{\mathbf{k}} \phi_{\mathbf{k}} \tag{B.21}$$

where the coefficients $\alpha_{\mathbf{k}}$ must be found by means of the boundary conditions.

It has been shown that the eigenvalue $z_{\mathbf{k}}$ satisfies the following equation:

$$z_{\mathbf{k}}(C-\sum_{i=1}^{m} N_i/2R_i)-\sum_{i=1}^{m} N_i/2(a_i+b_i) = \sum_{i=1}^{m}(k_i-N_i/2)\sqrt{(z_{\mathbf{k}}R_i + b_i - a_i)^2 + 4a_ib_i} \quad \text{(B.22)}$$

The eigenvector correponding to the eigenvalue $z_{\mathbf{k}}$ is given as

$$\phi_{\mathbf{k}} = \phi_{k_1} \bigotimes \phi_{k_2} \bigotimes \cdots \bigotimes \phi_{k_m} \quad \text{(B.23)}$$

where $\bigotimes$ denotes the Kronecker product and ϕ_{k_i} is given as the coefficients in the following polynomial:

$$\Phi_{k_i}(x) = (x - r_i)^{k_i}(x - s_i)^{N_i-k_i} \quad \text{(B.24)}$$

where

$$r_i = (-(z_{\mathbf{k}}R_i + b_i - a_i) + \sqrt{(z_{\mathbf{k}}R_i + b_i - a_i)^2 + 4a_ib_i})/2a_i \quad \text{(B.25)}$$

$$s_i = (-(z_{\mathbf{k}}R_i + b_i - a_i) - \sqrt{(z_{\mathbf{k}}R_i + b_i - a_i)^2 + 4a_ib_i})/2a_i \quad \text{(B.26)}$$

Specifically, the eigenvector corresponding to the zero eigenvalue, i.e., the vector of equilibrium probabilities is:

$$\pi_{\mathbf{k}} = \prod_{i=1}^{m} \frac{1}{(1 + a_i/b_i)^{N_i}} \binom{N_i}{k_i} (a_i/b_i)^{k_i} \quad \text{(B.27)}$$

The unknown constant $\alpha_{\mathbf{k}}$ is determined below. For the infinite buffer case, the solution to (B.19) given in (B.21) is only composed of a set of "stable" modes, i.e., those terms in (B.21) with negative eigenvalues. Accordingly, let

$$\mathbf{\Phi} = (\phi_1, \phi_2, ..., \phi_N) = (\mathbf{\Phi}_u, \mathbf{\Phi}_s)$$

$$\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)^T = (\alpha_u^T, \alpha_s^T)^T$$

The partitions $\mathbf{\Phi}_u$ and $\mathbf{\Phi}_s$ represent the eigenvectors associated with unstable(u) and stable(s) modes respectively, where the equilibrium mode π is included with the unstable modes. The vector α is partitioned similarly. In the infinite buffer case, only the coefficients α_s are needed in the solution. Furthermore, we partition the states into an overload(o) set, $\mathbf{S}_o$, and an underload(u) set, $\mathbf{S}_u$, where

$$\mathbf{S}_o = \{\mathbf{k} : \sum_{i=1}^{m} k_iR_i > C\}, \qquad \mathbf{S}_u = \{\mathbf{k} : \sum_{i=1}^{m} k_iR_i < C\}.$$

Then letting

$$\mathbf{\Phi}_s = (\mathbf{\Phi}_{us}^T, \mathbf{\Phi}_{os}^T)^T,$$

$$\pi = (\pi_1, \pi_2, ..., \pi_N)^T = (\pi_u^T, \pi_o^T)^T,$$

we have the following linear equations determining the unknown coefficients of the stable modes, α_s:

$$\mathbf{\Phi}_{os}\alpha_s = -\pi_o \tag{B.28}$$

Now there are no further obstacles to a complete solution and we can get the buffer overflow probability by

$$G(x) = \mathbf{1}^T(\pi - \mathbf{F}) = -\sum_{\mathbf{k}:z_\mathbf{k}<0} e^{z_\mathbf{k}x}\alpha_\mathbf{k}(\mathbf{1}^T\phi_\mathbf{k}) \tag{B.29}$$

Appendix C

The MMPP/D/1/K Queue

In this appendix, we study the performance of a finite-buffered statistical multiplexer with MMPP input. The MMPP/D/1/K queue is a useful tool for modeling service systems with complex nonrenewal arrivals and has been successfully used in the analysis of ATM multiplexers loaded with the superposition of bursty sources. An exact analysis of the MMPP/D/1/K queue is carried out, yielding the cell loss probability.

C.1 The MMPP

The MMPP is a special case of the versatile Markovian process, or the so called N-process [129]. Arrivals are governed by a continuous-time, discrete-state Markov chain, $\mathcal{M}$, in the following manner. Let M be the number of states in $\mathcal{M}$ labeled $m = 1, 2, ..., M$. When the process is in state m, packets arrive according to a Poisson process with parameter λ_m.

C.2 Queueing Performance Analysis

Here we focus on modeling a statistical multiplexer with finite capacity K. We use the MMPP model developed previously as the input process and the service is deterministic. Hence the ATM cell multiplexer can be modeled as an MMPP/D/1/K queue. The N/G/1 queue with infinite buffer was studied in detail by Ramaswami [130] and more recently, Blondia [131] carried out an analysis of the N/G/1/K queue(of which the MMPP/D/1/K queue is a special case).

We assume that each packet requires Θ units of service [1]. The arrival process is generated by an M-state MMPP as outlined before. Let $\mathbf{Q} = [q_{i,j}]$ be the infinitesimal

[1] For the MMPP/G/1/K queue, the cumulative distribution function of the service time and the mean service time can be denoted by $\tilde{H}(.)$ and Θ, respectively.

generator of $\mathcal{M}$ and let $\mathbf{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_M)$. The stationary probability vector $\mathbf{x}$ of $\mathbf{Q}$ is obtained by solving the linear equations $\mathbf{xQ} = \mathbf{0}, \mathbf{xe} = 1$, where $\mathbf{e}$ is the M-dimensional unit vector whose elements are all equal to 1.

