Université d'Ottawa · University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	Cristian LAMBIRI
ADRESSE POSTALE / MAILING ADDRESS:	62 Bateman Drive Ottawa, ON K2G 5H5
GRADE / DEGREE:	**ANNÉE D'OBTENTION / YEAR GRANTED**
Ph.D.(Electrical Engineering)	2003
TITRE DE LA THÈSE / TITLE OF THESIS:	
On the Estimation and Control of Packet Loss For VPN Services	

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photograhie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

April 22, 2005

DATE

(AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

LAMBIRI, Cristian

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

On the Estimation and Control of Packet Loss For VPN Services

Dan Ionescu

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

D. Petriu

N. Georganas

V. Groza

S. Pierre

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

On the Estimation and Control of Packet Loss For VPN Services

by

Cristian Lambiri
M.A.Sc., University of Ottawa, 1995
Dipl. Eng., Technical University of Timisoara, 1992

Doctoral Dissertation

Submitted to the Faculty of Graduate Studies and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology
University of Ottawa
2003

© Cristian Lambiri, Ottawa, Canada, 2003

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-79305-2

Canada

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Cristian Lambiri

I further authorize the University of Ottawa to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Cristian Lambiri

Abstract

Virtual Private Services (VPN) are an important service offering for any service provider. It can be justified that in a metro environment some networks will carry only such traffic. In addition, metro networks are characterized by small link spans, which make the propagation delays negligible. In this environment, this dissertation proposes and investigates a service aware network control platform for packet switched networks that is technology agnostic.

The basic premises of our work are that VPN are provider based, have *well defined QoS behavior* specified by the long term loss, and have a degree of *user-controllability.*

The dissertation proposes a number of aggregate traffic loss estimators, based on the theory of large deviations. Our assumptions are that the buffers at each multiplexor are finite, and therefore the loss ratios are computed. It has been noted that large deviations based estimators can be estimated in two main ways:

- assuming a traffic model and by computing the loss using that model
- estimating the moment generating function directly.

The work compares, through a large number of simulations, the performance of the proposed estimators in both cases. The results show that under the Gaussian model the model-based estimation is very accurate. This conclusion is important because the model-based estimator has better computational complexity when compared with the modeless estimator. Therefore, it is better to be used in practice.

The dissertation proposes a new method for atomically admitting VPN service instances in the network. Dubbed Service Admission Control, this generalizes the classical admission control methods by using an atomic operation over a set of class of service instances. Furthermore, we propose and investigate a method for dynamic loss control that is of use when the traffic injected in the network has long timescale changes.

Acknowledgment

This thesis is dedicated to my family: my parents, Constantin and Viorica, whom I miss; my wife Eugenia Mihaela, for her love and caring, and my son Darius William.

I would also like to express my sincere gratitude to my supervisor, Professor Dr. Dan Ionescu, for his constant guidance, encouragement, and support throughout my research.

Contents

List of Figures

List of Tables

Notation

Various symbols, superscripts, subscripts, and abbreviations used frequently in this proposal are summarized below. All symbols are fully defined where first used in the text.

Acronyms and Definitions

ABR	Available Bit Rate
ARMA	Auto-Regressive Moving Average
ARIMA	Auto-Regressive Integrated Moving Average
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BGP	Border Gateway Protocol
BM	Brownian Motion
CAC	Connection Admission Control
CDF	Cumulative Distribution Function
CLI	Command Line Interface
COS	Class of Service
DAC	Direct Admission Control
DiffServ	Differentiated Services
EB	Effective Bandwidth
ECDF	Empirical Cumulative Distribution Function
ENE	Electrical Network Element
ETE	End-to-End

FBM	Factional Brownian Motion
FIFO	First in first out
GPS	Generalized Processor Sharing
IETF	Internet Engineering Task Force
iid	Independent and Identically Distributed
IntServ	Integrated Services
ITB	in-the-box
LBC	Leaky Bucket Controller
LAN	Local Area Network
LDP	Label Distribution Protocol
LPM	Longest Match Prefix
LSP	Label Switched Path
MAN	Metropolitan Area Network
MC	Mostly Closed
MBAC	Measurement Based Admission Control
MBSAC	Measurement Based Service Admission Control
MMP	Markov Modulated Process
MMPP	Markov Modulate Poisson Process
MO	Mostly Open
MPLS	Multi-Protocol Label Switching
NE	Network Element

NCP	Network Control Plane
NMS	Network Management System
OTB	Out-of-Box
OSPF	Open Shortest Path First
QoS	Quality of Service
PDF	Probability Density Function
PSNE	Packet Switched Network Element
RED	Random Early Detection
SAC	Service Admission Control
SC	Strategic Control
SLA	Service Level Agreement
SML	Service Management Logic
SMM	Service Management Module
SP	Service Provider
TC	Tactical Control
VPN	Virtual Private Network
WAN	Wide Area Network
WRR	Weight Round Robin
WFQ	Weighted Fair Queuing

Chapter 1

Introduction

Many medium and large corporations, as well as a growing number of small ones, have geographically distributed premises that have to be connected together over public[1] networks. Companies with such a geographical structure need to provide a uniform communication service that hides the distributed nature of its communication infrastructure. The logical network that connects such sites is called a *Virtual Private Network (VPN)* [56].

The idea of site connection over a network that is under the control of another entity has been around for some time. Originally, the solution to the VPN problem involved the setup of circuit-switched leased lines between end-points that form a service. Such a solution is very effective because it is secure, and provides deterministic Quality of Service (QoS) characteristics to its user. The main drawback is its high price tag. The same connectivity problem can be solved in a packet switched network, at a lower cost. The price decrease results from the sharing of network resources with other customers, thus reducing the total cost per service instance. In contrast, in circuit-switched networks, the user has to reserve a full transmission line for each pair of end-points that it wants connected or, more recently, a lambda circuit.

Sometimes, the characteristics of the circuit-switched VPN solution, which generate its higher cost, can also be perceived as advantages. For example, such a solution ensures complete traffic separation between service instances, which cannot be duplicated in

[1] *Public* in this context describes a network that offers services to users that do not belong to the organization that operates the network.

packet switched networks. In addition, if the amount of bandwidth required for the service is close to the transmission capacity of the core links through which the service passes, then the circuit-switched solution can prove to be more economical. Elements of the service control architecture presented in this dissertation can be applied to the setup and control of circuit switched services too. Nevertheless, this case is not treated in detail.

Packet switched networks offer either connectionless data transport, of which IP based networks are the best example, or connection oriented data transport, as it is the case for ATM and MPLS based networks. In connectionless networks, the main service that is offered is the *access* service, which provides guarantees of connectivity to the network. Connection oriented packet switched networks offer variants of the *end-to-end connectivity.* Using these basic services, an end user can create a VPN solution without the knowledge of the service provider. Similarly, the VPN solution can be offered by the network operator. It is this latter case that is studied in this dissertation.

Provider based VPN services for packet switched networks are an active area of research and development. ATM Forum describes such a service in one of its specifications [20], while IETF is also actively pursuing this service category [45] as part of the work done by the *Provider Provisioned Virtual Private Networks (PPVPN)* group. In our work, we assume that the network carries only VPN services. This assumption allows us to consider the network that offers the VPN service in isolation and present a control solution for this problem setup.

1.1 Motivation

This dissertation presents a service aware network control platform that is technology agnostic. The focus of the presentation is on the *quantitative* aspects of the VPN service, as defined by their QoS and topological characteristics, rather than the *qualitative* service characteristics. The basic premises of this work is that VPN are provider based, have *well defined QoS behavior*, and have a degree of *user-controllability.*

VPN services are governed by customer-provider contracts, known as Service Level Agreements (SLAs) [42], which oversee both the traffic characteristics, to and from

the end-points of the service, as well as the provider duties and penalties for failing to provide the service guarantees. SLAs contain sets of procedures and service targets agreed upon by customers and providers and define services in all of their aspects. These definitions include the qualitative and quantitative parameters of the service, as well as business aspects that define the relationship between the customer and the provider (billing, support etc.)

There are a large number of possible *administrative* scenarios for the setup of VPN services. The simplest case is when all VPN end-points are connected to the same *contiguous* administrative domain. Other, more complex, situations arise when administrative domains are separated (i.e. when the VPN points span a number of networks). In this case, the setup and control of the service involves administrative cooperation between SPs on top of the technical cooperation. Although important, the administrative and business interactions between providers are considered to be outside of the scope of this dissertation. Furthermore, it shall be considered that all sites that are part of a VPN service are connected to a network that is under the administrative control of one entity.

The measure of user satisfaction with a telecommunication service is dependent on two classes of parameters:

- The first is a class of features that characterizes a telecommunication service relative to its traffic performance. Known under the generic nomenclature of *Quality of Service (QoS)* parameters [53], they describe commitments by the network operator for numerically quantifiable quality parameters.

- QoS parameters are defined for the times when the network does not experience topological failures. For these events, a different class of parameters is defined, in terms of either the probability of traffic outage or in terms of maximum time periods without connectivity during a year of operation, due to provider side network failures. To decrease the possibility of service outage to negligible levels, the provider presents the user with the possibility of enhancing its service through extra reservation. This is a specifiable characteristic of the service, which serves as a differentiator between service instances. Ideally, the service restoration in

case of fault should be achieved in a time frame that is smaller than the timeout period for applications, but such stringent temporal requirements are enforceable only by using a small number of link layer technologies like SONET and SDH [19]. In this case, resources are reserved at link level and all services that pass across the protected link are restored. The same problem can be stated from a service perspective. This links the restoration with the notion of service and, in the event of a topological failure only the services that are protected will be restored.

QoS assurances are intrinsically linked to the problem of congestion control in the network. It can be safely stated that congestion control represents the key to providing QoS guarantees in packet switched networks, although the reverse is not, generally true. This problem has been recognized for a long time in packet switched networks, and many solutions, that work at different levels of the network [57], have been proposed and implemented. It is to be noted that, in general, several congestion control mechanisms coexist in a network and that they operate on different timescales.

The first network wide control loop can be found at connection level, where protocols like TCP and ABR use a *reactive* model to control the amount of traffic that they inject in the network based on the amount of loss that is experienced by the source. This adaptation works well in networks where more complex methods of congestion control are not used, as is the case in the original Internet design. A large portion of the Internet traffic is using this type of mechanism, because of the use of TCP as the basis for web information exchanges. Nevertheless, using only this congestion control mechanism in a network does not permit the network to make any QoS guarantees for the traffic.

To support multiple QoS classes on the same network more complex control mechanisms are required. These mechanisms can be classified in two major categories called *proactive* and *reactive*. The *proactive* control mechanisms try to *anticipate* the effect of a change in the network and make *resource allocations* to accommodate the change. The allocation can be either *hard* allocation, when hardware resources are allocated to flows and services, or *soft allocation*, when the system keeps track of the traffic flows but does not allocate hardware resources for them explicitly. This type of control mechanism does not make any assumptions on the type of source that injects traffic in the network.

Hence, proactive control works on a macroscopic level by describing the long-term characteristics of traffic. The basic mechanism for service setup, in our platform, follows the *proactive* model. A new service's impact on existing services is determined *a priori* to service setup. Thus, the new service will be instantiated only if enough resources are present in the network, and once a new VPN instance is admitted, the *minimum* service QoS guarantees are enforced via reservations.

In practice only *estimates* of an upper traffic envelope are given by the user, as it is extremely difficult to give an accurate characterization of traffic. The service control platform can improve this characterization, by *observing* the actual traffic and building a model that describes its behavior. This is very useful in the following ways:

1. It can help the control system to increase the network utilization beyond the simple addition of user traffic requirements.

2. It can be used to signal to the user the actual traffic utilization, thus paving the way for more sophisticated payment methods, for example a pay per use method. The charge on demand method has been proposed by Courcoubetis in the context of ATM services [25].

Reactive control strategies are used to control the network congestion that appears because of unforeseen traffic behavior. The control mechanisms for dealing with such an event are two fold:

- Traffic Control Mechanisms, allow the control system to *throttle* traffic at the ingress of the network. In this case, the system throttles the aggregate of the VPN traffic that is inserted in the provider network, rather than individual applications as connection level congestion control does. Such a system still works in a reactive manner but at slower timescales that makes it possible for a network control system to react.

- Topological Control Mechanisms, allow the control system to *move* services, or service segments, inside the network, with the purpose of alleviating congestion.

In our platform, the *proactive* and *reactive* control strategies are used in complementary ways, with the ultimate goal of maximizing the network utilization, subject to the

QoS constraints.

During the lifetime of a service, the user needs may change. Therefore, a method for dynamic resizing of a service is necessary. Two modes of operation for the resizing mechanism have been envisioned:

- a *mostly closed (MC)* mode in which the user has to specifically request a resizing. The service management processes the request and takes the appropriate actions.
- a *mostly open (MO)* mode in which the user uses the CE based leaky-bucket regulators to indicate to the network the amount of desired bandwidth. The MO model works in conjunction with a reactive control system. Each user will receive an amount of extra traffic such that the level of congestion in the provider network is maintained below a preset level. When the congestion increases above this level, the control mechanism reduces the amount of user traffic that is allowed in the network.

In the MC model, it is possible for the user to ask for an increase in the available bandwidth that can only be satisfied by a different topology realization than the one that was in use when the request was made. This is not possible in the MO model. Therefore, the MC model has a more general semantic than the MO model.

1.2 Contributions

The main contributions described in this dissertation are as follows:

- The dissertation proposes a new service control framework for networks that carry VPN services. Our framework combines network control plane mechanisms with reactive control mechanisms in a combination of *proactive* and *reactive* control systems. The framework introduces two concepts for dynamic service resizing and presents a control architecture, together with algorithms, for the *mostly open* resizing concept, when the network is lightly loaded.
- Using the theoretical framework of large deviations, it proposes approximations for probability of traffic loss by an aggregate at FIFO and PQ schedulers with

finite buffers.

- The dissertation proposes a set of loss estimators for FIFO and PQ schedulers, where only the aggregate traffic that arrives at each queue has to be known.

- Introduce the concept of service admission control (SAC). This concept represents a generalization of the connection admission control at service level. This concept arises from the need of atomic admission or rejection of a service at a node in the network during the service setup phase. Qos-VPNs allow the aggregation of several QoS classes under one service, and as such, the service admission has to span several of these classes. Hence, the admission control mechanism will accept or deny the addition of a *service* at the node, rather than a simple connection.

- It presents an accurate algorithm for SAC based on model correction through loss observation. The observation of loss is based on the assumption that the link capacities are large, and therefore in a sample interval that is greater than one second, loss can be observed to a fair degree of accuracy.

- It presents a network control system that dynamically actuates the VPN ingress traffic in order to maintain the loss experienced at the core nodes below a given target.

- The dissertation contains a thorough verification of the proposed methods through large number of case studies and simulations.

1.3 Structure of the Dissertation

The dissertation attempts to follow a logical flow of ideas. With this aim in mind, Chapter 2 introduces the main networking concepts that are used in the rest of the material. The chapter starts with an introduction of the Network Element (NE) model. Each NE has a set of controllable elements that can be acted upon by the Service Control Framework. The main network resource that has to be managed is the bandwidth of the links that form the network. The process of proactive bandwidth management is known as Admission Control. Noting that the work in this area of network research is

extremely large, the review is by no means exhaustive, as only the work relevant to the dissertation has been reviewed. The presentation continues with the introduction of a node taxonomy for networks that carry VPN services, followed by the introduction of the main topological[2] models for VPN services.

Chapter 3 proceeds in two main sections. The chapter begins with a graph theoretic description of the network model. The latter part of the chapter introduces the VPN service model that is used latter in this dissertation.

Chapter 4 contains the presentation of the Service Control architecture that forms the basis of our work. The architecture is based on an out-of-box control paradigm and has elements of both the proactive and the reactive control models.

Chapter 5 presents the details of the theoretical framework to be applied to the measurement based, service admission control subsystem. The measure of interest in this chapter is the loss ratio of the multiplexer under the assumption that the buffer space is finite. The results in this chapter are based on extensions of results obtained by other researchers. The chapter starts with the formal description of the problem under study. It continues with a presentation of the scheduling element dynamics. The models are used in the remainder of the chapter to derive results for multiplexer loss ratios.

The results presented in Chapter 5 assume that the traffic can be modeled as a stationary stochastic process. If this is true, then the loss experienced at the multiplexer can be linked to the moment generating function of the arrival process. Chapter 6 deals with the problem of *estimating* the multiplexer loss if the traffic can be measured. The estimated loss is compared with the measured values obtained by simulation, using traces obtained from traffic measurements and from digitized video sequences. Results of the comparisons for both FIFO and Priority Queue estimators are presented in the chapter.

The results obtained over the previous two chapters are applied to the problem of service admission control. Chapter 7 details the admission of services that is used by the framework introduced in Chapter 4. The main idea of the chapter is to use the *measured* loss as a means for increasing the accuracy of the estimation for admission

[2]Through this dissertation, unless specifically stated, the term *topology* is used in its graph theoretic semantic.

control.

Chapter 8 presents a method for loss control in packet switched networks. The system allows the control framework to maintain a specified loss target by actuating the leaky bucket regulators at the ingress of the network. The main idea of the chapter is to use the *loss* as the measured output, rather than the buffer length. This opens up new avenues for the control of networks in general, and is directly applicable to networks that carry VPN services, which is the principal subject of this dissertation.

The final Chapter of the dissertation offers a number of conclusions that can be drawn from the work previously presented in each of the Chapters. In addition there are a number of research directions that can continue the work that has been done as part of this dissertation, which are also presented in Chapter 9.

The dissertation also has an Appendix that details the statistical characteristics of the high-speed MPEG2 traces that have been used for simulations in Chapters 6, 7 and 8.

Chapter 2

Background Concepts

2.1 Packet Switched Network Elements

A *Packet Switched Network* (PSN) is a collection of hardware elements, physical links, and control software. The controllable elements in such a system are the switching nodes, known as Network Elements (NEs). All *Packet Switched Network Elements (PSNE)* contain the following functional blocks (Figure 2.1):

- *data path* system, which is responsible for the data forwarding, from and to the IO interfaces and within the NE. The datapath of a PSNE is formed from a set of IO interfaces that feed into an internal packet routing component (that can be either a simple data bus or a switch).

- *internal NE control system (NECS)* that is responsible for the configuration and administration of the NE. The NECS usually has communication interfaces that are separate from the data path, which connect to a *command line interface* (CLI). The NECS can also be addressed through the data path, if administrative setups allow it. This is done by using a NE management protocol like SNMP or CMIP. For the purposes of this dissertation, the method of communication is not considered relevant; henceforth in the reminder, all communication methods are assumed to have the same expressing power. It is noted, that in practice this is not always the case.

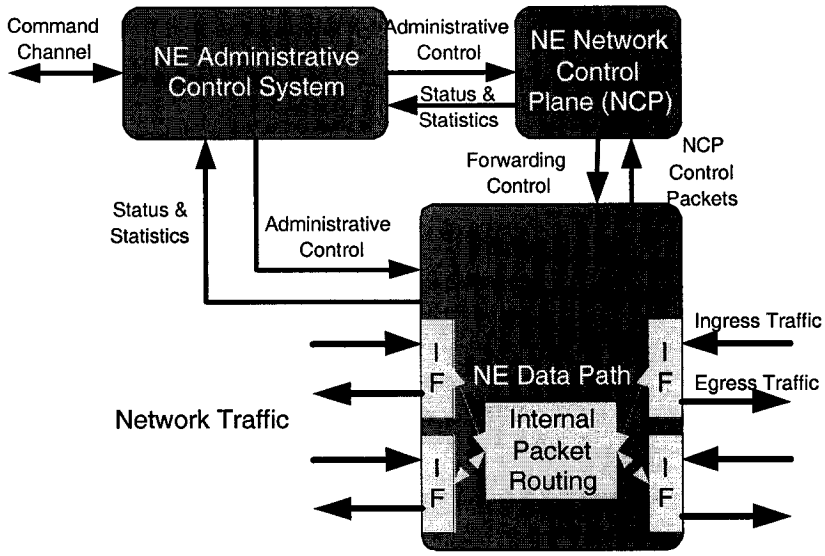

Figure 2.1: Packet Switched NE (PSNE) Model

- *internal NE network control.* This subsystem contains the part of the *network control plane* (NCP) that resides on the NE. Network control plane is defined as the totality of components that allow NEs to setup data forwarding paths in the network. Currently, network control planes are formed only by components that reside on the NE, in what will subsequently be called an *in-the-box (ITB)* control plane.

Traffic that arrives at the ingress of a NE is either terminated at the NE or forwarded to an egress interface. The forwarding decision is based on information that is carried in the header of the packer. All packet switched network data packets contain at least the following two elements in their headers[1]: an address field, which determines the *destination* of the packet; a traffic description field, which will be called *packet type*, and that describes network wide behavior independent of the destination address. When a packet is received at a PSNE the control part of the packet is checked against a set of predetermined *internal* destinations, which can be either egress interfaces or the NE control plane. The mapping of the packet into an internal destination is called *packet classification.* If the packet is to be forwarded, then the mapping result is an

[1]These two elements are present in all major packet switched technologies, IP[49], ATM [22], MPLS [87].

egress interface name and a queue number on that interface. For the packets that are forwarded, the NE can alter the addressing part of the packet or the type part of the packet, and the resulting packet is forwarded to the egress interface.

The main PSNE components that are of interest for our exposition are the *traffic policing* and the *scheduling policy*, because they represent the controllable elements of the PSNE data path that our framework acts upon.

2.1.1 Traffic Policing

Policing is defined as the set of actions taken by the system to control the user behavior with the purpose of maintaining user traffic within agreed values. This function protects the network from misappropriation of traffic resources, thus allowing the network control mechanisms to maintain preset control targets.

As enforcement to the policing action, the system can either mark the non-conforming traffic or drop it. Marked traffic serves as an indication to subsequent nodes on the path that it can drop that particular traffic first, if congestion arises.

The main traffic-policing device is the *leaky bucket controller* (LBC). This device forces traffic to conform with a linear, in case of the single leaky bucket, or concave in case of multiple leaky buckets, arrival curve. Using limiters at the CE-PE interface, for all the users and all QoS classes supported by the network, creates a network wide control mechanism that allows the provider to limit the amount of used bandwidth. In the absence of limiters, the amount of traffic that is injected in the provider network is limited only by the line rate.

Each LBC is a dynamic system, that emulates a storage system of capacity b which fills at constant rate r and empties at the speed of the arriving traffic. Let us present a model of a CE-PE interface with m single LBC controllers in a discrete time setup.

The arriving traffic, for each class of service $j \in [1..m]$, is modeled as a series of stochastic variables $[\mathbf{X}_1^j, \mathbf{X}_2^j \ldots \mathbf{X}_n^j]$ that take values over $\mathbb{N}$. In Equation 2.1 $s_k^j \in \mathbb{N}$ denotes the number of tokens that are present in the bucket for class j at time k. The vector $\mathbf{s}_t =< s_t^0 s_t^1 \ldots s_t^m >$ denotes the state of the LBC aggregate. The system completely described by the following equation:

$$\mathbf{s}_{t+1} = \alpha_t - \min(\mathbf{X}_t, \mathbf{s}_t + \min(\mathbf{r}, \mathbf{b} - \mathbf{s}_t)) \tag{2.1}$$

The amount of conforming traffic that is released at time t is:

$$\mathbf{y}_t = \min(\mathbf{X}_t, \alpha_t) \tag{2.2}$$

Similarly, the amount of non-conforming traffic (or discarded traffic) is given by:

$$\mathbf{z}_t = \mathbf{X}_t - \min(\mathbf{X}_t, \alpha_t) \tag{2.3}$$

The traffic modeled by the stochastic process $\mathbf{X}_t$ is conforming to the leaky bucket system $< \mathbf{r}, \mathbf{b} >$ if the random variables $\mathbf{z}(t)$ satisfy the following equation (a.s. stands for *almost surely* [92]):

$$\mathbf{z}_t = 0 \quad a.s. \quad \forall t \in \mathbb{Z} \tag{2.4}$$

The controllable parameters of the leaky bucket system are the *rate of fill (r)* and the *bucket depth (b)*. The control system modifies r and b to change the amount of traffic that is allowed in the network.

2.1.2 Packet Scheduling

Packet Scheduling is an important part of the ENE data path. It provides the NE the possibility of packet level control for several streams that are maintained in different queues.

The model of all scheduling subsystems is similar: a set of buffers is drained by a server using a particular service discipline. Scheduling disciplines are classified in two main categories:

- **Work Conserving** disciplines, which always send a packet if one is queued in *any* of the queues that are drained by it. Almost all of the scheduler instances that are implemented in today's packet switching NEs are of this type. Work conserving schedulers allow the user to make good utilization of the links through statistical multiplexing. Since all work-conserving schedulers transmit when there is data, all have *identical busy periods*. The difference between schedulers is given by the

fairness and latency characteristics of each version. Hence, although the behavior of all work-conserving servers, with respect to the flow aggregates, is identical they modify the characteristics of the individual flows. Most scheduler work has concentrated in this category (see [85, 71] and the references within).

- **Non work Conserving** disciplines [104], which do not necessarily send a packet even if one is present in the queues. This behavior is useful in maintaining inter-packet times within controlled limits, which in turn allows the control of the delay jitter. The drawback of the scheme is lower link utilization and increased burden on the user to exactly describe its traffic. Conceptually, non-work conserving schedulers can be implemented by a combination of shapers and work conserving schedulers [103]. This observation is helpful when trying to implement non-work conserving behaviors in already built NEs. While in the work conserving environments it is possible for a user to receive more bandwidth than what it has requested, if there is any spare along its route, in the non-work conserving case this is not true. In the reminder of the section, the non-work conserving case will not be considered, as our interest is in the support of services over currently available infrastructure.

All the schedulers are considered to have drop-tail queuing. Active queue management is not considered in this dissertation, because large scale studies report insignificant gains from it. In [74] May and his colleagues conclude after an extensive study of RED that it does not provide any benefits over tail-queuing disciplines.

One of the most utilized scheduling disciplines is the *static priority scheduler*. Its main advantages are given by the very good computational complexity, $\mathcal{O}(1)$ and the more simple analysis, under both deterministic and stochastic settings. The computational simplicity comes at a price, as this algorithm is not packet, delay or bandwidth fair.

Another class of work conserving service disciplines is represented by implementations of the *Generalized Processor Sharing (GPS)* policy. The first bandwidth and delay fair *packet* scheduling algorithm, called *Weighted Fair Queuing (WFQ)*, was presented in [32] by Demers et all and analyzed by Parekh and Gallager in a seminal paper [82].

One important result proved by Parekh was that the difference between the idealized model (GPS) and the WFQ algorithm is bounded. This allows the analysis of GPS to be carried to the WFQ implementation.

WFQ has a good computational complexity, which is $O(n)$ where n denotes the number of queues. The most important result regarding *any* packetized version of GPS policy can be summarized by Theorem 2.1, proven in [82]. The existence of a bounded difference between the GPS and the PGPS behavior allows a system to be analyzed under the GPS assumption.

Theorem 2.1 [82]
The departure time of a packet under the PGPS link is bounded above by the sum of the departure time of the corresponding packet under the GPS link and the ratio between the maximum packet length and the link capacity $\frac{L_{max}}{c}$. ■

Considering a range of line rates in the interval [1Gbps, 10Gbps] the values of the term $\frac{L_{max}}{c}$ are small. Precisely, they are between $[8.19 \times 10^{-6}, 8.19 \times 10^{-7}]$ seconds for a 1 KB packet and between $[6.55 \times 10^{-5}, 6.55 \times 10^{-6}]$ seconds for an 8 KB packet. Comparing these numbers with the propagation delay it results that, the link span that would generate a similar delay is between $[0.24, 2.4]$ km for the 1KB packet and between $[1.84, 18.4]$ Km for the 8 KB packet. Therefore, if the network links have values that are at least 10 times larger than these values, the packetization delay can be safely ignored.

2.2 Network Control Mechanisms

The purpose of all network control and management mechanisms in packet switched networks is to control congestion. The control and management planes of a packet switched network are composed of a small set of functional components. Some are intrinsic part of an NE, while others are out-of-box components (with respect to the NEs) that run on dedicated computing devices. The NE bound components are introduced in Section 2.1. The network control plane (NCP) is built as a distributed system whose components are the NEs. The management plane of modern networks is delegated to a *Network Management System (NMS) (Figure 2.2).* NMS is a network control element whose main functions [54, 55, 58] can be summarized as follows:

- Monitor the status and performance of the network.

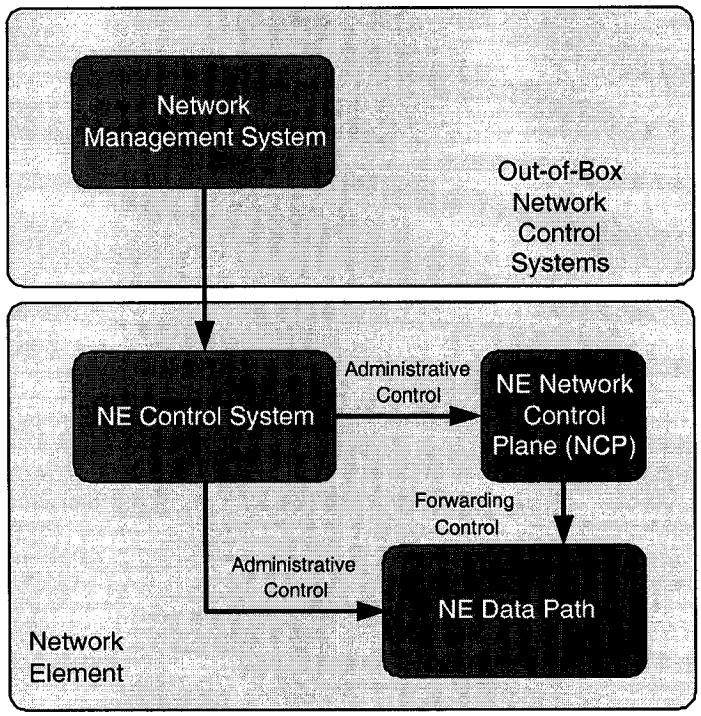

Figure 2.2: Block diagram of the control relationships between network subsystems. The link between two components describes this relationship: $a \rightarrow b$ is read as *a controls b*.

- Detect, investigate and identify the reasons for abnormal network operation.
- Take corrective actions for abnormal network conditions.
- Issue reports on the network status, network events and the corrective actions taken.

The network events that NMS systems have to react to are:

- Network topology failures, such as link or nodal loss.
- Traffic related events, such as high levels of traffic loss.
- NCP related events, such as high rates of circuit setups, which can lead to setup failures due to lack of control plane processing power, rather than lack of data plane resources.

Corrective actions to congestion control take place in a variety of places inside the network. In the network data paths, NEs attempt to correct congestion by queuing,

shaping and dropping data packets. These mechanisms are introduced in Section 2.1. At flow (connection) level, congestion control is assured by feedback mechanisms that signal the condition to the traffic source. When the congestion indication signal is received, the traffic source throttles its throughput. This indication can be as simple as dropping a packet that belongs to the source, which is the type of indicator used by TCP [96]. The importance of TCP traffic sources is raised by high percentage of this type of traffic in today's data networks. For example, Figure 2.3 shows that 93% of the traffic that is observed on a link of the CANET3 network is TCP traffic.

TCP control mechanisms have been modeled, and studied in a control theoretic framework by Padhye et al. [81] and by Holot [51]. The analysis of the protocol done in these papers show that the maximum transfer rate attained by a TCP connection is limited by three factors:

- window size (W_m), which represents the amount of unacknowledged data that a source can send
- round trip time (RTT)
- loss ratio

Based on the loss indication a TCP source will vary its rate between zero and a peak rate (r_{max}) which is equal to [81]:

$$r_{max} = \frac{W_m}{RTT} \tag{2.5}$$

Therefore, increasing the value of RTT decreases the maximum source rate. In general, VPN sites have internal communication lines that are of larger capacities than the connections that are available to the public network. In a typical VPN scenario, a number of users compete for bandwidth on an access line that has a capacity that is small relative to the internal link capacities. In such a scenario, most of the loss indication is received from the loss at the public network access point. However, if the allocated bandwidth between the VPN sites is less than r_{max} a TCP source will occupy all that bandwidth.

The main ATM congestion control mechanism at connection-level is the ABR protocol. There is a respectable body of literature that studies this protocol, of which we

mention Zhao [107] and Koralov [66] who have studied the behavior of the protocol using control theoretic frameworks.

The *network control plane (NCP)* of a packet switched network serves the following functions:

- Maintains the network state, both the network topology and traffic loads on network links. This function is done by a routing protocol [21, 78], which are used in all packet switched networks (IP, ATM, or MPLS). Traffic forwarding decisions have to be made prior to actual data transfer. In connection-oriented networks, ATM or MPLS, the trigger for the forwarding decision is a request for a new circuit. In connectionless networks (IP) the forwarding decisions are triggered by the appearance of a new address in the network or by network topology changes.
- Using the network state information, the NCP calculates forwarding paths and necessary resources for new traffic reservations.
- Makes the resource reservations that are required to support the traffic forwarding.
- Modifies the forwarding tables of NEs to support the new traffic.

Resource reservations in a PSN depend on the type of underlying network. If the network is connection oriented (ATM or MPLS) NCP allocates resources for the duration of a connection: in such a network, because the packet forwarding is done by using a link dependent, temporary address, the network control plane has to allocate these addresses and allocate entries in the forwarding table. This is a major constraining factor in connection oriented networks, because for a given pair of global addresses there can be many connections, say n which would require a number of entries in the forwarding tables of each NE of the order of $O(n)$. This is in contrast with the connectionless networks where between a pair of addresses there is only one forwarding table entry used along the path used for forwarding from source to destination.

Two control mechanisms allow the NCP to control network congestion:

- logical topology mapping, which is reviewed in Section 2.3.3. When a logical topology is given for mapping to the network physical layer the NCP takes into

account the amount of available bandwidth on the physical links. This mechanism allows the NCP to balance the traffic in the network.

- admission control, which is described in Section 2.3.

2.3 NCP Control Mechanisms

2.3.1 Admission Control

Admission Control (AC) is a set of mechanisms that accept or reject network traffic at either node or path level. *AC* [85] is a very important controlling element and it is crucial to providing guarantees under the *proactive* model that has been chosen as the basis for this thesis. Conventional admission control systems, known as *connection admission control (CAC)* work on individual flows, by applying the admission process to each flow. CAC systems need three types of primary information for their operation:

- An evaluation of the aggregate load (ie. the existing traffic that has been admitted). This can be obtained from either the descriptions of the individual flows that form the aggregate or by estimating the aggregate directly. An admission control system that obtains the aggregate load information from the descriptions of the admitted flows that have been provided by the user, will be called *Direct Admission Control* (DAC). If the aggregated load estimation is obtained from measurements, the resulting system will be called *Measurement Based Admission Control (MBAC).*

- The new flow description. The user is supposed to give some coarse description, of the amount of traffic that it plans to send.

- Information about the guarantees that have to be assured for the flow.

Direct Admission Control systems have been studied using both deterministic and stochastic traffic models. The work in [60] uses the deterministic analysis approach of Cruz [28, 29] and Parekh [82, 83]. It has been proven in [83] that in networks where traffic is ingress bounded, uses PGPS schedulers in *all* nodes, and schedules individual flows to separate queues at *all* nodes, that deterministic end-to-end delays can be calculated.

The most popular bounding functions, due to their simple interpretation, are interval linear [71]

The deterministic framework for network analysis has a number of limitations that restrict its applicability. In our opinion, the following restrictions are the most debilitating for the framework:

- Packet loss cannot be used as a service differentiator as all services that are considered are loss less.

- There exist good closed form results for systems that separate flows inside the network [82, 83]. However, if traffic from several sources is aggregated inside the network then, for general flow topologies, no deterministic guarantees can be made [71]. This is a result of the lack of information about what happens to the individual streams inside the aggregates. Since it is possible to have flow arrival synchronizations, which is more acute in synchronous type of streams like voice [15], to be able to offer *provable* no-loss guarantees the link utilization has to be decreased to very low levels. A more disruptive problem is that the flow setup procedure cannot be automated, since there is no general solution to the problem of proving delay bounds in general network topologies [71]. Hence, a human operator has to analyze and prove the bounds for each new topology. This is a time consuming procedure, that *is not guaranteed to succeed*, since the new topology might be intractable to this type of analysis.

- In the initial utilization of the WFQ schedulers by Parekh [82] the weights of the queues were assigned to be proportional to the average rates of the admitted flows. Such a weight assignment links the delay experienced by the packets of a flow to the amount of bandwidth that flow sends. Such a correlation is not always helpful. In a recent paper, Szabo et al. [97] propose the use of non-proportional weights to solve this problem and describe an algorithm that uses it. In [79] the authors present a weight assignment algorithm that is optimal with respect to the bandwidth utilization. Until now, it has been assumed that the weights of the queues are constant. Evidently, schemes where the weights are dynamically modified can be envisioned. Such schemes require hardware support and we are

not aware of any NEs that implement them. Hence, they will not be subsequently considered.

The above problems can be circumvented by adopting a stochastic modeling perspective. Such a modeling strategy has the advantage of introducing packet loss as a QoS metric. Admission control schemes designed in this model use *loss probability* as the target for the algorithms.

Consider a buffered multiplexer with a single queue, served in a FIFO order. If the queue is modeled by an infinite buffer, then the probability of loss is defined as $\mathbb{P}(Q > b)$, ie. the probability that the queue is longer than a given value b. If the buffer is considered finite, then the *loss probability* refers to the ratio between the average of the traffic that is lost and the average of traffic that is received at the input of the multiplexer.

In the statistical QoS approach to admission-control the controlling system tries to maintain a value of the loss parameter that is below a preset number. Therefore, the main problem to be solved is that of linking the input process to the loss probability. This problem has been studied, under various assumptions by many researchers. The seminal work of Anick et al. [1] studies the behavior of the FIFO scheduler when fed by a large number of on-off sources. The extensions in [95] extends Anick's work to more complicated Markov modulated rate models.

Elwalid [38] presents a simple approximation for the loss problem for a system fed with Markov Modulated Rate Processes (MMRP) when the length of the buffer is very large. This results in a direction of research that employs techniques from the *large deviations theory* [93, 31]. The work in [38] also makes a link with the notion of *effective bandwidth*, first proposed by Kelly [61] and generalized in [62]. Similar results for effective bandwidth approximation s obtained by Guerin [47].

Asymptotic results of the loss approximations have been studied under two types of limits:

- The *large buffer asymptote* operates by finding the value of $P(Q > b)$ when the buffer is asymptotically increased to infinity ($b \to \infty$) . The main results of this research direction are presented in [14] and the references within.

- Another possibility in linking the EB characterization to the loss curve is to multiplex a large number of *independent* sources over a link, while scaling *both* the capacity of the link *simultaneously* with the capacity of the buffer. In this case both the buffer size and the capacity of the link are considered proportional to the number of sources, while the number of sources is asymptotically increased to infinity. First proposed by Weiss [102], it has been expanded by Courcoubetis [26] and Duffield [9]. The large number of sources asymptote has been improved upon by Likhanov et al. [73] using the sample mean deviation theorem of Bahadur and Rao [2].

The area of stochastic modeling is heavily weighed towards the study of the FIFO multiplexer in isolation. Nevertheless, a number of studies have tackled the problem of networks of FIFO multiplexers. In particular, in [13] the authors obtain a large deviations result that is applicable to *in-tree* networks. Subject to some technical conditions, that are satisfied for a large class of traffic sources, the asymptotic loss characteristics, in the large buffer framework, of a node *inside* the network can be obtained from the characteristics of the traffic at the ingress of the network.

As is the case for deterministic frameworks, the problem of analyzing a general network in the effective bandwidth framework is unsolved. This state of affairs is a result of the same technical problem that arises in deterministic framework: because of cycles, the induction principle cannot be applied, severely limiting the mathematical apparatus that can be used for analysis

Large buffer asymptotes have also been applied to the admission control problem in multiplexers with multiple buffers. Berger and Whitt [5] heuristically apply a large buffer approximation to admission control of static priority schedulers.