The queueing system can be described by a two-dimensional state variable $S(t) = \{X(t), J(t)\}$, where $X(t)$ denotes the number of customers in the system while $J(t)$ represents the state of the underlying Markov process $\mathbf{Q}$. The classical imbedded semi-Markov sequence approach is therefore used to determine the limiting probability distribution of $S(t)$ whose state space consists of $\{0, 1, ..., K\} \times \{1, 2, ..., M\}$. Let $(\tau_n : n \geq 0)$ be the successive epochs of departure. Then the sequence $\{S(\tau_n), \tau_{n+1} - \tau_n\}$ forms a semi-Markov sequence. We can write down the transition probability matrix for the imbedded Markov chain and then solve for the joint distribution of the number in the queue and the state of the MMPP at epochs of departure. This, in turn, can be used to get the performance measures of interest. The details of the solution will be discussed in the following.

Let (M_i, N_i) denote the state of the queue where M_i is the state of $\mathcal{M}$ immediately after the i-th departure and N_i is the number of packets left behind by the departing packet. We are interested in the behaviour of $\lim_{i \to \infty}(M_i, N_i)$, when it exists. This chain is aperiodic, recurrent and irreducible provided that $\mathcal{M}$ is aperiodic, recurrent, irreducible and $\lambda_m < \infty, 1 \leq m \leq M$. Let $\mathbf{T} = [t_{m,n;k,l}]$ be the transition probability matrix of the imbedded Markov chain, i.e., $t_{m,n;k,l} = \Pr[M_{i+1} = k, N_{i+1} = l | M_i = m, N_i = n]$.

We now define $M \times M$ matrices $\mathbf{A}_i$ and $\mathbf{B}_i, i \geq 0$ which define the transition probability matrix $\mathbf{T}$ [131]. Let $\mathbf{A}_i = [a_{i;m,k}]$, where $a_{i;m,k}$ is the conditional probability that i packets arrive and the resulting state of the uniformized version of $\mathcal{M}$ is k at the end of a service interval given that it was m at the beginning of the service interval. Let $\mathbf{A} = \sum_{i=0}^{\infty} \mathbf{A}_i$. The (i, j)th element of $\mathbf{A}$ gives the conditional probability of reaching state j at the end of a service time, starting from state i. Let $\mathbf{B}_i = [b_{i;m,k}]$, where $b_{i;m,k}$ denotes the probability of i packet arrivals and the resulting state of the uniformized version of $\mathcal{M}$ is k given that an idle period preceded the service period and that the state of the arrival process was m at the start of the idle period. Also, let $\mathbf{U} = (\mathbf{\Lambda} - \mathbf{Q})^{-1}\mathbf{\Lambda}$, whose (i, j)th element denotes the conditional probability of reaching state j at the end of an idle period, starting from state i. Then we have

$$\mathbf{B}_i = \mathbf{U}\mathbf{A}_i \tag{C.1}$$

The transition probability matrix can now be written as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \sum_{i=K-1}^{\infty} \mathbf{B}_i \\ \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \sum_{i=K-1}^{\infty} \mathbf{A}_i \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \sum_{i=K-2}^{\infty} \mathbf{A}_i \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sum_{i=1}^{\infty} \mathbf{A}_i \end{bmatrix} \tag{C.2}$$

Note that obtaining $\mathbf{T}$ requires the computation of $\mathbf{A}, \mathbf{U}$, and $\mathbf{A}_n$ for $n = 0, ..., K-2$.

The matrix $\mathbf{A}$ can be given as

$$\mathbf{A} = \int_0^{\infty} e^{\mathbf{Q}t} d\tilde{H}(t) \tag{C.3}$$

Since $\mathbf{xQ} = \mathbf{0}$, we have $\mathbf{xA} = \mathbf{x}$.

The matrices $\mathbf{A}_n$ can be efficiently computed by means of an iterative procedure based on the "technique of randomization" [132]. Define $\lambda = \max\{\lambda_m - q_{m,m}\}$,

$$r_{i,j} = \begin{cases} q_{i,j}/\lambda, & j \neq i, \\ (\lambda + q_{i,i})/\lambda, & j = i, \end{cases} \tag{C.4}$$

and

$$p'_{i,j} = \begin{cases} 0, & j \neq i, \\ \lambda_i/(q_{i,i} + \lambda), & j = i. \end{cases} \tag{C.5}$$

Let $\mathbf{P}$ be the matrix with elements $(1 - p'_{i,j})r_{i,j}$ and $\mathbf{P}'$ be the matrix with elements $p'_{i,j}r_{i,j}$. Then it is proved that

$$\mathbf{A}_i = \sum_{j=i}^{\infty} (\lambda\Theta)^j e^{-\lambda\Theta} \mathbf{R}_i^j / j! \tag{C.6}$$

where $\mathbf{R}_k^l$ is defined recursively by

$$\mathbf{R}_k^l = \begin{cases} \mathbf{I}, & k = 0, l = 0 \\ \mathbf{R}_k^{l-1}\mathbf{P} + \mathbf{R}_{k-1}^{l-1}\mathbf{P}', & 1 \leq k \leq l \end{cases} \tag{C.7}$$

where $\mathbf{I}$ is the identity matrix. For the numerical implementation, we can make a truncation of (C.6) for the computation of $\mathbf{A}_i$.

Let $\pi(i)$ be the M-dimensional vector whose jth element is the limiting probability at the imbedded epochs of having i users in the system and being in the state j of the MMPP, $i = 0, 1, ..., K-1$. Define $\mathbf{\Pi} = (\pi(0), \pi(1), ..., \pi(K-1))$. Then we have the following linear equations:

$$\mathbf{\Pi} = \mathbf{\Pi}\mathbf{T}, \qquad \sum_{i=0}^{K-1} \pi(i)\mathbf{e} = 1 \tag{C.8}$$

Let $\mathbf{y}(i)$ be the M-dimensional vector whose jth element is the limiting probability at an arbitrary time instant of having i users in the system and being in the state j of the MMPP, $i = 0, 1, ..., K-1$. We note that $\mathbf{y}(i)\mathbf{e}$ is the stationary probability that there are i users in the system at an arbitrary epoch. It can be proved that [131]:

$$\mathbf{y}(0) = \pi(0)(\mathbf{\Lambda} - \mathbf{Q})^{-1}[\Theta + \pi(0)(\mathbf{\Lambda} - \mathbf{Q})^{-1}\mathbf{e}]^{-1} \tag{C.9}$$

To determine the loss probability P_l in the MMPP/D/1/K queue, we use the principle of flow conservation. The effective arrival rate must be equal to the effective service rate. Let λ_{av} be the average arrival rate of the MMPP and λ_{av} is given by

$$\lambda_{av} = \mathbf{x}\mathbf{\Lambda}\mathbf{e} \tag{C.10}$$

Then, we have the relation

$$\lambda_{av}(1 - P_l) = \Theta^{-1}(1 - \mathbf{y}(0)\mathbf{e}) \tag{C.11}$$

Substitution of (C.9) into above yields

$$P_l = 1 - \lambda_{av}^{-1}[\Theta + \pi(0)(\mathbf{\Lambda} - \mathbf{Q})^{-1}\mathbf{e}]^{-1} \tag{C.12}$$

Therefore, the loss probability P_l can be computed as long as $\pi(0)$ is known.

C.3 An Efficient Way for Computing $\pi(0)$

In order to get $\pi(0)$, solving the MK-dimensional linear equation system (C.8) directly is computationally expensive (with the asymptotic complexity of $O(M^3K^3)$). Here we propose an alternative approach to reduce the computational complexity greatly. From (C.8), we observe that

$$\pi(i) = \pi(0)\mathbf{B}_i + \sum_{n=1}^{i+1} \pi(n)\mathbf{A}_{i-n+1}, \qquad 0 \leq i \leq K-2 \tag{C.13}$$

which suggests that there exists a $M \times M$ matrix sequence $\{\mathbf{F}_i\}$, independent of K, such that

$$\pi(i) = \pi(0)\mathbf{F}_i, \qquad 0 \leq i \leq K-1 \tag{C.14}$$

The matrix sequence $\{\mathbf{F}_i\}$ can be obtained recursively from

$$\mathbf{F}_0 = \mathbf{I}, \qquad \mathbf{F}_1 = (\mathbf{I} - \mathbf{B}_0)\mathbf{A}_0^{-1} \tag{C.15}$$

$$\mathbf{F}_{i+1} = (\mathbf{F}_i - \mathbf{B}_i - \sum_{n=1}^{i} \mathbf{F}_n\mathbf{A}_{i-n+1})\mathbf{A}_0^{-1}, \qquad i = 1, ..., K-2 \tag{C.16}$$

Next our task is to find $\pi(0)$. By summing (C.13) over $0 \leq i \leq K-2$, we get

$$\sum_{i=0}^{K-1} \pi(i)(\mathbf{I}-\mathbf{A}) = -\pi(0)(\mathbf{I}-\mathbf{U})\mathbf{A} \tag{C.17}$$

Adding $\sum_{i=0}^{K-1} \pi(i)\mathbf{ex} = \mathbf{x}$ to both sides of the above equation gives

$$\sum_{i=0}^{K-1} \pi(i)(\mathbf{I}-\mathbf{A}+\mathbf{ex}) = -\pi(0)(\mathbf{I}-\mathbf{U})\mathbf{A}+\mathbf{x} \tag{C.18}$$

From (C.14), we have

$$\pi(0)[\sum_{i=0}^{K-1} \mathbf{F}_i(\mathbf{I}-\mathbf{A}+\mathbf{ex})+(\mathbf{I}-\mathbf{U})\mathbf{A}] = \mathbf{x} \tag{C.19}$$

By noting that $\mathbf{xA} = \mathbf{x}$, we have $\mathbf{x}(\mathbf{I}-\mathbf{A}+\mathbf{ex}) = \mathbf{x}$. Multiplying $(\mathbf{I}-\mathbf{A}+\mathbf{ex})^{-1}$ to both sides of (C.19), we have

$$\pi(0)[\sum_{i=0}^{K-1} \mathbf{F}_i + (\mathbf{I}-\mathbf{U})\mathbf{A}(\mathbf{I}-\mathbf{A}+\mathbf{ex})^{-1}] = \mathbf{x} \tag{C.20}$$

The matrix within square brackets in (C.20) is nonsingular because the steady-state imbedded Markov chain probability distribution of the finite queueing system always exists and is unique. Therefore $\pi(0)$ can be computed from (C.20), in which the matrices $\mathbf{F}_i, i = 1, .., K-1$ can be obtained from (C.16). It should be pointed out that the above algorithm is particularly efficient for computing $\mathbf{\Pi}$ for different values of the buffer size since the sequences $\{\mathbf{A}_i\}$ and $\{\mathbf{F}_i\}$ need only be computed once.

As an example, Figure C.1 shows the loss probability as a function of the buffer size for a two-state MMPP. Assuming the constant value of the service time as the time unit, the MMPP parameters are given by

$$\mathbf{\Lambda} = \begin{bmatrix} 1.0722 & 0 \\ 0 & 0.48976 \end{bmatrix} \tag{C.21}$$

$$\mathbf{Q} = \begin{bmatrix} -8.4733 \times 10^{-4} & 8.4733 \times 10^{-4} \\ 5.0201 \times 10^{-6} & -5.0201 \times 10^{-6} \end{bmatrix} \tag{C.22}$$

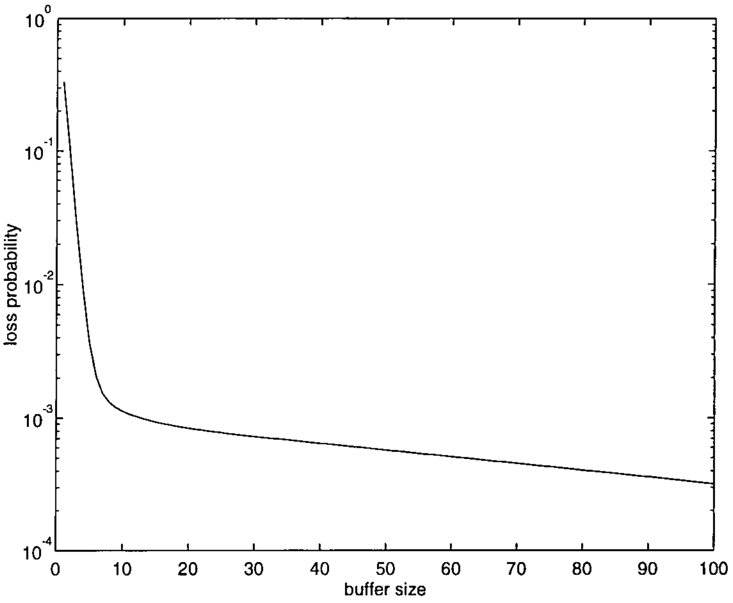

Figure C.1: Loss probability vs. buffer size

Appendix D

A Brief Introduction to Large Deviations

Large deviations refers to a collection of techniques for estimating properties of rare events such as their frequency and most likely manner of occurrence. It applies to certain types of rare events, which are caused by a large number of unlikely things occuring together, things that look in retrospect like a conspiracy, rather than a single event of small probability. Recently large deviations has been widely used in the area of ATM networks. ATM is a packet switching standard that is generally understood to have cell loss probabilities on the order of 10^{-6} to 10^{-9}. This means that many aspects of operating ATM networks need to have remarkably small error rates. Therefore, admission control, buffer dimensioning, and even the simulation of ATM models have stochastic behaviour falling in the domain of large deviations. In this appendix, we give a brief review of some of the basic theorems of large deviations. For a complete reference on the subject, see [111] [133].