Admission control under the WFQ queuing discipline has been studied in [108, 69]. Kumar et al. [68] proposes a system that is input shaped before the WFQ scheduler. For such a system, a method for weight assignment is proposed. In [108] the authors study the admission control in a multi-queue GPS system by making the queues independent. By separating the queues, the system is made analytically tractable at the expense of accuracy. The system WFQ weight assignment makes this a variant of the proportional

system proposed by Parekh [82].

2.3.2 Measurement Based Admission Control

Direct Admission Control Systems, as detailed in the previous subsection, are limited by the ability of the user to provide tight traffic characterizations, and the inability of the network to extract statistical multiplexing information from those descriptions. In all cases where a temporally tight characterization of traffic is desired some measurement mechanism has to be employed. Since it is undesirable to force the customer to institute a measurement process, such a mechanism can be employed on the provider side of the network. The admission of services in the network utilizes in this case information about the traffic that it obtains from the measurement process itself and supplements it with the user stated data. Admission Control techniques that follow this architecture are called *Measurement Based Admission Control (MBAC).*

Relevant work in this area includes [36, 7, 84, 91, 70, 98, 10, 12, 24], with work in both for the ATM and the IP based networks.

The effective bandwidth asymptotes, described in the previous section have been applied to the MBAC strategy too. In [33] an estimator is proposed for obtaining the effective bandwidth value from traffic measurements. Recently, in [24] the large number of sources asymptote has been proposed to be used for the same purpose. In this work, Courcoubetis et al. uses the notion of traffic equivalence, to speed up the numerical solution process. The method proposed is heuristic at this point, since the authors state that they could not find a proof for the convergence property of the proposed algorithm.

Qiu and Knightly [84] propose a system based on measurement of the *aggregate traffic envelope.* The envelope is a function that returns the peak rate of the traffic over a given timescale. The purpose is to detect the time scale over which the maximum loss occurs. The idea of multi-time scale analysis first appears in [99], in the context of analyzing the multiplexing of multiple time-scale Markov streams. The same line of reasoning is developed in [64] as well as [77].

Tse and Grossglauser [98] present a bufferless admission control framework that operates over several time scales. The time scale of interest is determined to be the flow lifetime. In contrast, in virtual private networks, the service lifetime can be assumed

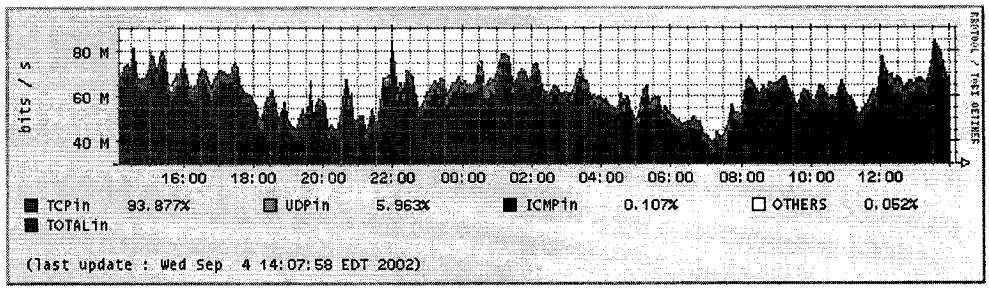

Figure 2.3: Hourly Traffic Rates as 5 minute averages from one of the CANET3 Network Links (www.canet3.net)

infinite. Nevertheless, variability in traffic can be observed, due to the aggregation at the customer to provider interface. Hence, although time scale determination remains important some other means have to be employed.

Another approach to MBAC is presented in [36], where Dziong et al. use a Kalman Filter framework to estimate online the mean and variance of traffic. In [44] Gibbens and Kelly introduce Bayesian estimation as the means of determining if a flow should be admitted or not. In [98] the role of the measurement window is studied, in the framework of single queue servers. Grossglauser [46] and Duffield [34] also take into account the effect of sampling the traffic data on the accuracy of the admission control system.

Breslau and his colleagues have reported in [10] that a number of MBAC algorithms [41, 27, 84] seem to have identical performance, and infer from this that further work on this issue might not be needed. It has to be noted that the algorithm comparison does not seem very consistent since the algorithms presented in the paper use different QoS targets. Furthermore, the behavior of the algorithms which have loss as a target is not very consistent. Hence, we consider that the conclusion of the study is not warranted.

The case of virtual private services based networks has some crucial differences from the classical connection admission control problem that we will try to exploit:

- services setup and tear down rates are assumed to be rather small compared to the call setups that are experienced in connection oriented networks. This works to the advantage of a measurement based system because it allows it to acquire the necessary information before another request comes in. Since the call setup

rate is linked to the speed of computation of the AC procedure, a low setup rate implies that more computationally complex algorithms can be used, if the increase on robustness and conciseness justify the computational overhead. It is important in this context for the algorithm to be consistent as usually there are penalties for breaching the QoS targets.

- Traffic from a customer site is itself an aggregate of many sources, and as such presents different characteristics from the individual source traffic.

- Most of the network traffic presents strong temporal dependencies such as the ones depicted in the Figure 2.3.

  Real traffic traces, if taken over longer time periods (see Figure 2.3), exhibit variability on several time scales. The traffic of Figure 2.3 is more intense during business hours and less intense at other times. Hence, it exhibits a slow varying component that has a strong *temporal correlation*, and a rapidly varying component whose properties are not determinable by visual inspection.

  This behavior is emphasized in networks where the majority of traffic is generated by VPN services, since for these networks most of the traffic is related to human activity of some kind and therefore it is generated during business hours. The traffic in Figure 2.3, that is used to exemplify the reasoning, is pertinent because the CANET network provides a service that is localized to a group of users, which is akin to what happens in a network that serves VPN customers. The aggregation of traffic from a site follows similar patterns, for the same reason.

Crucial to any measurement based traffic control system is the issue of traffic characterization. The traffic characterization in this work will be obtained on-line, by passing the information from a *measurement* process to a modeling and prediction module. The main rationale for this approach is to try to *predict* the long term characteristics of the network traffic and use the resulting information to allow *timed* service requests for the customers that have already been allowed in the network. Such a request would be of use if a user wants to utilize extra bandwidth for a period that he can predetermine, and the network has the capacity to carry it. For example if extra video conferencing

has to be set up between sites and the time of the conference is known, and it usually is, such dynamic requests would allow the provider to generate extra revenue from its infrastructure. Sang et all [90], test the predictability of the ARMA and the Cumulant Markov Modulated Process traffic models against Ethernet and Internet traffic. Their results, over a number of traffic traces obtained from live network, are encouraging. Notably, aggregated Internet traffic seems to have good predictability characteristics.

An important issue, regarding traffic modeling, that has received attention in the past decade is that of statistical self-similarity [4]. Since it was first reported by Leland and his colleagues in [72] it has been studied by several research groups because of the perceived implications of the phenomenon. In the large buffer asymptotic, it has been proven by a number of researchers, most notably Tsybakov and Georganas [100, 101], that the loss of a FIFO buffer that is fed with long-range dependent traffic decreases only *hyperbolically* with the buffer length rather than exponentially. The implications of this theoretical result would be that increases in buffer sizes are not very helpful when it comes to loss reduction in NEs. Nevertheless, the behavior predicted under the large buffer asymptotic has not been observed in practice. The reason is the different behavior of the long-range dependent traffic when the NE buffer sizes are small in comparison to the link capacities. Ryu [89] and Heyman [50] argue that long-range dependent traffic - for realistic values of buffer sizes does have an exponential behavior, which is what has been practically observed.

2.3.3 Topological VPN Service Models and Their Instantiation

Virtual Private Network services provide customers with seamless, long term, connectivity of geographically dispersed sites across a public network. The nodes that participate in the creation of a VPN service can be separated in four distinct categories [59]:

- customer nodes [C]. These nodes are internal to the customer sites; they are owned and managed by the customer.
- customer edge [CE]. Such nodes are the interface between the provider and customer sites. They are owned by the customer, but they can be managed by either the customer or the provider.

- provider edge [PE]. These nodes connect customer sites to the provider network. They are owned and managed by the provider. The customer sees only these nodes from his side of the network.

- provider nodes [P]. Provider owned and managed nodes that represent the core of the provider's network.

CE and PE nodes can be connected in a variety of ways [11]. The simplest connection, described in the beginning paragraph, connects each CE to one PE only. In this connectivity model, each PE has a set of CEs from *distinct* VPNs attached to it. Such a solution has the disadvantage that is not tolerant to link failures between the CE and the PE device, nor to node failure of the PE or CE devices. Fault tolerance of the CE-PE connection can be attained by connecting the CE to more than one PE or by using multiple CEs to connect to the network from the same site.

Since these connectivity problems are easily solvable and are contained at the interface between the customer and the provider, they will not be considered in the subsequent presentation, without loss of generality.

The data plane connectivity that is realized by a VPN service also allows control plane connectivity for the user sites. Note that although data plane connectivity implies control plane connectivity, the two functionalities can be separated. For example, if the control planes of both the customer and the provider are compatible, some of the control plane work that would be done by the CEs can be exported to the control plane of the provider. Such a solution is applied by the BGP/MPLS [88] specification from IETF. The advantages of such a solution versus a customer managed control plane are not obvious: this solution just moves some control plane processing from the CE devices to the PE devices. As such, it increases the computational requirements of the PE devices, which is not desired for the reasons described in the first section of this chapter. Furthermore, similar results can be obtained by peering CE devices directly. In our work we assume that the control planes of the customer and the provider do not interoperate for the purpose of forwarding information dissemination.

From the perspective of a VPN customer, the process of setting up a service terminates when the CE device is programmed with a set of virtual interfaces, one for each

virtual link to another site that is part of the VPN.

Hence, the customer receives from the provider a logical network that connects its sites. The way the user will determine the forwarding paths over this logical topology is not considered. What is of importance for us is the manner in which the provider can realize the logical topology.

The second mechanism of control that is available at the NCP is the *topology instantiation.* The topological model that is used for the service definition becomes essential for this. Two topological models are used to define VPN services. They offer different flexibility of service to the user and to the provider. Let us consider that we want to instantiate a service between n customer nodes. The traffic that is passed between these sites can be specified by the user in two different ways:

- Traffic is specified between pairs of nodes. Services that use this type of specification are instances of the *pipe service model* [59].
- Only the total ingress and egress traffic from each node to the provider network is specified. Services of this kind are classified as *hose service model* instances [35].

The *type* of user-traffic information that is available to the provider is crucial to the resource allocation strategies:

- If the traffic matrix is known, the provider can allocate resources *individually* for each of the connections realized between CE pairs.
- Another possibility, dubbed the *hose model* in [35], is to obtain only the bandwidths that are necessary at the CE-PE border. In this case a different allocation strategy is employed, which results the realization of a logical network between the CEs that is optimally a tree.

VPN Pipe Model

The *pipe model* is a representation, in packet switched networks, of the model used to realize VPN services in circuit switched networks[76]. An example of this model is presented in Figure 2.4. First, let us define some notation. Let P_1, P_2, P_3, P_4 denote

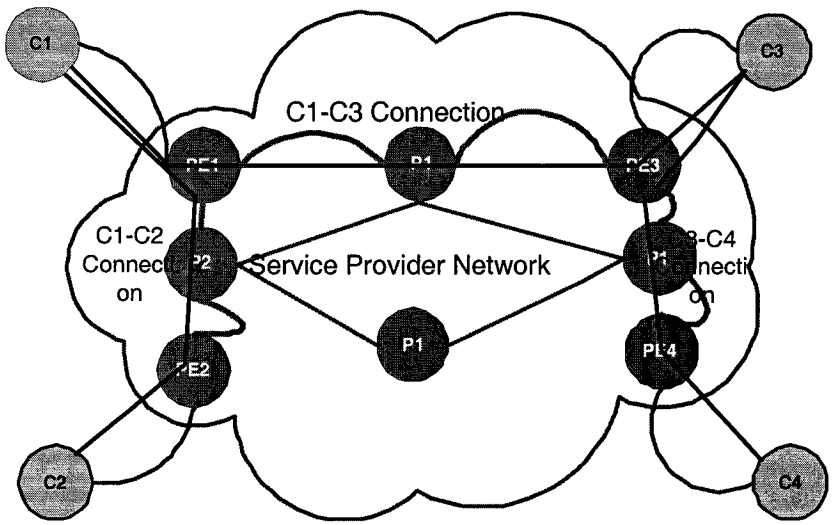

Figure 2.4: Pipe Model Example. Four sites CE_1, CE_2, CE_3, CE_4 connected through a set of three point-to point virtual pipes. The provider network has four P nodes that are connected together with the red links. Three of the P nodes are connected to the PE nodes which in turn connect to the CE nodes. The black lines denote the *logical* connections that are established between the CE nodes.

the nodes that are under the providers administrative control. In this subsection, no distinction is made between the P and PE nodes, to keep the presentation simpler and focused on the main issues. The nodes that are not under the provider's administrative control are called *customer nodes* and are denoted by CE_j.

In Figure 2.4 four customer sites that form one single VPN are connected to a single provider network. The sites are *logically* connected through point-to-point *logical* pipes. The problem that has to be solved on the provider side is to find a set of paths in the network that provides connectivity between the following points: $\{(P_1, P_2), (P_1, P_3), (P_3, P_4)\}$. The pipes are dimensioned according to the traffic values provided by the customer between the points: $\{(CE_1, CE_2), (CE_1, CE_3), (CE_3, CE_4)\}$. This example presents a logical network that is not a full mesh. Therefore, in order to have connectivity between the nodes that are not connected through logical links (like CE_1 and CE_4 for example) the customer logical network has to have a higher layer routing running on it. In this case information from CE_1 can reach CE_4 via CE_3.

Since the traffic pipes between the nodes (CE_1, CE_3) and (CE_1, CE_2) are separate,

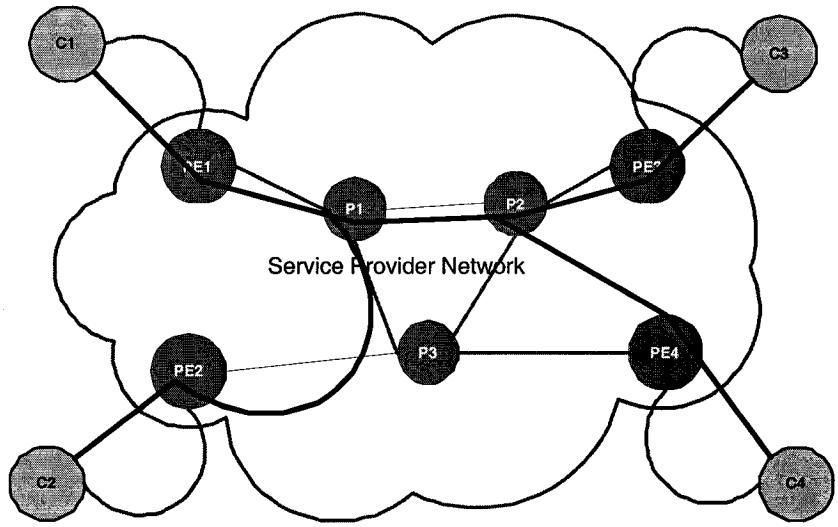

Figure 2.5: Hose Model Example. Four sites CE_1, CE_2, CE_3, CE_4 connected through a hose model. Three P nodes form a fully connected core network has four PE nodes attached to them. Each of the CE nodes is attached to one of the PE nodes. The VPN instance is realized by a connection *tree* that is depicted by the black lines that link the PE nodes.

the total traffic that CE_1 can inject in the provider network is limited by two separate set of constraints. In this case, even if the (CE_1, CE_3) connection is not utilized, the capacity cannot be used on the (CE_1, CE_2) link.

Each of the virtual connections returned by the provider to the customer is individually addressable and has resources reserved for it in the provider's network. The resource reservation is in fact the main distinction between this model and the VPN hose model.

In the *pipe model* there are two main problems that have to be solved in order to instantiate a new VPN service:

- obtain a logical topology that satisfies the user, while optimizing some cost or delay criteria. This direction of research has been studied by Farago [39] in the context of ATM VP optimization. The similarities with the problem at hand are obvious, therefore the optimization procedure presented in the article can be used for the logical topology optimization of VPNs.

- Map the logical topology to the physical topology. Since each pair of nodes that

have to be connected in the logical topology can be solved in separation, this problem can be solved as a single-commodity constrained problem. There is a large body of work in this area, which is related mostly to the unicast routing in general graphs (see for example [16]. Chlamtac [17] casts the problem in an interesting manner: it tries to obtain a logical topology mapping that balances the loads of the physical links. The algorithm presented uses randomization to obtain a solution and has good properties for large (with greater than 40 links) networks. The same problem can be cast as a multicommodity optimization problem, which is the approach taken by Mitra et al. [75, 76] and Gupta et al. [48].

VPN Hose Model

Let us consider the same example as above in a *hose model* framework. In this case the provider receives only the *total* amounts of bandwidths that each of the sites will send and receive. From the customer stand point such a model has the following advantage over a pipe model realization:

- It is possible to use the total traffic available between sites among any two sites. Again considering the example of Figure 2.4, consider that the connection (CE_3, CE_1) has allocated bandwidth CE_{31} while the connection (CE_3, CE_4) has allocated bandwidth CE_{34}. Denote the bandwidth available to the connection (CE_2, CE_1) by CE_{21}. If site CE_3 does not use the (CE_3, CE_4) connection, but needs to use some extra bandwidth to communicate with CE_1 then a resize request has to be sent to the network control system. In the hose model such a case is automatically handled, as the site CE_3 can send as much as it wants, provided that it is less than the hose capacity, to any of the other sites that are part of the VPN service. Hence this increases the flexibility of the service from a user point of view.

Instances of services that use this model are trees. Duffield et all [35] obtain solutions using Steiner trees. This solution is improved upon by Kumar et all [67]. The last reference is expanded by Italiano et al. in [52] where the problem of providing topological fault tolerance for the services is studied.

A comparison between the *pipe* and the *hose* models reveals that they have different expressive power. While with the *pipe* model it is possible to make traffic guarantees for *all* the traffic that belongs to a service instance, this is not the case for the *hose* model. To illustrate this, let us go back to the example presented in Figure 2.5 and consider that for each of the sites an equal amount of bandwidth, denoted by bw, is allocated for the connection of the VPN sites to the network. Furthermore, to simplify the discussion, consider that both the ingress and the egress bandwidth allocations are equal for all VPN sites. Therefore, regardless of the adopted *hose model* solution, on the (P_2,PE_3) segment we need to allocate only bw of the link capacity in both directions for this service. If both C_1, C_2 and C_4 want to use all their egress bandwidth to send information to the node C_3 over an interval t, then bwt traffic will be lost at the egress interface that connects P_1 to P_3 and the same amount will be lost at the egress interface that connects P_2 to PE_3. It becomes evident that no QoS characteristics can be guaranteed by the network operator for this model. Hence, the improvement in bandwidth utilization that is reported in [35] versus the *pipe* model, comes at the cost of a lesser guarantee semantic. Therefore, this model is not suitable for the type of provider guaranteed framework that is proposed in this dissertation. The *hose model* can be successfully applied if traffic is elastic, as the TCP generated traffic is. For the TCP case, the loss of data is smaller because the source will adapt its rate to the loss experienced by the data flow in the network. Another application of this model is when VPN users do some form of bandwidth management inside the VPN. Consider the case when the bandwidth between two VPN points is allocated a priori to data transmission. In this case, the situation presented above does not arise.

A more pressing problem in the *hose model* is represented by the interaction between different service instances. Since several service instances pass the same P node it is possible for users to experience high loss even if they do bandwidth management, in the presence of users that do not. This problem can be remedied with NE support in two manners:

- if some form of egress policing is available for each service instance. In this case, a number of policers equal to the number of traffic instance that pass through the

egress interface are necessary.

- if each service instance can be mapped to a different set of queues.

Both of these solutions require hardware support that is beyond the available capabilities of NE. Both solutions have scalability issues for the P nodes that were deemed too limiting in the integrated network solutions. Based on the arguments of this section, the hose model is considered to have a semantic that is too restrictive for the instantiation of provider guaranteed VPNs.

2.4 Summary

Regardless of their functions, all ENEs that are part of a PSN can be modeled using a small number of elements, which are usually implemented as hardware components. Of these, two are crucial for support of traffic classes in PSN: the traffic policers, who allow network control mechanisms to limit the amount of traffic injected in the network by a user and the scheduler policy which allows the network owner to build differentiating traffic classes.

Network control is achieved at several levels by different mechanisms. As our view of the network is service centric only network control plane mechanisms and network management plane mechanisms are studied in the subsequent chapters. The network control plane uses two main mechanisms for traffic and congestion control: Admission Control is used in conjunction with topology based controls to instantiate services in a manner that guarantees the correct behavior off all services already setup and the one that is under instantiation. There is a large body of work in this domain, mostly for connection-oriented technologies like ATM and MPLS. This chapter introduces the concept and reviews some of the extensive literature on the subject. The realization of VPN services can be modeled in two manners that offer different service flexibility to the user and to the provider. The first model present in literature is known as the *pipe* model and has traffic that is specified between pairs of nodes. When only the total ingress and egress traffic from each node to the provider network is specified the service is instantiated in a *hose* model. Both models have strengths and weaknesses that are discussed in the sections above.

NMS level control assures the operation of the network in the face of unforeseen events such as traffic overloads or topology failures. The types of controls that are used at this level depend entirely on the level of control that is available at the network control plane. If the network control plane is based on a very conservative admission approach in a deterministic framework then, although theoretically still possible, the probability of traffic overflows is extremely low. Hence, a control system that takes actions against these occurrences will not help since, as explained above, if the deterministic guarantees cannot be proven it is possible for the event to appear again regardless of the traffic distribution. Similarly if the topological failures are handled as part of the NCP then such a control at this level would be redundant.

Our interest lays in the control of packet switched networks that carry VPN services. This subject builds on many networking concepts, the most important of which are reviewed in the current chapter. Networks that support mostly VPN services have specific node taxonomy. Nodes that compose such a network are CE, PE or P depending on their functional role in the network.

Chapter 3

Network and Service Models

This chapter considers communication networks with nodes that can be described functionally as in Section 2.3.3. All PE and P nodes that compose the network are assumed to be under the administrative control of a single entity. Furthermore, since the same functionality can be duplicated in both the CE and the PE devices, we assume, without loss of generality, that CE devices are under the same administrative control as the P and PE nodes. For this setup, the next section introduces a graph theoretic model.

This dissertation is concerned with a special type of communication service. It is characterized by both topological[1] and traffic properties. The second part of the current chapter introduces the characteristics of the service, as they will be used in the reminder of the dissertation.

3.1 Network Model

The model of the network described informally in Section 2.3.3, is a double (G, Q), where G is a model of the network topology, while Q is a model of the classes of service that are supported by the network. G is a graph, $G = (V, E)$, and as such it is composed of a set of nodes, V, and a set of *bidirectional* links ,E. The set V is the union of four sets, V_{ce}, V_{pe}, V_p, V_a, the first three modeling the types of nodes described in Section 2.3.3, while the last represents the set of nodes that allow the connection to other networks. Such a model encompasses VPN only networks, as well as networks that offer VPNs and access to the Internet. The first three sets of nodes are disjoined. Using set theory

[1]The term *topology* is used in this chapter in its graph theoretic semantic.

notation, the following can be summarized as follows, where ϕ denotes the empty set:

$$V = V_{ce} \cup V_{pe} \cup V_p \cup V_a$$

$$V_{ce} \cap V_{pe} = \phi$$

$$V_{pe} \cap V_p = \phi$$

$$V_{ce} \cap V_p = \phi$$

Following the nomenclature of Section 2.3.3, nodes that belong to the set V_{ce} are called *customer edge nodes*; nodes that belong to the set V_{pe} are called *provider edge nodes* while the nodes in V_p are the *provider nodes.*

The possible graphs G that are realizable in this model is restricted by two constraints that are imposed on the set E.

- As no CE node can be connected to a P node, and vice-versa, the set E does not contain any such links.

$$\{e(x,y) \mid \forall x \in V_{ce}, \forall y \in V_p\} = \phi$$

- It is considered that the CE nodes are not connected directly. Since we are interested in finding logical topologies over the PE and P nodes, we impose this restriction in order to simplify the specification of finding a logical topology.

$$\{e(x,y) \mid \forall x \in V_{ce}, \forall y \in V_{ce}\} = \phi$$

Each link (i,j) is endowed with two properties:

- a cost denoted by $\xi_{ij} \in \mathbb{R}^+$
- a link capacity c_{ij}.

The network supports a number of Classes of Service (COS), all with statistical Quality of Service, denoted by $\mathcal{C}$. Each class of service j is completely defined by a loss guarantee P_j which is viewed as the upper bound of the loss experienced by traffic of that class in any part of the network over a prescribed time interval. Hence, $\mathcal{C} = \{P_j \mid j \in [1, n]\}$.

Another QoS parameter that can be used as a COS differentiator is the end-to-end (ETE) delay between two CE devices. ETE delay in packet switched networks is the sum of three terms:

- Propagation delay. It is a function of physical constants such as the speed of light and the total cable distance between the transmission points. For a network that has cables built from the same materials (ie all copper or all fiber optic) if the total cable length that has to be traversed by a packet between two points n_1 and n_2 is d_{12} and if the speed of light in the medium is denoted by v_m then the propagation delay, p_{12} is equal to $p_{12} = \frac{d_{12}}{v_m}$.

- Transmission delay. This is a function of the packet length and the interface transmission speed. For a packet composed of L bits, sent over an interface of capacity c_e, the transmission delay over n nodes is: $n\frac{L}{c_e}$.

- Queuing delay. This delay component depends on the amount of buffer space that is available at the nodes through which the data packet passes on route to destination. The maximum queuing delay, as well as the maximum delay jitter, is a function of the buffer length, the scheduler structure and the egress interface capacity.

It can be observed that even if the queuing delay is zero, the delay of the same type of traffic differs depending on the length of the path that the packets travel from source to destination and the packet length. This implies that there are physical limitations in offering low delay across communication networks. Therefore, traffic of the same class can have different delay characteristics, depending on distances between the source and the destination. This limits the applicability of the absolute ETE delay as a QoS differentiator for COS specifications, and for this reason it has been decided not to

consider it in our definition.

Queuing delay is a function of the total amount of buffer space that is present at a multiplexer. It is shown in subsequent chapters that in high capacity links that multiplex a large number of streams, high link utilization can be attained with very small losses, even with small buffer length relative to the link capacity. Therefore, when this is the case, and in most current practical cases it is, NEs can be configured with short buffers without any effect on the maximum attainable link utilization.

Even though increasing buffer sizes in the case presented here is not large, it is possible to use the buffer sizes as a *relative* differentiator among COS. In such a setup, each COS j is defined by a double (R_j, P_j), in which R_j represents the *ratio* between the delay of the class $j-1$ and the class j or 1 if $j = 1$. Such a definition can help the provider differentiate the COS offerings based on the user perception of value: lower delay is perceived as being *better* than higher delay. The next section presents an economic based procedure for the determination of the maximum buffer sizes in the provider-administered network.

For a given node $n_k \in V$ all traffic that belongs to a class of service is *aggregated* in the same queue. This *class aggregation* model has been chosen because the availability of NE with limited egress queues makes this model realizable.

3.2 VPN Service Definition

The process of service creation (Figure 3.1) starts with the definition of the service based on a *service specification.* The specification is used by the control system to generate a *service implementation*, which is used to allocate the necessary resources for the service and generate a new *service instance.* From this description it is apparent that we differentiate between the *service specification* which is the input data, and the *service instance* which describes the fully functional service. To show that our service specification is general enough, we deal with both the *pipe* and the *hose* models that were introduced in Chapter 2. However, our interest resides with the *pipe* model, which, as shown, has more encompassing semantic than the *hose* model.

The VPN service specification is a quad $(V_s, T_s, B_{AS}, B_{RS})$ where each of the letters

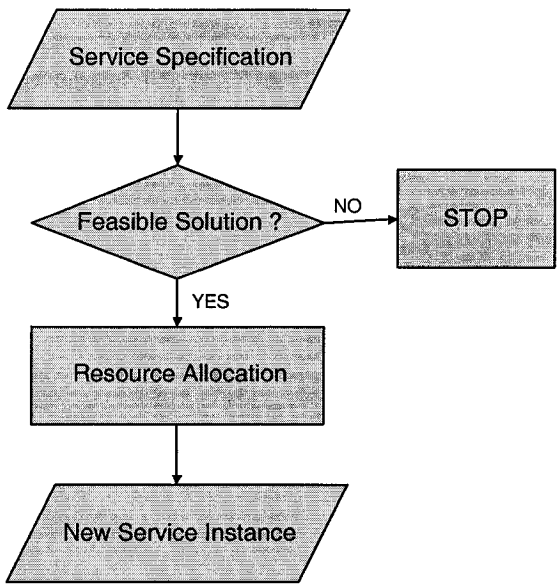

Figure 3.1: Service Creation Process

denotes one of the following elements:

Topological Specification (V_s). The service specification has to contain the names (addresses) of the nodes that are part of the service. To make this possible, all nodes $n_{ce} \in V_{ce}$ are considered to be addressable, by some global naming scheme. Using set theoretic notation:

$$V_s = \{n_i \mid n_i \in V_{ce}\}$$

If the service requires access to the Internet, then the set V_s contains some nodes that belong to the V_a set as part of the service specification:

$$V_s = \{n_i \mid n_i \in V_{ce} || n_i \in V_a\}$$

Traffic Specification (T_s). In order to assure that the QoS measure (loss) is not infringed, it is necessary for the user to make a declaration of the amount of traffic that it wants transported by the provider. How this quantity is specified depends on the VPN model that is adopted for the service: if a pipe model is used then the traffic is specified between pairs of nodes; while if a hose model is used the traffic

is specified according to the exchanges between the CE and PE nodes. For each of the COS that are supported by the network, the Traffic Specification describes the bandwidths that are needed to fulfill the service. To make this precise, consider a service specified using a pipe model as described by Figure 2.4. In this case the customer states that is wants to link sites C_1 and C_3. On this link there are k classes of service available. Hence the traffic specification has to state the amounts that are required of each class. These amounts form a set denoted by *tspec*:

$$tspec = \{bw_i \mid i \in C\} \tag{3.1}$$

In the *pipe* model service the traffic specification is defined by the following set:

$$T_s = \{(n_i, n_j, tspec_{ij}) \mid n_i \in V_{ce}, n_j \in V_{ce}\} \tag{3.2}$$

This type of traffic specification is represented by a set of traffic matrixes, one for each of the classes of service defined in the network. For a given service S each traffic matrix for a class of service k $M_k = [m_{ij}^k]$ contains the amount of bandwidth between the nodes i and j of the service.

The traffic specification in the *hose model* is different, and reflects the information that the only information that is available from the user is the aggregated access traffic of the nodes given by the Topological specification to and from the provider network. Here, too, we have a multi COS network, and therefore the aggregates are specified with respect to these classes. Hence, the service is modeled as a set:

$$T_s = \{(n_i, tspec_i) \mid n_i \in V_{ce}\}$$

The *tspec* sequences describe the bandwidth for each of the COS used by the service, for both the ingress and egress directions.

A VPN service can request only a subset of the network wide classes of service. Therefore, some services might have only one class of service type while others might use multiple classes.

Service Adaptability (B_{AS}). This property specifies if the service can be moved inside the provider network after it has been set up. A service that has this characteristic can experience periods of loss which are due to the service movement process, rather than the P node congestion or topological failure. The main differences between this type of service unavailability and the one generated by topological failures refer to the frequency of event occurrence and the duration of outage. Adaptive movements of service have higher frequency than the topological failures, because they are generated by the network control system, and are short in duration. This property is useful to the network control system, if it wants to re-optimize the service layout of the network. It is considered that the acceptance of this criterion is strongly user dependent. Hence, adaptability is modeled as a Boolean variable that is *true* if the service is adaptable and *false* if it is not.

Service Resilience (B_{RS}). This service specification element expresses the service durability against provider side topological failures. Consider the case when a node in the provider network becomes unavailable. If no topological resilience is specified for a service that passes through the affected node, then the service will be affected for the duration of the outage. The services that have been specified as resilient to topological failures will be set-up in a manner that allows the network to reroute them through other nodes for the duration of the outage. Such an option allows the user to retain extremely high availability rates even when there are unforeseen topological faults, while the provider has a service differentiation mechanism. In our specification, resilience is a property of the service instance rather than of the class of service. This choice is observed on the observation that in today's business environment all classes of a service, regardless of their QoS requirements, have connectivity necessities that are of equal importance. For example, in most of today's businesses when the data service that links the business units fails, the effect is of greater magnitude than the cessation of the phone services. In some of the cases, for example in e-commerce businesses, such an interruption is equivalent to a complete loss of business for the period, regardless of the availability of phone services. Hence, it is considered that the topological

connectivity characteristics are important, and therefore, it will be considered as such.

Both VPN services that follow the pipe model as well as services that follow the hose model can be made resilient against both link and node failures. In both the pipe and the hose models the tolerance to topological failures is a characteristic of the service and not of the connection. Hence this property can be modeled as a Boolean variable that takes the value *true* if the service is resilient to topological faults and *false* if it is not.

In the pipe model, the logical network to be realized by the provider can be designed by the customer and presented to the provider for realization, or it can be suggested by the provider, based on price considerations. Hence, in the pipe model there are potentially two optimization problems to be solved: one to solve the logical topology and one to map the logical topology to the physical topology. By contrast, in the hose model there is a single problem to be solved, namely the realization of a logical topology that connects the given end-points.

3.3 Summary

The first section of the current chapter, introduces the graph theoretic model of a network that carries VPN services. This is the network model that is used throughout the dissertation. The chapter continues with the presentation of the VPN service specification. Such a service is viewed as having four main elements: topological characteristics, traffic description, service adaptability characteristics and resilience to topological faults.

Chapter 4

Service Centric Network Control Architecture

This chapter introduces a general network control framework that decouples the service definition from the network-element based network control. The architecture is general enough to be used with any type of packet-switched network and, with some restrictions, can be used for the control of services in circuit-switched networks, although this case is not addressed.

The functional description of the network control plane in Chapter 2 reveals that some of the network control plane functions are better handled in a distributed fashion while others have better performance if realized in centralized fashion. In addition, this separation makes it possible to create a control system that is more general than the current ITB NCP with respect to service definition.

4.1 NCP ITB Paradigm Limitations

The network control plane maintains topological and traffic information. Together this information forms the network state. If the control plane is completely distributed, there are two fundamental ways of maintaining this information:

- to have identical network state information at all nodes that form the NCP. In this case, all nodes in the NCP have to synchronize their view of the network state. Each node can then use this information to find feasible solutions to network wide resource allocation problems.

- federate partial network state information, that is maintained by the individual nodes, and uses a distributed algorithm to solve the problem of end-to-end resource reservation.

Topological state information can be maintained in both ways efficiently. For the traffic related information distributed NCPs use the first type of solution. In networks with explicit traffic requests most of the control plane computational burden is in the edge nodes. This can become a major bottleneck in the network scalability due to limits of the egress nodes CPU power, as shown by Petriu et all [37]. The measurements mentioned in the paper were made in the context of simple shortest path calculations and setup. This represents the simplest possible topological calculation and hence it is a *lower bound* on the CPU power that is needed for any path calculations.

Network control planes of packet-switched networks with traffic-resource reservations use extensions of link state routing protocols to carry network load related information between the nodes that form the NCP. Link state routing protocols [21, 78] use flooding algorithms to synchronize the network state information of the nodes that run them. This solution limits the number of state changes that can occur in the network, and still allow the NEs to synchronize their view of the network state. For example, consider a network with l links. Then, any network state change will generate $O(l)$ update messages. If the time needed for the messages to propagate to all nodes is τ, then the number of changes to the network state cannot be larger than $\frac{1}{\tau}$. If more than $\frac{1}{\tau}$ reservations per second occur, then the network control plane nodes will not converge in state between two changes. As a result, NEs will calculate resource reservation solutions with incomplete information, increasing the probability of finding a solution that is not implementable in the network. This in turn, generates a resource reservation failure message that leads to a recalculation and retry.

Moving the traffic related information and the decisions about the service layout out of the NE and centralizing then solves the problems presented above. Such a control plane will be called *out-of-box (OTB)* network control plane, to distinguish it from the existing solutions. Although such a system can be designed using only centralized components, some of the ITB components, such as signaling, can still be used by the

OTB control plane.

The following information and functions are centralized:

- maintenance of traffic related network state. Such a model has the inherent advantage of better network state knowledge than any ITB network control plane. In the OTB case, the controller can make near optimal decisions about the service setup, with respect to the network state.

- logical topology mapping for service instantiation. As it was described in Section 2.3.3 this is service specific. Therefore, to allow the provider to specify new services irrespective of the data transport type, it is necessary to move this part of the NCP out of the NE. From a computational standpoint, the logical topology mapping is the most computationally intensive process in an ITB. Moving this task outside the NE is a simple and effective solution to the CPU overload situation presented in [37]. This strategy is also economical, since general-purpose computing systems can be orders of magnitude cheaper than the NE subsystems that run the NE control software.

- service admission control. This is also service specific and as such is should be part of the NE independent NCP.

Centralizing these control plane elements allows better integration with the service definition process. Such a network wide control loop has been proposed and studied by Benmohammed and collegues [3] and by Blanchini et al. [8]. Both works consider the queue levels as the congestion indicator and use a traffic actuation mechanism to maintain them at preset levels. Using the queue levels as the congestion indicator has the drawback of not taking into account that, in most networks, a small amount of loss is tolerated.

4.2 Service Aware Network Control Platform

The reasoning presented in the previous section has directed us towards the design of a centralized network control architecture for general communication services. Although

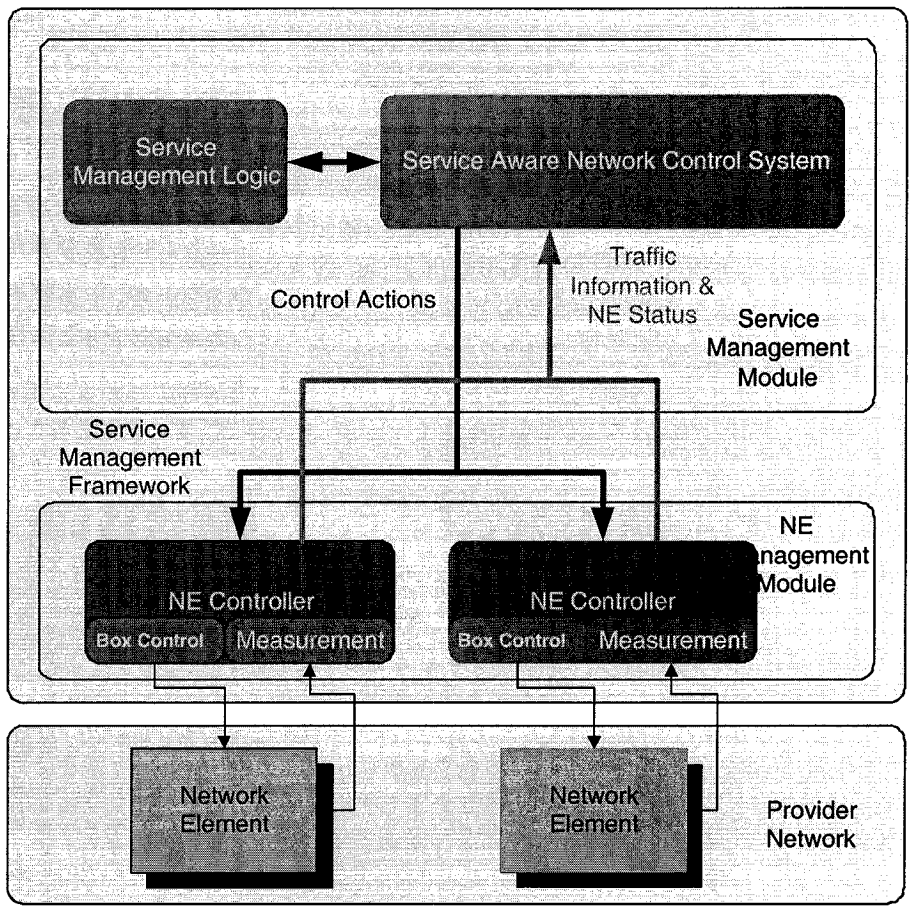

Figure 4.1: System Architecture

the platform is presented in the context of the service specified in Chapter 3, its generality permits it to be used for other types of communication services as well. The architecture has been designed with the following goals in mind:

- Allow the setup of the VPN service specified in Chapter 3.
- Make both the network control and network management planes aware of the services that are provisioned in the network, and hence allow the control system to use service related information when attempting to control congestion.
- Allow customer control over the instantiated services.
- Control the network in a manner that maximizes the provider profit.