D.1 Cramer's Theorem

Let's begin with the simplest large deviations question: what is

$$\Pr(\frac{1}{n}\sum_{i=1}^{n} x_i \geq a)$$

for i.i.d. random variables x_i, where $a > \mathrm{E}(x_i)$? Cramer's theorem(also known as Chernoff's theorem) establishes both upper and lower bounds that are the same to leading order. Let the i.i.d random variables $x_1, x_2, ...$ have common distribution function, and assume the mean $\mathrm{E}x_1$ exists. Define

$$M(\theta) = \mathrm{E}e^{\theta x_1}, \varphi(\theta) = \log M(\theta), \varphi^*(a) = \sup_{\theta}(\theta a - \varphi(\theta)) \qquad \text{(D.1)}$$

The transformation applied to φ in (D.1) is called the convex transform or Legendre transform. Cramer's theorem says:

Theorem 1 (Cramer) Let $x_1, x_2, ...$ be i.i.d. random variables. Then the function φ^* defined in (D.1) is lower semicontinuous[1] and convex. For any closed set F

$$\limsup_{n\to\infty} \frac{1}{n} \log \Pr(\frac{x_1 + \cdots + x_n}{n} \in F) \leq -\inf_{a\in F} \varphi^*(a) \tag{D.2}$$

and for any open set G

$$\liminf_{n\to\infty} \frac{1}{n} \log \Pr(\frac{x_1 + \cdots + x_n}{n} \in G) \geq -\inf_{a\in G} \varphi^*(a) \tag{D.3}$$

Now applying the above theorem to our basic question, we have

Theorem 2 Let $x_1, x_2, ...$ be i.i.d. random variables. For every $a >$Ex_1 and positive integer n

$$\Pr(\frac{x_1 + \cdots + x_n}{n} \geq a) \leq e^{-n\varphi^*(a)} \tag{D.4}$$

Assume that $M(\theta) < \infty$ for θ in some neighborhood of 0 and that (D.4) holds for some θ^* in the interior of that neighborhood. Then for every $\epsilon > 0$ there exists an integer n_0 such that whenever $n > n_0$

$$\Pr(\frac{x_1 + \cdots + x_n}{n} \geq a) \geq e^{-n(\varphi^*(a)+\epsilon)} \tag{D.5}$$

Next we introduce a refinement of Cramer's theorem due to Bahadur and Rao (see [111], pp. 94). It involves a more accurate estimate of the law μ_n of $S_n = \frac{1}{n}\sum_{i=1}^n x_i$. Specifically, for a "nice" set A, one seeks an estimate J_n^{-1} of $\mu_n(A)$ in the sense that $\lim_{n\to\infty} J_n \mu_n(A) = 1$.

Theorem 3 (Bahadur-Rao) Let $S_n = \frac{1}{n}\sum_{i=1}^n x_i$, where x_i are i.i.d. real valued random variables with logarithmic moment generating function $\varphi(\theta) = \log \mathrm{E}e^{\theta x_1}$. Let μ_n denote the law of S_n, i.e., μ_n is the probability distribution of S_n. Consider the set $A = [q, \infty)$, where $q = \varphi'(\eta)$ for some positive $\eta \in \mathcal{D}_\varphi^o$.[2]
(a) If the law of x_1 is non-lattice, then

$$\lim_{n\to\infty} J_n \mu_n(A) = 1 \tag{D.6}$$

where $J_n = \eta\sqrt{\varphi''(\eta) 2\pi n} e^{n\varphi^*(q)}$.
(b) Suppose x_1 has a lattice law, i.e., for some x_0, d, the random variable $d^{-1}(x_1 - x_0)$ is (a.s.) an integer number, and d is the largest number with this property. Assume

[1] That is, if $y_1, y_2, ...$ is a sequence so that $y_n \to y$ as $n \to \infty$, then $\liminf_n \varphi^*(y_n) \geq \varphi^*(y)$.
[2] $\mathcal{D}_\varphi^o$ is the interior of the effective domain $\mathcal{D}_\varphi = \{\theta : \varphi(\theta) < \infty\}$ of $\varphi(\theta)$.

further that $1 > \Pr(x_1 = q) > 0$. Then

$$\lim_{n\to\infty} J_n \mu_n(A) = \frac{\eta d}{1 - e^{-\eta d}} \tag{D.7}$$

D.2 The Large Deviations Principle

Theorems in large deviations are usually stated in terms of a Large Deviations Principle(LDP). A general statement of this principle is given in the following.

Definition 1 We are given a sequence of processes or random variables $z_1, z_2, \ldots$ on a state space X. We say that $z_1, z_2, \ldots$ satisfies an LDP with rate function I if, for every closed set $F \subset X$, we have

$$\limsup_{n\to\infty} \frac{1}{n} \log \Pr(z_n \in F) \leq - \inf_{x \in F} I(x) \tag{D.8}$$

and for every open set $G \subset X$, we have

$$\liminf_{n\to\infty} \frac{1}{n} \log \Pr(z_n \in G) \geq - \inf_{x \in G} I(x) \tag{D.9}$$

Definition 2 A real valued function I on X is called a *rate function* if it satisfies the following requirements.

(a) $I(x) \geq 0$,

(b) I is lower semicontinuous.

If in addition the set $\{x : I(x) \leq a\}$ is compact for every real a, then I is called a *good rate function.*

In Cramer's theorem, we consider the random variables $z_n = \frac{1}{n}(x_1 + \cdots + x_n)$. The function φ^* defined in (D.1) is a rate function. If $\lim_{|a|\to\infty} \varphi^*(a) = \infty$, then it is a good rate function.

D.3 The Gartner-Ellis Theorem

The Gartner-Ellis theorem establishes the existence of an LDP with convex good rate function for a large class of sources. Given a sequence $s_1, s_2, \ldots$ of random variables, define

$$\varphi_n(\theta) = \frac{1}{n} \log \mathrm{E} e^{\theta s_n} \tag{D.10}$$

We will estimate the asymptotics of $\{s_n/n\}$. First we have following assumptions.