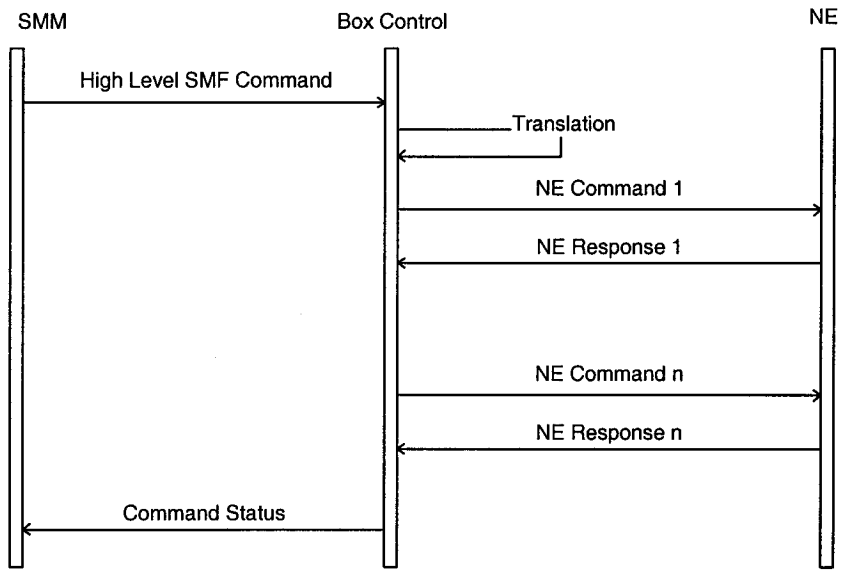

Figure 4.2: Box Control Sequence

The architecture has two main modules (Figure 4.1): the *Service Management Module (SMM)* which contains all the service management logic and the network control systems and the *NE Management Module (NEMM)* which is responsible with the interaction with the NEs.

NEMM has a number of NE Controllers (NETC) equal to the number of NEs that are under the administration of the system. Therefore, there is an NETC module for each P node and a NETC module for each PE node. Furthermore, all CE nodes that are administrated by the provider are also controlled by a NETC. Each NETC module has two main functional uses, which are reflected in its architecture.

It can be observed that the service activation commands that are generated by the SMM have to be translated into commands that NEs can execute. The command translation module and the machine dependent responses are processed by the *Box Control (BC)* system. This part is highly NE dependent, and insulates the SMM from the NE command languages.

SMM commands are translated into NE commands. Figure 4.2 depicts the sequence of events that take place when the NETC receives a command from the SMM. The high level command is translated into the set of NE commands. Each is sent to the NE and

the state of the machine after its execution is assessed. At the end of the process, a reply is sent back to the SMM relaying status of the executed commands. As a heuristic, the BCs should be as close as possible (communication wise) to the NEs that they control. To explain why, consider the case where the BC and the SMM systems run on different machines. Let us denote by t_h the message latency between the two machines. Also, let us denote by t_{ne} the latency from the BC machine to the NE that it controls. If an SMM command translates into n NE commands, the execution of the SMM command has a latency that is generated in part by the message passing between the machines, and it is equal to $2t_h + 2nt_{ne}$. Therefore, in order to optimize the system for response time, it is necessary to reduce the latency of communication between the NETC and the NEs rather than the communication latency between the SMM and the NE.

The framework monitors the status of the network, both from a topological perspective and from a traffic perspective. For the latter it is assumed that it is possible to gather traffic statistics from each of the egress interfaces that are part of the controlled network. Furthermore, it is assumed that either the PE or the CE nodes have a device that permits traffic conditioning by the control platform. The platform collects traffic data from each of the egress interfaces of all P nodes under its supervision. As such, each BC contains a Measurement Agent that collects traffic information from all the interfaces that are part of the NE.

The traffic monitoring system performs continuous observations of the traffic arrivals over pre-determined intervals. The information collected by these processes is used for detection and notification of QoS violations, for the global traffic rebalancing and for statistics collection. The information made available by the Monitoring Agent is used in a two-fold manner:

- To make service admission control decisions. For this purpose, the traffic information collected in the core nodes is forwarded to the service admission control module.

- Since the user traffic patterns can change over time, admission control based on measurements can lead to future traffic violations. Hence, it is necessary to have a mechanism that corrects this type of events.

4.2.1 Service Management Module

The core of the network control platform is the *Service Management Module (SMM).* Figure 4.3 shows that the component of the SMM which takes the final control decisions is the *Service Management Logic (SML).* The main functions of the SML are:

- Interact with the administrative users on both the provider and customer sides.
- Maintain all the service related information.
- Take the final network control decisions, based on the scenarios presented by the other modules.

In this architecture, the service knowledge becomes crucial because it allows the control system to take decisions based on service related information, rather than simple network state information. For example, consider a network that has some VPN services, that follow the definition of Chapter 3, already instantiated. If the strategic control system determines that a rebalancing is necessary, having information about the *adaptability* of the service is important. If some services are adaptable, then they will be given the primary consideration for movement.

Two types of user interactions are envisioned for systems that implement the architecture. The first is the normal administrative interaction from the administrative entity of the system. The second interaction is between the system and the customers of services that have already been instantiated. The customer side interface allows a limited set of commands that relate to the control of its service. One of the commands available for this interface is the service resizing command. This automates the process of service changes by integrating a service resize process with the SMM.

The provider administrative interface allows a user to request new service instances. A new service instantiation request follows the sequence described in Figure 4.4. The new request is passed from the GUI to the SML which then requests a solution from the optimization module. If a feasible solution to the request is found, then the result is returned to the SML. The SML will try to activate the service by setting up the NEs. If the setup is successful then the service is considered instantiated. All the service definition data, the request data together with the solution information, are maintained

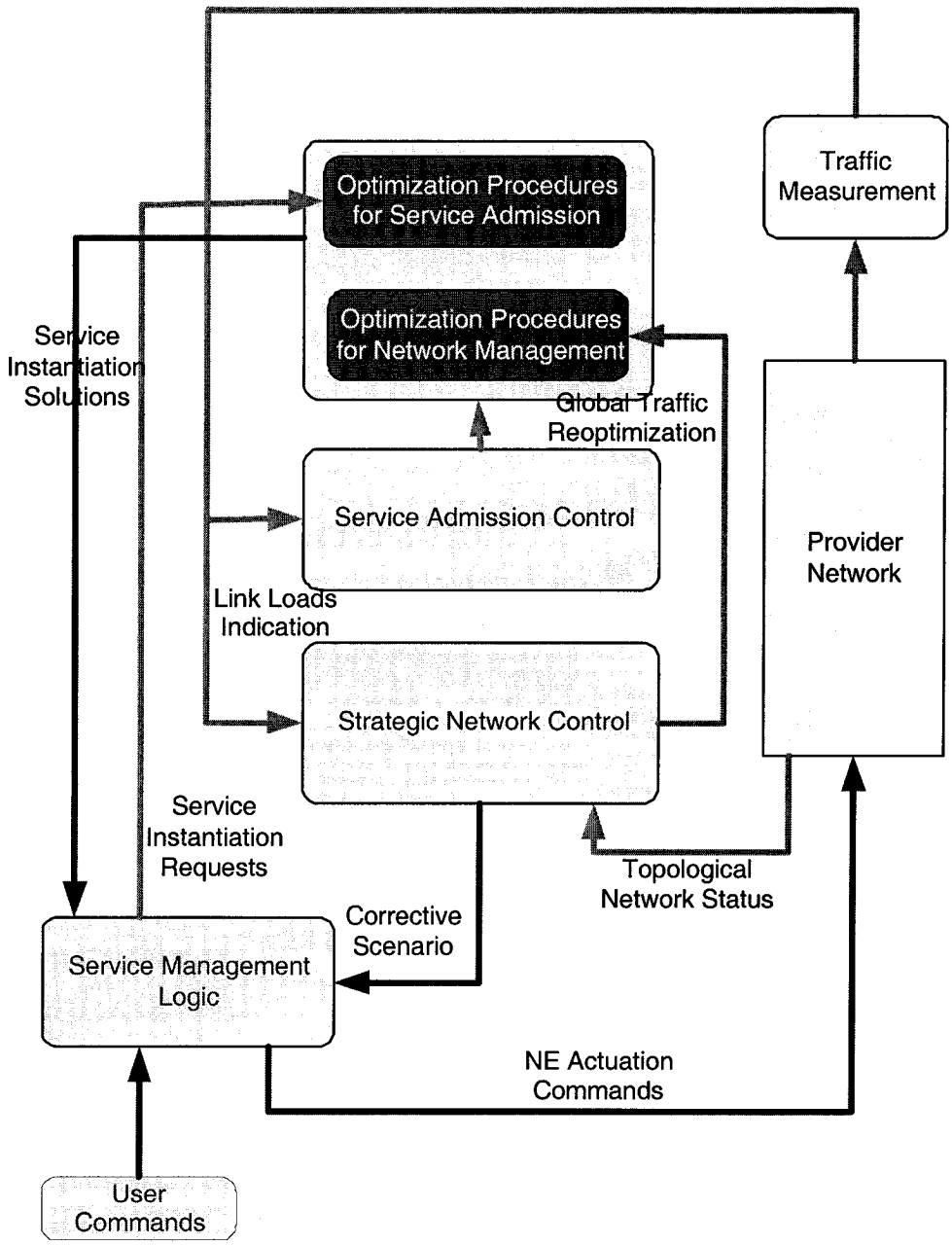

Figure 4.3: Service Management Module Architecture

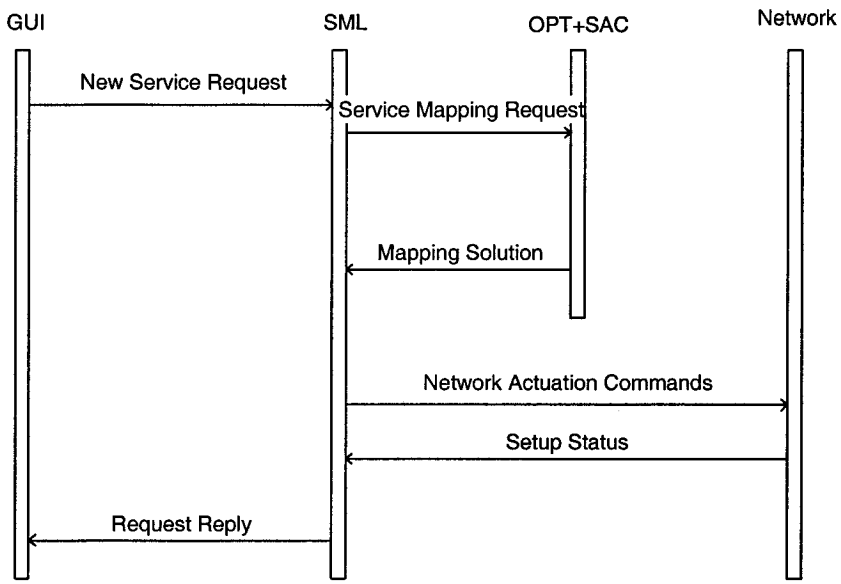

Figure 4.4: New Service Request Sequence

by the SML while a notification is passed back to the requesting entity. If the service activation fails due to some NE related issue, then the next course of action is decided by the SML. The courses of actions taken in case of failure are not considered crucial for our exposition, and therefore are disregarded.

The OTB control plane is concerned with the mapping of the service to the physical topology together with the initial service admission control. This part of the SMM has two components that work in conjunction:

- *Service Topology Realization (STR)* obtains topological state information about the network and calculates a mapping of the logical service topology to the physical topology. The centralized nature of the system decreases the amount of service recalculations (cranckbacks) that result from the inability of the system to activate a solution due to complete lack of information about the state of the network. However, this problem cannot be eliminated entirely even in this scenario, because faults can always appear between the time a feasible solution is passed back to the SML and the actual programming of that solution into the NE. A centralized solution is justified based on the service instantiation workflow. All new service requests are centralized before any instantiation procedures are started. Therefore,

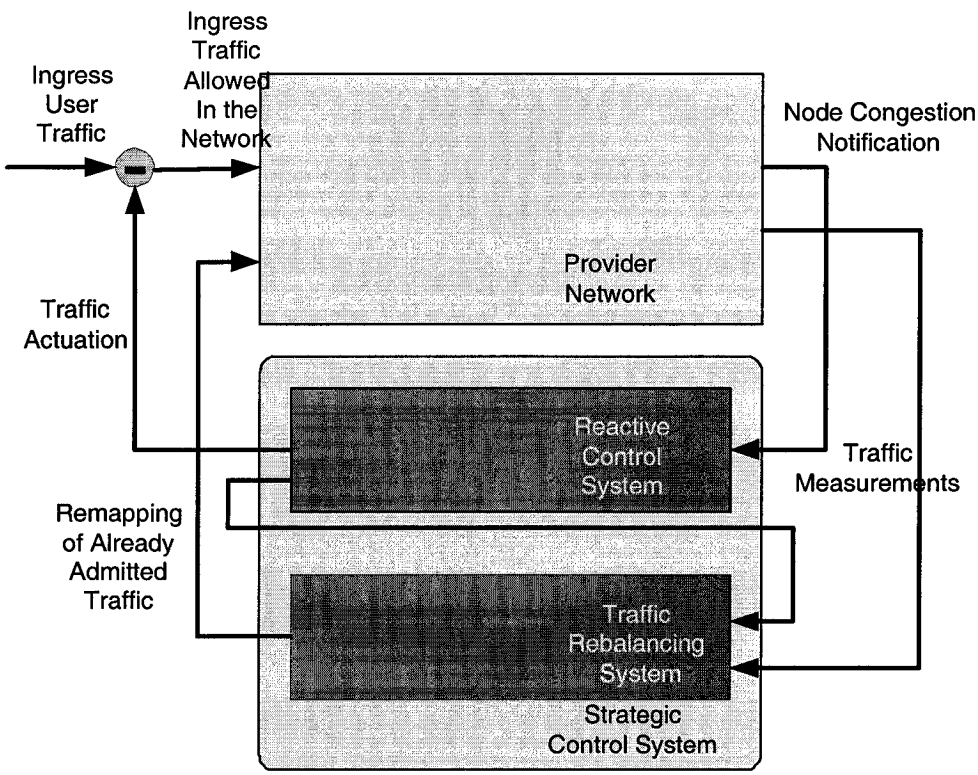

Figure 4.5: Strategic Control Mechanism

the STR can be done at this stage without any overhead.

- *The Service Admission Control Subsystem (SAC)* solves the problem of nodal service admission based on knowledge of the service requests and the traffic loads that are seen at an interface. The STR subsystem uses the SAC subsystem to make the admission determinations.

The tradeoff between network link utilization and admission control has been discussed in Chapter 2. If a measurement based admission control is used, it is possible that although the QoS parameters of all admitted services are satisfied at the time of admission, changes in user traffic patterns might make some nodes to breach the QoS guarantees. Therefore, a mechanism for remedying this occurrence is required. The control mechanism that fulfills this prerogative is called *Strategic Level Controller.* (Figure 4.5). The control system contains two control loops that try to repair QoS violations that mirror the two possible scenarios for restoration of the SLA guarantees.

Consider an egress interface of a packet switched NE, which is shared by flows of several services; if the node experiences a larger amount of loss than the amount allowed by the SLA it is possible to repair this situation by:

- reducing the traffic allowed in the network by some or all of the services that send traffic to the node.
- moving the traffic of a subset of the services that use the node to other parts of the network, hence decreasing the overall load of the node.

Since reducing the ingress bandwidth is revenue affecting for the provider this strategy is useful when the network it heavily loaded in all nodes. As Chapter 3 details, some of the service instances are adaptive, while some are not. If no service instances that pass through the affected node are adaptive, then the optimal control strategy depends on the penalties that the provider has to pay for traffic interruptions that are suffered during service movements, as opposed to the penalties that are to be paid due to the reduction of bandwidth. Therefore, the optimization criterion for the control system is to minimize the revenue loss.

It should be stressed that in our architecture the strategic controller only presents a solution to the SML, which takes the final decision on actuating the network. The controller is then informed of the decision by the SML. This mode of operation has been chosen because the SML contains information about the service instantiations that are being setup, and hence it avoids the possibility of concurrent control over the interior P nodes by both the strategic control and the SML.

4.3 Service Resizing Paradigms

The user driven service-resizing feature increases the level of service control automation in provider supported virtual private services. Our work has identified two main paradigms for the interaction between the users and the provider network after the VPN service has been instantiated. The two types of interactions will be named *Mostly Open Interation (MO)* and *Mostly Closed Interaction (MC).*

The MC paradigm (Figure 4.6) needs a single point of traffic policing, which has to be under the provider administrative control. It is assumed, without loss of generality,

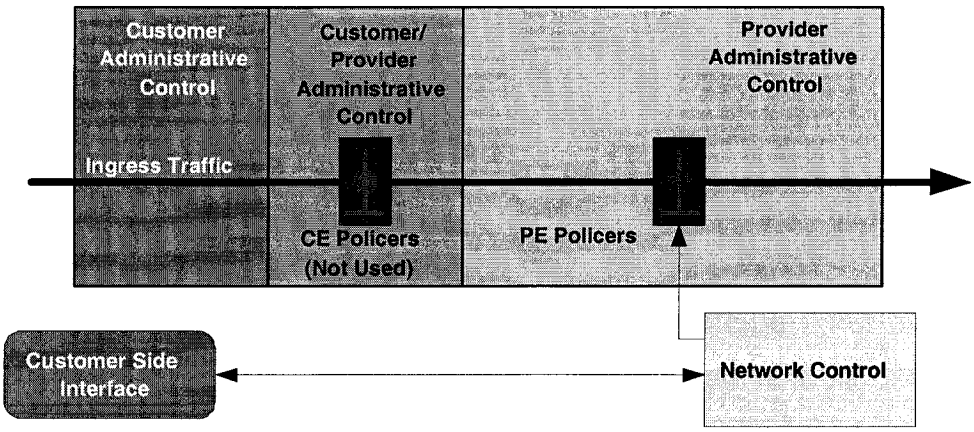

Figure 4.6: Mostly Closed Resizing Control Paradigm

that the PE NE has policing capabilities for all of the QoS classes that are part of the service instance. Therefore, in this paradigm, the CE can be under either customer or network administration. Immediately after the service instantiation the PE based policers are set to the values agreed upon in the SLA. Furthermore, it is considered that the NC supports a mechanism that allows the customer to send resizing requests to it. Such a mechanism can be easily implemented in a web-based application.

The interaction between the customer and the network control system proceeds as follows:

- The customer asks for a service increase. The request can be timed, in which case the starting time, as well as its duration, has to be specified. If no duration bounds are known, then an untimed request is issued, in which case the control system considers it as a permanent service resizing.

- The NC processes the request sends an indication of acceptance/rejection to the customer. If there is a feasible solution to the resizing request the NC system will set the PE policers to their new values for the duration of the request.

When this type of interaction is used, the price of the extra bandwidth can be set either a-priori or through auction type mechanisms. Both pricing schemes have merits, and choosing one over another is a complex economic matter that is out of the scope of this dissertation.

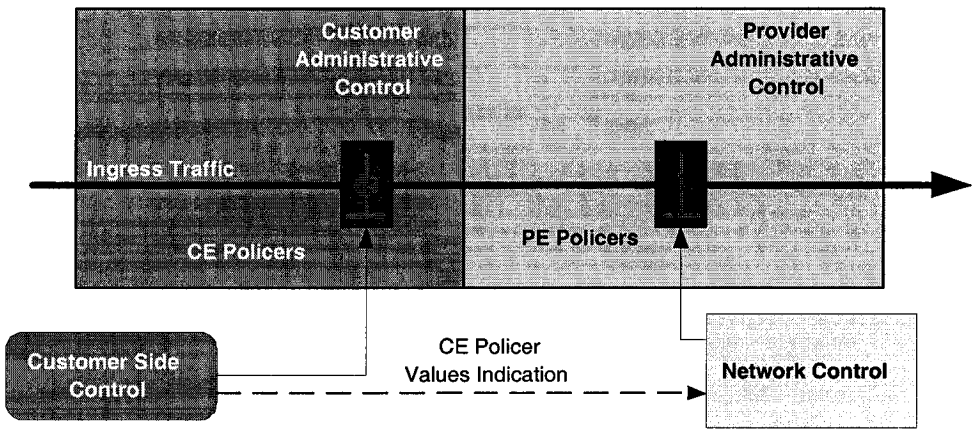

Figure 4.7: Mostly Open Resizing Control Paradigm

The *MO* concept (Figure 4.7) approaches the dynamic resizing problem from a different direction. In this case, the customer-provider SLA contains two types of pricing aggrements:

- The price for a minimum bandwidth that the provider will support for each class of the service instance, together with that minimum bandwidth value.
- A set of price functions $f^j(t, bw)$ for the extra bandwidth per class of service j. Hence, the following set is defined:

$$\mathcal{P} = \{f^j : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid j \in \mathcal{C}\}$$

  The first parameter of the pricing function denotes the time. This is introduced in order to compensate for the uneven network utilization that appears due to the temporal variations that have been detailed in Chapter 2.3.2. Since there is a higher demand for bandwidth during regular business hours it is natural that the price for this *finite* resource to be increased during this period. This type of pricing has been used for a long time in POTS networks, where the cost of a phone call varies as a function of the time of day.

  In addition, the pricing functions can be negotiated on a per customer basis, and therefore are considered to be different for each service instance. The reasoning behind this assumption is that each service instance spans a different number of

network links. Consider a class j of a service instance s. From the service instance, consider two nodes n_1^s and n_2^s. If the number of links that are used in order to assure a connection between n_1^s and n_2^s is $s(e)$ (in either pipe or hose models), then the total amount of network bandwidth that is used in order to provide a the service is $\mathcal{O}(s(e))$. Therefore, the longer the service distances the higher the amount of network resources needed and the higher the price per bandwidth for that service.

Furthermore, it is assumed that the network will provide extra bandwidth, at the prescribed loss parameters, only if it is available.

MO systems have an administrative delimitation between the provider and the customer sites at the PE-CE boundary. Precisely, the PE NE is under the provider administrative control, while the CE NE is under the customer administrative control.

In this control paradigm the customer sets its CE policers to the values that reflect the amounts of traffic that it wants to receive. The NC has to set the PE policers in a manner that satisfies the following constraints:

- Satisfy the loss limits for each of the classes that are part of the network.
- Satisfy the minimum bandwidth requirements of all service instances that are running on the network.
- Distribute the extra capacity, in order to maximize the provider profit. Since the price per service is usually different for each user this constraint is different from a normal fairness constraint towards users.

4.4 Dynamic Resizing of Instantiated Services in Lightly Loaded Networks

The provider side of the network has two different zones of operation. The network can be lightly loaded, which is the case when for all egress interfaces of all network nodes the sum of the peak rates of all services that pass through the interface is less than the link capacity. When this is not the case, the network will be considered heavily loaded.

It is clear that in the former case, the network loss is insignificant and un-quantifiable, while in the latter loss occurs with higher frequency.

If the network is lightly loaded then the resizing control mechanisms are different than in the second operational zone. For this case, to make the process of dynamical resizing as responsive as possible, the resizing control system is separated from the normal service request path. The new control units that are responsible for the resizing of data are called *resizing control units (RCU).* In this architecture, the RCUs act as concurrent controllers independent of the service instantiation process. The service instantiation is handled separately as described in Chapter 4. This part of the control system will be referred to, in this section, as the *system controller (SC).* In this section we address only the FIFO scheduler case. In the PQ cases the same reasoning can be applied by using separate controllers for each COS.

For each link under its control, the RCU unit requests from the service management unit amounts of bandwidth for its own use. It will then utilize this bandwidth to fulfill service-increasing requests. The maximum amount of bandwidth that the RCU unit is allowed to fulfill is fixed at a constant. This method has the advantage of making reasonable increases to be approved or denied fast, while large increases are deferred to the SC because they might entail topological changes. The amount of bandwidth that a RCU unit can control is also upper-bounded. Then, if this value is surpassed, the RCU unit will return the excess bandwidth to the SC. This arrangement is convenient for our control system for the following reasons:

- large increases in bandwidth might not be solvable within the available link capacity.
- the SC has to admit new services; if the RCUs have too much available bandwidth under control and the requests do not use it, then the system will work in a suboptimal manner.

Algorithm 1 describes the process of an increase in the service requirements, while algorithm 2 describes the opposite procedure.

The notation utilized in the Algorithms 1 and 2 is as follows:

- bw the amount of requested bandwidth.

- l_{RCU} the upper limit for the bandwidth increase that is solved by this unit.
- $\mathcal{L}$ the set of affected links
- SAC service admission control
- m_b amount of new bandwidth that is needed by the RCU unit to fulfill the request.
- q_{RCU} bandwidth quota for a link that can be maintained by the RCU.
- abw available link bandwidth after the resizing operation

Algorithm 1 Service Increase Procedure

```
1: if bw > l_RCU then
2:    pass request to the SML unit
3: end if
4: for all l ∈ L do
5:    if SAC(bw) ≠ OK then
6:       m_b ← Necessary(bw)
7:       if m_b > q_r then
8:          pass request to the SML unit
9:          exit
10:      end if
11:   end if
12: end for
13: update the egress traffic policer with new traffic parameters
```

Algorithm 2 Service Decrease Procedure

```
1: update the egress traffic policer with new traffic parameters
2: for all l ∈ L do
3:    if abw > q_RCU then
4:       pass the excess bw back to the SC
5:    end if
6: end for
7: update the egress traffic policer with new traffic parameters
```

The basic procedures have to be adapted to the type of service model that is used. The adaptation is simple and is limited to a procedure that determines the links that are affected by change. Consider one of the graph links that are part of a service instance. Potentially, several services pass through this link. As the requests for service resizing are asynchronous, several requests for resizing of services that are on the same link can

be received by the RCU. Therefore, it is necessary to make sure that resize requests are atomically updated along the whole path for one service before other requests are considered.

Since the communication between the nodes along the path and the RCU can take a significant amount of time, serializing *all* of the resizing requests is not optimal with respect to response time. The proposed solution is to parallelize as many of the requests as possible. Consider that requests are queued at the controller in the order in which they are received. Denote by $\mathcal{R}_n$ the n request in the list, where $\mathcal{R}_1$ denotes the first request.

For each request build the set of links that are affected $\mathcal{L}_n = \{l \mid l \in \text{path of } \mathcal{R}_n\}$. The aim is to build dependencies lists in order that requests can be processed in parallel. Commands are executed in rounds, each round being considered atomic.

This is done by using the following Algorithm 3. The notations are as follows:

- D_k denotes the set of commands that can be executed in the k round.
- $\|D_k\|$ the cardinality of the set D_k.
- D_{jk} a service whose command will be executed in round k.
- $\mathcal{L}(D_{jk})$ the links that are affected by the service resizing request.
- n denotes the number of queued commands.

Proposition 4.1 Algorithm 3 terminates in $\mathcal{O}(n^2)$ operations, where n is the number of queued commands. ■

Proof: The first for loop takes n iterations to complete, while the second for loop takes another n iterations. The body of the second loop has the worst case of n iterations. Therefore the second for takes $\mathcal{O}(n^2)$ operations.

The whole algorithm terminates in a maximum of $n + n^2$, therefore the complexity is $\mathcal{O}(n^2)$.

■

The RCU system can be distributed among several entities if each controller has its domain of authority over a set of services and links that are disjoint from the other

Algorithm 3 Build Parallel Commands Groups

```
1: for all j ∈ [1..n] do
2:   D_j ← φ
3: end for
4: for all j ∈ [1..n] do
5:   k ← 0
6:   while D_k ≠ φ do
7:     if L_| ∩ (⋃_{k=1}^{||D_j||} L(D_jk)) = φ then
8:       break
9:     else
10:      k ← k + 1
11:    end if
12:    D_k ← D_k ∪ {L_j}
13:  end while
14: end for
```

controllers. This separation ensures that any request to the controller can be solved within that domain.

The partition among separate tactical RCUs can be made in an *a priori* manner, if the strategic controller will set up the services such that they respect the tactical control rule. Or in an *a posteriori* manner, if the service setup is done without knowledge of this rule and the assignment to tactical units is done independently.

For the *a posteriori* process the architecture utilizes Algorithm 4. The notation used in the description of the algorithm is:

- C_k denotes the set of services that are under the management of tactical controller k.
- $\mathcal{S}$ the set of all services that are setup in the system.
- s a service instance.

Proposition 4.2 Algorithm 4 has $\mathcal{O}(n^2)$ complexity, where n denotes the number of service instances that are in service within the network. ■

Proof: The first for loop takes n iterations to complete, while the second for loop takes at most n iterations, because there can be as many tactical controllers as there are services. Therefore the second for takes $\mathcal{O}(n^2)$ operations.

■

Algorithm 4 Service and Link Partition

```
1: for all s ∈ S do
2:   S ← S − {s}
3:   for all C_k ∈ C do
4:     if L(s) ∩ L(C_k) ≠ φ then
5:       C_k ← C_k ∪ {s}
6:     else
7:       C ← C ∪ {C_new}
8:       C_new ← {s}
9:     end if
10:  end for
11: end for
```

4.5 Summary

This chapter presents an overview of the proposed control architecture for QoS-VPN services. Because of the limitations inherent in distributed control planes, our framework is based on an OTB architecture that assembles proactive and reactive control elements into one unit. Each new service request is instantiated in the network by using a proactive approach: the load of the network links are evaluated at the time of admission and the service is instantiated, or not, based on this.

The reactive control system thrives to maintain the class loss targets for all the instantiated services. To achieve this goal it utilizes two main mechanisms: ingress rate reductions of the services that have been instantiated and service movements if there is spare capacity in the rest of the network. The two types of control actions are used in conjunction, based on the service management logic and the type of resizing mechanism that is used. In the case of the MO a dynamic rate reduction mechanism is implied by the structure of the system.

Chapter 5

Loss Determination for Scheduling Disciplines

Work conserving schedulers can be modeled as state-space dynamic systems, where the state of the system is the tuple determined by the instantaneous lengths of the queues[1] that are attached to the multiplexer. The model can be endowed with either finite or infinite buffer space. The former system experiences actual loss, which is the measure of interest in this chapter, while for the latter this measure is estimated through the length of the queue that is attained in a sample path. If the model has an infinite queue, all ingress traffic that enters the multiplexer is eventually sent on the egress link. In this case loss does not actually occur, but it is assumed that packets that encounter a queue that is longer than a fixed finite) value b are of no use and are considered lost by the application.

5.1 Problem Formulation

Consider a multiplexer with $n \geq 1$ queues attached to it. Each queue is indexed by a natural number $j \in [1, n]$. The input signal of the multiplexer is a vector of stochastic processes, $\mathbf{X} = [\mathbf{X}^1, ..., \mathbf{X}^n]$, where each vector component is a stochastic process $\mathbf{X}^j = \{\mathbf{X}_t^j \mid t \in T\}$. Each random variable is defined over a probability space $(\Omega, \mathcal{F}, P) = (\mathbb{R}, \mathcal{B}(\mathbb{R}), P)$, where $\mathcal{B}(\mathbb{R})$ denotes a system of Borel sets over the set $\mathbb{R}$. Time is considered to be discrete, hence isomorphic with $\mathbb{Z}$.

The main assumption on $\mathbf{X}^j$ is that it is stationary in the wide sense [92]. Therefore,

[1] Throughout this chapter *queue* and *buffer* are considered synonymous.

$\forall t \in T\ \mathbb{E}(X_t^j) = \mu$, and $\mathbb{E}(X_{t_1+s}^j X_{t_1}^j) = \mathbb{E}(X_{t_2+s}^j X_{t_2}^j)$. As discussed in Chapter 2, these two conditions do not hold, for real traffic, over all timescales. However, they hold over shorter time scales, and as such the arriving process will be examined over a window of length T, in which the stationarity assumption is assumed to hold.

The total amount of traffic arriving at a queue j in an interval of length t, $(-t, 0]$, is denoted by $A_j(t)$. In the discrete time setting this random variable is equal to:

$$A_j(t) = \sum_{k=-t}^{0} X_k^j$$

Hence, $A(t)$ is equal to:

$$A(t) = \sum_{j=1}^{n} A_j(t) = \sum_{j=1}^{n} \sum_{k=-t}^{0} X_k^j$$

In a similar fashion, let us denote by $C(t)$ the amount of traffic that the multiplexer can output in an interval of length t , $(-t, 0]$. Similarly, by $C_j(t)$ we denote the amount of traffic that is outputted from queue j in the same time interval. If the scheduler is coupled directly to an egress link, $C(t)$ is constant. It should be noted, though, that for more complex scheduling constructions, this might not be the case. For example when the output of one scheduler can be the input to another. In the discrete time semantic we denote by c_t the amount of traffic that exits the multiplexer in the interval $(t, t+1]$. Similarly c_t^j denotes the amount of traffic that leaves queue j in the interval $(t, t+1]$. Thus, in a discrete time environment:

$$C_j(t) = \sum_{k=-t}^{0} c_k^j$$

The value of $C(t)$ is equal to:

$$C(t) = \sum_{j=1}^{n} C_j(t) = \sum_{j=1}^{n} \sum_{k=-t}^{0} c_k^j$$

The next section details how the length of a queue is linked to the amount of traffic that enters it for different types of multiplexers. Because of the direct dependency between the two and because the traffic is modeled stochastically, the length of a queue is also modeled as a random variable $Q_j, \forall j \in [1, n]$.

5.2 State Models For Work Conserving Schedulers

In the current section, let us denote by $\mathbf{q}_t$ the length of the queue at time t and by $\mathbf{c}_t$ the amount of traffic that can exit the multiplexer in the interval $(t-1, t]$[2]. The amount of traffic that exits the multiplexer in the same interval is denoted by z_t.

5.2.1 FIFO Scheduler

A finite buffer FIFO scheduler is a non-linear dynamic system that can be modeled in a state-based framework as a system with *two* inputs *two* outputs (MIMO). The state evolution is described by Equation 5.1[3]

The amount of traffic that exits the scheduler can be quantified by observing that it is the minimum between the link capacity at time t, and the sum between the amount of traffic already present in the queue, together with the amount of traffic that arrives at time t. This description is made precise by Equation 5.2. The amount of traffic lost at time t is equal to the difference between the first term of the min expression in Equation 5.1 and the second term of the same expression. Precisely, if we denote by this difference by l_t then Equation 5.4 holds.

$$\mathbf{q}_{t+1} = \min([\mathbf{q}_t + \mathbf{X}_t - c_t]^+, B) \tag{5.1}$$

$$\mathbf{z}_t = \min(\mathbf{q}_t + \mathbf{X}_t, \mathbf{c}_t) \tag{5.2}$$

$$\mathbf{l}_t = [[\mathbf{q}_t + \mathbf{X}_t - c_t]^+ - B]^+ \tag{5.3}$$

$$= [\mathbf{q}_t + \mathbf{X}_t - c_t - B]^+ \tag{5.4}$$

The loss experienced by the multiplexer during each period is not a useful measure. A more appropriate quality of service measure is the given by the probability of traffic loss which is defined as the ratio between the expectation of loss to the first moment of the traffic input.

[2]This is the multiplexer capacity in the interval $(t, t+1]$

[3]The semantic of the notation $[a]^+$ is defined as $[a]^+ = \max(a, 0)$.

$$\mathbf{P}_{loss} = \frac{\mathbb{E}[l_t]}{\mathbb{E}[X_t]} \tag{5.5}$$

For FIFO schedulers that aggregate traffic from a large number of sources, it is of interest to quantify the loss experienced by the individual sources, as well as the loss of the traffic aggregate. To do this, let us first consider the simplest case, from the point of view of the number of sources: a two-source system that feeds a FIFO multiplexer. With this case we can model more complex situations with larger number of aggregated sources by considering two sets: one for the individual source that we want to study whose arrival traffic is denoted by $A_2(t)$ and, another that contains all the other sources whose traffic is denoted by $A_1(t)$. Let us also assume that the capacity of the link is significantly larger than the maximum source rate (peak rate) of the source under study; the ratio of the buffer size available at the multiplexer relative to the source peak rate is large; the traffic of the aggregate is assumed to be Gaussian. Under these hypotheses we approximate the ratio of traffic lost by the second stream to the traffic lost by the first stream to be equal to the ratio between the averages of the two sources:

$$L_{ratio} \approx \frac{\mathbb{E}[A_2(t)]}{\mathbb{E}[A_1(t)]}$$

In this case, the loss experienced by source number two is:

$$P_{(loss,2)} = \frac{\mathbb{E}[l_2(t)]}{\mu_2} \tag{5.6}$$

$$= \frac{\mathbb{E}[\frac{l_{total}(t)}{1+1/L_{ratio}}]}{\mu_2} \tag{5.7}$$

$$= \frac{\mathbb{E}[l_{total}(t)]}{\mu_2(1+1/L_{ratio}} \tag{5.8}$$

$$= \frac{\mathbb{E}[l_{total}(t)]}{\mu} \tag{5.9}$$

$$= P_{loss} \tag{5.10}$$

Therefore, under the above assumptions, the per-source loss *ratio* is equal to the aggregate loss *ratio*. The validity of this argument is verified via simulations in Chapter 6.

5.2.2 Multi-Queue Service Disciplines

In multi-queue scheduling disciplines, individual queue dynamics is strongly dependent on the scheduler service discipline. Although the class of work conserving disciplines is large, the types of implementations that can be found in NEs fall mostly in two categories: the priority queuing system and packetized implementations of GPS service discipline. Hence, only these two service disciplines are discussed in the reminder of the chapter.

In the multi-queue service disciplines, we define the state of the multiplexer to be equal to the lengths of the individual queues that compose the system. Therefore, for a scheduler with n queues in a discrete time setting, the state at time t, $\mathbf{q}_t$ is a vector composed of the individual queue lengths:

$$\mathbf{q}_t =< \mathbf{q}_t^1, \mathbf{q}_t^2, ..., \mathbf{q}_t^n >$$

For all work conserving queuing disciplines that have infinite buffers (under the assumptions stated at the beginning of the section) the infinite buffer Lyndley recursion holds. Hence, in this case the following can be stated about the evolution, in discrete time, of the *aggregated queue*:

$$\sum_{k=1}^{n} \mathbf{q}_{t+1}^k = [\sum_{k=1}^{n} \mathbf{q}_t^k + \sum_{k=1}^{n} \mathbf{X}_t^k - c_t]^+ \tag{5.11}$$

The amount of traffic that exits the scheduler from all queues is also invariant and can be stated as:

$$\sum_{k=1}^{n} \mathbf{z}_t^k = \min(\sum_{k=1}^{n} \mathbf{q}_t^k + \sum_{k=1}^{n} \mathbf{X}_t^k, c_t) \tag{5.12}$$

Equation 5.11 denotes a system invariant for all work multi-queue work conserving scheduler. However, its practical value is limited by the lack of information about the *individual* queue dynamics. The evolution of the individual queues is strongly dependent on the scheduling discipline and therefore has to be stated separately for each scheduler.

Priority Queue Service Discipline

Let us index the queues present at the multiplexer with a subset of natural numbers $[1, n]$ with the higher priority queues having a smaller index than the lower priority queues. Based on the system definition, since only the higher priority queues have an influence on the lower priority queues, it said that a queue q_j influences the dynamics of queue q_k iff $j < k$. Thus, the highest priority queue, q_1, does not have any dependencies[4], and as such, it can be analyzed in separation. Therefore, for this queue, all of the results of Section 5.2.1 hold.

Let us analyze the dynamic of the lower priority queues, in a discrete time environment. Over any period, all queues $(q_j, j \in [2, n])$ other than the highest priority one, can utilize only the residual capacity that is left from the higher capacity queues. If c_t^j denotes the residual bandwidth at priority j, at time t, then $c_t^j = c(t) - \sum_{k=0}^{i-1} c_t^k$.

Therefore in the infinite queue model, the dynamic of each queue can be stated as in Equation 5.13.

$$\mathbf{q}_{t+1}^j = [\mathbf{q}_t^j + \mathbf{X}_t^j - c_t^j]^+ \tag{5.13}$$

The traffic that exits on the link from queue j is capped by the residual capacity available to that queue. Hence:

$$\mathbf{z}_t^j = \min(\mathbf{q}_t^j + \mathbf{X}_t^j, c_t^j) \tag{5.14}$$

The dependencies on the higher priority queues can be solved in the above equations. Hence, they can be rewritten as follows:

$$\mathbf{q}_{t+1}^j = [\mathbf{q}_t^j + \mathbf{X}_t^j - [c - \sum_{r=1}^{j-1}(q_t^r + \mathbf{X}_t^r)]^+]^+ \tag{5.15}$$

If the maximum amount of traffic that the queue j can hold is finite and denoted by B_j, then the equation has to be modified accordingly, and the result is Equation 5.16.

[4]Because priority queue systems used in NEs are non-preemptive, there is a lag of at most $\frac{L_{max}}{C}$ between the time that the a high priority packet is received at an empty queue and the moment the scheduler starts to process it. Therefore the argument in Section 2.1.2 applies and this delay is not considered.

$$\mathbf{q}_{t+1}^j = \min([\mathbf{q}_t^j + \mathbf{X}_t^j - [c - \sum_{r=1}^{j-1}(q_t^r + \mathbf{X}_t^r)]^+]^+, B) \tag{5.16}$$

$$\mathbf{z}_t^j = \min(\mathbf{q}_t^j + \mathbf{X}_t^j, [c - \sum_{r=1}^{j-1}(q_t^r + \mathbf{X}_t^r)]^+) \tag{5.17}$$

$$\mathbf{l}_t^j = [[\mathbf{q}_t^j + \mathbf{X}_t^j - [c - \sum_{r=1}^{j-1}(q_t^r + \mathbf{X}_t^r)]^+]^+ - B^j]^+ \tag{5.18}$$

$$= [\mathbf{q}_t^j + \mathbf{X}_t^j - [c - \sum_{r=1}^{j-1}(q_t^r + \mathbf{X}_t^r)]^+ - B^j]^+ \tag{5.19}$$

The traffic output equation in the finite buffer case remains the same as in the infinite buffer model.

GPS service discipline

It has been recognized for some time, that GPS scheduler systems are easier to analyze in the two queue setup than they are in the multi-queue setup [30, 105, 80, 106, 6]. In this subsection we will show that the GPS scheduler changes its characteristics in a subtle way when the number of queues increases to more than two, hence explaining some of the analysis difficulties that this system presents.

Consider a system with n queues, each queue having a weight of $\phi_j \in [0, 1], \forall j \in [1, n]$. Therefore, the weights are normalized to one, and thus Equation 5.20 holds.

$$\sum_{j=1}^{n} \phi_j = 1 \tag{5.20}$$

Definition 5.1 A set of scheduling weights are said to be *well-defined* if the scheduler behavior under that set is *unique* for a given stream of input traffic. ■

The individual queue dynamics of a two-queue GPS system, in a discrete time setting ($t \in \mathbb{Z}$), is described by the following two equations:

$$\mathbf{q}_{t+1}^1 = [\mathbf{q}_t^1 + \mathbf{X}_t^1 - max(c - \mathbf{q}_t^2 - \mathbf{x}_t^2, \phi_1 c)]^+ \tag{5.21}$$

$$\mathbf{q}_{t+1}^2 = [\mathbf{q}_t^2 + \mathbf{X}_t^2 - max(c - \mathbf{q}_t^1 - \mathbf{x}_t^1, \phi_2 c)]^+ \tag{5.22}$$

If the scheduler has more than three queues attached to it, ($n > 3$) then the following state equation also holds in discrete time:

$$\mathbf{q}_{t+1}^j = [\mathbf{q}_t^j + \mathbf{X}_t^j - max(c - \sum_{j=1, j\neq i}^{n} \mathbf{q}_t^j - \sum_{j=1, j\neq i}^{n} \mathbf{X}_t^j, \phi_i c)]^+ \tag{5.23}$$

For finite buffer systems, the queue dynamics are modified accordingly. Therefore, we can state the following:

$$\mathbf{q}_{t+1}^1 = \max([\mathbf{q}_t^1 + \mathbf{X}_t^1 - max(c - \mathbf{q}_t^2 - \mathbf{x}_t^2, \phi_1 c)]^+, B_1) \tag{5.24}$$

$$\mathbf{q}_{t+1}^2 = \max([\mathbf{q}_t^2 + \mathbf{X}_t^2 - max(c - \mathbf{q}_t^1 - \mathbf{x}_t^1, \phi_2 c)]^+, B_2) \tag{5.25}$$

For the case when ($n > 3$) the queue dynamics is modified as follows:

$$\mathbf{q}_{t+1}^j = \max([\mathbf{q}_t^j + \mathbf{X}_t^j - max(c - \sum_{j=1, j\neq i}^{n} \mathbf{q}_t^j - \sum_{j=1, j\neq i}^{n} \mathbf{X}_t^j, \phi_i c)]^+, B_j) \tag{5.26}$$

In a GPS system, all queues that have $\phi_k \neq 0$ induce dependency on them from all the other queues. Hence for a system where $\phi_k \neq 0$ is valid for all queues, the dependency graph is a full mesh. Such dependencies are depicted for the two queue system in Figure 5.1 and the three queue system in Figure 5.2.

From the definition of the GPS scheduler, it results that in order to have a well-defined behavior, at most one of the weights can be zero. Let us consider such a system. First, let us consider the two-queue system. Here, $n = 2$ and the sum of weights becomes: $\phi_1 + \phi_2 = 1$. It can be proven, as Theorem 5.1 shows, that a two queue GPS system can be made to emulate a priority queuing system with the same number of queues.

Theorem 5.1 There exists a weight assignment for a GPS system with two queues, that allows the scheduler to behave like a priority queue system with the same number of queues. ■

Proof: Consider that $\phi_1 = 1$ while $\phi_2 = 0$. Since $\phi_1 = 1$, whenever the first stream has something to send, it will be sent. The second queue traffic will be served whenever

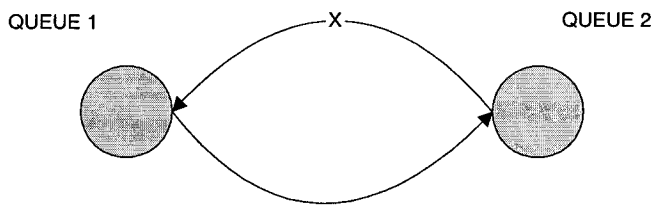

Figure 5.1: Dependencies in the two queue WFQ system. The head of the arrow indicates the direction of dependency: $q_1 \rightarrow q_2$ is read as q_2 *depends* on q_1.

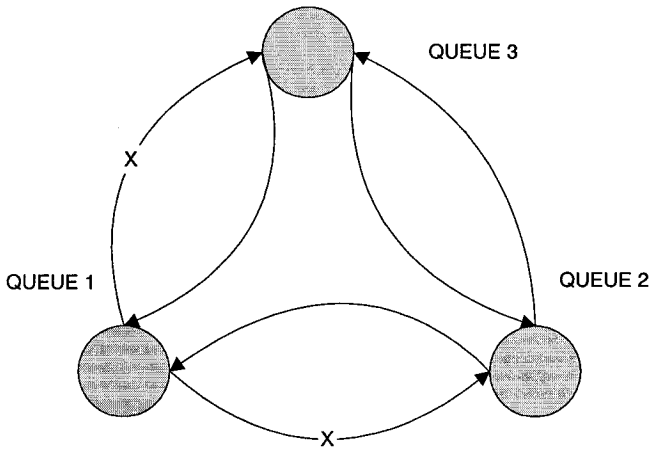

Figure 5.2: Dependencies in the three queue WFQ system. The head of the arrow indicates the direction of dependency: $q_1 \rightarrow q_2$ is read as q_2 *depends* on q_1.

there is nothing to serve for queue one, because the scheduler is *work conserving.*

■

The dependencies of the queues can be represented by a graph. Hence, the two queue case can be depicted by a directed graph with two states as in Figure 5.1. If both $\phi_1 \neq 0$ and $\phi_2 \neq 0$ then the dependency graph is cyclic. If one of the weights is zero, one of the graph arcs is removed and the graph becomes acyclic.

If the scheduler has more than two queues, no acyclic dependency graph can be formed for any well-defined weight assignment. Such a system is only *asymptotically* equivalent with a priority queue scheduler with the same number of queues.

Theorem 5.2 There is no weight assignment in a WFQ system with more than three queues that allows it to completely emulate a PQ system. ■

Proof: Consider a WFQ system with $n > 3$ queues. Denoting, as before, the weights associated with each queue ϕ_j, and considering them normalized at one, we have:

$$\sum_{j=1}^{n} \phi_k = 1$$

To ensure a strict dependency among the queues, there cannot be two queues q_j and q_k such that $\phi_j = \phi_k$. As a direct consequence, there cannot be more than one queue in the system that has its weight equal to zero. Since $n > 3$ there exists at least *two* queues that have weights different from zero. Such queues have their dynamics dependent on each other. Therefore, a GPS weight assignment that creates simple queue dependencies does not exist.

■

If a weight is zero, the dependencies imposed by that node are removed. Hence, in a system with n queues we can remove at most $n - 1$ dependencies. For example, in the case of a scheduler with three queues ($n = 3$), which Figure 5.2 depicts, only two of the links can be removed. Thus, the resulting graph still contains a cycle, and therefore it is fundamentally different from the graph resulting from the same operation in the two-queue system.

Some of the other dependencies in the $n \geq 3$ system can be reduced significantly. However, they cannot disappear. Consider the same system with more than three queues ($n > 3$), with the following weight assignments: all queues $j \in [2, n]$ have weights equal to $\phi_j = \left(\frac{1}{m}\right)^{j-1}$, where m is a large positive number. The series $S = \sum_{j=2}^{n} \phi_j$ is finite, because it is a finite sum of terms, and less than one: $S < 1$. Therefore, $\exists \phi_1 \in \mathbb{R}^+ \quad s.t. \quad S + \phi_1 = 1$. Furthermore, it can be proven that $\frac{\phi_1}{\phi_2} > m$. With a weight assignment as described, some of the queue interdependencies can be made asymptotically small. In practice, such an assignment will not be usable due to the limitation of the number representation systems on NE.

The link between the PQ and WFQ system is important because it can help with the validation of results obtained for a GPS system. In particular, any results for the

two-queue system should have the PQ behavior as an extremal case. Furthermore, since the GPS system can approximate a PQ system, for the purposes of this dissertation we will analyze only the PQ system. GPS schedulers will have the weight assignment presented above and hence the same analysis applies.

5.3 Large Deviations Analysis of FIFO Multiplexers

Consider a FIFO multiplexer fed by an arrival process with the properties described in Section 5.1. We would like to obtain parsimonious formulas for the loss attained by such a system under different traffic models. Using the moment generating function of the arrival process, Kelly [62] defined the *effective bandwidth (EB)* of a traffic source to be:

$$eb(\theta, t) := \frac{1}{\theta t} log\mathbb{E}[e^{\theta A(t)}] \tag{5.27}$$

The right hand side of Equation 5.27 is found in most of the results presented in this section, because of the *large deviations theory* that is applied to obtain them.

5.3.1 Large Buffer Asymptotic

The first model under study considers that the buffer is large. Then, asymptote is used to extrapolate the value of loss attained in the low buffer region.

The assumptions made about the input traffic are crucial for this type of asymptote. The asymptotic results are different for long-range dependent traffic than they are for short-range or Gaussian traffic.

For large buffers, the loss-buffer curve of a constant rate FIFO system is asymptotically linear, as proven by Theorem 5.3. The result is similar to Theorem 9.1 in Chang [14], where it is proven in a discrete time setting. The same type of result has been proven by Duffield [33] in a continuous time setting. Our interest in this type of asymptotes is due to their applicability to aggregate arrival processes, which is the way we approach loss estimation in our control framework.

From the definition of the effective bandwidth, given by Equation 5.27 if time is increased asymptotically to infinity, the effective bandwidth becomes equivalent to:

$$eb(\theta, \infty) = \lim_{t \to \infty} \frac{1}{t\theta} \log(\mathbb{E}[e^{\theta A(t)}]) \tag{5.28}$$

If the Gartner-Ellis Theorem [31] hypothesis are satisfied, then existence of the limit of Equation 5.28 is assured, possibly as an extended real number. By restricting the limit to a finite value, the following can be proven.

Theorem 5.3 [14] Consider a constant rate FIFO system fed by a stochastic process that satisfies the conditions in Section 5.1. Furthermore, assume that the Gartner-Ellis type limit

$$eb(\theta, \infty) = \lim_{t \to \infty} \frac{1}{\theta t} log\mathbb{E}e^{\theta A(t)}$$

exists $\forall \theta \in \mathbb{R}^+$, is *finite*, and it is differentiable. Then:

$$\lim_{b \to \infty} log\mathbb{P}(Q(t) > b) = -\theta^* \tag{5.29}$$

where θ^* is the solution to the equation

$$eb(\theta, \infty) = c \tag{5.30}$$

■

The hypotheses of this theorem are not satisfied in all traffic cases. For example, as shown in the reminder of this subsection, they are not satisfied for long-range dependent processes.

To show why this is true, let us consider a Gaussian Noise Model (GNM) [85]. In the GNM the traffic that arrives over an interval of length t is described by the equation:

$$A(t) = \mu t + Z(t) \tag{5.31}$$

In the above, $Z(t)$ is normally distributed with zero mean. It is dully noted that the model is problematic due to the probability of appearance of negative increments. Nevertheless, since the model has wide acceptance and is parsimonious, this problem is accepted. For such a model the effective bandwidth of this source is given by:

$$eb(\theta, t) = \mu + \frac{\theta}{2t} VarZ(t) \tag{5.32}$$

Let us instantiate $VarZ(t)$ for the two cases presented in [62] by Kelly.

- In the first case traffic is composed of independent Gaussian increments. Therefore, $VarZ(t) = \sigma^2 t$ and the corresponding EB of the source becomes:

$$eb(\theta, t) = \mu + \frac{\theta}{2}\sigma^2 \tag{5.33}$$

From Equation 5.33 it results that $eb(\theta, \infty)$ exists and it is finite, $\forall \theta \in \mathbb{R}^+$, iff μ and σ are finite. Therefore this model satisfies the hypothesis of Theorem 5.3.

- On the other hand, if $VarZ(t) = \sigma^2 t^{(2H-1)}$, where H is the Hurst Parameter [4], then the effective bandwidth of the source can be shown to be:

$$eb(\theta, t) = \mu + \frac{\theta}{2}\sigma^2 t^{2H-1} \tag{5.34}$$

Therefore, if $H \in (0.5, 1)$ then $eb(\theta, \infty) = \infty$. Hence the finiteness hypothesis of Theorem 5.3 is breached.

For long-range dependent data it has been observed that the queue process, if the buffer length is large is not exponential. Results for this have been proven by Duffield and O'Connell [33] and Tsybakov and Georganas [100, 101]. Although acknowledged, this type of results is not considered in this work, because, as it has been shown in [89, 50] the small buffer behavior of long-range dependent traffic is close to an exponential queue decay.

This has important practical implications. It shows that the behavior of the long-range dependent traffic in small buffer settings can be approximated by a logarithmically linear behavior.

Theorem 5.3 encourages the use of a linear approximation for the relationship between tail probability and the length of the buffer. Such linear relationships have been proposed many times before. For example in [18, 27, 14]. Therefore if Equation 5.29 holds, the tail probability for *any* buffer length is approximated by the following expression:

$$\exists b_0 \in \mathbb{R}^+, \exists \zeta \in \mathbb{R}^+ \quad s.t. \quad \forall b > b_0 \quad \log(\mathbb{P}(Q(t) > b)) = -\theta^* b - \zeta$$

Therefore, the tail probability is exponential and can be described by:

$$P(Q(t) > b) = e^{-\theta b - \zeta} \tag{5.35}$$

Theorem 5.3 describes the asymptotic link between the tail distribution and the buffer length, whilst we are interested in the loss probability at the multiplexer. For this we consider a discrete time environment, in which Theorem 5.3 also holds as proven by Chang [14]. In this case the loss ratio at the FIFO multiplexer is given by Equation 5.5. Under the linear approximation described above the following Theorem holds:

Theorem 5.4 Under the assumptions of Theorem 5.3 and using the approximation for the tail loss probability of Equation 5.35, the loss ratio at a FIFO multiplexer, denoted by P_{loss} is:

$$P_{loss} = \frac{e^{-\theta b - \zeta}}{c\rho\theta} \tag{5.36}$$

In Equation 5.36 θ is calculated as in Theorem 5.3. ■

Proof: Assuming that the queue at $t = 0$ does not contain any traffic, then under the same realization of the input process, the length of the queue in an infinite buffer system is less than the length of the queue in any finite buffer system.

Denoting by q_t^f the length of the queue in the infinite buffer system and by q_t^i the length of the queue in the finite buffer system, then:

$$\forall t \in \mathbb{N} \quad q_t^f \leq q_t^i \quad a.s.$$

By comparing Equation 5.1 with the dynamic evolution equation of an infinite buffer FIFO scheduler, we approximate $\mathbb{E}[l_t]$ by:

$$\mathbb{E}[l_t] \approx \mathbb{E}[max(0, \mathbf{q}_t - B)]$$

The right hand side of the above expression can be readily obtained as follows:

$$\mathbb{E}[max(0, \mathbf{q}_t - B)] = \int_B^\infty \mathbb{P}(\mathbf{q}_t > x)\mathbf{d}x$$

Using the linear approximation of Equation 5.36 the integral can be calculated and the result is:

$$\int_B^\infty \mathbb{P}(\mathbf{q}_t > x)\mathbf{d}x = \frac{e^{-\theta B - \zeta}}{\theta}$$

By plugging the result in Equation 5.5 the desired result ensues.

■

Applying the logarithm function to both sides of Equation 5.36 gives us the traffic loss approximation, under the large buffer assumption, to be:

$$log(P_{loss}) = -\theta B - log(c\rho\theta) - \zeta \tag{5.37}$$

This relation will form the basis for large buffer estimator studied in Chapter 6.

5.3.2 Large Number of Sources Asymptotics

This section studies the FIFO multiplexer behavior under the assumption that the capacity of the link and the buffer length are scaled together in proportion with the number of sources that are fed to the multiplexer. This scaling was first proposed by Weiss [102] where a result for small buffers is proven using an argument of sample path large deviations.

The *n-asymptotic* results are based on homogeneity of the sources that feed the multiplexer. As such, consider n identical sources that are multiplexed together in a FIFO system. The strongest results of this category are those of Lykhanov and Mazumdar [73]. They prove, in a discrete time setting, the following result.

Theorem 5.5 [73] Consider a FIFO multiplexer of Section 5.2.1 fed by a large number of independent streams (n) that are instances of the same stochastic process X_t, with capacity c and buffer b *per source*. Define:

$$I_t(c,b) = (ct+b)\theta - log(\mathbb{E}[e^{\theta A(t)}]) \tag{5.38}$$

$$\frac{\mathbb{E}[A(t)e^{\theta A(t)}]}{\mathbb{E}[e^{\theta A(t)}]} = (ct+b) \tag{5.39}$$

$$\omega^2 = \frac{\frac{\partial^2 \mathbb{E}[e^{\theta A(t)}]}{\partial \theta^2}}{\mathbb{E}[e^{\theta A(t)}]} - (ct+b)^2 \tag{5.40}$$

θ is the solution to Equation 5.39.
Furthermore, assume that:

$$\exists! t_0 < \infty \quad s.t. \quad I_{t_0}(c,b) = \min_{t\in\mathbb{N}} I_t(c,b) > 0$$

and

$$\liminf_{t\to\infty} \frac{I_t(c,b)}{log(t)} > 0$$

Then the loss ratio of the multiplexer, when $n \to \infty$, is equal to:

$$P_{loss} = \frac{e^{-NI_{t_0}(c,b)}}{\theta^2 c\rho\sqrt{2\pi\omega^2 n^3}} \tag{5.41}$$

■

It is interesting to investigate the link between the *n-asymptote* and the *b-asymptote*. Considering that $I_{t_0}(c,b)$ exists and it is finite, let us rewrite Equation 5.41 as:

$$log(P_{loss}) = -NI_{t_0}(c,b) - log(\theta^2 c\rho\sqrt{2\pi\omega^2 n^3}) \tag{5.42}$$

$$= -N((ct+b)\theta - log(\mathbb{E}[e^{\theta A(t)}])) - log(\zeta) \tag{5.43}$$

Dividing the Equation 5.43 by N and applying the limit:

$$\lim_{N\to\infty} \frac{1}{N} log(P_{loss}) = -(ct+b)\theta$$

Dividing by b and applying the limit when $b \to \infty$ the following results:

$$\lim_{b\to\infty} \lim_{N\to\infty} \frac{1}{N} log(P_{loss}) = -\theta$$

This is the same as the result of Theorem 5.3.

From theorem 5.5 the following lemma can be proven for the case of Gaussian traffic.

Lemma 5.1 For any large capacity multiplexer fed with a large number (n) of Gaussian streams, the product of the number of streams with the relative buffer size is a constant. ■

Proof: To prove this result it is useful to observe that Theorem 5.5 predicts an asymptotically linear regime for the loss as the number of sources increases. Therefore, for large n

$$\log(P_{loss}) \approx -NI(c,b)$$

Consider two multiplexers that are loaded such that

$$\log(P^1_{loss}) = log(P^2_{loss})$$

For such a case:

$$N_1 I_1(c,b) = N_2 I_2(c,b)$$

The expression for $I_x(c,b)$ can be obtained analytically as:

$$I(c,b) = -2\frac{b(c-\mu)}{\sigma^2}$$

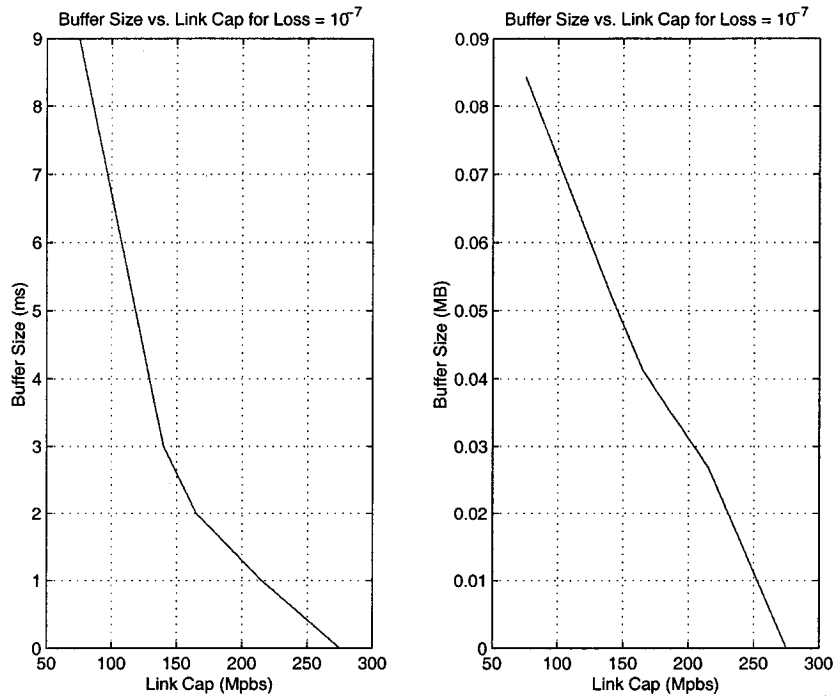

Figure 5.3: Buffer Length versus Capacity for a Fixed Loss

If we express the length of the buffer relative to the link capacity, then $b = c\tau_{sec}$. Therefore:

$$I(c,b) = -2\frac{c^2\tau_{sec}(1-\rho)}{\sigma^2}$$

Henceforth: $N_1\tau_1 = N_2\tau_2$. Therefore, $N\tau_{sec} = constant$

■

A result similar to Lemma 5.1 can be obtained for aggregated streams obtained by multiplexing a large number of statistically identical substreams. Figure 5.3 shows the relationship between the buffer size and the link capacity, for a fixed loss and fixed link utilization. The absolute amount of buffer space (in MB) decreases linearly with respect to the link capacity, as expected. Since the maximum length of the buffer induces the maximum queuing delay as well as the delay jitter, this observation has important practical implications. We can conclude that multiplexing a large number of streams on a link *is an effective way of controlling* **delays** *in a network.* This shows that better delays can be obtained on a link of large capacity than on n links of $1/n$ the capacity,

and provides a justification for using the largest link capacities available rather then using other capacity enhancing techniques such as Wavelength Division Multiplexing (WDM).

5.3.3 Aggregate Traffic Approximation

To reduce the computational complexity of the traffic estimation system, traffic is evaluated as an aggregate at each of the P-nodes interfaces. Therefore, an approximation on the loss ratio obtained from a *traffic aggregate* that closely follows the loss curve is desired. As previously mentioned, the traffic at the P-nodes is considered to be an *aggregate* of a number of sources, and each of the VPN sources is itself an aggregate of the traffic generated by the users that populate that VPN. Hence, it is a safe assumption that the number of individual user sources is unknown and only information about the number of aggregates at the core is known. Since each of these aggregates can have traffic characteristics that are unique, the large number of sources asymptotic is not directly applicable. Therefore, an estimation of the loss from the aggregate traffic is sought.

It is known from the Bahadur-Rao Theorem [2] that the asymptotic tail distribution of the sum X^n of n identically distributed non-lattice random variables, when $n \to \infty$, can be calculated with the following theorem:

Theorem 5.6 [2] Given a set of iid random variable $X_t, t \in N$, with finite moment generating functions, and denoting by $\Pi(n) = \sum_{t=1}^{n} X_t$ the sum of n such variables, then, if $a > 0$:

$$\mathbb{P}(\Pi(n) > na) = \frac{e^{-nI(a)}}{\theta\sqrt{2pi\omega^2 n}}(1 + O(\frac{1}{n})) \tag{5.44}$$

Where the variables that appear in the Equation are defined as follows:

$$I(a) = a\theta - log(\mathbb{E}(e^{\theta X_t}))$$

θ is a solution to the equation:

$$\frac{\frac{\partial \mathbb{E}(e^{\theta X_t}}{\partial \theta}}{\mathbb{E}(e^{\theta T_t})} = a$$

Finally, ω is equal to:

$$\omega = \frac{\partial \mathbb{E}(e^{\theta X_t})}{\partial \theta} - a^2$$

■

The reader will certainly recognize the similarities with the conditions in Theorem 5.5. This is due to the fact that the derivation of Theorem 5.5 in [73] was also based on Theorem 5.6.

From the Bahadur-Rao theorem we can obtain an aggregate traffic approximation, by identifying $na = (Ct + B)$, where Ct is the maximum amount of traffic that the multiplexer can output over a period of length t, while B is the buffer size available at the multiplexer.

$$\mathbb{P}(A(t) > (Ct + B)) \approx \frac{e^{-I(C,B)}}{\theta\sqrt{2\pi\omega^2}} \tag{5.45}$$

Denoting, as before, the amount of traffic that is received in an interval of length t by $A(t)$, the values of $I(C, B), \theta, \omega$ are defined as follows:

$$I(C, B) = \theta(Ct + B) - log(\mathbb{E}[e^{\theta A(t)}]) \tag{5.46}$$

As before θ is the point at which Equation 5.47 is solved:

$$\frac{\frac{\partial \mathbb{E}[e^{\theta A(t)}]}{\partial \theta}}{\mathbb{E}[e^{\theta A(t)}]} = Ct + B \tag{5.47}$$

Finally, ω is calculated by using Equation 5.48:

$$\omega = \frac{\frac{\partial^2 \mathbb{E}[e^{\theta A(t)}]}{\partial \theta^2}}{\mathbb{E}[e^{\theta A(t)}]} - (Ct + B)^2 \tag{5.48}$$

We can now state the following theorem.

Theorem 5.7 Consider a FIFO multiplexer of Section 5.2.1 with capacity c and buffer b, fed by a stream that satisfies Equation 5.45. Assume that:

$$\exists! t_0 < \infty \quad s.t. \quad I_{t_0}(c, b) = \min_{t \in \mathbb{N}} I_t(c, b) > 0$$

and

$$\liminf_{t \to \infty} \frac{I_t(c, b)}{log(t)} > 0$$

Then the loss ratio of the multiplexer is equal to:

$$P_{loss} = \frac{e^{-I_{t_0}(c,b)}}{\theta^2 c\rho\sqrt{2\pi\omega^2}} \tag{5.49}$$

ω is defined by Equation 5.48, while θ is the solution to Equation 5.47. ■

Proof: Follows the same line of reasoning Lykhanov [73], with different initial hypothesis.

■

In Equation 5.49 we have a way in which we can approximate the loss ratio given an input process. We will use this approximation to calculate the loss at a multiplexer for the experiments presented in Section 6.3 Let us rewrite the equation in a logarithmic form. Applying the logarithm function to both sides of Equation 5.49 it results in the following:

$$log(P_{loss}) = -\inf_{t>0}\sup_{\theta>0} I_t(c,b) - log(\theta^2 c\rho\sqrt{2\pi\omega^2}) \tag{5.50}$$

The main difficulty in evaluating Equation 5.50 results from the unconstrained optimization problem of the first term of the right hand side. The issues related to solving this problem will be discussed in Section 6.1.

5.4 Large Deviations Analysis of Priority Queue Multiplexers

As the analysis of Section 5.2.2 shows, the high priority queue can be considered in isolation. All the results for the FIFO scheduler carry over to the analysis of this queue without any changes. Therefore, this section will contain the behavior analysis for the lower priority queues.

5.4.1 B-Asymptote Results

In this section, we consider a PQ system with buffers scaled to infinity. Due to their linear dependencies, PQ systems can be analyzed by using mathematical induction. To start the process, let us consider a system with two queues. Therefore, all queue dynamic Equations presented in Section 5.2.2 are particularized for $j = [1, 2]$. For such a system, for the lower priority queue, the following can be proven:

Theorem 5.8 [14] Consider a priority queue system that receives traffic on both inputs (to avoid trivialities). Let us define the *effective bandwidths* of each of the sources as:

$$\alpha_1(\theta, \infty) = \lim_{t\to\infty} \frac{1}{\theta t} log(\mathbb{E}e^{\theta A_1(t)})$$

$$\alpha_2(\theta,\infty) = \lim_{t\to\infty} \frac{1}{\theta t} log(\mathbb{E}e^{\theta A_2(t)})$$

Furthermore, consider that the ingress traffic satisfies the following assumptions:

1. The two inputs are independent.
2. The arrival processes $A_j(t)$ are stationary and ergodic.
3. Each of the arrival processes $A_1(t)$ and $A_2(t)$ are bounded, non-negative.
4. $\alpha_1(\theta,\infty)$ and $\alpha_2(\theta,\infty)$ exist $\forall\theta \in \mathbb{R}^+$.
5. $\theta\alpha_1(\theta,\infty)$ and $\theta\alpha_2(\theta,\infty)$ are differentiable $\forall\theta \in \mathbb{R}^+$.
6. Each of the arrival processes $A_1(t)$ and $A_2(t)$ are adapted to filtrations $\mathcal{F}_{j,t}$, such that for all $\theta \in \mathbb{R}$, $s,t \geq 0$, there is a positive valued function $\Gamma_j(\theta)$ that satisfies the following:

$$\theta t\alpha_j(\theta,\infty) - \Gamma_j(\theta) \leq log(\mathbb{E}(e^{\theta(A_j(t+s)-A_j(t))} \mid \mathcal{F}_{j,s})) \leq \theta t\alpha_j(\theta,\infty) + \Gamma_j(\theta) \quad a.s. \tag{5.51}$$

Furthermore, let us define the function $f : \mathbb{R}^+ \to \mathbb{R}$ as:

$$f(\theta) = \theta(c - \alpha_1(\theta,\infty))$$

Denoting by θ_1 the solution to the equation, if it exists ($\theta_1 = \infty$ otherwise):

$$\frac{\partial f(\theta)}{\partial\theta} = 0$$

Then, the process $\mathbf{q}_t$ is ergodic and satisfies the following relation:

$$\lim_{x\to\infty} \frac{1}{b} log\mathbb{P}(q_\infty \geq b) = -\theta_2^* \tag{5.52}$$

Where θ_2^* is the solution to the following equation:

$$\theta\alpha_2(\theta,\infty) - f(\min(\theta,\theta_1)) = 0$$

■

The above theorem can be used to obtain the tail probabilities in systems with $n > 2$ queues. Such a system can be solved iteratively, by solving all queues up to $j < n$ and then considering a new system in which all the traffic from queues 1 to j is added together and considered to have priority 1, while queue $j+1$ has priority 2. For this system Theorem 5.8 can be applied. Note that as in the FIFO case, the theorem assumptions do not hold for long-range dependent traffic.

Theorem 5.8 shows that the same type of linear approximation between tail probability and the length of the buffer that has been used in the FIFO case can be used for all queues composing a PQ system. Therefore if Equation 5.52 holds, the tail probability for *any* buffer length is approximated by the following expression:

$$\exists b_0 \in \mathbb{R}^+, \exists \zeta \in \mathbb{R}^+ \quad s.t. \quad \forall b > b_0 \quad \log(\mathbb{P}q_\infty > b)) = -\theta^* b - \zeta$$

Therefore, the tail probability is exponential and can be described by:

$$P(q_\infty > b) = e^{-\theta b - \zeta} \tag{5.53}$$

Under the linear approximation described above, for all of the queues comprising a PQ system the loss ratios can be computed as follows:

Theorem 5.9 Under the assumptions of Theorem 5.8 and using the approximation for the tail loss probability of Equation 5.53, the loss ratio of queue j at a PQ multiplexer, denoted by P^j_{loss} is:

$$P_{loss,j} = \frac{e^{-\theta_j b_j - \zeta_j}}{c\rho_j\theta_j} \tag{5.54}$$

In Equation 5.54 θ^j is calculated as in Theorem 5.8, for $j > 1$ or as in Theorem 5.3 for $j = 1$. ρ^j is defined as:

$$\rho_j := \frac{\mathbb{E}[X_{j,t}]}{c}$$

for each of the queues, while ζ_j is a constant, different for each queue. ■

Proof:

The proof follows the lines of the proof in Theorem 5.4.

Assuming that the queue at $t = 0$ does not contain any traffic, then under the same realization of the input process, the length of the queue in an infinite buffer system is less than the length of the queue in any finite buffer system.

Denoting by q^f_t the length of the queue in the infinite buffer system and by q^i_t the length of the queue in the finite buffer system, then:

$$\forall t \in \mathbb{N} \quad q^f_t \leq q^i_t \quad a.s.$$

Similar to the argument of Theorem 5.4 we approximate $\mathbb{E}[l_{j,t}]$ by:

$$\mathbb{E}[l_{j,t}] \approx \mathbb{E}[max(0, \mathbf{q}^j_t - B_j)]$$

The right hand side of the above expression can be readily obtained as follows:

$$\mathbb{E}[max(0, \mathbf{q}_t^j - B_j)] = \int_{B_j}^{\infty} \mathbb{P}(\mathbf{q}_t^j > x)\mathrm{d}x$$

Using the linear approximation of Equation 5.36 the integral can be calculated and the result is:

$$\int_{B}^{\infty} \mathbb{P}(\mathbf{q}_t^j > x)\mathrm{d}x = \frac{e^{-\theta_j B_j - \zeta_j}}{\theta_j}$$

By plugging the result in Equation 5.5 the desired result ensues.

■

5.4.2 Aggregate Traffic Approximations

In this subsection we will introduce aggregate traffic approximations for PQ systems. Since the high traffic does not have any dependencies, all of the results in Section 5.3.3 apply.

For the lower priority queues we apply the same work strategy as in the previous subsection: the system of queues will be solved iteratively, starting with the queue number 2 and finishing with the lowest priority queue.

Since the amount of traffic that the lower priority queue can send depends on the amount of *spare capacity* that is available from the higher priority queue, it is important to characterize the output process of the high priority queue.

Such a characterization can be obtained for a system with a large number of statistically identical sources. The systems we are studying are also composed of large number of multiplexed sources, although heterogeneous in nature. Nevertheless, the same result is conjectured to be true for this class of systems too, and the validity of this hypothesis is verified by simulations. The result regarding the output of a FIFO multiplexer that is fed with a large number of identical sources can be stated as follows:

Theorem 5.10 In an N-Asymptote system fed by n identical stochastic processes with moment generating functions

$$\Lambda(\theta) = \log(\mathbb{E}[e^{\theta A(t)}])$$

the moment generating function of the output process Λ_n^* is

$$\Lambda_n^*(\theta) = \Lambda_n(\theta)$$

■

Proof: Section 5.3.2 shows that the overflow probability in such a system is:

$$P(q > b) \approx e^{-nI}$$

Let us define the event $\{A^*(t) \neq A(t)\}$ of the output process being different from the input process over an interval of length t. The probability of this event is equal to the probability of overflow. Hence:

$$P(A^*(t) \neq A(t)) \approx e^{-nI}$$

Therefore, as $n \to \infty$ the above value tends asymptotically to zero. Hence, the input and the output processes of an N-asymptote system are equal (a.s.). Therefore, the moment generating functions of these two processes are equal, almost surely.

■

What Theorem 5.10 does is to allow us to quantify the amount of work that is available at the multiplexer for the lower priority queue.

The amount of work at the lower priority queue in a multiplexer with infinite queue is then:

$$\mathbb{P}_{\not\prec}(A_2(t) > (Ct - \alpha_1(\theta_1, t_1^*) + B_2)) \approx \frac{e^{-I(C,B_2)}}{\theta\sqrt{2\pi\omega_2^2}} \tag{5.55}$$

$$\begin{aligned} I_{t,2}(C,B) &= \theta((Ct + B_2) - log(\mathbb{E}[e^{\theta(A_2(t)+\alpha_1(\theta_1,t_1^*))}]) && (5.56)\\ &= \theta(((Ct - \alpha_1(\theta_1, t_1^*)) + B_2) - log(\mathbb{E}[e^{\theta A_2(t)}]) && (5.57)\\ &\approx \theta(((Ct - \alpha_1(\theta, t)) + B_2) - log(\mathbb{E}[e^{\theta A_2(t)}]) && (5.58)\\ &= \theta((Ct + B_2) - log(\mathbb{E}[e^{\theta(A_2(t)+A_1(t)}]) && (5.59) \end{aligned}$$

$$\frac{\frac{\partial \mathbb{E}[e^{\theta(A_1(t)+A_2(t))}]}{\partial\theta}}{\mathbb{E}[e^{\theta(A_1(t)+A_2(t))}]} = Ct + B_2 \tag{5.60}$$

Finally, ω is calculated by using Equation 5.61:

$$\omega = \frac{\frac{\partial^2 \mathbb{E}[e^{\theta(A_1(t)+A_2(t))}]}{\partial\theta^2}}{\mathbb{E}[e^{\theta(A_1(t)+A_2(t))}]} - (Ct + B_2)^2 \tag{5.61}$$

Theorem 5.11 Consider a PQ multiplexer of Section 5.2.2 with capacity C. Furthermore assume that the priority 2 queue has a buffer of length B_2 and it is fed by a stream that satisfies Equation 5.55. Assume that:

$$\exists! t_0 < \infty \quad s.t. \quad I_{t_0,2}(C, B_2) = \min_{t \in \mathbb{N}} I_{t,2}(C, B_2) > 0$$

and

$$\liminf_{t \to \infty} \frac{I_t(C, B_2)}{log(t)} > 0$$

Then the loss ratio of the multiplexer is equal to:

$$P_{loss,2} = \frac{e^{-I_{t_0,2}(C,B_2)}}{\theta^2 c \rho_2 \sqrt{2\pi\omega^2}} \tag{5.62}$$

ω is defined by Equation 5.61, while θ is the solution to Equation 5.60. ■

Proof: Assuming that at $t = 0$ the queue of priority j does not contain any traffic, then under the same realization of the input process, the length of the queue in an infinite buffer system is less than the length of the queue in any finite buffer system.