Assumption 1 The limit $\lim_{n\to\infty} \varphi_n(\theta) = \varphi(\theta)$ exists(possibly infinite) pointwise.

Assumption 2 φ is differentiable on $\mathcal{D}_\varphi$.

Let $\varphi^*(.)$ be the Legendre transform of $\varphi(.)$, with $\mathcal{D}_{\varphi^*} = \{x : \varphi^*(x) < \infty\}$.

Theorem 4 (Gartner-Ellis) (a) Under Assumption 1, for $-\infty < a < b < \infty$, if $[a, b] \cap \mathcal{D}_{\varphi^*} \neq \emptyset$, then

$$\limsup_{n\to\infty} \frac{1}{n} \log \Pr(\frac{S_n}{n} \in [a, b]) \leq - \inf_{x\in[a,b]} \varphi^*(x) \tag{D.11}$$

(b) Under Assumption 1 and Assumption 2, let $-\infty < a < b < \infty$ and assume that for any $v \in (a, b)$, there exists θ_v such that $\varphi'(\theta_v) = v$. Then

$$\liminf_{n\to\infty} \frac{1}{n} \log \Pr(\frac{S_n}{n} \in (a, b)) \geq - \inf_{x\in(a,b)} \varphi^*(x) \tag{D.12}$$

This result applies to i.i.d sequences with $\mathrm{E}e^{\theta x_1} < \infty$ for all θ, which corresponds to the original large deviation estimate of Cramer. The result also applies to sequences with weak dependencies. For example, (random) coordinate functions of Markov chains satisfying strong uniformity conditions on the transition kernel and tails will satisfy an LDP. For stationary sequences satisfying appropriate mixing and tail conditions similar results hold.

Appendix E

Publications

1. A critical review of congestion control in ATM networks, *2nd Communication Networks Symposium*, Manchester, UK, July 1995, pp. 137-140.

2. ATM traffic prediction using FIR neural networks, *3rd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley, UK, July 1995, pp. 34/1-34/10. extended version in second volume of *ATM Networks — Performance Modelling and Evaluation*, D. Kouvatsos Editor, Chapman and Hall, London, 1996, pp. 74-91.

3. Access flow control for ATM networks using a neural network traffic predictor, *IEE 13th UK Teletraffic Symposium*, Glasgow, UK, March 1996, pp. 3/1-3/7.

4. Large deviations approximation for ATM multiplexers fed by Markov fluid sources, *IEE 13th UK Teletraffic Symposium*, Glasgow, UK, March 1996, pp. 5/1-5/11.

5. Performance analysis of an ATM cell multiplexer with MMPP input and a neural connection admission control approach, *International Conference on Communication Technology(ICCT'96)*, Beijing, China, May 1996, pp. 916-919.

6. Application of artificial neural networks to effective bandwidth estimation in ATM networks, *IEEE International Conference on Neural Networks(ICNN'96)*, Washington DC, USA, June 1996, pp. 1951-1956.

7. A congestion controller for ATM networks using reinforcement learning, *9th Yale Workshop on Adaptive and Learning Systems*, Yale University, CT, USA, June 1996, pp. 77-82.

8. A neural network approach to admission control in ATM networks, *World Congress on Neural Networks(WCNN'96)*, San Diego, CA, USA, Sept. 1996.

9. An effective bandwidth approach to connection admission control for multimedia traffic in ATM networks, *Electronics Letters*, vol. 32, no. 16, pp. 1438-1439,

August 1996.

10. An accurate approximation of cell loss probability for self-similar traffic in ATM networks, *Electronics Letters*, vol. 32, no. 19, pp. 1749-1751, Sept. 1996.

11. The impact of the Hurst parameter and its crossover effect on long-range dependent traffic engineering, *IEE 14th UK Teletraffic Symposium*, Manchester, UK, March 1997, pp. 10/1-10/8.

12. Dynamic routing in ATM networks with effective bandwidth estimation by neural networks, *IEEE International Workshop on Applications of Neural Networks to Telecommunications(IWANNT'97)*, Melbourne, Australia, June 1997, pp. 45-53.

13. Multiplexing gains in ATM networks, *5th IFIP Workshop on the Performance Modelling and Analysis of ATM Networks*, Ilkley, UK, July 1997, pp. 56/1-56/10.

14. An NN-based dynamic time-slice scheme for bandwidth allocation in ATM networks, accepted by *IEEE First International Conference on Information, Communications and Signal Processing(ICICS'97)*, Singapore, September 1997.

15. Fast and accurate estimation of ATM quality of service parameters with applications to call admission control, accepted by *IEEE First International Conference on Information, Communications and Signal Processing(ICICS'97)*, Singapore, September 1997.

16. Self-similar traffic generation and parameter estimation using wavelet transform, accepted by *IEEE GLOBECOM*, Phoenix, AZ, USA, November 1997.

17. An access flow control scheme for ATM networks using neural-network-based traffic prediction, accepted by *IEE Proceedings-Communications*, April 1997.

References

[1] ITU-T Recommendation I.113. *Vocabulary of terms for broadband aspects of ISDN*, 1993.

[2] ITU-T Recommendation I.211. *BISDN service aspects*, 1993.

[3] R. O. Onvural. *Asynchronous Transfer Mode Networks.* Artech House, 1994.

[4] M. De Prycker. *Asynchronous Transfer Mode.* Ellis Horwood, 1993.

[5] C. T. Lea. What should be the goal for ATM. *IEEE Network*, pages 60–66, September 1992.

[6] M. De Prycker, R. Peschi, and T. Van Langdegem. B-ISDN and the OSI protocol reference model. *IEEE Network*, pages 10–18, March 1993.

[7] ITU-T Recommendation I.362. *BISDN ATM Adaptation Layer(AAL) functional description*, 1993.

[8] R. Jain. Congestion control and traffic management in ATM networks: recent advances and a survey. *Computer Networks and ISDN Systems*, 1995. to appear.

[9] S. E. Minzer. Broadband ISDN and asynchronous transfer mode(ATM). *IEEE Commun. Magazine*, pages 17–24, September 1989.

[10] H. Gilbert, O. Aboul-Magd, and V. Phung. Developing a cohesive traffic management strategy for ATM networks. *IEEE Commun. Magazine*, pages 36–45, October 1991.

[11] M. Wernik, O. Aboul-Magd, and H. Gilbert. Traffic management for B-ISDN services. *IEEE Network*, pages 10–19, September 1992.