Denoting by $q_t^{f,j}$ the length of the priority queue j in the infinite buffer system and by $q_t^{i,j}$ the length of the priority queue j in the finite buffer system, then:

$$\forall t \in \mathbb{N} \quad q_t^{f,j} \leq q_t^{i,j} \quad a.s.$$

By comparing Equation 5.15 and Equation 5.16 we approximate expectation of the priority j loss $\mathbb{E}[l_t^j]$ as:

$$\mathbb{E}[l_t^j] \approx \mathbb{E}[max(0, \mathbf{q}_t^j - B_j)]$$

The right hand side of the above expression can be readily obtained as follows:

$$\mathbb{E}[max(0, \mathbf{q}_t^j - B_j)] = \int_{B_j}^{\infty} \mathbb{P}(\mathbf{q}_t^j > x) \mathbf{d}x$$

Using the expression of loss in an infinite buffer at priority j given by Equation 5.55 and integrating in parts the desired result ensues.

■

Let us rewrite Equation 5.62 in a logarithmic form. Applying the logarithm function to both its sides results in the following:

$$log(P_{loss,2}) = -\inf_{t>0} \sup_{\theta>0} I_{t,2}(C, B_2) - log(\theta^2 c \rho \sqrt{2\pi\omega^2}) \tag{5.63}$$

All queues of lower priority can be solved iteratively, by replacing $A_1(t)$ with:

$$A(t) = \sum_{k=1}^{j-1} A_k(t)$$

Therefore the loss of any priority less or equal to 2 can be calculated using Theorem 5.11.

5.5 Summary

Chapter 3 presents a service model that uses loss as the main QoS class differentiator. To be usable in the context of service admission control, such a parameter has to be linked with the amount of traffic that enters the scheduling device. In this chapter, the problem is solved by extending results from the Large Deviation Theory.

As stated in Chapter 2 our service platform considers only work conserving service disciplines. Most of the schedulers of this type, that are implemented in current NEs, fall into one of the following three categories: FIFO, priority queuing and GPS. In the first part of the chapter it is proven that there exists a GPS scheduler weight assignment such that the resulting scheduling system *approximates* the PQ system behavior. Because of this result, the problem of loss estimation is treated only for the FIFO and the PQ queuing disciplined. All schedulers considered in this chapter have finite buffers, and as such the loss ratios are calculated for them.

The analysis of the FIFO scheduler presents two approximations that link the arrival process to the loss ratio: one based on the large buffer asymptotic and one based on the aggregate traffic approximation. The next chapter will check the validity of these approximations using trace-based simulations. The analysis of the PQ scheduler follows a pattern similar to the FIFO case. Simulations that verify the applicability of results obtained for the PQ queuing discipline to measurement based loss estimation are presented in Chapter 6.

Chapter 6

Loss Estimation from Traffic Measurements

The results of the previous chapter assumed that the traffic can be modeled as a stochastic process and linked the loss at the multiplexer to the moment generating function of this process. This chapter deals with the problem of *estimating* the multiplexer loss if the traffic can be measured. The crux of the problem, for the results presented in Chapter 5 lays in the estimation of the moment generating function. The estimation of this measure is achievable in two different ways: assume a model for the arrival process and identify the parameters, and then use this model to calculate the moment generating function. The second option estimates the moment generating function directly without a model for the process. This chapter compares two methods for estimating this measure.

The venue for comparison is simulation; using traces obtained from traffic measurements and from digitized video sequences. The results of the comparisons are presented in detail in the subsequent sections.

6.1 Loss Estimation For FIFO Multiplexers

All the large deviation results presented in the sections above are pertinent *if* the moment generating function of the arrival process $A(t)$ is known. If a realization of such a process is obtained from traffic measurements, the determination of its moment generating function can be done via two routes (Figure 6.1):

- Identify the process type and the process parameters for the arrival process. For

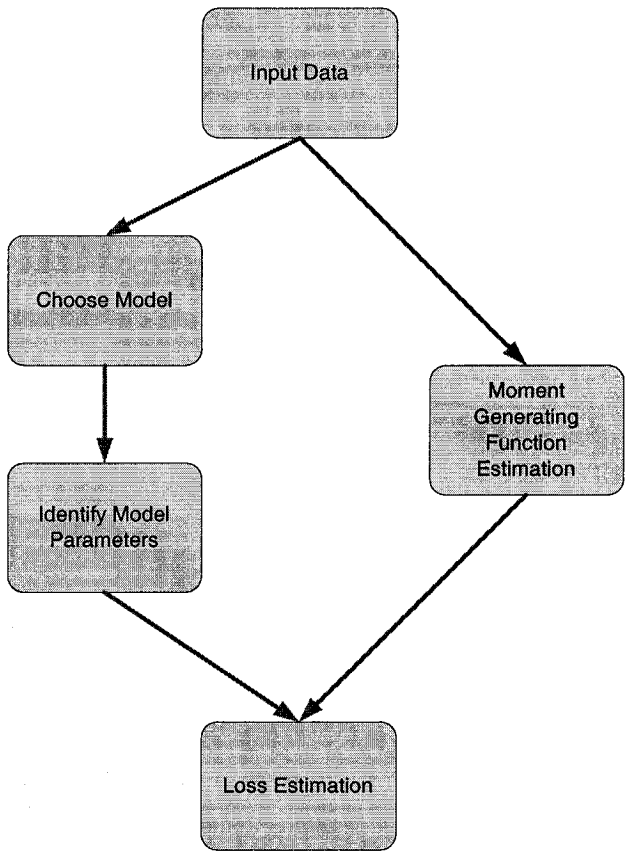

Figure 6.1: Loss Estimation from Measured Traffic Data

this method to be useful, the moment generating function has to be analytically solvable for the model chosen for the process. When this condition is satisfied, then the model parameters have to be estimated, hopefully through a recursive method.

- Estimate the moment generating function directly. This method has been introduced by Duffield et all. [33], though the idea is credited to Amir Dembo. Such a method bypasses the process modeling and represents a fundamentally different loss estimation method.

The second option has received some attention both in the context of *b-asymptotes* [27] and in the context of n-asymptotes [94]. The method has become so engrained that papers which report trace based results for these types of asymptotes (for example

[65, 63] use it by default. When applying any loss estimation method to measurement-based systems it is desired that the method will have the following characteristics:

- has accurate predictions within the system accepted tolerances.
- it is not very demanding computationally.

Therefore, since the same result can be obtained from the same probabilistic underpinnings through two different paths, it is important to compare the two and choose between them.

6.1.1 Estimation of the Moment Generating Function

To introduce the direct moment estimator method, let us consider a stochastic process with iid increments $\mathbf{X}_k$. In this restricted case, from the definition of effective bandwidth in Equation 5.27 it results that:

$$\begin{aligned} eb(\theta,t) &= \frac{1}{\theta t} log\mathbb{E}[e^{\theta A(t)}] = \frac{1}{\theta t} log\mathbb{E}[e^{\theta \sum_{k=0}^{f(t)} X_k}] \\ &= \frac{1}{\theta t} log\mathbb{E}[\prod_{k=0}^{f(t)} e^{\theta X_k}] = \frac{1}{\theta t} log \prod_{k=0}^{f(t)} \mathbb{E}[e^{\theta X_k}] \\ &= \frac{1}{\theta t} \sum_{k=0}^{f(t)} log\mathbb{E}[e^{\theta X_k}] = \frac{n}{\theta t} log\mathbb{E}[e^{\theta X}] \\ &= \frac{1}{\theta \frac{t}{n}} log\mathbb{E}[e^{\theta X}] = \frac{1}{\theta \tau} log\mathbb{E}[e^{\theta X}] \end{aligned}$$

One can arrive at identical results by postulating the independence of the *samples* as done in [33]. Here we make assumptions on the stochastic process first to stress the point of the equivalence between the two assumptions. Therefore, the effective bandwidth for iid variables is given by:

$$eb(\theta,\tau) = \frac{1}{\theta \tau} log\mathbb{E}[e^{\theta X}] \tag{6.1}$$

In Equation 6.1 τ represents the *width* of the temporal interval that contains the variable X.

Hence, in this case the moment generating function $\varphi(\theta) = \mathbb{E}[e^{\theta X}]$ of a random variable X is to use a sample average estimator defined as:

$$\hat{\varphi}(\theta) := \frac{1}{n}\sum_{k=0}^{n} e^{\theta \hat{X}_k} \tag{6.2}$$

The estimator presented in Equation 6.2 is unbiased, as it results from the following derivation:

$$\begin{aligned}\mathbb{E}\hat{\varphi}(\theta) &= \mathbb{E}[\frac{1}{n}\sum_{k=0}^{n} e^{\theta X_k}] = \frac{1}{n}\sum_{k=0}^{n}\mathbb{E}[e^{\theta X_k}] \\ &= \mathbb{E}[e^{\theta X_k}] = \varphi(\theta)\end{aligned}$$

Using Equation 6.2, the authors of [33] propose it to be used for the estimation of the effective bandwidth as detailed by Equation 6.3.

$$\hat{eb}(\theta,\tau) := \frac{1}{\theta\tau} log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta \hat{X}_k}) \tag{6.3}$$

The main issue that arises from the use of Equation 6.3 is that due to the concave nature of the *log* function, the resulting estimator is *biased*, as the following derivation shows.

$$\begin{aligned}\mathbb{E}[\hat{eb}(\theta,\tau)] &= \mathbb{E}[\frac{1}{\theta\tau} log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta X_k})] \\ &= \frac{1}{\theta\tau}\mathbb{E}[log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta X_k})] \\ &\leq \frac{1}{\theta\tau} log\mathbb{E}[(\frac{1}{n}\sum_{k=0}^{n} e^{\theta X_k})] \\ &= \frac{1}{\theta\tau} log\mathbb{E}[e^{\theta X_k}] \\ &= eb(\theta,\tau)\end{aligned}$$

Furthermore, the estimator is also asymptotically biased. This drawback cannot be corrected, as it results from the logarithmic nature of the measure. Other possibilities for estimation are possible. For example considering a sample of data denoted by a

vector $\hat{X} =< \hat{X}_1, \hat{X}_2, ..., \hat{X}_n >$. Instead of dividing the sample in *disjoint* blocks that are large enough to be independent as proposed in [33] one can create the sum:

$$\hat{A}_k = \sum_{j=0+k}^{n} \hat{X}_j$$

In other words, use a *rolling window* to accumulate the samples used in the estimation of the moment generating function. The estimation process has increased computational complexity and has not been considered at this time. Hence, it has been decided to use the independent block estimator with both the *b-asymptote* approximation from Equation 5.37 and with the aggregate estimator of Equation 5.49.

For the *b-asymptote* of Equation 5.37 we have to solve the effective bandwidth equation provided by Theorem 5.3. The estimator is used to determine the value of θ for the large deviations approximation by solving the following equation:

$$\theta c\tau = log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta \hat{X}_k}) \tag{6.4}$$

Equation 6.4 can be re-written as:

$$\sum_{k=0}^{n}[e^{\theta(\hat{X}_k - c\tau)} - 1] = 0 \tag{6.5}$$

Let us define a function g as the left hand side of Equation 6.5

$$g : \mathbb{R}^+ \rightarrow \mathbb{R}$$

Hence:

$$g(\theta) := \sum_{k=0}^{n}[e^{\theta(\hat{X}_k - c\tau)} - 1] \tag{6.6}$$

Function $g(\theta)$ (see Figure 6.2) can be readily shown to be continuous in θ over the range $\theta \in [0, \infty)$. Since $g(0) = 0$, Equation 6.5 has a trivial solution at $\theta = 0$. Since we are interested in a solution for $\theta > 0$, we would like to find out if such a solution exists. Taking the first derivative with respect to θ of the function $g(\theta)$ we obtain:

$$\frac{\partial g(\theta)}{\partial \theta} = \sum_{k=0}^{n}(\hat{X}_k - c\tau)[e^{\theta(\hat{X}_k - c\tau)}] \tag{6.7}$$

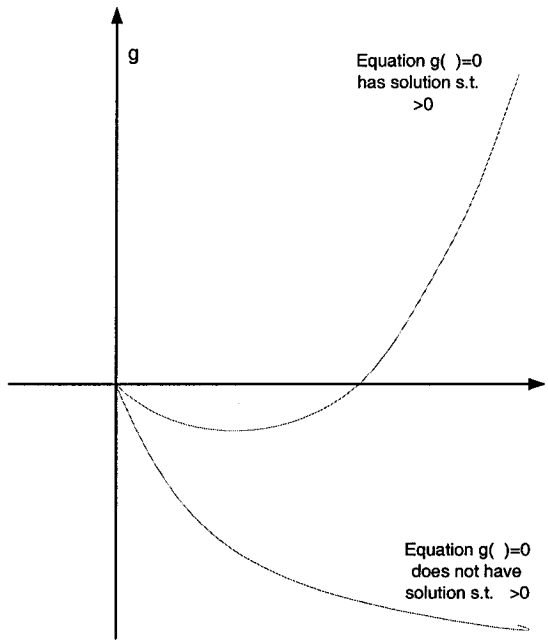

Figure 6.2: Shape and direction of the function $g(\theta)$

The value of the derivative in zero is $g'(0) < 0$ because of the stability condition[1]. The second derivative of $g(\theta)$ with respect to theta gives the following expression:

$$\frac{\partial^2 g(\theta)}{\partial \theta^2} = \sum_{k=0}^{n} (\hat{X}_k - c\tau)^2 [e^{\theta(\hat{X}_k - c\tau)}] \tag{6.8}$$

Since all the terms of the right hand side of 6.8 are positive, it results that function $g(\theta)$ is convex. Hence, if $g(\theta) \leq 0 \quad \forall \theta \in \mathbb{R}^+$, then Equation 6.5 does not have a solution such that $\theta \neq 0$. Furthermore, the function

Therefore, the following lemma holds:

Lemma 6.1 A necessary and sufficient condition for Equation 6.5 to have a solution[2] such that $\theta > 0$ is that:

$$\exists k \in K \quad s.t. \quad \hat{X}_k > c\tau$$

■

[1]The stability condition for any multiplexer states that the average amount of work received by the multiplexer is less that the link capacity

[2]Finding a solution to Equation 6.5 can be done in a very efficient way using the Newton-Ralphston algorithm

Proof:

Necessity. As above, if all $\hat{X}_k \leq c\tau$ then $g'(\theta) < 0, \quad \forall\theta > 0$.

Sufficiency. If there exists one $\hat{X}_k > c\tau$, then

$$e^{\theta(\hat{X}_k - c\tau)} > 1 \quad \forall\theta > 0$$

Applying Laplace's argument with respect to the largest exponent it results that for $\theta \to \infty$ the right hand side of Equation 6.5 goes to infinity. Since the first derivative is negative in the vicinity of $\theta = 0$ and since the equation has a trivial solution in zero it results that $\exists\theta > 0$ such that right hand side of Equation 6.5 is less than zero. Therefore, by the continuity of the equation there exists a $\theta > 0$ such that the equation is satisfied.

■

The same estimator defined in 6.3 is also used in the estimation of the moment generating function of the aggregate traffic approximation. Estimation of loss through this method is more computationally demanding because of the unconstrained optimization problem, in two dimensions, from Equation 5.50[3]. Thus, the resulting loss estimator for Equation 5.50 is:

$$log\hat{P}_{loss} := -\inf_{t\geq 0}\sup_{\theta>0}[\theta(Ct+B) - log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta\hat{X}_k})] - log(\beta) \tag{6.9}$$

In Equation 6.9, β is defined by the Expression 6.10, with ρ being the utilization of the line.

$$\beta := (\frac{\sum_{k=0}^{n}\hat{X}_k^2 e^{\theta\hat{X}_k}}{\sum_{k=0}^{n} e^{\theta\hat{X}_k}} - (Ct+B)^2)\theta^2 C\rho \tag{6.10}$$

Let us define a new function $g : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}$ as follows:

$$g(\theta, t) := [\theta(Ct+B) - log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta\hat{X}_k})] \tag{6.11}$$

The $\sup_{\theta>0}$ takes place when:

$$\frac{\partial g(\theta,t)}{\partial\theta} = 0$$

[3]The computational complexity is low enough to allow its usage for real-time loss estimation. To achieve this, the inner optimization is solved using Equation 5.47, as opposed to the delta computation approach of [94] which yields worse computational times

The partial derivative with respect to θ of $g(\theta, t)$ is:

$$\frac{\partial g(\theta,t)}{\partial \theta} = (Ct + B) - \frac{\sum_{k=0}^{n} \hat{X}_k e^{\theta \hat{X}_k}}{\sum_{k=0}^{n} e^{\theta \hat{X}_k}}$$

After some simple algebraic manipulations the problem of finding the value of θ becomes:

$$\sum_{k=0}^{n} (\hat{X}_k - (Ct + B)) e^{\theta \hat{X}_k} = 0 \tag{6.12}$$

Lemma 6.2 A necessary and sufficient condition for Equation 6.12 to have a solution such that $\theta > 0$ is that:

$$\exists k \in K \quad s.t. \quad \hat{X}_k > Ct + B$$

■

Proof:

Necessity. As above, if all $\hat{X}_k \leq Ct + B$ then the right hand side of Equation 6.12 is less than zero $\forall \theta > 0$.

Sufficiency. If there exist one $\hat{X}_k > Ct + B$, then

$$e^{\theta(\hat{X}_k - (Ct+B))} > 1 \quad \forall \theta > 0$$

Applying Laplace's argument with respect to the largest exponent it results that for $\theta \to \infty$ the right hand side of Equation 6.5 goes to infinity. Since the first derivative is negative in the vicinity of $\theta = 0$ and since the equation has a trivial solution in zero it results that $\exists \theta > 0$ such that right hand side of Equation 6.12 is less than zero. Therefore, by the continuity of the equation there exists a $\theta > 0$ such that the equation is satisfied.

■

Rewriting the expression $(Ct + B)$ to use the relative value for $B = C\tau$ results in $C(1 + \frac{\tau}{t})t$. The aggregate approximation will consider loss periods only if the rate of the arriving traffic is greater than $C(1 + \frac{\tau}{t})$. Therefore, the aggregate approximation might not give a result to the loss estimation *even if the multiplexer experiences loss.* A typical example can be seen in Figure 6.3. In the figure, the difference in length between the estimated loss and the observed loss is due to the situation described above. This limitation implies that the modeless aggregate estimator error is *unbounded.*

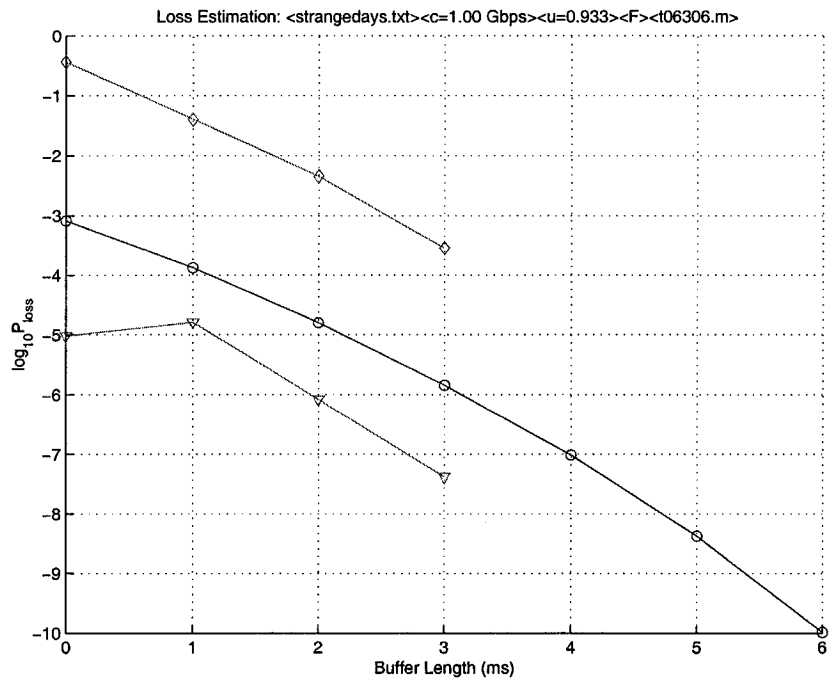

Figure 6.3: Loss curve and Loss Estimation using the Aggregate Traffic Estimator. The upper estimator curve shows the results without the logarithmic correction factor of Equation 5.50. The lower curve is calculated using both terms of Equation 5.50

6.1.2 Model Based Estimation of the Loss Curve

In this section we use the left hand side branch of Figure 6.1 as our estimation strategy. Because, as stated previously in this dissertation, in a network that supports mostly VPN services the traffic on the core links is highly aggregated and we assume that traffic can be modeled by a Gaussian representation. To be useful, model based estimation has to have two main characteristics:

- To be parsimonious. This characteristic is important the more complex the model the more computationally complex it is to identify it. Since the main issue with the modeless based estimation regards exactly this point, we want to use the simplest possible model that has the loss behavior of the traffic trace.

- To have analytical solutions to the loss expression. This point is especially important for the aggregate traffic approximation, where most of the computational complexity of estimating the loss is used in solving the infsup problem.

B-Asymptote Estimator

The Gaussian Traffic model, introduced in Section 5.3.1 satisfies both these properties. Let us first calculate the loss using such a model in the b-asymptote model. As in Section 5.3.1, the model will have a Hurst Parameter between $[0.5, 1)$ hence covering all the range of Gaussian stationary processes.

For the large buffer based estimator we can use only a value of H equal to 0.5 for the reasons detailed in Section 5.3.1. The traffic model is defined by the Equation 5.31. For this model the moment generating function has an analytical solutions, as given by Equation 5.32.

Replacing this in Equation 5.30 results in the following:

$$\mu + \frac{\theta\sigma^2}{2} = c$$

Solving in theta it results that:

$$\theta = \frac{2(c-\mu)}{\sigma^2}$$

Replacing θ in Equation 5.37 with the value calculated above, we obtain the following expression that links the loss to the main model parameters (μ and σ^2):

$$log(P_{loss}) = -\frac{2(c-\mu)B}{\sigma^2} - log(c\frac{2(c-\mu)}{\sigma^2}\rho)$$

If we use the relative buffer size relation $B = c\tau$ then above expression can be reduced to:

$$log(P_{loss}) = -\frac{2c^2(1-\rho)\tau}{\sigma^2} - log(c\frac{2c(1-\rho)}{\sigma^2}\rho) \tag{6.13}$$

Replacing ρ and σ^2 by their *estimated* values the following loss *estimator* is obtained in from the b-asymptote approximation:

$$log(\hat{P}_{loss}) = -\frac{2c^2(1-\hat{\rho})\tau}{\hat{\sigma}^2} - log(\frac{2c^2(1-\hat{\rho})}{\hat{\sigma}^2}\hat{\rho}) \tag{6.14}$$

Aggregate Traffic Based Estimator

For the general fractional Gaussian Noise model with $h \in [0.5, 1)$ the solution of the constrained optimization problem, with $B = \tau c > 0$, is obtained at:

$$t^* = \frac{h}{1-h}\frac{\tau}{1-\rho} \tag{6.15}$$

$$\theta^* = c\tau^{1-2h}\frac{(1-h)^{2h-1}(1-\rho)^{2h}}{h^{2h}\sigma^2} \tag{6.16}$$

Note that the extremizing point exists only if $B > 0$. This is not a limitation since all packet-switched NEs function on a store-and-forward principle and therefore have buffering.

The effect of h on t^* is to increase the timescale at which the traffic has to be measured. This has a beneficial effect as it decreases the number of data updates that have to be gathered from the NEs.

At this operating point, by replacing (θ, t) in Equation 5.50 the following expression for the loss ratio at a FIFO multiplexer fed by fractional Gaussian Noise is obtained:

$$log(P_{loss}) = -\frac{2c^2\tau^{2-2h}(1-\rho)^{2h}}{2\sigma^2}\frac{(h-1)^{2h-2}}{h^{2h}} - log(\sqrt{2\pi}\frac{c^3\tau^{2-3h}\rho}{\sigma^3}\frac{(1-\rho)^{3h}h^{-3h}}{(1-h)^{2-3h}}) \tag{6.17}$$

In Equation 6.17 we estimate the variance σ^2, the average μ of the process and h. The average is, of course, used to calculate the utilization ρ.

Therefore, by substituting the average and the variance of the process with their empirical counterparts, the following expression results for the estimation of the loss ratio. Denoting, as usual $B = \tau C$ and using the empirical average in the calculation of the empirical utilization $\hat{\rho}$, the following holds:

$$log(\hat{P}_{loss}) = -\frac{2c^2\tau^{2-2\hat{h}}(1-\hat{\rho})^{2h}}{2\hat{\sigma}^2}\frac{(\hat{h}-1)^{2\hat{h}-2}}{\hat{h}^{2\hat{h}}} - log(\sqrt{2\pi}\frac{c^3\tau^{2-3\hat{h}}\hat{\rho}}{\hat{\sigma}^3}\frac{(1-\hat{\rho})^{3\hat{h}}\hat{h}^{-3\hat{h}}}{(1-\hat{h})^{2-3\hat{h}}}) \tag{6.18}$$

For the case when $h = 0.5$, the time to overflow and the extremizing value of θ are:

$$t^* = \frac{\tau}{1-\rho} \tag{6.19}$$

$$\theta^* = 2\frac{c(1-\rho)}{\sigma^2} \tag{6.20}$$

The same algebraic process allows us to calculate the loss ratio given by Theorem 5.7 as follows:

$$\mathbb{P}_{loss} = \frac{e^{\frac{-2CB(1-\rho)}{\sigma^2}}}{\frac{2c^2(1-\rho)\rho}{\sigma^2}\sqrt{4\pi\frac{2CB(1-\rho)}{\sigma^2}}} \tag{6.21}$$

In Equation 6.21 σ^2 represents the variance of the process over the timescale defined by Equation 6.19

To estimate the loss using Equation 6.21 both the variance σ^2 and the average μ of the arrival process have to be estimated. The average is, of course, used to calculate the utilization ρ.

Therefore, by substituting the average and the variance of the process with their empirical counterparts, the following expression results for the estimation of the loss ratio. Denoting, as usual $B = \tau C$ and using the empirical average in the calculation of the empirical utilization $\hat{\rho}$, the following holds:

$$\log \hat{\mathbb{P}^*}_{loss} = -\frac{2C^2\tau(1-\hat{\rho})}{\hat{\sigma}^2} - \frac{1}{2}\log(4\pi\frac{2C^2\tau(1-\hat{\rho})}{\hat{\sigma}^2}) - \log(\frac{2C^2(1-\hat{\rho})\hat{\rho}}{\hat{\sigma}^2}) \tag{6.22}$$

It is dully noted that Equation 6.22 *cannot* be used to estimate the loss at a bufferless multiplexer. This is not a practical problem, since this case does not arise in store-and-forward NEs. For the bufferless multiplexer we used Equation 6.13 in simulations.

It is interesting to compare the expression of the loss estimator given by Equation 6.22 with the one given by Equation 6.13. It can be observed that the term outside the logarithm is identical in both cases. Therefore, the difference between the two results yields:

$$\hat{\Delta} = \log \hat{\mathbb{P}^*}_{loss} - \log \hat{\mathbb{P}^*}_{loss} = \frac{1}{2}\log(4\pi\frac{2C^2\tau(1-\hat{\rho})}{\hat{\sigma}^2})$$

Hence, the difference between the values estimated by the two estimators is dependent on both the characteristics of the multiplexer and the traffic conditions.

6.2 Loss Estimation of Priority Queuing Systems

Priority queues have dependencies that make the determination of loss for the individual queues more difficult than for a completely separable system. In this section we will consider the Priority Queuing Scheduling Discipline that was described in Section 5.2.2. Let us consider that the scheduler has n queues attached to it, with $n > 2$ in order to avoid trivialities, and that the link capacity is c. As stated in that section the highest priority queue has the same behavior as a FIFO system attached to a link of the same capacity c as the one for the PQ multiplexer. Therefore, all of the results from the above sections hold for this queue.

6.2.1 B-Asymptote Estimators

The lower priority queues have to be solved iteratively, because their behavior is influenced by the traffic that arrives as the higher priority queues. Theorem 5.8 contains the basis for an algorithm for determining the loss at these queues.

Proposition 6.1 Algorithm 5 has $\mathcal{O}(n)$ complexity, where n denotes the number of queues that have to be solved. ■

Proof: The algorithm executes the loss estimation loop once for each queue for priorities 2 to n. Hence the complexity in the number of queues is $\wr(n)$.

■

Solutions to steps 8, 17 and 26 of Algorithm 5 can be found in both the model based and modeless estimation.

Modeless Estimation

In the case of modeless estimation, using the estimator for the moment generating given by Equation 6.2 we can solve step 3 by using Theorem 5.8.

$$f(\theta) = \theta c - \frac{1}{\tau} log(\frac{1}{n}\sum_{k=0}^{n} e^{\theta \hat{X^1}_k}) \tag{6.23}$$

Deriving with respect to θ, and equating with zero we obtain:

$$c - \frac{1}{\tau} \frac{\frac{1}{n}\sum_{k=0}^{n} \hat{X^1}_k e^{\theta \hat{X^1}_k}}{\frac{1}{n}\sum_{k=0}^{n} e^{\theta \hat{X^1}_k}} = 0 \tag{6.24}$$

Algorithm 5 Loss Estimation For a PQ System under the B-Asymptote Assumptions

obtain θ_1 for priority 1 queue
if $\theta_1 < \infty$ **then**
  obtain θ_1^*
else
  $\theta_1^* = \infty$
end if
if $\theta_1^* < \infty$ **then**
  $\exists \theta_2 \quad s.t. \quad \alpha_2(\theta, \infty) - f(\theta) = 0$
  if $\theta_2 < theta_1^*$ **then**
    if $\exists$ lower priority queue **then**
      $A_1(t) = A_1(t) + A_2(t)$
      goto 1
    else
      stop
    end if
  else
    $\exists \theta_2 \quad s.t. \quad \alpha_2(\theta, \infty) - f(\theta_1^*) = 0$
    if $\exists$ lower priority queue **then**
      $A_1(t) = A_1(t) + A_2(t)$
      goto 1
    else
      stop
    end if
  end if
else
  $\exists \theta_2 \quad s.t. \quad \theta_2 \alpha_2(\theta_2, \infty) - f(\theta_2) = 0$
  if $\exists$ lower priority queue **then**
    $A_1(t) = A_1(t) + A_2(t)$
    goto 1
  else
    stop
  end if
end if

It is interesting to notice that the existence of a solution for Equation 6.24 is linked to the existence of a solution for $f(\theta) = 0$.

Lemma 6.3 There exist a $\theta > 0$ such that $f(\theta) = 0$ iff there exist a $\theta^* > 0$ such that

$$\frac{\partial f(\theta)}{\partial \theta} = 0$$

■

Proof: If $\exists \theta > 0$ such that $f(\theta) = 0$ then by Lemma 6.1:

$$\exists \hat{X}_k^1 \quad s.t. \quad \hat{X}_k^1 > Ct$$

Therefore,

$$\frac{1}{n}\sum_{k=0}^{n}(Ct - \hat{X}_k^1)e^{\theta \hat{X}_k^1} = 0$$

has a solution with $\theta > 0$.

■

What Lemma 6.3 shows is that we have to solve step 3 in Algorithm 5 only if the loss solution to the highest priority queue is not zero. This simplifies the algorithm in the cases when the high priority queue does not experience any loss.

Model Based Estimator

Considering a Gaussian traffic model the value of θ_1^* is:

$$\theta_1^* = \frac{C - \mu_1}{\sigma_1^2}$$

The value of θ_2 can be deduced from the following result:

$$\theta_2^* = \frac{C - (\mu_1 + \mu_2)}{(\sigma_1^2 + \sigma_2^2)}$$

If $\theta_1^* < \theta_2$ then the loss of the second priority queue is expressed as:

$$log(P_{loss,2}) = -\frac{2c^2(1 - \frac{\mu_1+\mu_2}{c})\tau_2}{(\sigma_1^2 + \sigma_2^2)} - log(c\frac{2c(1 - \frac{\mu_1+\mu_2}{c})}{(\sigma_1^2 + \sigma_2^2)}\frac{\mu_1 + \mu_2}{c}) \qquad (6.25)$$

6.2.2 Aggregate Traffic Estimators

Based on the results obtained in the FIFO case we will consider only the model-based estimation for the aggregate traffic estimator. Since the higher priority queue does not have any dependencies, it can be solved in isolation, similarly to the FIFO case. The loss at each queue can be estimated *iteratively* by using Algorithm 6.

Algorithm 6 Loss Estimation For a PQ System under the Aggregate Traffic Approximations

1: solve priority 1 queue as for the FIFO system
2: **for all** $k \in [2, n]$ **do**
3: solve priority k queue
4: $A_1(t) = A_1(t) + A_k(t)$
5: **end for**

As a model we consider the Gaussian traffic described by Equation 5.32. Similar to the FIFO queue, we will treat the general case first, $h \in [0.5, 1)$, and then particularize to the solution for $h = 0.5$. For the priority $j > 1$, $B_j = \tau_j c$, the optimizing point of (t^*, θ^*) is obtained at:

$$t_j^* = \frac{h}{1-h} \frac{\tau_j}{1 - \sum_{k=1}^{j} \rho_k} \tag{6.26}$$

$$\theta_j^* = c\tau_j^{1-2h} \frac{(1-h)^{2h-1}(1 - \sum_{k=1}^{j} \rho_k)^{2h}}{h^{2h} \sum_{k=1}^{j} \sigma_k^2} \tag{6.27}$$

Let us define the following notations:

$$\rho = \sum_{k=1}^{j} \rho_k \tag{6.28}$$

$$\sigma^2 = \sum_{k=1}^{j} \sigma_k^2 \tag{6.29}$$

At the operating point $t^*, \theta^*)$ the loss of attained at the priority j queue is computed as:

$$log(P_{loss,j}) = -\frac{2c^2\tau_j^{2-2h}(1-\rho)^{2h}}{2\sigma^2} \frac{(h-1)^{2h-2}}{h^{2h}} - log(\sqrt{2\pi} \frac{c^3\tau_j^{2-3h}\rho_j}{\sigma^2\sqrt{\sigma^2}} \frac{(1-\rho)^{3h}h^{-3h}}{(1-h)^{2-3h}}) \tag{6.30}$$

By substituting the average and the variance of the process with their empirical counterparts, the following expression results for the estimation of the loss ratio.

$$log(\hat{P}_{loss,j}) = -\frac{2c^2\tau_j^{2-2\hat{h}}(1-\hat{\rho})^{2h}}{2\hat{\sigma}^2}\frac{(\hat{h}-1)^{2\hat{h}-2}}{\hat{h}^{2\hat{h}}} - log(\sqrt{2\pi}\frac{c^3\tau_j^{2-3\hat{h}}\hat{\rho}_j}{\hat{\sigma}^2\sqrt{\hat{\sigma}^2}}\frac{(1-\hat{\rho})^{3\hat{h}}\hat{h}^{-3\hat{h}}}{(1-\hat{h})^{2-3\hat{h}}}) \tag{6.31}$$

For the case when $h = 0.5$, the time to overflow and the extremizing value of θ are:

$$t^* = \frac{\tau}{1-\sum_{k=1}^{j}\rho_j} \tag{6.32}$$

$$\theta^* = 2\frac{c(1-\sum_{k=1}^{j}\rho_k)}{\sum_{k=1}^{j}\sigma_k^2} \tag{6.33}$$

Simplifying the expression given by Equation 6.30

$$\mathbb{P}_{loss,j} = \frac{e^{\frac{-2CB_j(1-\rho)}{\sigma^2}}}{\frac{2c^2(1-\rho)\rho_j}{\sigma^2}\sqrt{4\pi\frac{2CB_j(1-\rho)}{\sigma^2}}} \tag{6.34}$$

Again, by substituting the average and the variance of the process with their empirical counterparts, we obtain our model based aggregate traffic estimator of the loss at priority j if the traffic at all of the higher priority queues is Guassian:

$$\log\hat{\mathbb{P}}^*_{loss,j} = -\frac{2C^2\tau_j(1-\hat{\rho})}{\hat{\sigma}^2} - \frac{1}{2}\log(4\pi\frac{2C^2\tau_j(1-\hat{\rho})}{\hat{\sigma}^2}) - \log(\frac{2C^2(1-\hat{\rho})\hat{\rho}_j}{\hat{\sigma}^2}) \tag{6.35}$$

6.3 Experimental Results for FIFO multiplexers

This section compares performance of three of the FIFO estimators introduced in the previous Sections 6.1 6.2.2. The values of loss predicted by the estimators are compared to the observed loss values in a series of trace base simulations. The traffic that has been injected in the multiplexer was of two types:

- **High Resolution MPEG2 traces.** The MPEG2 traces, described in detail in Appendix A, have been used to simulate different mixes over a number of line capacities.

- **Ethernet traffic.** An Ethernet traffic stream were mixed to simulate aggregation of traffic in networks that carries VPN services.

The available traffic traces have been combined in two ways in order to produce some variability in the traffic mixes:

- **Homogeneous Mixes.** In this series of simulations, for a given line rate and a given utilization, a single MPEG2 or Ethernet trace has been mixed such that the resulting mix has an average equal to the product of the line rate and the line utilization. Several experiments were conducted by varying the link capacity while keeping the utilization fixed. It should be noted that since only a integer number of streams can be mixed there is some variability in the utilization, between different line rates.

- **Heterogeneous Mixes.** Since a large number of traces were available, different mixes were generated by combining, at random, MPEG2 and Ethernet mixes. Again, the line rate and utilization were given, while the simulation program would generate a mix by randomly choosing from the available traces. Similarly to the homogeneous case, the actual utilization obtained for the simulations varies slightly from the chosen number due to the fact that only an integer number of flows can be added, each having predefined characteristics. The ratio between the average rates of each flow versus the line rate is very small. Therefore, a large number of flows have been mixed, and as such the difference between the chosen utilization and the actual utilization was small.

A number of experiments were performed for buffer lengths between $[0, 10]$ ms. The lower bound on the buffer length has been chosen such that it is larger than zero, for the reasons enunciated in Section 6.2.2. The differences between the estimated loss values and the observed loss values, denoted by ε_k^{ij} for each value k obtained for the estimator i at buffer length j, were recorded. The statistics of each of the sets of values defined as $\{\varepsilon_k^{ij} \mid k \in [0, n]\} \quad \forall i \in [1, 3] \quad \forall j \in [0, 10] ms$.