[12] ITU-T Recommendation I.371. *Traffic control and congestion control in B-ISDN*, 1993.

[13] A. E. Eckberg, B. T. Doshi, and R. Zoccolillo. Controlling congestion in B-ISDN/ATM: issues and strategies. *IEEE Commun. Magazine*, pages 64–70, September 1991.

[14] A. E. Eckberg. B-ISDN/ATM traffic and congestion control. *IEEE Network*, pages 28–37, September 1992.

[15] D. Hong and T. Suda. Congestion control and prevention in ATM networks. *IEEE Network*, pages 10–16, July 1991.

[16] G. M. Woodruff and R. Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE J. Selected Areas in Commun.*, 8(3):437–446, 1990.

[17] I. W. Habib and T. N. Saadawi. Controlling flow and avoiding congestion in broadband networks. *IEEE Commun. Magazine*, pages 46–53, October 1991.

[18] M. Decina and T. Toniatti. On bandwidth allocation to bursty virtual connections in ATM networks. In *IEEE ICC*, pages 844–851, 1990.

[19] G. Gallassi, G. Rigolio, and L. Fratta. ATM: Bandwidth assignment and bandwidth enforcement policies. In *IEEE GLOBECOM*, pages 1788–1793, 1989.

[20] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE J. Selected Areas in Commun.*, 9(7):968–981, 1991.

[21] J. S. Turner. Managing bandwidth in ATM networks with bursty traffic. *IEEE Network*, pages 50–58, September 1992.

[22] M. Ilyas and H. T. Mouftah. Performance evaluation of congestion avoidance in broadband ISDN. In *IEEE ICC*, pages 727–731, 1990.

[23] H. Saito and K. Shiomoto. Dynamic call admission control in ATM networks. *IEEE J. Selected Areas in Commun.*, 9(7):982–989, 1991.

[24] J. S. Turner. New directions in communications(or which way to the information age?). *IEEE Commun. Magazine*, pages 8–15, October 1986.

[25] M. Butto, E. Cavallero, and A. Tonietti. Effectiveness of the "leaky bucket" policing mechanism in ATM networks. *IEEE J. Selected Areas in Commun.*, 9(3):335–342, 1991.

[26] V. Bemmel and M. Ilyas. A novel congestion control strategy in ATM networks. *Computers and Industrial Engineering*, 25:549–552, 1993.

[27] E. Rathgeb. Modelling and performance comparison of policing mechanisms for ATM networks. *IEEE J. Selected Areas in Commun.*, 9(3):325–334, 1991.

[28] P. E. Boyer, F. M. Guillemin, M. J. Servel, and J. P. Coudreuse. Spacing cells protects and enhances utilization of ATM network links. *IEEE Network*, pages 38–49, September 1992.

[29] L. Trajkovic and S. J. Golestani. Congestion control for multimedia services. *IEEE Network*, pages 20–26, September 1992.

[30] T. Y. Huang and J. L. C. Wu. Performance analysis of a dynamic priority scheduling method in ATM networks. *IEE Proc. -I*, 140(4):285–290, 1993.

[31] P. Newman. Traffic management for ATM local area networks. *IEEE Commun. Magazine*, pages 44–50, August 1994.

[32] H. T. Kung and R. Morris. Credit-based flow control for ATM networks. *IEEE Network*, pages 40–48, March 1995.

[33] F. Bonomi and K. W. Fendick. The rate-based flow control framework for the available bit rate ATM service. *IEEE Network*, pages 25–39, March 1995.

[34] K. Y. Siu and H. Y. Tzeng. Intelligent congestion control for ABR service in ATM networks. *Computer Communication Review*, 24(5):81–106, 1994.

[35] K. K. Ramakrishnan and P. Newman. Integration of rate and credit schemes for ATM flow control. *IEEE Network*, pages 49–56, March 1995.

[36] P. E. Boyer and D. P. Tranchier. A reservation principle with applications to the ATM traffic control. *Computer Networks and ISDN Systems*, 24:321–334, 1992.

[37] M. E. Anagnostou, M. E. Theologou, and E. N. Protonotarios. Cell insertion ratio analysis in asynchronous transfer mode networks. *Computer Networks and ISDN Systems*, 24:335–344, 1992.

[38] I. Chlamtac and T. Zhang. A counter based congestion control(CBC) for ATM networks. *Computer Networks and ISDN Systems*, 26:5–27, 1993.

[39] I. Khan and V. O. K. Li. Traffic control in ATM networks. *Computer Networks and ISDN Systems*, 27:85–100, 1994.

[40] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching networks: the case of a single congested node. *IEEE/ACM Trans. Networking*, 1(6):693–708, 1993.

[41] A. A. Tarraf, I. W. Habib, and T. N. Saadawi. Intelligent traffic control for ATM broadband networks. *IEEE Commun. Magazine*, pages 76–82, October 1995.

[42] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Macmillan College Publishing Company, 1994.

[43] D. E. Rumelhart, J. L. McClelland, and the PDP Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, 1986.

[44] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[45] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1(1):4–27, 1990.

[46] X. Chen and I. M. Leslie. Neural adaptive congestion control for broadband ATM networks. *IEE Proceedings-I*, 139(3):233–240, 1992.

[47] M. K. Mehmet Ali and F. Kamoun. Neural networks for shortest path computation and routing in computer networks. *IEEE Trans. Neural Networks*, 4(6):941–954, 1993.

[48] R. Fantacci, M. Forti, and M. Marini. Efficient fast packet switching fabric using neural networks. *Electronics Letters*, 30(13):1077–1078, 1994.

[49] R. J. T. Morris and B. Samadi. Neural network control of communications sytems. *IEEE Trans. Neural Networks*, 5(4):639–650, 1994.

[50] Y. K. Park and G. Lee. Applications of neural networks in high-speed communication networks. *IEEE Commun. Magazine*, pages 68–74, October 1995.

[51] X. Chen and J. F. Hayes. Access control in multicast packet switching. *IEEE/ACM Trans. Networking*, 1(6):638–649, 1993.

[52] O. Gallmo and L. Asplund. Reinforcement learning by construction of hypothetical targets. In J. Alspector, R. Goodman, and T. X. Brown, editors, *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications 2*, pages 300–307, 1995.

[53] E. Nordstrom and J. Carlstrom. A reinforcement learning scheme for adaptive link allocation in ATM networks. In J. Alspector, R. Goodman, and T. X. Brown, editors, *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications 2*, pages 88–95, 1995.