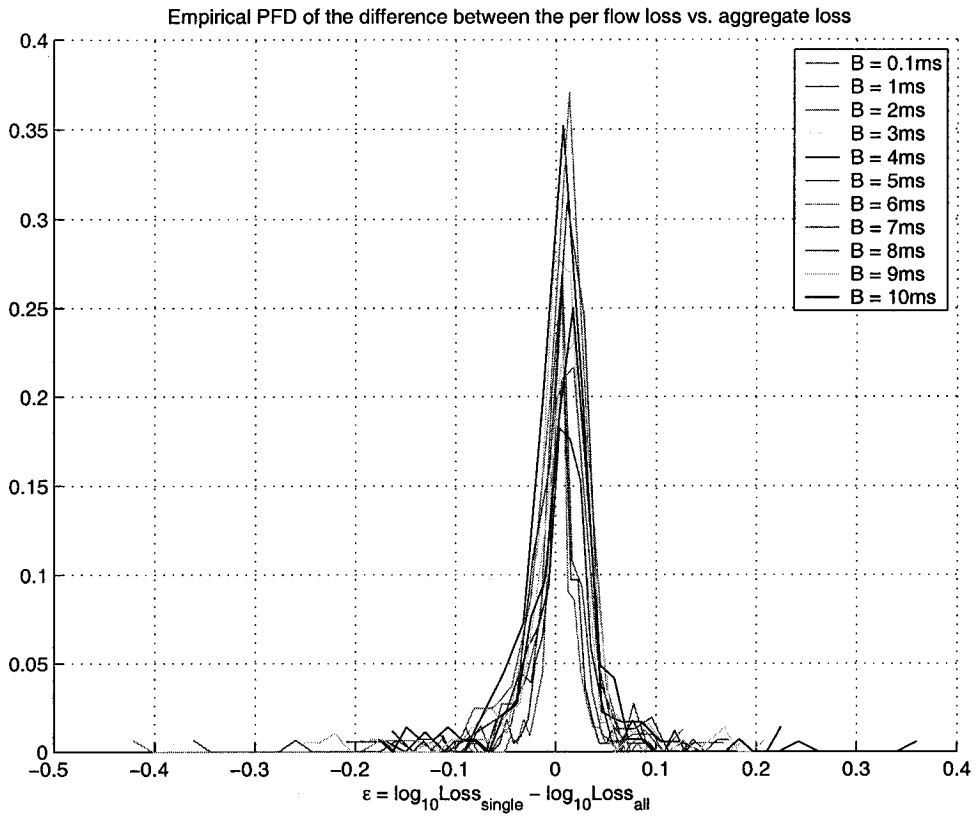

Figure 6.4: Individual flow loss vs. Aggregated Flow loss in highly aggregated traffic at FIFO multiplexers

6.3.1 Per Source Loss in Aggregated Traffic Networks

Service guarantees in our framework are on a per service basis. Since all the results in this section deal with the *aggregated loss* at a multiplexer, we investigate, by simulation, if Equation 5.10 holds or not. If we denote by $\log_{10} \hat{P}^s_{loss}$ the estimated loss probability of the single source, and by $\log_{10} \hat{P}^a_{loss}$ the estimated loss probability of the aggregate, we are interested in the following measure:

$$\hat{\varepsilon}^s := \log_{10} \hat{P}^s_{loss} - \log_{10} \hat{P}^a_{loss} \tag{6.36}$$

Figure 6.4 shows the empirical PDFs of the results obtained for $\hat{\varepsilon}^s$ during a large number of simulations. Each of the curves in the figure represents the EPDF for a different buffer size, relative to the link capacity.

The results show that Equation 5.10 holds with a high degree of accuracy. Therefore the loss ratio (P^{js} that a service instance s will experience for a class j along a path of n nodes, each node k experiencing a loss ratio P_k^j is:

$$P^{js} = \sum_{k=1}^{n} P_k^j \tag{6.37}$$

$$\approx n \max_{k\in[1,n]}(P_k^j) \tag{6.38}$$

Applying log_{10} on both sides of the above approximation we obtain:

$$log_{10}P^{js} \approx log_{10}(\max_{k\in[1,n]}(P_k^j)) + \log_{10}(n) \tag{6.39}$$

Therefore, the number of intermediate nodes between two service end-points affects the QoS of the service. Hence, since the service target QoS levels are established a-priori, the *maximum* number of nodes (N_{max}) that a service can pass through has to be specified. Thus, the target loss at a P-node given the target loss ratio for a service Γ^s becomes:

$$log_{10}P_k^j \leq \Gamma^s - \log_{10}(N_{max}) \tag{6.40}$$

6.3.2 Modeless Aggregate Traffic Estimator

The estimator based on Equation 5.49, called (AGE), has the highest computational requirements of all the investigated estimators. This is a side effect of the unconstrained optimization problem that is to be solved to obtain the estimation of loss.

Figure 6.5 details the empirical probability density functions for each buffer size from 0 to 10 ms, while Figure 6.6 displays the same information as the empirical cumulative distribution function.

The mean of the error and the standard deviation are summarized in Table 6.3.2. The results show that this estimator follows the shape of the loss curve well over the buffer range considered in this dissertation.

Nevertheless, the second column of Table 6.3.2 shows that the estimator does not have a constant bias. The mean error increases, almost linearly over the buffer of

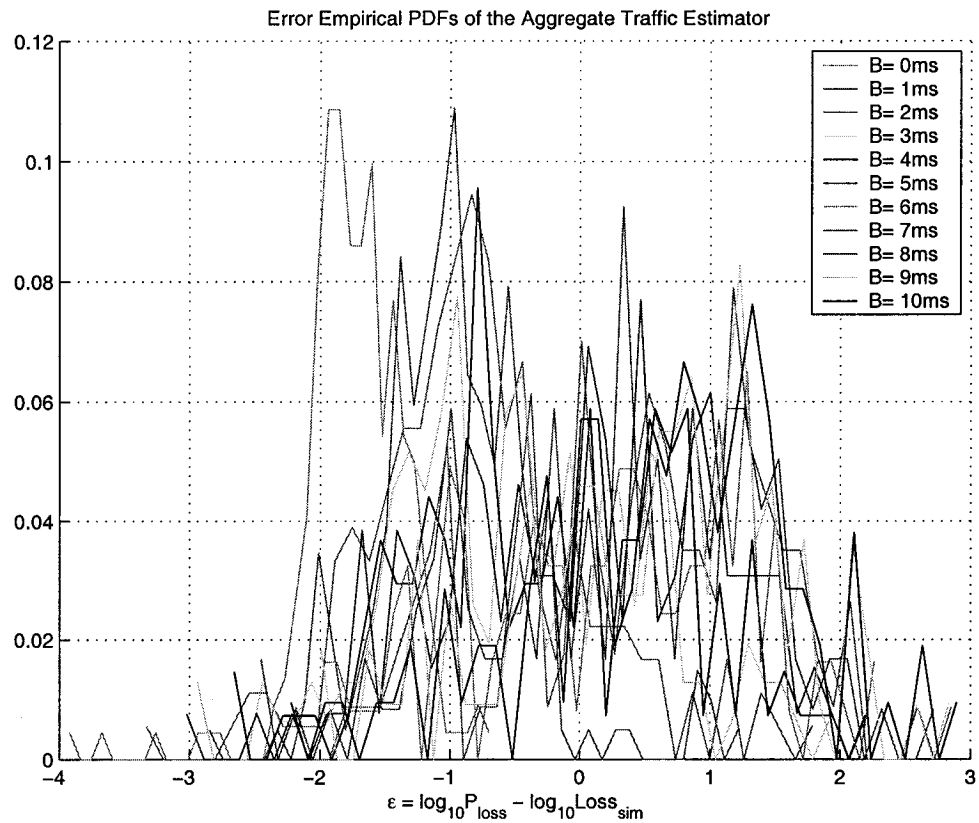

Figure 6.5: Empirical PDFs for the Aggregate Traffic Estimator at different buffer sizes

Buffer (ms)	Error Mean	Error Variance	Min Error	Max Error	Error Range
0	-1.730830	064244	-3.961580	-0.657772	3.303808
1	-0.956254	0.363724	-2.897530	1.268380	4.165910
2	-0.794566	0.701082	-3.395670	1.852070	5.247740
3	-0.463156	0.926804	-2.998810	1.977160	4.975970
4	-092640	1.044117	-2.719720	2.254290	4.974010
5	-0.005525	1.124308	-3.086850	2.272390	5.359240
6	089068	1.109809	-2.034760	2.315130	4.349890
7	0.273371	1.180146	-2.519350	2.784580	5.303930
8	0.401922	1.117145	-2.251090	2.919020	5.170110
9	0.519694	1.084787	-2.054820	2.896790	4.951610
10	0.589555	1.083661	-2.288140	2.960040	5.248180

Table 6.1: Aggregate Traffic Estimator (AGE) Statistics Synopsis

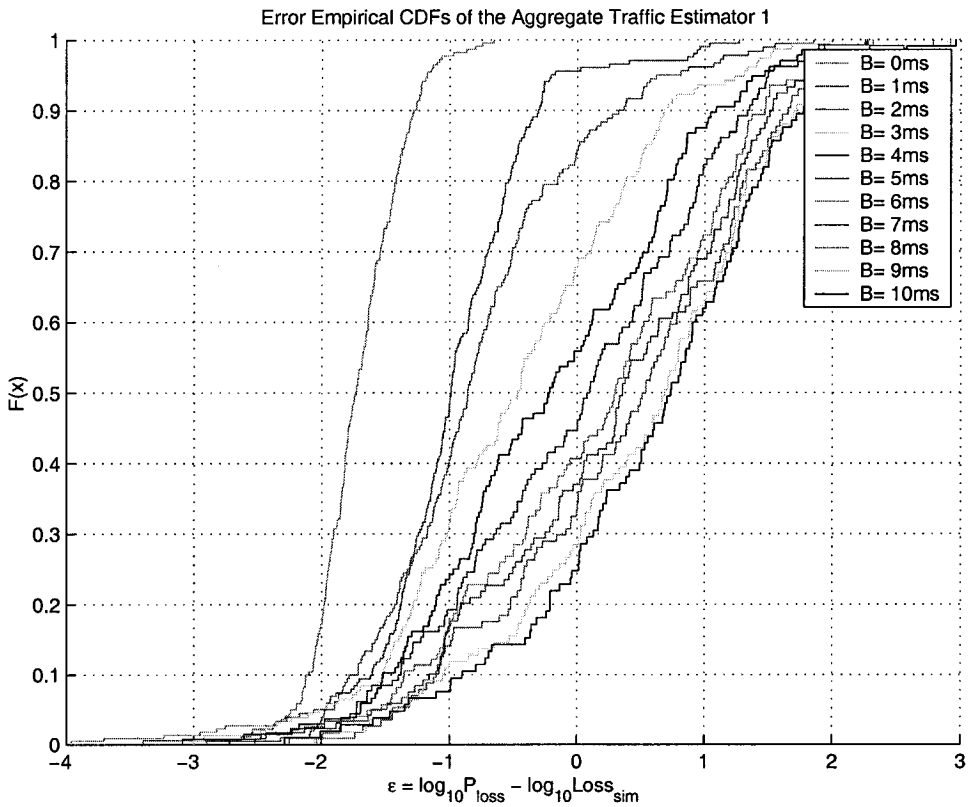

Figure 6.6: Empirical CDFs for the Aggregate Traffic Estimator at different buffer sizes

range $[0, 10]$ ms. The error range is rather large, with errors spanning over 5 orders of magnitude. Also, the error ranges are large for all buffer sizes. This means that the use of this estimator without run-time corrections can lead to large estimation errors and, depending on the sign of the error, to either large violations of the loss limit or to some network underutilization.

6.3.3 B-Asymptote based Estimator

The large buffer based loss estimator (LBE) is based on Equation 5.36. The estimator has average computational complexity, as compared to the other two proposals. In practice, a solution is usually found within a few iterations, resulting in a fast estimation that can be used for real-time estimation.

Figure 6.7 details the empirical probability density functions for each buffer size from 0 to 10 ms, while Figure 6.8 displays the same information as the empirical cumulative

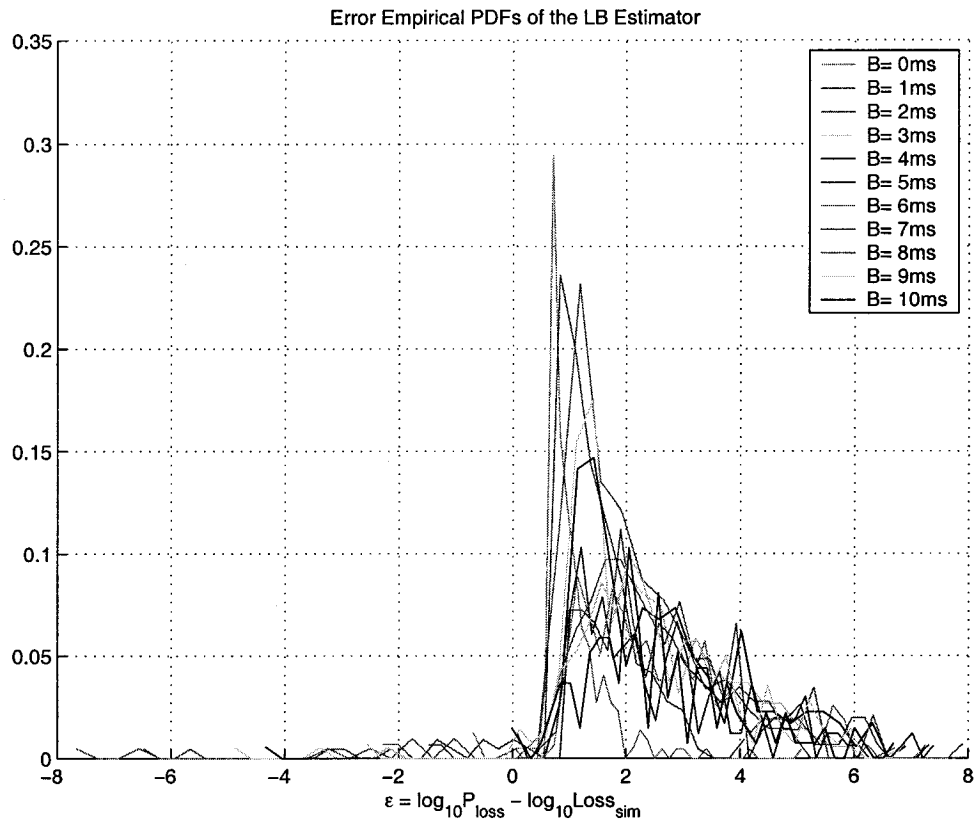

Figure 6.7: Empirical PDFs for the Large Buffer Estimator at different buffer sizes

distribution function.

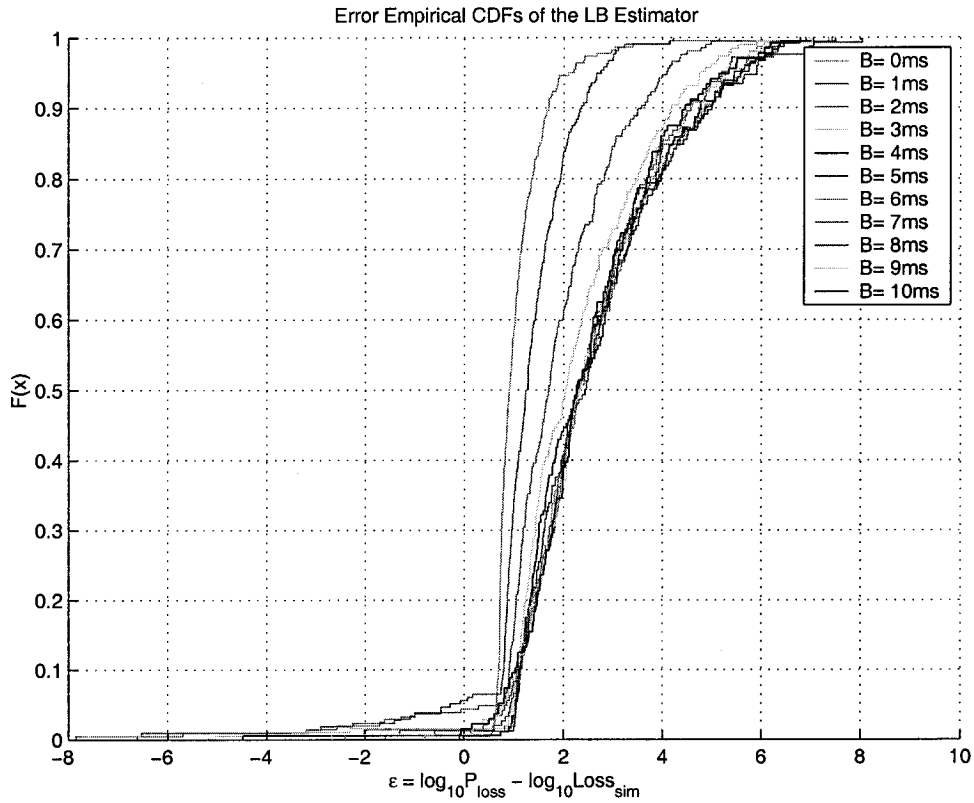

Figure 6.8: Empirical CDFs for the Large Buffer Estimator at different buffer sizes

Buffer (ms)	Error Mean	Error Variance	Min Error	Max Error	Error Range
0	1.087539	0.336202	0.000251	5.146730	5.146981
1	1.256143	1.316083	-6.609040	4.213960	10.823000
2	1.800431	2.310986	-7.847690	6.898990	14.746680
3	2.276573	2.115893	-5.031710	6.969060	12.000770
4	2.585060	2.345883	-4.478960	7.043590	11.522550
5	2.681347	2.216880	-0.009853	7.489950	7.499803
6	2.659552	2.072298	-0.791778	7.355700	8.147478
7	2.675745	2.283503	-1.594390	7.427530	9.021920
8	2.567470	2.202688	-2.408410	8.031820	10.440230
9	2.565073	2.009058	-2.674100	7.027800	9.701900
10	2.564880	1.847453	-0.098401	6.778140	6.876541

Table 6.2: LBE Estimator Statistics Synopsis

Table 6.3.3 summarizes the same statistics as for the AGE estimator. The comparisons of the estimated loss values with the results of simulations, show that this estimator does *not* follow the shape of the loss curve well, because in general, the loss curve is not linear. This estimator has a positive error bias. Note that a positive bias is more useful than the negative one. This is due to the asymmetry of consequences in the two cases: if the loss is overestimated then the overall result is that the system will operate at a lower utilization. The decrease in utilization is not, in general, dramatic.

On the other hand, the violation of the loss targets usually has more negative economic consequences because of either possible penalties that have to be paid or because end users might switch services altogether. This estimator has a very large error range. This is due to three factors:

- Estimation errors. Because in this case we estimate the moment generating function of the arrival process directly, there is a bias introduced by the method. Nevertheless, this is not very large.

- Operating zone. Since the LBE is based on *large* buffer asymptotics, it is possible for the process behavior at large buffers to be completely different from the behavior in the low buffer range. Hence, large estimation errors can occur.

- Traffic does not satisfy the hypothesis under which the estimator has been obtained. As detailed in Section 5.3.1 long-range dependent traffic falls in this category. Since the behavior of this traffic at the multiplexer is not quantified, applying the estimator to this case can generate large errors.

6.3.4 Model Based Aggregate Traffic Estimator

The *Model Based Aggregate Traffic Estimator (MBE)* is based on Equation 6.22. This is the simplest to compute of all three of the investigated estimators. As stated, the decreased computational complexity is the result of solving the infsup problem analytically and estimating only $\hat{\mu}$ and $\hat{\sigma^2}$.

The results of comparing the loss predicted by the estimator with the traffic loss observed during the simulations are aggregated in Figure 6.9. In that figure, the empirical

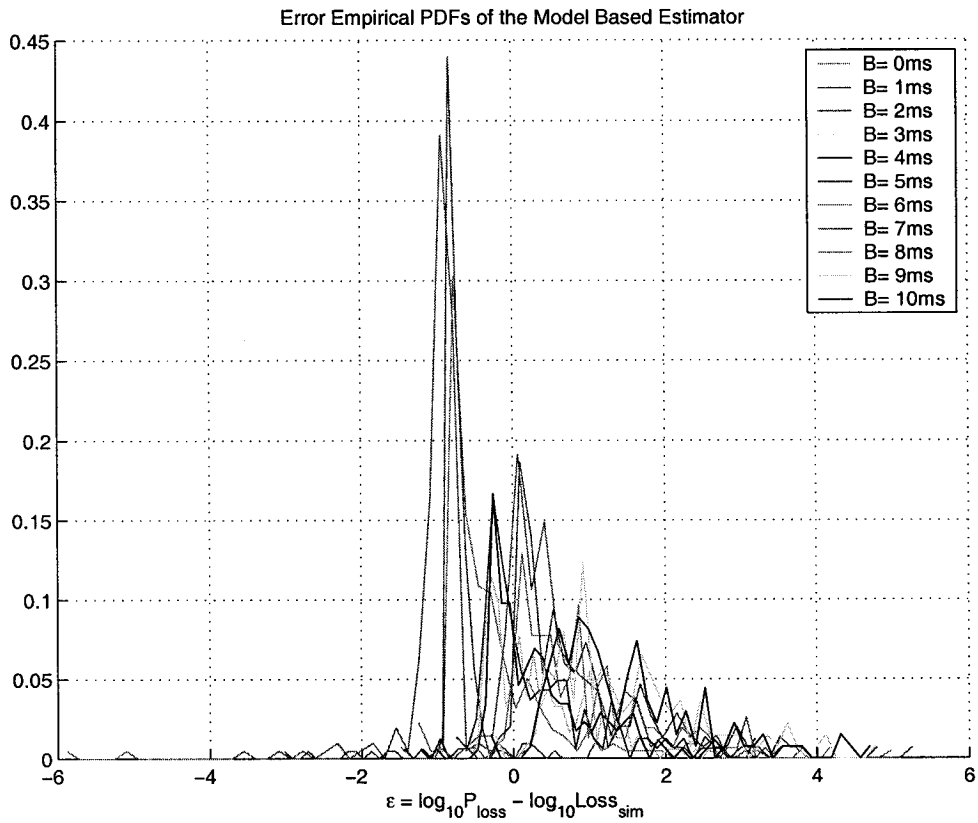

Figure 6.9: Empirical PDFs for Aggregate Traffic Estimator at different buffer sizes

probability density functions for each buffer size from 0 to 10 ms are displayed. Figure 6.10 displays the same information as the empirical cumulative distribution function. As for the other estimators, the main statistics are tabulated. Table 6.3.4 shows that, similar to the AGE estimator, the error bias is not constant. While the bias is negative for the small buffer multiplexer, it is positive for larger buffer sizes. The maximum error range is larger than the one for AGE, but significantly smaller than in the case of LBE.

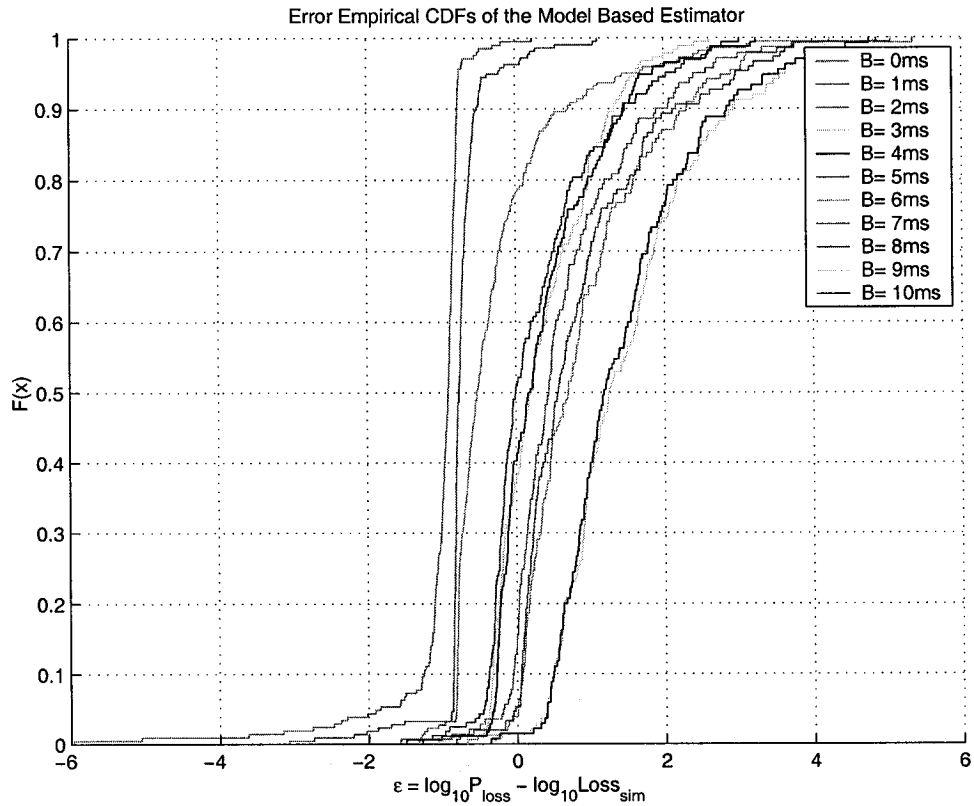

Figure 6.10: Empirical CDFs for the Aggregate Traffic Estimator at different buffer sizes

Buffer (ms)	Error Mean	Error Variance	Min Error	Max Error	Error Range
0	-0.245694	0.713577	-1.325820	5.055800	6.381620
1	-0.735097	047710	-3.075990	1.113320	4.189310
2	-1.039383	0.351360	-5.956730	0.232477	6.189207
3	0.363553	0.455989	-1.066410	2.613110	3.679520
4	0.391634	0.550551	-1.273920	3.017620	4.291540
5	0.301357	0.672910	-1.495990	3.240230	4.736220
6	0.917688	0.813414	-0.673140	4.233690	4.906830
7	0.856098	0.935722	-0.823099	4.861900	5.684999
8	0.680859	0.860602	-1.572670	5.349060	6.921730
9	1.520226	0.970492	-0.297447	4.823510	5.120957
10	1.467417	0.951053	-0.357951	4.761850	5.119801

Table 6.3: MBE Estimator Statistics Synopsis

6.3.5 Comparison of Estimators Performance

The FIFO estimators presented in the current chapter are compared using a set of performance measures. The estimators are compared using the following quantitative criteria:

- Error Mean. This criterion is the sample mean of each of the estimators at each of the buffer sizes.

$$\bar{\varepsilon^{ij}} := \frac{1}{n+1}\sum_{k=0}^{n}\hat{\varepsilon}_k^{ij}$$

- Error Variance. This metric is defined by the following equation:

$$\sigma^2(\varepsilon^{ij}) := \frac{1}{n}(\sum_{k=1}^{n}(\hat{\varepsilon}_k^{ij} - \bar{\varepsilon^{ij}})^2)$$

- Error Range. It is defined as the difference between the maximum value, from a given data set, and the minimum value from the same set.

$$\delta := max\{\varepsilon_k^{ij} \mid k \in [0,n]\} - min\{\varepsilon_k^{ij} \mid k \in [0,n]\}$$

- Mean Square Error. This is simply the average of the squared individual values from each data set.

$$\mathbb{M} := \frac{1}{n}(\sum_{k=1}^{n}(\hat{\varepsilon}_k^{ij})^2)$$

The above criteria are computed for all data sets obtained from experiments.

The PDF comparisons show that the distributions, for all buffer sizes, of the MBE estimator have light tails, while the other two estimators (AGE and LBE) do not. The LBE estimator has the worst overall behavior over all of the four criteria of comparison. Furthermore, it mostly has a positive bias, which means that in most of the simulated cases it overestimates the loss at the multiplexer. Since the actual loss value experienced at the link has to be lower than a preset value, the unidirectional bias is useful. The effect of overestimating loss is a decrease in the link utilization. The decrease is usually minute, because at high link capacities, due to the high degree of multiplexing, even utilization decreases of 1 percent lead to decreases in loss of one order of magnitude.

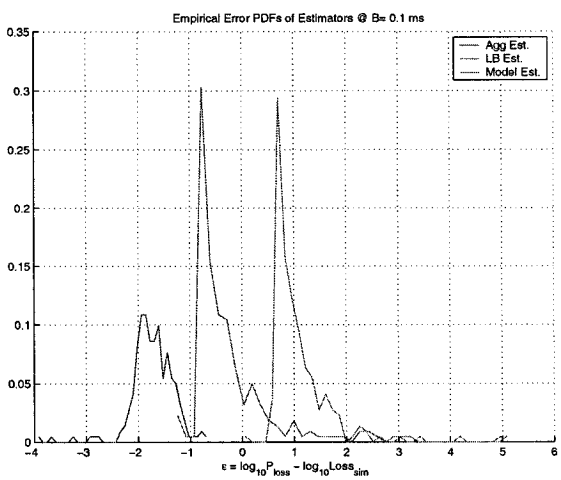

Figure 6.11: Empirical PDF Comparison of the FIFO Estimators at Buffer = 0 ms

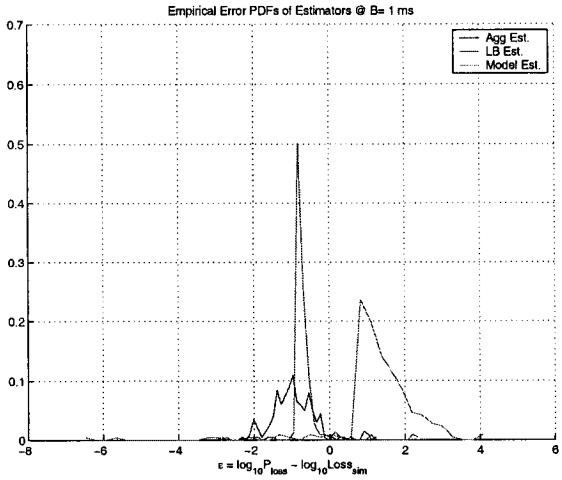

Figure 6.12: Empirical PDF Comparison of the FIFO Estimators at Buffer = 1 ms

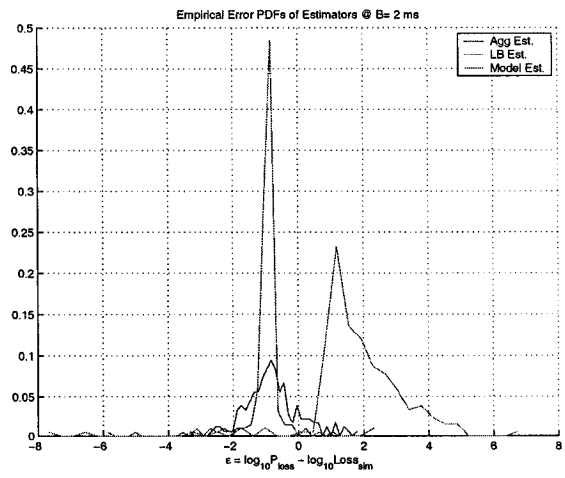

Figure 6.13: Empirical PDF Comparison of the FIFO Estimators at Buffer = 2 ms

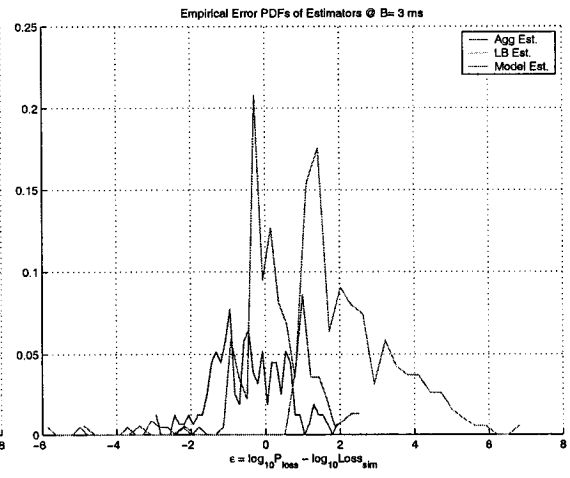

Figure 6.14: Empirical PDF Comparison of the FIFO Estimators at Buffer = 3 ms

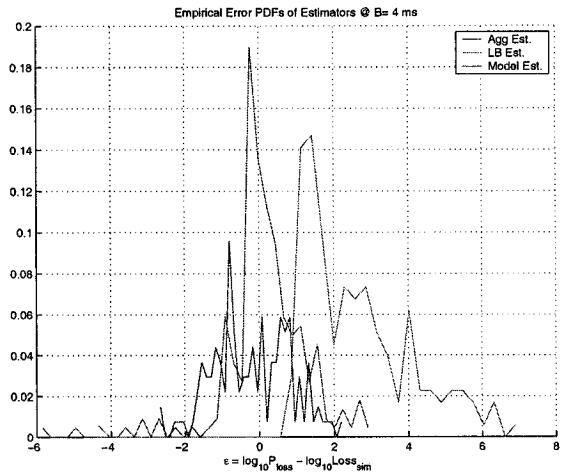

Figure 6.15: Empirical PDF Comparison of the FIFO Estimators at Buffer = 4 ms

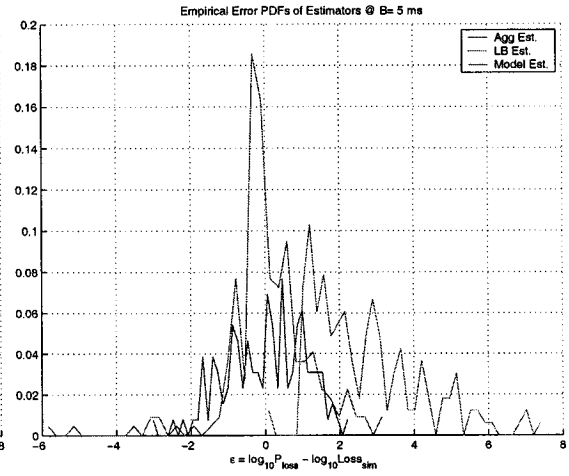

Figure 6.16: Empirical PDF Comparison of the FIFO Estimators at Buffer = 5 ms

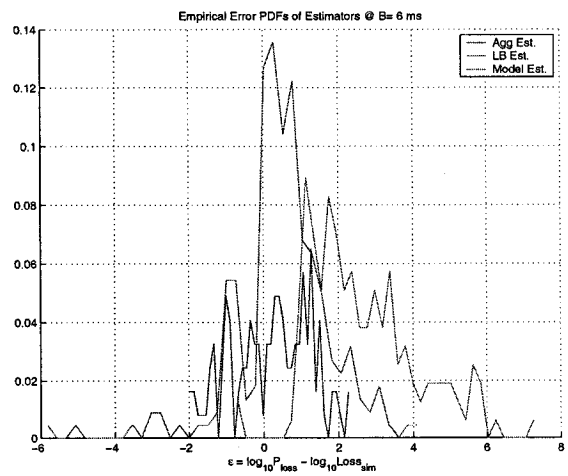

Figure 6.17: Empirical PDF Comparison of the FIFO Estimators at Buffer = 6 ms

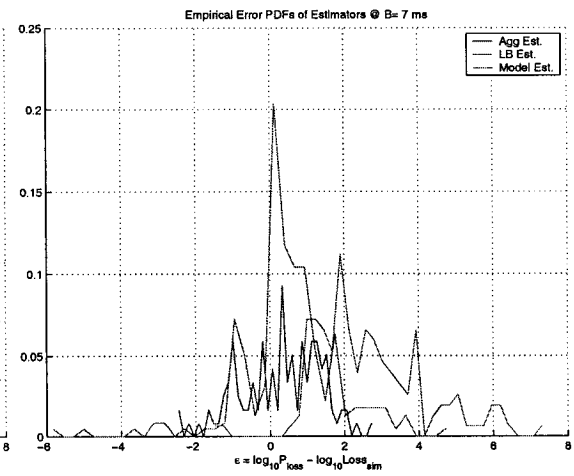

Figure 6.18: Empirical PDF Comparison of the FIFO Estimators at Buffer = 7 ms

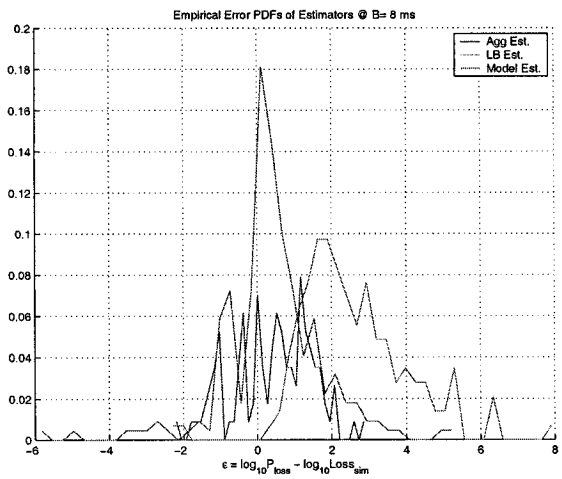

Figure 6.19: Empirical PDF Comparison of the FIFO Estimators at Buffer = 8 ms

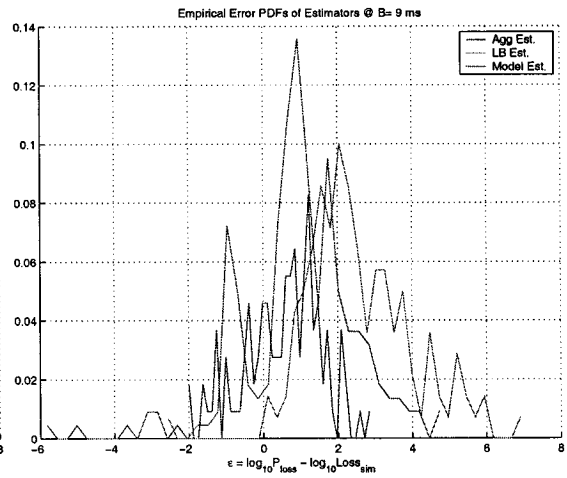

Figure 6.20: Empirical PDF Comparison of the FIFO Estimators at Buffer = 9 ms

Therefore, if the loss is estimated in open loop (the loss value is not adjusted with the actual observed data), then LBE is more useful than the AGE or MBE.

The AGE and the MBE estimators have similar performances, over the four criteria enumerated above. Both have non-constant biases, which increased almost linearly over the buffer range that was used in the experiments.

The main drawbacks of the AGE estimator are three-fold:

- it is not capable of estimating the loss in all situations. Hence, the usable range is smaller than the one for the LBE estimator, especially for larger buffer sizes.
- it is computationally intensive.
- when new data is received, the none of the old computations can be used, and the solutions search has equal or higher complexity[4].

[4]The computational complexity of the solution search is greater is all the data including the new sample is used to obtain a solution, or of equal complexity if a data window is utilized, and therefore the length of data vector is maintained constant.