[54] 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*, 2(1):1–15, 1994.

[55] A. Hiramatsu. ATM communications network control by neural networks. *IEEE Trans. Neural Networks*, 1(1):122–130, 1990.

[56] A. Hiramatsu. Training techniques for neural network applications in ATM. *IEEE Commun. Magazine*, pages 58–67, October 1995.

[57] J. E. Neves, M. J. Leitao, and L. B. Almeida. Neural networks in B-ISDN flow control: ATM traffic prediction or network modeling? *IEEE Commun. Magazine*, pages 50–56, October 1995.

[58] A. Hiramatsu. Integration of ATM call admission control and link capacity control by distributed neural networks. *IEEE J. Selected Areas in Commun.*, 9(7):1131–1138, 1991.

[59] A. A. Tarraf, I. W. Habib, and T. N. Saadawi. A novel neural network traffic enforcement mechanism for ATM networks. *IEEE J. Selected Areas in Commun.*, 12(6):1088–1096, 1994.

[60] Y. C. Liu and C. Douligeris. Static vs. adaptive feedback congestion controller for ATM networks. In *IEEE GLOBECOM*, pages 291–295, 1995.

[61] T. Yamada and T. Yabuta. Neural network controller using autotuning method for nonlinear functions. *IEEE Trans. Neural Networks*, 3(4):595–601, 1992.

[62] J. Filipiak. *Modelling and Control of Dynamic Flows in Communication Networks.* Springer-Verlag, 1988.

[63] R. G. Cheng and C. J. Chang. Design of a fuzzy traffic controller for ATM networks. *IEEE/ACM Trans. Networking*, 4(3):460–469, 1996.

[64] V. Catania, G. Ficili, S. Palazzo, and D. Panno. A comparative analysis of fuzzy versus conventional policing mechanisms for ATM networks. *IEEE/ACM Trans. Networking*, 4(3):449–459, 1996.

[65] L. D. Chou and J. L. C. Wu. Parameter adjustment using neural-network-based genetic algorithms for guaranteed QOS in ATM networks. *IEICE Trans. Commun.*, 78(4):572–579, 1995.

[66] K. Sriram and W. Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE J. Selected Areas in Commun.*, 4(6):833–846, 1986.

[67] H. Heffes and D. M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Selected Areas in Commun.*, 4(6):856–868, 1986.

[68] I. W. Habib and T. N. Saadawi. Multimedia traffic characteristics in broadband networks. *IEEE Commun. Magazine*, pages 48–54, July 1992.

[69] J. N. Daigle and J. D. Langford. Models for analysis of packet voice communications systems. *IEEE J. Selected Areas in Commun.*, 4(6):847–855, 1986.

[70] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[71] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[72] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks: prediction and system modeling. Technical report, Los Alamos National Laboratory, 1987.

[73] K. J. Lang and G. E. Hinton. The development of the time-delay neural network architecture for speech recognition. Technical report, Department of Computer Science, Carnegie-Mellon University, 1988.

[74] E. A. Wan. Time series prediction by using a connectionist network with internal delay lines. In A. S. Weigend and N. A. Gershenfeld, editors, *Time Series Prediction*, pages 195–217. Addison-Wesley Publishing Company, 1994.

[75] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. Commun.*, 36(7):834–843, 1988.

[76] J. J. Bae and T. Suda. Survey of traffic control schemes and protocols in ATM networks. *Proceedings of the IEEE*, 79(2):170–189, 1991.

[77] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[78] S. N. Rasband. *Chaotic Dynamics of Nonlinear Systems.* Wiley, 1990.

[79] S. Chong, S. Q. Li, and J. Ghosh. Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM. *IEEE J. Selected Areas in Commun.*, 13(1):12–23, 1995.

[80] J. T. Amenyo, A. A. Lazar, and G. Pacifici. Cooperative distributed scheduling for ATS-based broadband networks. Technical report, CTR, Columbia University, 1991.

[81] A. Atai and J. Y. Hui. A rate-based feedback traffic controller for ATM networks. In *IEEE ICC*, pages 1605–1615, 1994.

[82] S. Yazid and H. T. Mouftah. Congestion control methods for BISDN. *IEEE Commun. Magazine*, pages 42–47, July 1992.

[83] I. W. Habib and T. N. Saadawi. Access flow control algorithms in broadband neworks. *Computer Communications*, 15(5):326–332, 1992.

[84] T. Murase, H. Suzuki, S. Sato, and T. Takeuchi. A call admission control scheme for ATM networks using a simple quality estimate. *IEEE J. Selected Areas in Commun.*, 9(9):1461–1470, 1991.

[85] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal*, 61:1871–1894, 1982.

[86] D. Mitra. Stochastic theory of a fluid model of producers and consumers coupled by a buffer. *Advances in Applied Probability*, 20:646–676, 1988.

[87] R. C. F. Tucker. Accurate method for analysis of a packet-speech multiplexer with limited delay. *IEEE Trans. Commun.*, 36(4):479–483, 1988.

[88] R. Nagarajan, J. Kurose, and D. Towsley. Approximation techniques for computing packet loss in finite-buffered voice multiplexers. *IEEE J. Selected Areas in Commun.*, 9(3):368–377, 1991.

[89] J. W. Roberts. Variable-bit-rate traffic control in B-ISDN. *IEEE Commun. Magazine*, pages 50–56, September 1991.

[90] T. E. Stern and A. I. Elwalid. Analysis of separable Markov-modulated rate models for information-handling systems. *Advances in Applied Probability*, 23:105–139, 1991.

[91] A. I. Elwalid and D. Mitra. Effective bandwidth of general Markovian traffic sources and admission control of high speed networks. *IEEE/ACM Trans. Networking*, 1(3):329–343, 1993.

[92] R. J. Gibbens and P. J. Hunt. Effective bandwidths for the multi-type UAS channel. *Queueing Systems*, 9:17–28, 1991.

[93] G. L. Choudhury, D. M. Lucantoni, and W. Whitt. Squeezing the most out of ATM. *IEEE Trans. Commun.*, 44(2):203–217, 1996.

[94] K. Sriram. Methodologies for bandwidth allocation, transmission scheduling, and congestion avoidance in broadband ATM networks. *Computer Networks and ISDN Systems*, 26:43–59, 1993.

[95] S. Gupta, K. W. Ross, and M. El Zarki. On routing in ATM networks. In M. Steenstrup, editor, *Routing in Communications Networks*, pages 49–74. Prentice Hall, 1995.