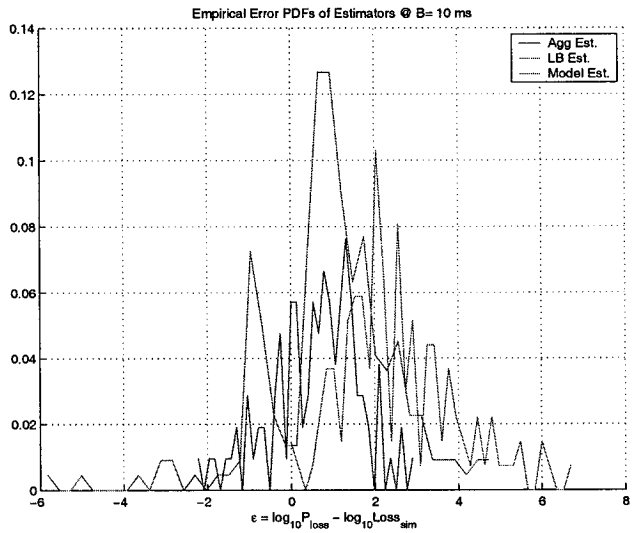

Figure 6.21: Empirical PDF Comparison of the FIFO Estimators at Buffer = 10 ms

Buffer Size (ms)	AGE Estimator	LBE Estimator	MBE Estimator
0	3.159273	1.517422	0.770714
1	1.276345	2.887885	0.742747
2	1.328522	5.541373	1.503381
3	1.135339	7.287422	1.156800
4	1.073550	9.015164	1.256631
5	1.115690	9.393064	1.304750
6	1.136533	9.132313	1.956748
7	1.244960	9.428093	2.057632
8	1.268887	8.779292	1.979185
9	1.344916	8.574310	3.231967
10	1.420915	8.412479	3.181783

Table 6.4: Mean Square Errors of the Estimators for Different Buffer Sizes

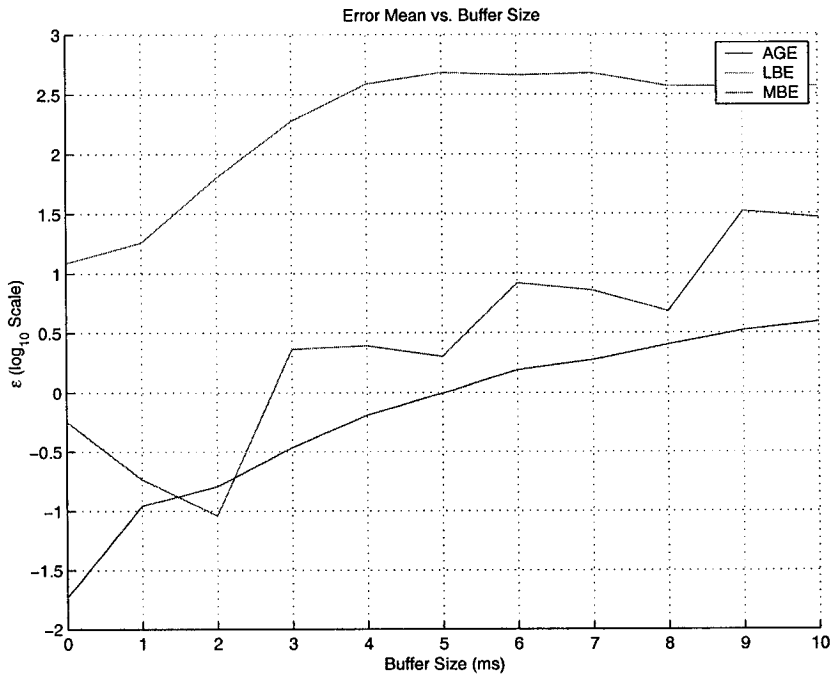

Figure 6.22: Estimators' mean error versus the buffer size

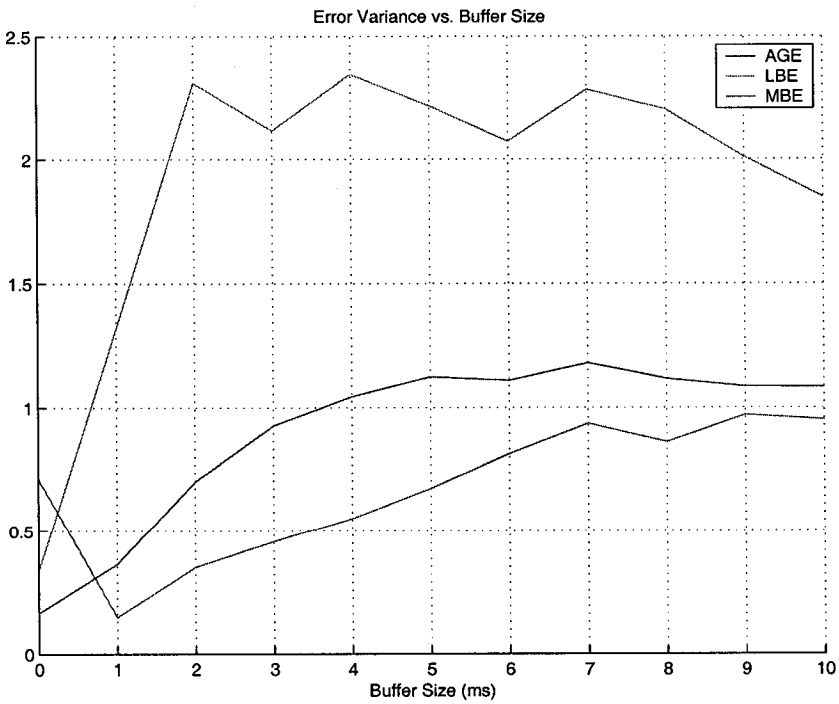

Figure 6.23: Estimators' variance error versus the buffer size

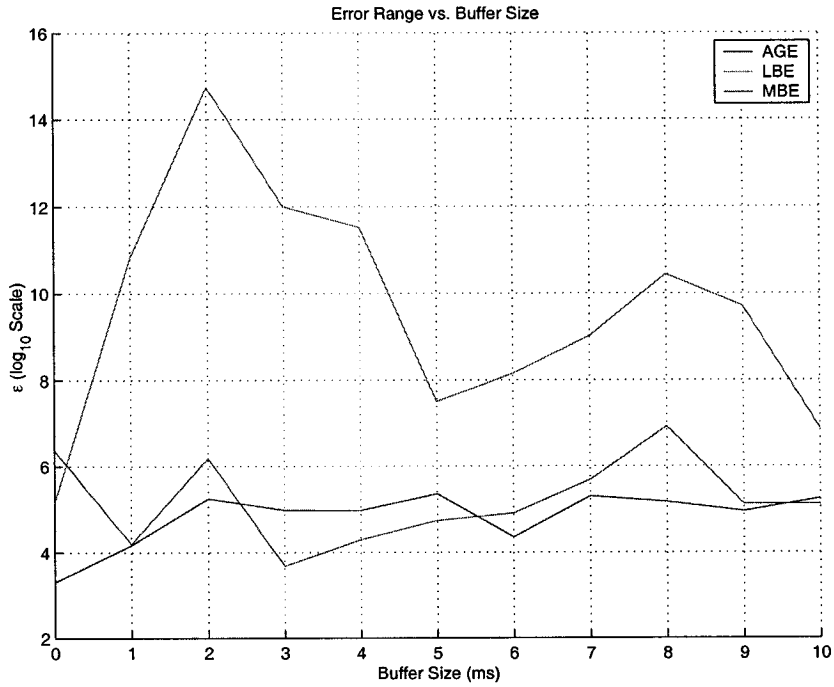

Figure 6.24: Estimators' range error versus the buffer size

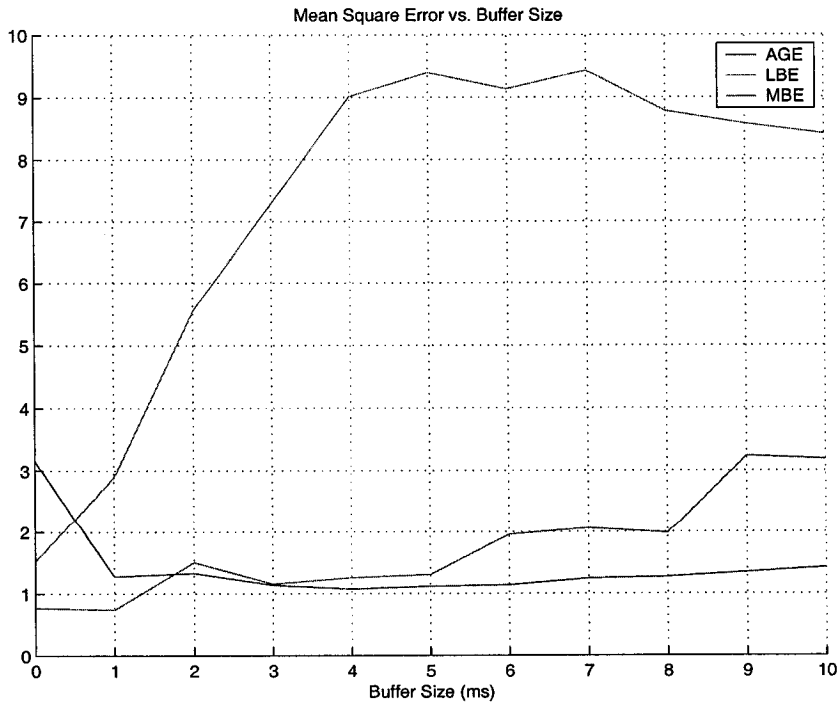

Figure 6.25: Estimators' mean square error versus the buffer size

The MBE estimator has performance metrics that are close to the ones of the AGE estimator. The error mean of MBE is larger than the one for the AGE. Nevertheless, since the bias is not constant, it results that the estimator behaves better for the region of smaller buffers and slightly worse in the larger buffer region. Things are reversed when variance is considered: in this case, the MBE has the more desired behavior because its variance is smaller.

Computational complexity of the MBE estimator is very good. Hence, it is recommended for use in large capacity multiplexers, when large numbers of sources share the same egress buffer. Furthermore, this estimator can be used for off-line traffic engineering, when network dimensioning against traffic traces is desired, in the stated conditions.

6.4 Experimental Results for PQ Service Disciplines

To verify the behavior of the estimator described by the Equation 6.35 a two queue PQ scheduler has been fed with the same traffic mixes as the ones described in Sections 6.3. The loss values predicted by the estimator, at the lower priority queue, are compared to the observed loss values.

The lower priority queue has a buffer with a length between $[0, 10]$ ms. The lower bound on the buffer length has been chosen such that it is larger than zero, for the reasons enunciated in Section 6.2.2. The differences between the estimated loss values and the observed loss values, denoted by ε_k^j for each value k obtained for the estimator, at buffer length j, were recorded. The statistics of each of the sets of values defined as $\{\varepsilon_k^j \mid k \in [0, n]\} \quad \forall j \in [0, 10] ms$.

6.4.1 Per Source Loss at the lower Priority Queue

Service guarantees in our framework are on a per service basis. Since all the results in this section deal with the *aggregated loss* at a multiplexer, we investigate, by simulation, if Equation 5.10 holds or not. If we denote by $\log_{10} \hat{P}_{loss}^s$ the estimated loss probability of the single source, and by $\log_{10} \hat{P}_{loss}^a$ the estimated loss probability of the aggregate, we are interested in the following measure:

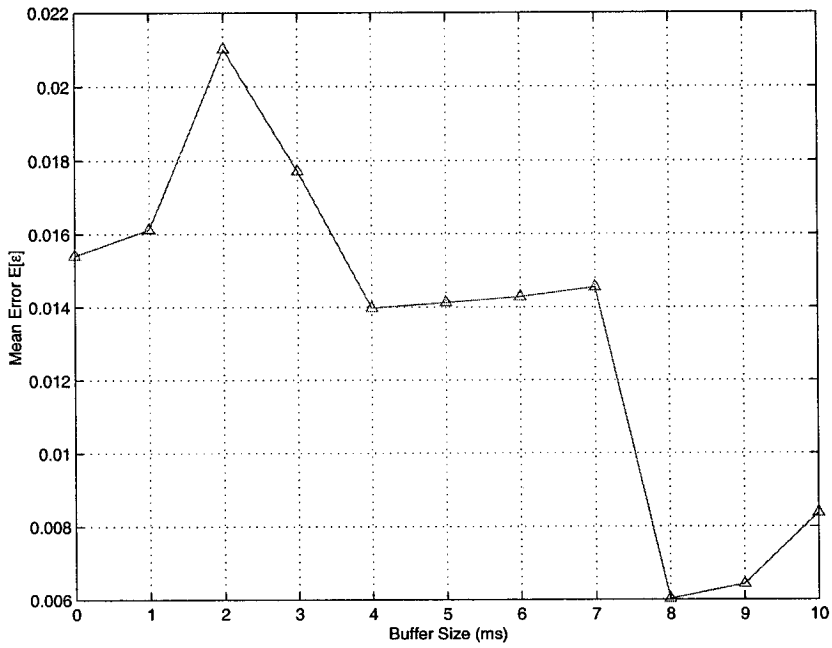

Figure 6.26: Mean of the Loss Difference between individual flow loss and the aggregated flow loss in highly aggregated traffic at the lower priority queue in a two queue PQ scheduler

$$\hat{\varepsilon}^s := \log_{10} \hat{P}^s_{loss} - \log_{10} \hat{P}^a_{loss} \tag{6.41}$$

Figure 6.26 plots the results of the average of $\hat{\varepsilon}^s$ over a large number of simulations. Similarly, Figure 6.27 details the results of the variance of the error. Both are very small over the range of studied buffers. Therefore, it can be concluded that the relative loss of the individual flows is equal to the relative loss of the aggregate, at multiplexers that aggregate large number of flows.

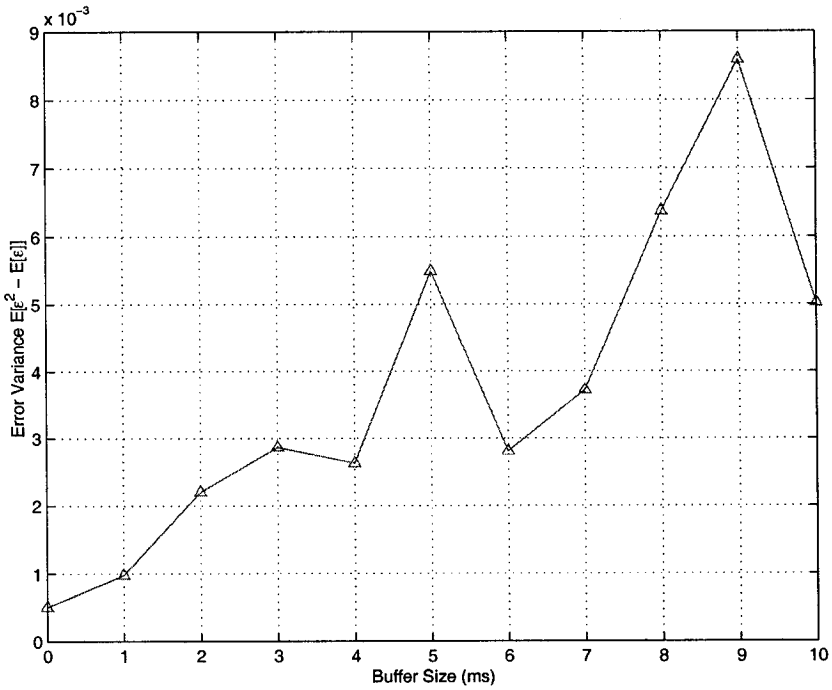

Figure 6.27: Variance of the Loss Difference between individual flow loss and the aggregated flow loss in highly aggregated traffic at the lower priority queue in a two queue PQ scheduler

Buffer Size (ms)	Mean	Variance	Minim	Maxim	Mean Square Error
0	0.015405	0.000506	-0.036152	012467	0.000740
1	0.016122	0.000979	-043275	026021	0.001234
2	0.021008	0.002201	-026902	0.343318	0.002631
3	0.017712	0.002867	-0.237904	0.213606	0.003164
4	0.013973	0.002629	-0.271681	0.210440	0.002808
5	0.014124	0.005491	-0.631974	0.367612	0.005655
6	0.014280	0.002808	-0.239983	0.238526	0.002993
7	0.014541	0.003718	-083779	0.360976	0.003903
8	0.006019	0.006372	-0.350774	0.406669	0.006361
9	0.006421	0.008588	-0.536762	0.450362	0.008563
10	0.008373	0.005021	-0.321699	0.263031	0.005051

Table 6.5: Individual Flow loss versus Aggregate Loss in PQ schedulers

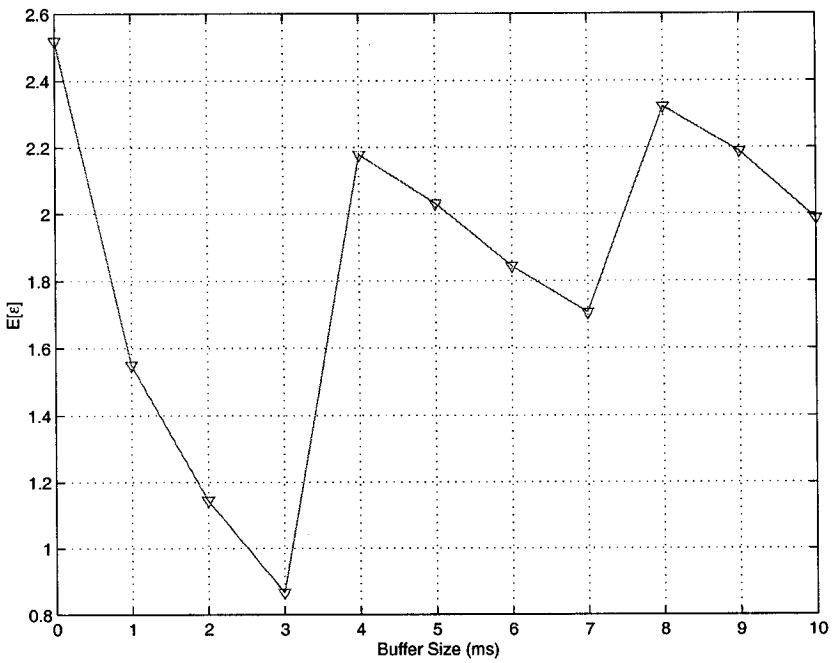

Figure 6.28: Mean of Error for the PQ estimator

6.4.2 Model-Based Aggregate Traffic Estimator

The estimator based on Equation 5.49, called (AGE), has the highest computational requirements of all the investigated estimators. This is a result of the unconstrained optimization problem that has to be solved to obtain the estimation of loss.

The mean of the error and the standard deviation are summarized in Table 6.4.2. The results show that this estimator follows the shape of the loss curve well, over the buffer range considered in this dissertation. However, the second column of Table 6.4.2 shows that the estimator does not have a constant bias. The mean error increases, almost linearly over the buffer of range $[0, 10]$ ms. The error range is rather large, with errors spanning over 5 orders of magnitude. Also, the error ranges are large for all buffer sizes. This means that the use of this estimator without run-time corrections can lead to large estimation errors and, depending on the sign of the error, to either large violations of the loss limit or to some network underutilization.

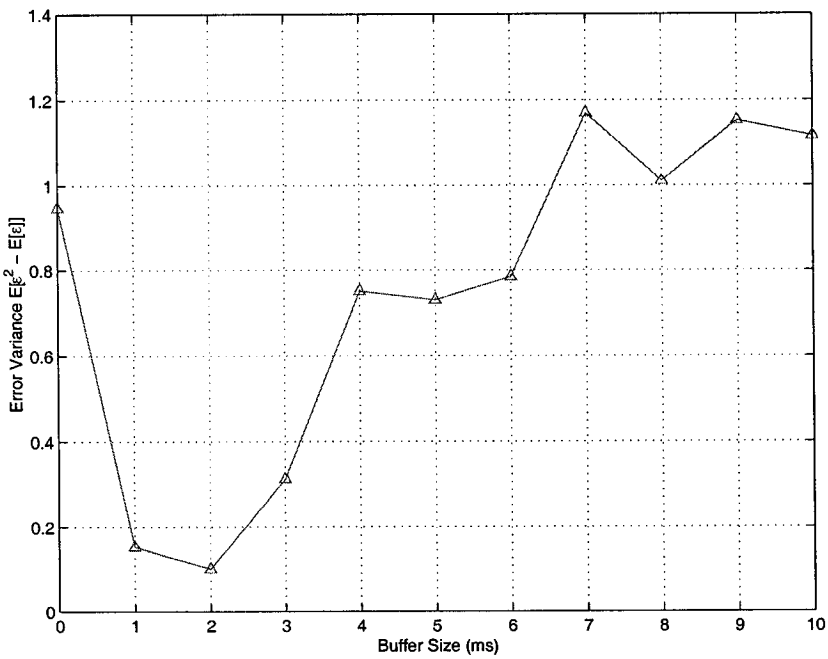

Figure 6.29: Variance of Error for the PQ Estimator

Buffer Size (ms)	Mean	Variance	Minim	Maxim	Mean Square Error
0	2.518221	0.945728	1.354900	7.741680	7.282597
1	1.549529	052641	0.859316	3.310370	2.552927
2	1.144971	000729	-0.248548	2.802930	1.411161
3	0.866049	0.312165	-2.370560	2.454530	1.060432
4	2.178298	0.751016	0.875967	5.199300	5.491390
5	2.030697	0.730486	0.401198	4.843570	4.849471
6	1.843799	0.784567	-0.359330	5.037620	4.178823
7	1.706056	1.169416	-1.117180	5.203890	4.071865
8	2.322807	1.010031	0.990013	5.457050	6.398039
9	2.188601	1.151960	0.653526	6.193540	5.933141
10	1.985697	1.115701	005470	6.409430	5.049838

Table 6.6: Main Statistics of the PQ Estimator

6.5 Summary

One of the side effects of applying the Large Deviation Theory is that results rely heavily on moment generating functions of random variables. Since all the results presented in the previous chapter fall into this category we are presented with the problem of estimating the moment generating functions of the arrival processes for the traffic that enters the multiplexer.

Since there are two main ways of dealing with this problem, it is important to check which method, model or modeless, yields better results. The comparison between the methods in the FIFO case reveals that the model based estimation is as accurate as the modeless one, with the added advantage of small computational complexity.

Using this conclusion, for the PQ case only the model-based estimator has been studied.

The results for both the PQ and the FIFO schedulers show that the measurement based loss estimation can yield results that are not very accurate. This conclusion mirrors findings by other researchers in the domain, for other type of estimators. This shows that direct application of loss estimation to admission control can yield QoS target violations.

In the next chapter, we present a solution to this problem that is based on on-line estimator correction.

Chapter 7

Service Admission Control

This chapter details the service admission control mechanism used by the VPN service platform presented in Chapter 4. For this purpose a *service* centric point of view on the admission control problem is presented and detailed. In this case, the service and traffic requirements are solved together. Thus, this represents a generalization of the *connection admission control* problem. The chapter is structured as follows: it presents the egress interface model and starts with the presentation of the service admission control for the FIFO queuing discipline. It then introduces the problem of service admission control, presents a solution to it and expands on the problem by presenting a model correction technique with application to the SAC problem.

The requirements of the admission control system that is presented in this chapter, can be summarized as follows:

- To be able to support multiple QoS classes, differentiated by loss.
- To be able to work with buffered egress interfaces.
- To have small enough computational complexity, such that it can be implemented by an OTB control framework.

7.1 Egress Interface Model

The service model presented in Chapter 3 has to be mapped in available hardware resources onto the NEs that form the network. Each egress interface of an NE is modeled as a set of queues $\mathcal{Q}$ that are served by a work-conserving scheduler. Flows that belong

to the same traffic class in the VPN model are terminated in the same queue at NE level.

We denote by $\mathcal{Q}_{jk}$ the set of queues that are present at the interface k of a NE j. The number of queue instances used at each egress NE interface is equal to $min(\|\mathcal{Q}_{jk}\|, \|\mathcal{C}\|)$. Hence, whenever possible each COS is mapped onto a separate queue. Queues are realized by a buffer that stores the egress bound traffic. The maximum buffer length of a queue k is therefore given by this fixed buffer value B_k.

When a service is mapped to the network, some of the nodes the service is routed through might not have egress interface semantics that are powerful enough for the number of classes of service requested by the service. Therefore, the service will have to be mapped into the available resources. The mapping is done by *up-classing*, which means that each class of service that cannot be individually served will be served by the closest highest class available.

This tradeoff allows the operator to offer QoS on the network, with the tradeoff of lower link utilization. As noted in Chapter 5 for highly multiplexed links such an utilization decrease is not an issue.

Therefore the service control framework provides a set of mappings $\mathcal{S} = \{\mathcal{M}_k \mid \mathcal{M}_k : \mathcal{C} \rightarrow \mathcal{Q}_{jk}, \forall k \in [1..n] \forall j \in NE_k\}$ that projects each class of service into the set of queues that are available at a node. The mapping is restricted by the fact that for any two COS, c_1 and c_2 with the loss requirements $P_1 < P_2$ have to be mapped into queues such that $\mathcal{M}_k(c_1) \leq \mathcal{M}_k(c_2)$. This definition allows to "up-class" any COS when mapping them to the egress interface queues.

7.2 Admission Control of VPN Services

Figure 7.1 depicts the general service admission control process at a network node. The process uses the measurement information obtained from each queue of the egress interface. This information together with the user provided traffic description is used as the basis for admission of traffic at the node.

The service admission control uses large deviations based results to estimate the loss at each node for the service request.

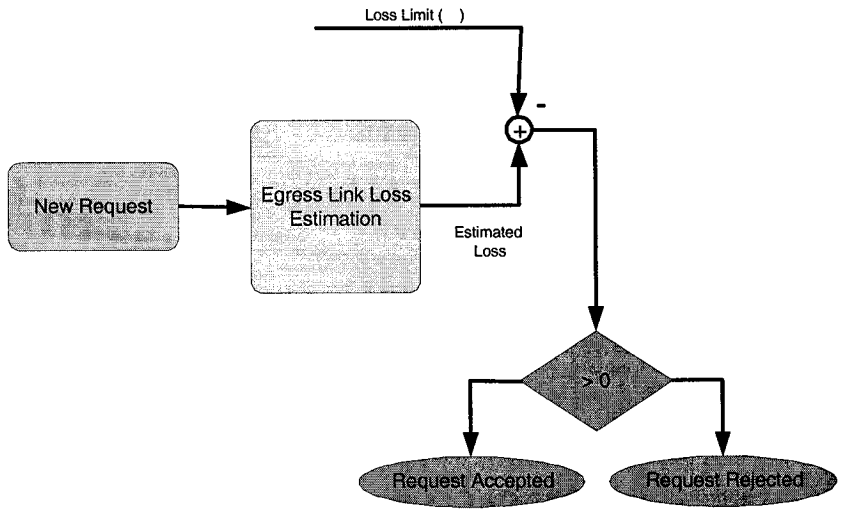

Figure 7.1: Nodal Service Admission Control Procedure

Using the interface mode defined in the previous section, we define a *service admission control (SAC)* function, which allows a new service request to be admitted or rejected at an egress P node interface. Hence, SAC is a Boolean function that is given a part of the service request and answers *yes* if the request can be admitted at the interface and *no* otherwise.

The traffic information that is passed to SAC follows the *pipe model* traffic specification defined in Chapter 3 at page 40. Each service request contains a traffic specification for each of the node pairs that are part of the VPN. Each request is a tuple that specifies the peak rates that are required for each COS supported by the network [1]

Consider an egress interface that runs a PQ scheduler with a number of queues equal to the number of classes of service that the network control system supports. The loss at a queue k, of such a node, for the preset buffer length B_k has to satisfy the following inequality:

$$\mathbb{P}_{loss}^{kj} \leq \gamma^k \tag{7.1}$$

Equation 7.1 represents the target of the admission control system for each individual queue.

Let us denote by $\mathbb{P}_{loss}^{j} =< \mathbb{P}_{loss}^{1j}, \mathbb{P}_{loss}^{2j}, ..., \mathbb{P}_{loss}^{nj} >$ the vector of loss ratios at a node j.

[1] Average rates are not considered because, in general, they are not stationary.

In addition let us denote by $\Gamma =< \gamma^1, \gamma^2, ..., \gamma^n >$ the vector the preset loss targets.

Definition 7.1 A loss vector $\mathbb{P}^j_{loss}$ is said to *satisfy* the loss targets, denoted by $\mathbb{P}^j_{loss} \leq \Gamma$ if

$$\forall k \in [1, n] \quad \mathbb{P}^{kj}_{loss} < \gamma^k$$

■

The admission control algorithm is based on Equation 6.35 at page 104. Given the current estimated loss, the new estimated loss at priority j is calculated as:

$$\log \hat{\mathbb{P}^*}_{new,j} = -\frac{2C^2\tau_j(1-\hat{\rho_n})}{\hat{\sigma}^2} - \frac{1}{2}\log(4\pi\frac{2C^2\tau_j(1-\hat{\rho_n})}{\hat{\sigma}^2}) - \log(\frac{2C^2(1-\hat{\rho_n})\hat{\rho_{n,j}}}{\hat{\sigma^2}}) \quad (7.2)$$

In Equation 7.2 σ^2 is defined by Equation 6.29 at page 103 while ρ_n and $rho_{n,j}$ are defined by the following two relations:

$$\rho_n = \rho + \sum_{k=1}^{j} \frac{r_k}{c} \quad (7.3)$$

$$\rho_{n,j} = \rho_j + \frac{r_j}{c} \quad (7.4)$$

Algorithm 7 SAC($\mathbf{r}, \mathbf{P}, \mathbf{\Gamma}$)

1: **if** $\sum_{k=1}^{n}(\rho_k + \frac{r_k}{c}) \geq 1$ **then**
2: return Rejected {Stability condition is not satisfied}
3: **end if**
4: **for all** $k \in [1, n]$ **do**
5: **if** $\log \hat{\mathbb{P}^*}_{new,k} > \gamma_k$ **then**
6: return Rejected
7: **end if**
8: **end for**
9: return Accepted

The correctness of the algorithm is easily proven in the following proposition.

Proposition 7.1 Algorithm 7 terminates in $\mathcal{O}(n)$ operations, where n is the number of queues that are found at the multiplexer. ■

Proof: The algorithm contains only one loop that terminates after *at most* n iterations. Therefore the algorithm complexity is $\mathcal{O}(n)$ in the number of queues that have to be

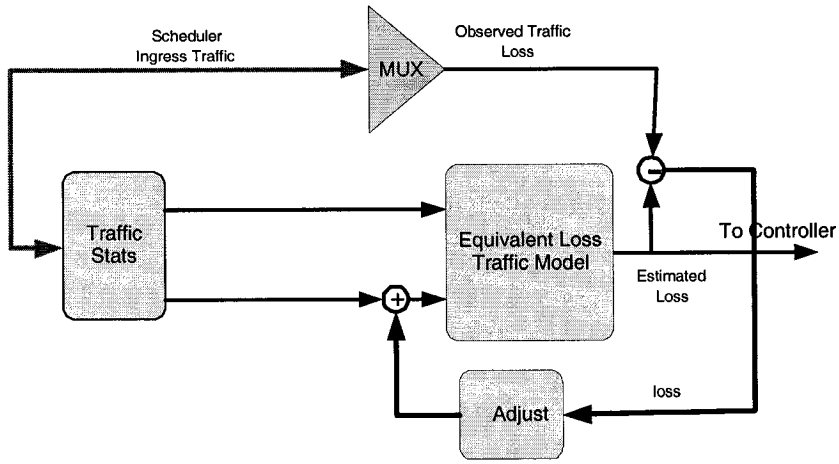

Figure 7.2: Reactive Model Correction through real loss observation

checked.

■

The same algorithm can be used in the FIFO case by checking the admission of $\sum_{k=1} nr_k$ against the traffic that is already admitted at the multiplexer.

7.3 Reactive Model Correction

As shown in the previous chapter, model based estimation of loss performs well when compared to the modeless estimation technique. However, both the FIFO and the PQ estimators exhibit large error ranges. Because the reason for obtaining a traffic model is to estimate the loss of that traffic at a given multiplexer, it can be inferred that *any* model will do as long as the loss estimated by it is close to the values experienced during operation. Our approach for this matter is inspired by the system identification techniques that are widely used by the control community. Let us assume that the amount of traffic that is lost over a given time interval can be *measured*[2]. It is generally accepted [85] that highly multiplexed traffic can be reasonably modeled as a constant stream summed with a Gaussian variable, as presented in Chapters 5 and 6. In general, there are three parameters that have to be specified, in order to fully determine such a

[2]This information has to be available at NEs in some form, because it is only the *measured* loss that can be used for SLA validation.

model:

- the traffic average, denoted by $\hat{\rho}$.[3]
- the traffic variance, denoted by $\hat{\sigma}^2$
- the Hurst parameter $\hat{h}$

Because of its additive nature, the traffic average can be determined with good accuracy over any time interval. The variance is difficult to calculate reliably because it implies a fine characterization of traffic. The same obstacle is faced in the determination of the Hurst Parameter. As Beran [4] details, the *accurate* determination of the Hurst parameter faces extra impediments in addition to the ones faced by the calculation of $\hat{\sigma}^2$. Therefore, let us consider a simple Gaussian model. For this model, in the case of a FIFO multiplexer, Equation 6.22 at page 99 describes the *estimated* loss that the multiplexor will experience. In the previous chapter the left hand side of the equation was considered to be unknown and solved by using the right hand side. If, on the other hand, we consider the left hand side to be *known* and $\hat{\sigma}^2$ to be unknown an exponential equation is generated that can be solved numerically. The *calculated* value of $\hat{\sigma}^2$ can then be used to correct the value used by the algorithm.

To correct the variance consider Equation 6.35 at page 104. Consider that the equation unknown is the term $x = \frac{2C^2\tau_j(1-\hat{\rho_n})}{\hat{\sigma}^2}$, and that the right hand side is obtained by measurements. We rewrite the equation in x as:

$$\log(\mathbb{P}) = -x - \frac{1}{2}log(4\pi) - \frac{1}{2}log(x) - log(x\frac{\rho_j}{\tau_j}) \tag{7.5}$$

Solving Equation 7.5 in x gives an expression that involves the first branch of the Lambert's W function[23]. Although it is a transcendent function with no analytical form, it can be evaluated numerically. In our work we used the C code generated by Maple as a solution for the equation.

$$x = \frac{3}{2}W[\frac{2}{3}e^{-\frac{2}{3}[log(\mathbb{P}+\frac{1}{2}log(4\pi)+log(\frac{\rho_j}{\tau_j})]}] \tag{7.6}$$

[3]The hat notation is used, to indicate that the values are *estimated*

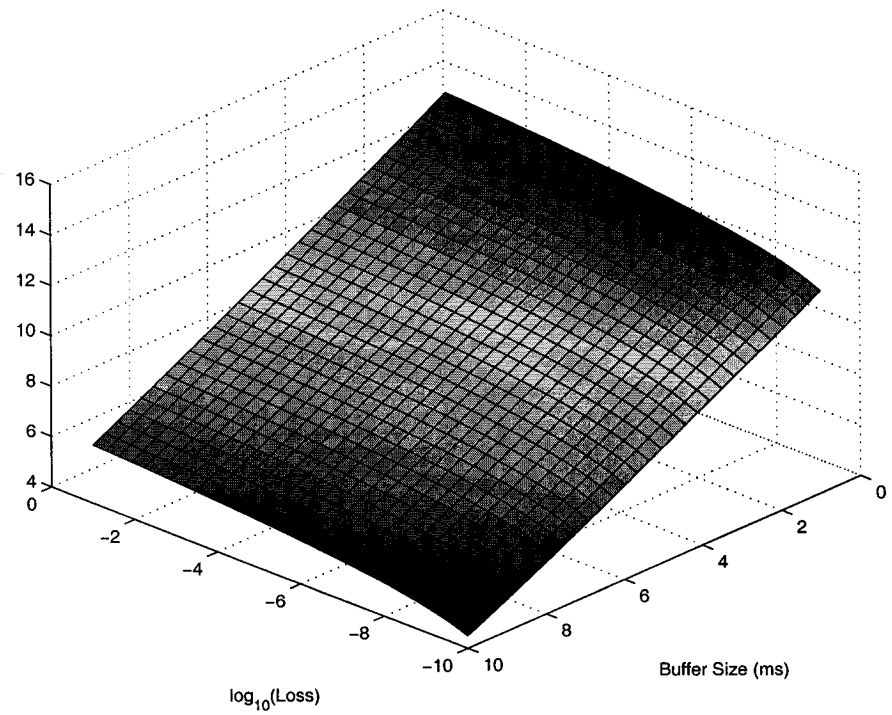

Figure 7.3: Lambert W Function

To obtain an idea about the range of x we plotted it against a range of values for loss and for the buffer size, while the utilization ρ was fixed at 0.9. Figure 7.3 details the results of this numerical work.

From Equation 7.6 we can readily obtain the corrective value for $sigma^2$ as:

$$\hat{\epsilon}_{\sigma^2} = \frac{2c^2\tau_j(1-\rho)}{\frac{3}{2}W[\frac{2}{3}e^{-\frac{2}{3}[log(\mathbb{P}+\frac{1}{2}log(4\pi)+log(\frac{\rho_j}{\tau_j})]}]} - \hat{\sigma}^2_{msmt} \tag{7.7}$$

Where $\hat{\sigma}^2_{msmt}$ denotes the measured value of the traffic variance.

This result can be used to cast the admission control problem in a different manner. All known measurement based admission control systems try to measure the first, and sometimes second, order moments of the arrival process. Of the two first order moments (average and variance), the average can be accurately determined on-line. The variance presents a computational problem for on-line processing, because to be able to calculate it, it is necessary to maintain some history about the stochastic process from which the

average has to be determined first. For on-line processes, this is problematic because it makes the determination process dependent on the data window length. Therefore, in general, the calculation of the variance of the arrival process suffers from errors. This can make the loss estimator imprecise, which adds to the modeling errors that are inherent because of the assumptions that we make regarding it. Since additive measures have greater accuracies, the admission control process can be made more precise by using additive measures only. Equation 6.35 contains two such measures: one is the running average while the other is represented by the loss ratio. Therefore, similar to the model correction process described above, we can determine the *equivalent* variance for a Gaussian model that has the same loss as the one experienced by the multiplexer.

Hence, the variance can be considered as a model tuning parameter. Using all three values $(\rho, \sigma^2, log(\mathbb{P}))$ we can apply Algorithm 7 for the service admission control.