[96] H. W. Chu and D. H. K. Tsang. Dynamic routing algorithms in VP-based ATM networks. In *IEEE GLOBECOM*, pages 1364–1368, 1995.

[97] J. W. Lee and B. G. Lee. Performance analysis of ATM cell multiplexer with MMPP input. *IEICE Trans. Commun.*, 75(8):709–714, 1992.

[98] G. Kesidis, J. Walrand, and C. S. Chang. Effective bandwidths for multiclass Markov fluids and other ATM sources. *IEEE/ACM Trans. Networking*, 1(4):424–428, 1993.

[99] N. G. Duffield, J. T. Lewis, N. O'Connell, R. Russell, and F. Toomey. Entropy of ATM traffic streams: a tool for estimating QOS parameters. *IEEE J. Selected Areas in Commun.*, 13(6):981–990, 1995.

[100] S. Crosby, I. Leslie, J. T. Lewis, N. O'Connell, R. Russell, and F. Toomey. Bypassing modelling: an investigation of entropy as a traffic descriptor in the Fairisle ATM network. In *12th UK Teletraffic Symp.*, pages 23/1–23/10, 1995.

[101] A. Simonian and J. Guibert. Large deviations approximation for fluid queues fed by a large number of on/off sources. *IEEE J. Selected Areas in Commun.*, 13(6):1017–1027, 1995.

[102] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. R. Weber. Admission control and routing in ATM networks using inferences from measured buffer occupancy. *IEEE Trans. Commun.*, 43:1778–1784, 1995.

[103] N. G. Duffield and N. O'Connell. Large deviations and overflow probabilities for the general single-server queue, with applications. *Math. Proc. of the Cambridge Philosophical Society*, 118:363–374, 1995.

[104] F. P. Kelly. Effective bandwidths at multi-type queues. *Queueing Systems*, 9:5–15, 1991.

[105] C. S. Chang. Stability, queue length and delay of deterministic and stochastic queueing networks. *IEEE Trans. Auto. Control*, 39(5):913–931, 1994.

[106] W. Whitt. Tail probabilities with statistical multiplexing and effective bandwidths in multi-class queues. *Telecomm. Systems*, 2:71–107, 1993.

[107] D. N. C. Tse, R. G. Gallager, and J. N. Tsitsiklis. Statistical multiplexing of multiple time-scale Markov streams. *IEEE J. Selected Areas in Commun.*, 13(6):1028–1038, 1995.

[108] G. de Veciana, G. Kesidis, and J. Walrand. Resource management in wide-area ATM networks using effective bandwidths. *IEEE J. Selected Areas in Commun.*, 13(6):1081–1090, 1995.

[109] G. de Veciana and J. Walrand. Effective bandwidths: call admission, traffic policing and filtering for ATM networks. *Queueing Systems*, 20:37–59, 1995.

[110] C. S. Chang, P. Heidelberger, S. Juneja, and P. Shahabuddin. Effective bandwidth and fast simulation of ATM intree networks. *Performance Evaluation*, 20:45–65, 1994.

[111] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Jones and Bartlett Publishers, 1993.

[112] C. S. Chang and J. Thomas. Effective bandwidth in high-speed digital networks. *IEEE J. Selected Areas in Commun.*, 13(6):1091–1099, 1995.

[113] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[114] K. Sohraby. On the asymptotic behavior of heterogeneous statistical multiplexer with applications. In *IEEE INFOCOM*, pages 6c.3.1–6c.3.9, 1992.

[115] J. Y. Hui. Resource allocation for broadband networks. *IEEE J. Selected Areas in Commun.*, 6(9):1598–1608, 1988.

[116] I. Hsu and J. Walrand. Admission control for ATM networks. Technical report, Department of EECS, University of California, Berkeley, CA, 1994.

[117] E. Buffet and N. G. Duffield. Exponential upper bounds via martingales for multiplexers with Markovian arrivals. *J. Applied Probability*, 31(4):1049–1060, 1994.

[118] C. I. Ani, R. Ahmad, and F. Halsall. Methodology for derivation of network resources to support video-related services in ATM-based private wide-area networks. *IEE Proc. Commun.*, 142(4):233–237, 1995.

[119] D. Heyman, A. Tabatabai, and T. V. Lakshman. Statistical analysis and simulation study of video teleconference traffic in ATM networks. *IEEE Trans. Circuits and Systems for Video Tech.*, 2(2):49–59, 1992.

[120] S. Y. Li, T. Yang, J. Ni, and D. H. K. Tsang. Source modeling and queueing analysis of VBR video teleconference traffic in ATM networks. Technical report, Technical University of Nova Scotia, Canada, 1994.

[121] J. Ni and T. Yang. Source modeling, queueing analysis and bandwidth allocation of VBR MPEG-2 video traffic in ATM networks. Technical report, Technical University of Nova Scotia, Canada, 1995.

[122] R. Drossu, T. V. Lakshman, Z. Obradovic, and C. Raghavendra. Single and multiple frame video traffic prediction using neural network models. Technical report, Department of EECS, Washington State University, WA, USA, 1996.

[123] A. Pitsillides, Y. A. Sekercioglu, and G. Ramamurthy. Effective control of traffic flow in ATM networks using fuzzy explicit rate marking(FERM). *IEEE J. Selected Areas in Commun.*, 15(2):209–225, 1997.

[124] S. Youssef, I. W. Habib, and T. N. Saadawi. A neurocomputing controller for bandwidth allocation in ATM networks. *IEEE J. Selected Areas in Commun.*, 15(2):191–199, 1997.

[125] A. Sarajedini and P. M. Chau. Cumulative distribution estimation with neural networks. Technical report, Department of ECE, University of California, San Diego, CA, 1996.

[126] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *IEEE Trans. Commun.*, 43(2):1566–1579, 1995.

[127] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.

[128] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. Networking*, 4(2):209–223, 1996.

[129] M. F. Neuts. A versatile Markovian point process. *J. Appl. Prob.*, 16:764–779, 1979.

[130] V. Ramaswami. The N/G/1 queue and its detailed analysis. *Adv. Appl. Prob.*, 12:222–261, 1980.

[131] C. Blondia. The N/G/1 finite capacity queue. *Commun. Statist.-Stochastic Models*, 5:273–294, 1989.

[132] D. M. Lucantoni and V. Ramaswami. Efficient algorithms for solving the nonlinear matrix equations arising in phase type queues. *Commun. Statist.-Stochastic Models*, 1:29–51, 1985.

[133] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis.* Chapman and Hall, 1995.