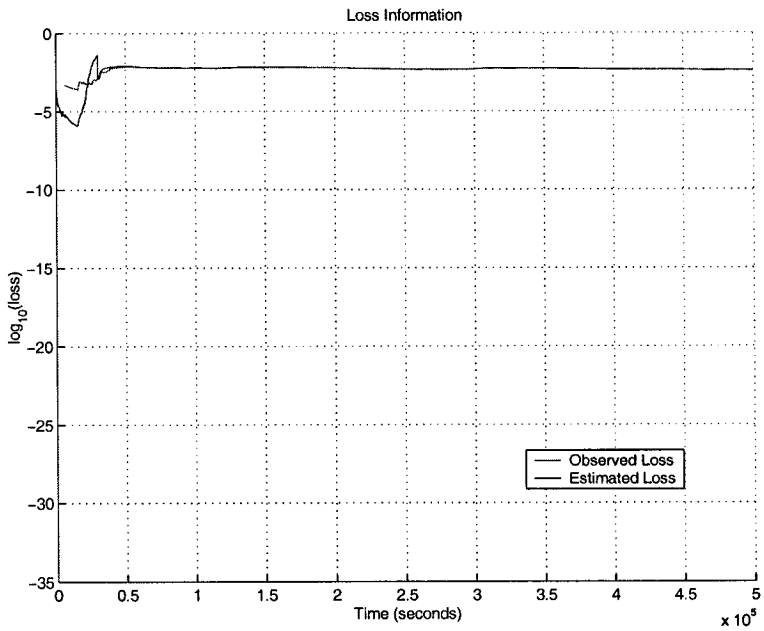

Figure 7.4: Real and Estimated Traffic Loss for Stationary Traffic. The traffic mix is a heterogeneous combination of Ethernet and MPEG2 traces. Link Capacity = 1Gbps, Buffer Size = 5ms

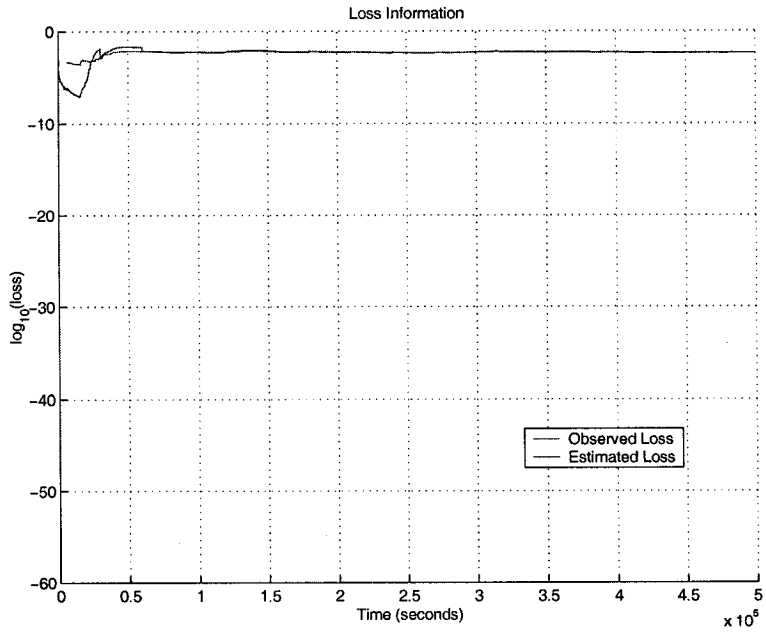

Figure 7.5: Real and Estimated Traffic Loss for Stationary Traffic. The traffic mix is a heterogeneous combination of Ethernet and MPEG2 traces. Link Capacity = 1Gbps, Buffer Size = 8ms

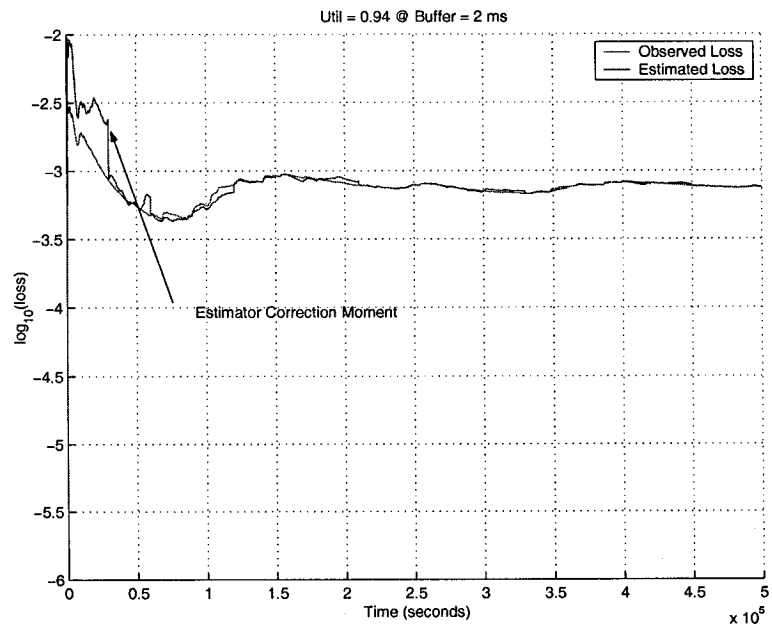

Figure 7.6: Real and Estimated Traffic Loss for Non-Stationary Traffic. Link Capacity = 3Mbps, Buffer Size = 2 ms

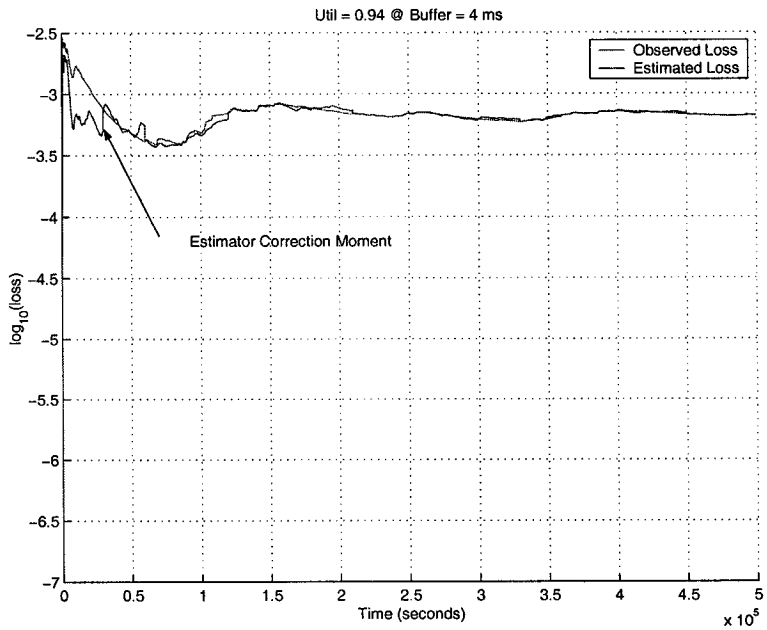

Figure 7.7: Real and Estimated Traffic Loss for Non-Stationary Traffic. Link Capacity = 3Mbps, Buffer Size = 4 ms

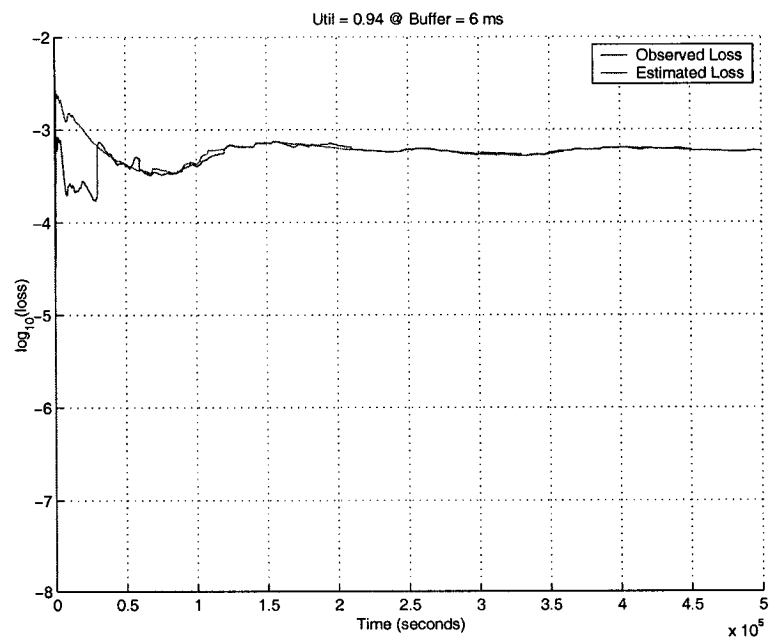

Figure 7.8: Real and Estimated Traffic Loss for Non-Stationary Traffic. Link Capacity = 3Mbps, Buffer Size = 6 ms

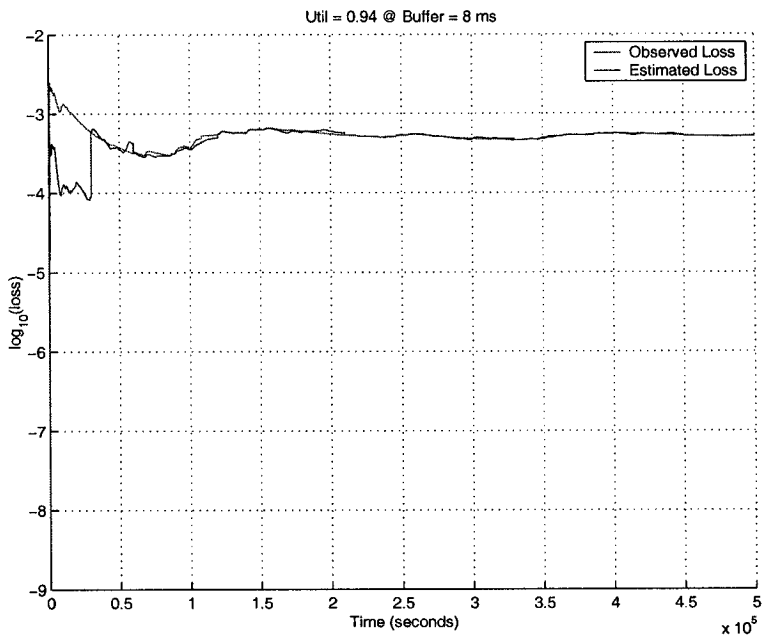

Figure 7.9: Real and Estimated Traffic Loss for Non-Stationary Traffic. Link Capacity = 3Mbps, Buffer Size = 8 ms

7.4 Experimental Results for the Model Correction Mechanism

To check the validity of the model-correction solution we have used the same type of traffic mixes as the ones presented in the previous chapter. In addition, a *non-stationary* traffic trace obtained from NLANR[4] has been used to see if the solution can handle such a case.

The experimental setup is a single node that is fed with traffic. The aggregate ingress traffic is measured and the loss is estimated. After a preset time, to allow the system to accumulate loss statistics, the model is corrected by calculating the variance corrector. The model correction has been recalculated every 86400 simulated seconds (i.e. once a day).

Figures 7.4 to 7.9 show the actual and the estimated loss for a number of the experiments. The results show that the model follows the real loss very closely, demonstrating its effectiveness.

Two different sets of experiments were conducted. The dynamic models for the FIFO, PQ and Leaky bucket regulators have been implemented in Matlab. The reason for this choice has been the availability of filters and control elements that were used in the simulations of the next chapter. In the first set of experiments the NLANR traffic trace has been sent through a FIFO multiplexer with link capacity fixed at $3Mpbs$. Several buffer sizes from $[1, 10]$ ms were tried. The results for four of the buffer sizes are displayed in the Figures 7.6 to 7.9. The running average and running variance of the source has been measured with elements provided in Simulink. The results show that after the model corrections, the real and estimated losses are very close. Furthermore, variations in the long term loss are also followed.

The next set of experiments involved the traffic traces that were used for the verification of the loss estimators in Chapter 6. A trace has been obtained by randomly adding traffic traces such that the resulting average is 0.94 of the link capacity. The

[4]The NLANR web site (http://www.nlanr.net) contains many traffic traces (http://pma.nlanr.net/Traces/) that were collected from different POPs around the world. Since most of the trace sizes, when unpacked, surpassed the maximum file sizes that were available on the machines at our disposal, only a trace from the New-Zeeland to US link has been decoded and reassembled. This trace contains a weeks worth of traffic in *one direction* from NZ to US.

resulting trace has been fed into a multiplexer with link capacity of $1Gbps$ and buffer lengths from $[1, 10]$ ms. The results for two buffer sizes are displayed below. As in the non-stationary case, the loss estimator follows the real loss curve closely.

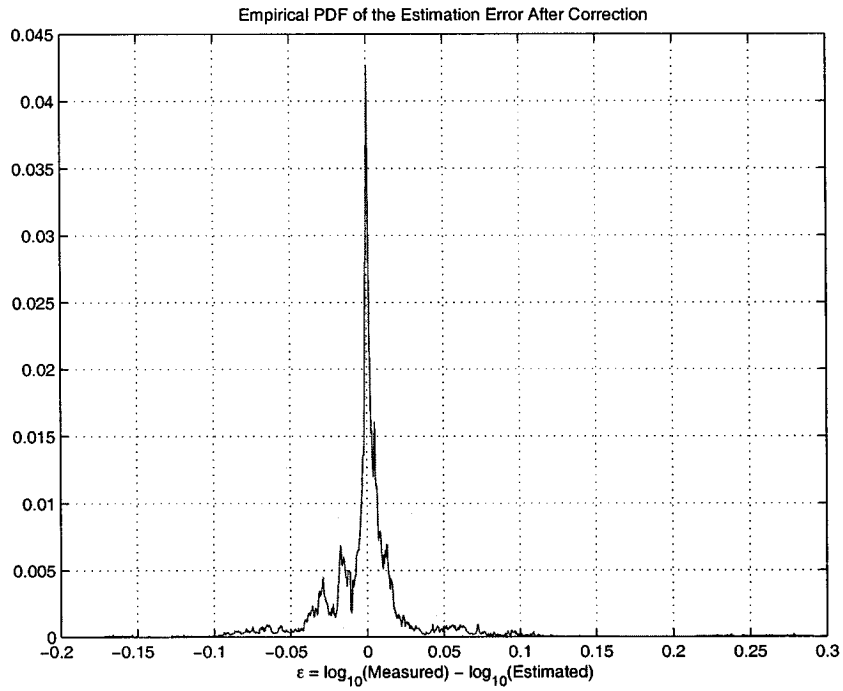

Figure 7.10: Empirical PDF of the difference between the Measured Loss and the Estimated Loss *after* the first model correction.

The differences between the base 10 logarithm of the measured loss and the base 10 logarithm of the estimated loss are aggregated in the empirical probability density function of the estimator error after correction. The results, displayed in Figure 7.10 show that the error is very close to zero most of the time. Comparing these results with the results obtained in Chapter 6 it can be safely concluded that the technique improves the estimation of the loss probability.

7.5 Summary

The SAC mechanism, together with the topology instantiation mechanisms are used to *proactively* control the network. Each part of a service request has to be mapped into a P-node egress interface. Since our framework maps service segments, a service admission

control procedure is proposed. The algorithm uses the aggregate traffic estimators that were introduced in Chapter 5 and 6. The algorithm uses the traffic measurement process to estimate if a new request can be admitted or not. The computational complexity of the algorithm is good, being $\mathcal{O}(n)$ and hence polynomial in the number of queues.

As the simulations in Chapter 6 revealed the value of the loss predicted by the estimator can differ from the real loss by several orders of magnitude. Therefore, in order to achieve good results with such a system it is necessary to *correct* the estimator with information inferred from the actual loss of the system. It is shown in this chapter that this model correction can be extremely effective. Simulations with both stationary and *non-stationary* traffic show that after correcting the estimator its value follows the observed loss very closely.

Chapter 8

Dynamic Service Control

The main purpose of the service control architecture is to maximize the operating profit, subject to the QoS constraints. It is noted that when the network is heavily loaded, small changes in the user traffic can affect the QoS targets. In networks with high rates of service setups and tear-downs such a problem can be easily remedied by dropping some of the calls, or by not accepting any more calls until the situation is corrected through the natural dropping of some of the existing flows, with no overall effect on the provider revenue. In a network that carries VPN services neither of these strategies work. Completely dropping one of the services can have large economic consequences for the provider. Furthermore, since each of the VPN services is composed of traffic from many users, and hence it is an aggregate, the event can correct itself through a natural traffic decrease in the customer aggregate. On the other hand, there is a possibility that the problem worsens. To manage this behavior we propose an *active* control system for the VPN based networks.

8.1 Dynamic QoS Control for Instantiated Services

The reactive control system follows the block diagram presented in Figure 8.1. In our model only the P-node egress interfaces have to be monitored for the control system to operate properly. Each of the links in the provider network has a number of controllers equal to the queues that are attached to it. Therefore for a graph with $||E||$ links and $||O||$ classes of service there are at most $||E||||O||$ controllers. The fact that the maximum number of controllers is a product of two numbers deserves some consideration. It has

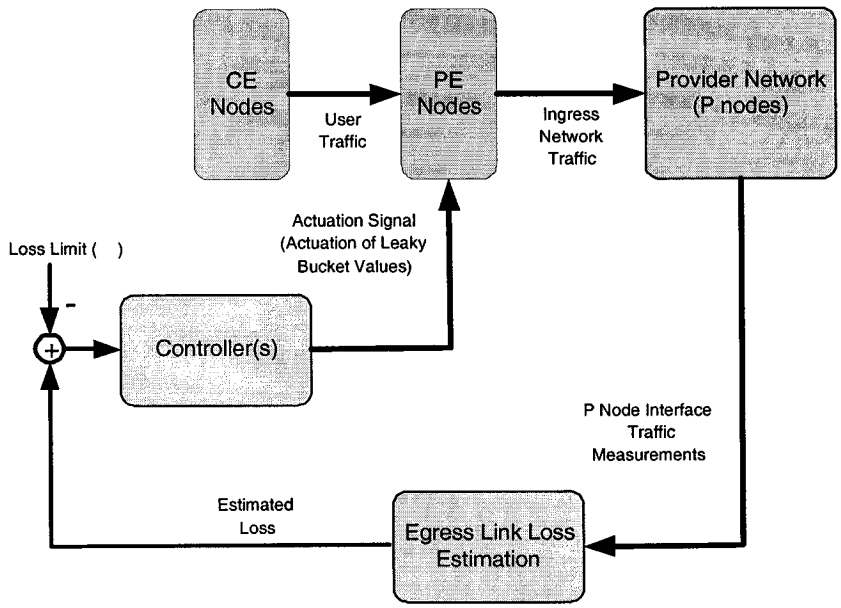

Figure 8.1: Reactive Control of QoS Assurances

to be noted that the maximum number of COS that are defined in most network is less than four. The main reason is that the traffic requirements of applications can be classified in a small number of classes of service. Hence, the values attained by $\|O\|$ are, in practice, very low. Although the number of controllers might look like a deterrent because of the simplicity of the control systems, this architecture is scalable to large number of links.

Each COS is served separately. Hence, the classes are controlled independently and as such the exposition can proceed by describing the FIFO case only. Each of the service instances has a number of flows that are routed through the provider nodes. We will consider that each such flow has a unique identifier which is a natural number and that all the flows that are instantiated use consecutive numbers starting at 1. Let us consider that there are s such flows set up in the network. A similar notation is used for the links, each link having a natural number identified between $[1, \|E\|]$. The reactive control system has access to the service instantiation information and as such to the flow setup information. Therefore, the system can build a traffic matrix M, with the property that $m_{ij} = 1$ if the service component j passes through link i.

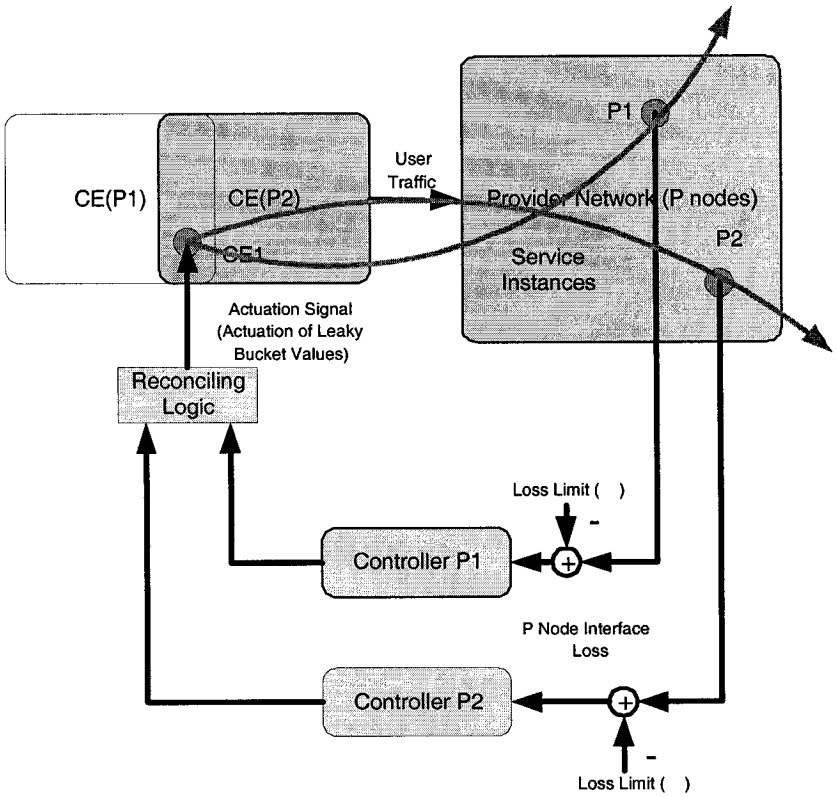

Figure 8.2: Concurrent Control of CE nodes

$$M = \begin{pmatrix} m_{11} & \cdots & m_{1s} \\ \vdots & \ddots & \vdots \\ m_{||E||1} & \cdots & m_{||E||s} \end{pmatrix}$$

From the traffic matrix, for each link i of the provider network, the set of CE nodes that sends traffic to that link is defined as:

$$S_i := \{j \mid m_{ij} = 1 \forall j \in [1, s]\}$$

The sets s_i represent the collections of CE nodes on which a given controller has to act. The sets S_i are not, in general, pair disjoint. Therefore, it is possible for two or more controllers to try to act on the same CE node simultaneously.

A situation like this is shown in Figure 8.2. This problem is solved by using a *reconciling logic* that takes all the actuation signals generated by the individual controllers and merges them into one value.

8.2 Controller System Design

The first thing to notice when designing the control element is that the controlled process is a non-linear system. Considering the input-state-output description given by Equations 5.1 , 5.2 and 5.4 at page 64. The problem is compounded by the control system lag. The lag between the observed event and the actuation has several reasons:

- measurement delays. Interface statistics are updated by the NE at time intervals that are chosen by the manufacturer. It is therefore reasonable to assume that the statistics are always read with a delay δ_s.
- processing delays. The information obtained from the NEs has to be processed in a format usable by the NEs. This delay is denoted by δ_p.
- actuation delays. Each NE has an actuation delay that depends on the NE hardware. In practice, this delay can be quite substantial and limits the actuation rate that can be achieved by the control system. (denoted by δ_a
- propagation delays. The temporal distance between the CE source and the measured node is called the propagation delay of the CE source and is denoted by δ_d

The total control delay is therefore $\delta = \delta_s + \delta_p + \delta_a + \delta_d$. The larger this value is the higher the probability of a large amount of traffic being injected in the network, which is not accounted for by the next control action. All the works that treat congestion control from a control theoretic perspective [3, 8] use the buffer length as their congestion indication mechanism. The resulting plant has fast dynamics and as such, the applicability of such schemes is doubtful, due to the delays incurred.

In our work we consider the control signal to be the loss ratio, which we compare with a preset target (see Figure 8.1). This signal has *slow* dynamics, for which an approximation can be obtained from the input-state-output equations.

The plant is connected to a modified PID controller. The classical PID controller [43] is a three term system which has the following z-transform expression:

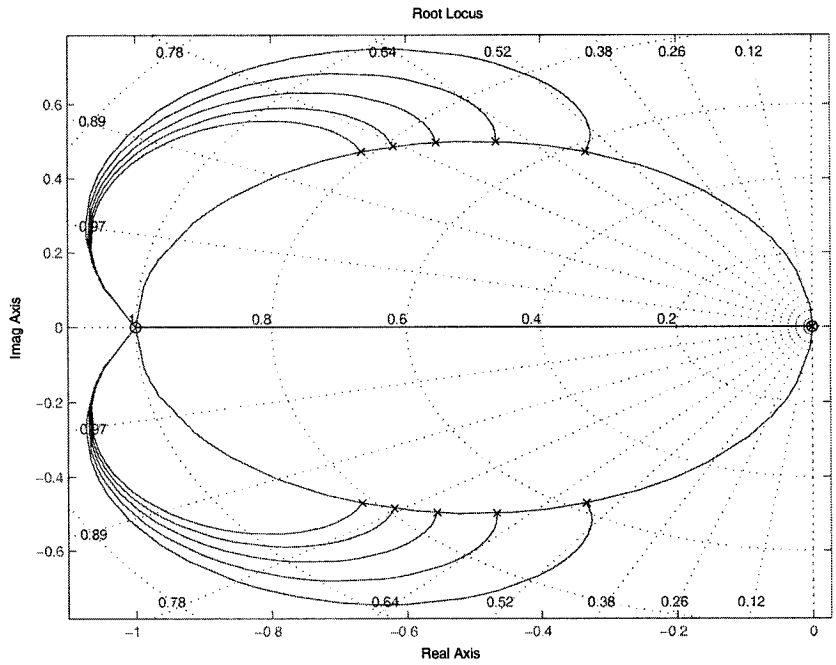

Figure 8.3: Double Locus for Equation 8.2 with $\delta = 10$

$$D(z) = K_p + K_I \frac{z}{z-1} + K_d \frac{z-1}{z} \tag{8.1}$$

The introduction of non-linear elements in the plant dynamics leads to changes to the control system. Due to the slow output signal dynamics, the derivative component of the controller can be set to zero. Furthermore, since we are interested in maintaining *inequalities* with the target signal, rather than equality, it is important that the integrator output is limited. Hence, a modified integrator is used, which has this property. Unfortunately, introducing such an element complicates the design of the K_p and K_I values. To overcome this, we proceed by considering the PID regulator, as described by Equation 8.1 and then use the modified I integrator in the implementation that is used to obtain the experimental results.

For the FIFO case, after making Equations 5.1 and 5.2 linear and after applying the z-transform to the plant-controller system we obtain the following expression for the ratio between the measured loss $L(z)$ and the target loss $L_0(z)$:

$$\frac{L(z)}{L_0(z)} = -\frac{\delta[K_p(1-z^{-1}) + K_i}{(1-z^{-1})^3 + \delta(K_p(1-z^{-1}) + K_I)} \tag{8.2}$$

Equation 8.2 has two parameters that have to be calculated: K_p and K_I. They can be obtained by the well-known root-locus method [43]. There exist an infinite number of solutions that satisfy Equation 8.2 as Figure 8.3 shows for a $\delta = 10$. Choosing a proper double depends on the behavior that is desired for the system.

Another design variable for the control system is given by the *minimum* amount of traffic that a source can be allowed to send in the system. Here several policies can be applied. The simplest, from the control system standpoint is the proportional policy. In this case, let us define by $r_{min,j}$ the minimum sending rate of a source, by c_I the capacity of the P-node I through which the CE node j sends traffic, c_{ce} the capacity of the link between the CE and PE, and by n_I the cardinality of the set S_I defined in the previous section. With this notation, we can write that:

$$r_{min,j} = min(\frac{C_I c_{ce}}{\sum_{k=1}^{n_i} c_{ce}}) \quad \forall i \quad s.t. m_{ij} = 1 \tag{8.3}$$

There are other possible policies, such as profit maximization, which were not considered due to their computational complexity. All our experiments were carried out using the proportional reduction policy.

8.3 Experimental Results

Two different topologies have been used to validate the control system presented in the previous section. First, we considered a single node in isolation. Second, a more complex architecture, with two P-nodes has been considered.

8.3.1 Single Node Case

To check the behavior of the control system in the single node case the topology presented in Figure 8.4 has been realized. The egress link capacity of the controlled node has been chosen based on the traffic trace that has been inserted in the node. Similarly, the system has been checked for different egress buffer sizes.

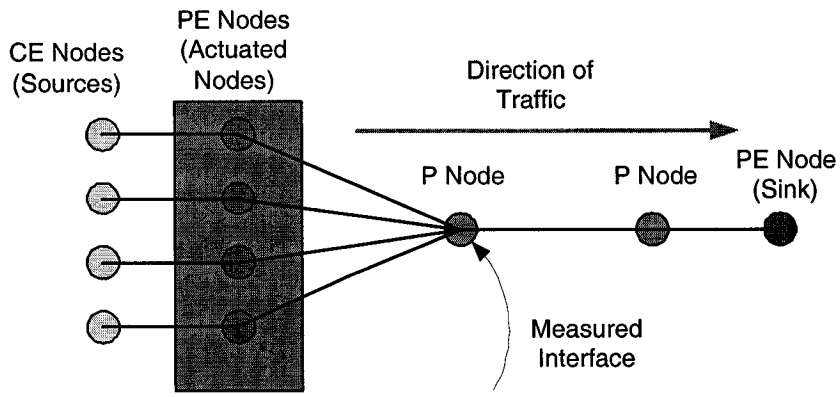

Figure 8.4: Single Controller Example

	K_p	K_I
1	0.2	0.0004
2	0.4	0.04

Table 8.1: K_p and K_I values for the single node experiments

The control is done by actuation of leaky bucket controllers at the PE nodes. Each of the CE nodes injects non-elastic traffic based on the traces used in the previous chapters. A value for the QoS loss target is chosen and the controller has to throttle the traffic to ensure that the target is not breached. Only the *proportional* source rate reduction has been implemented and tested.

The results obtained for several simulations are shown in Figures 8.5 to 8.11.

We have chosen two sets of parameters for K_p and K_I, as detailed by Table 8.3.1 to see how the system behavior varies. The value for δ has been fixed at $10s$. The link propagation delays are fixed to 20ms. All link capacities is $1Gbps$, all buffer sizes are $4ms$ and the traffic is a mixture of Ethernet and MPEG2 traffic.

The first set of values from Table 8.3.1 corresponds to the *Observed Loss 01* in the Figures 8.5 to 8.8 while the second set of value corresponds to *Observed Loss 02*. The differences in system dynamics are evident from the plotted curves. Again, it must be stressed that although optimal criterions like the minimization of the square error can create an order relation between these processes, practical considerations (ie. the number of device actuations) reverse the choice.

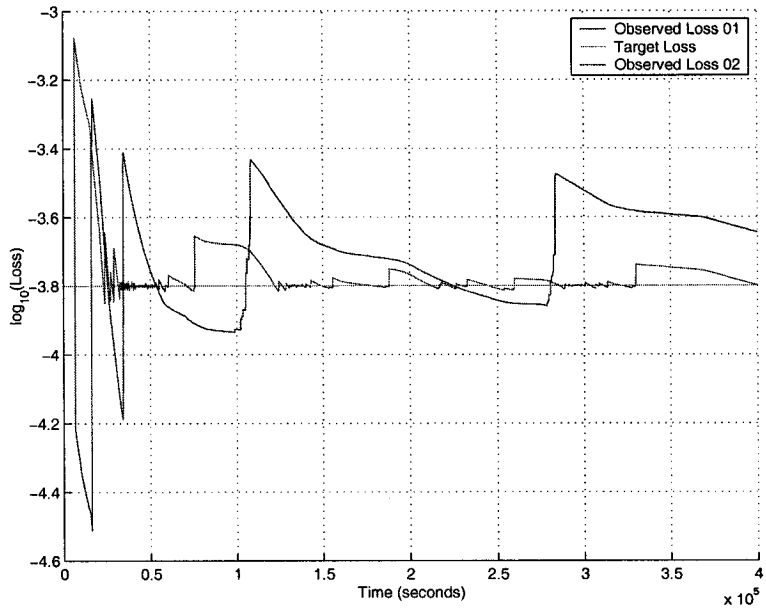

Figure 8.5: Measured Loss versus Target Loss at the Controlled Interface. Target Loss = -3.5

The analysis of the empirical probability density functions show that in both cases the control system keeps the actual loss close to the target loss. Both the EPDF and the ECDF show that there is a control error but that the expectation of the error is very close to zero.

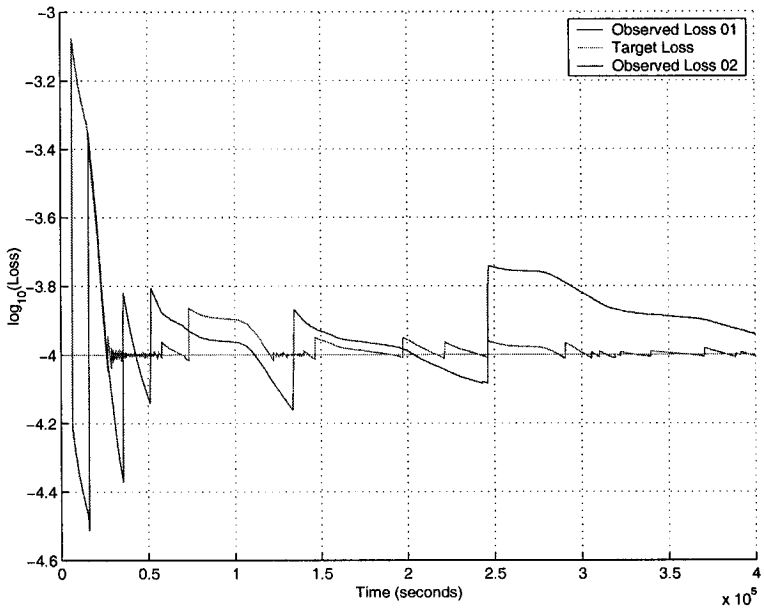

Figure 8.6: Measured Loss versus Target Loss at the Controlled Interface. Target Loss = -4

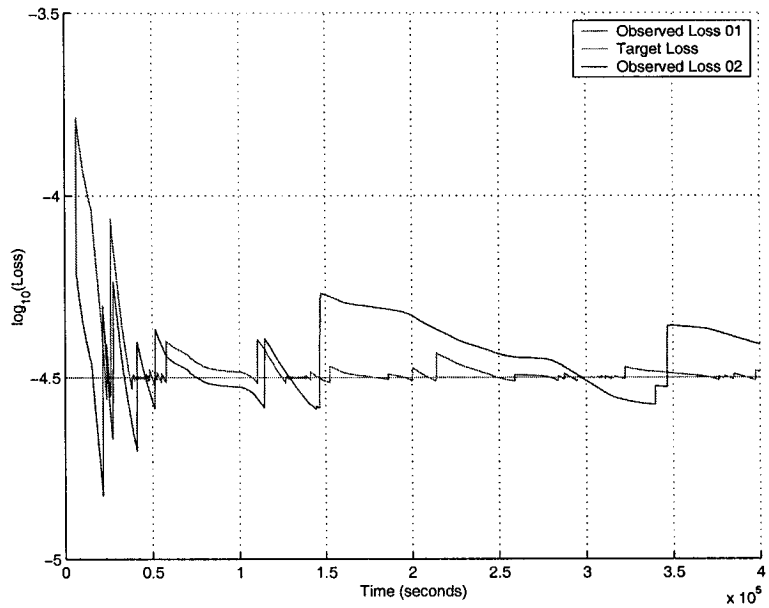

Figure 8.7: Measured Loss versus Target Loss at the Controlled Interface. Target Loss = -4.5

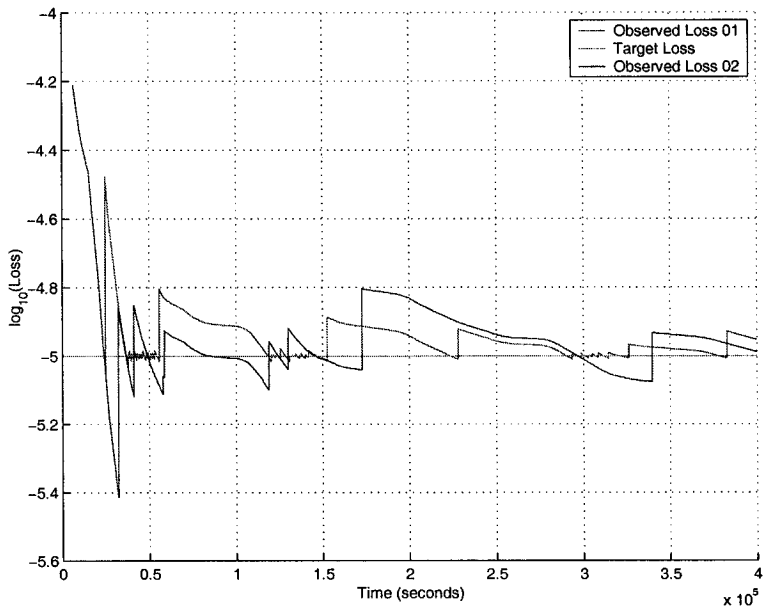

Figure 8.8: Measured Loss versus Target Loss at the Controlled Interface. Target Loss = -5

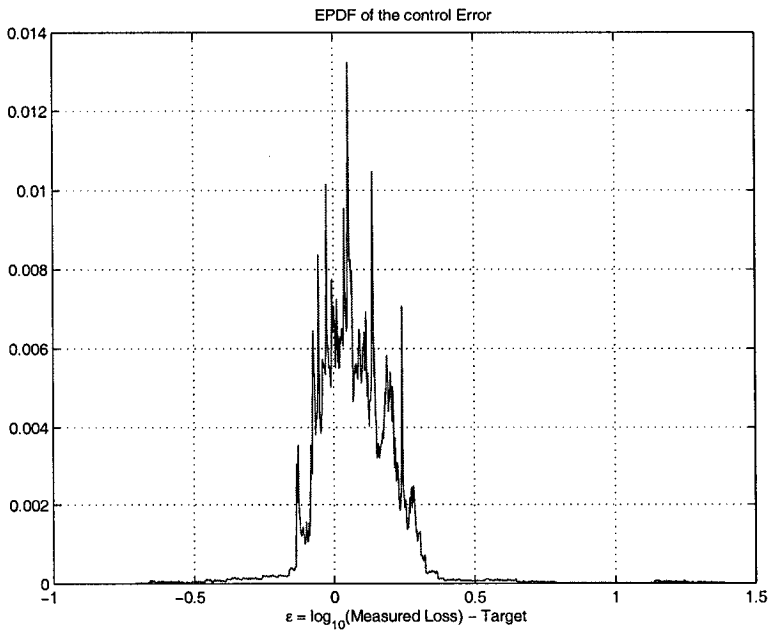

Figure 8.9: Empirical probability density function of the measured Error for the first set of control parameters

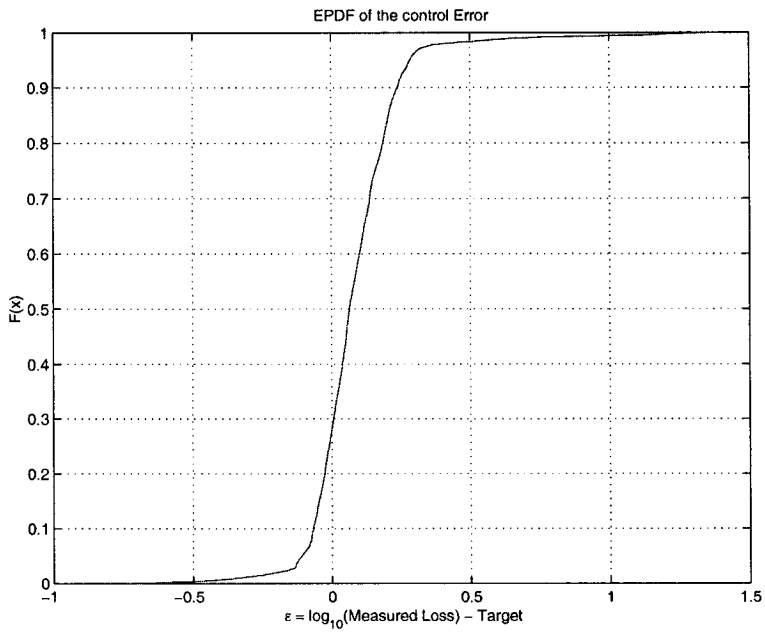

Figure 8.10: ECDF of the measured Error for the first set of control parameters

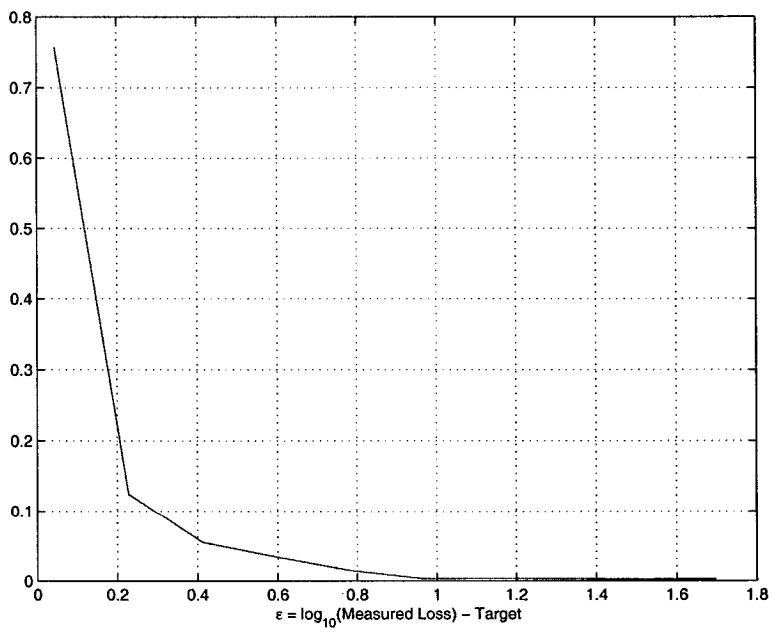

Figure 8.11: Empirical probability density function of the measured Error for second set of the control parameters

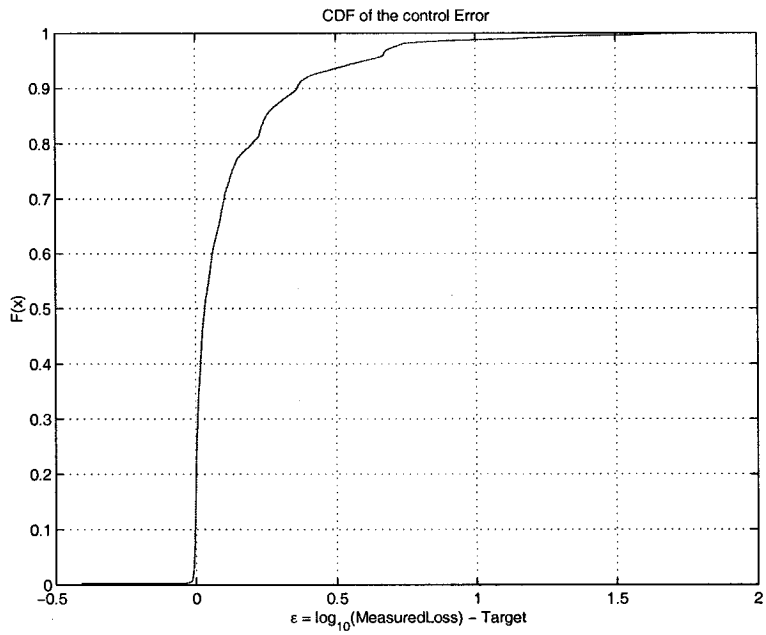

Figure 8.12: ECDF of the measured Error for the second set of control parameters

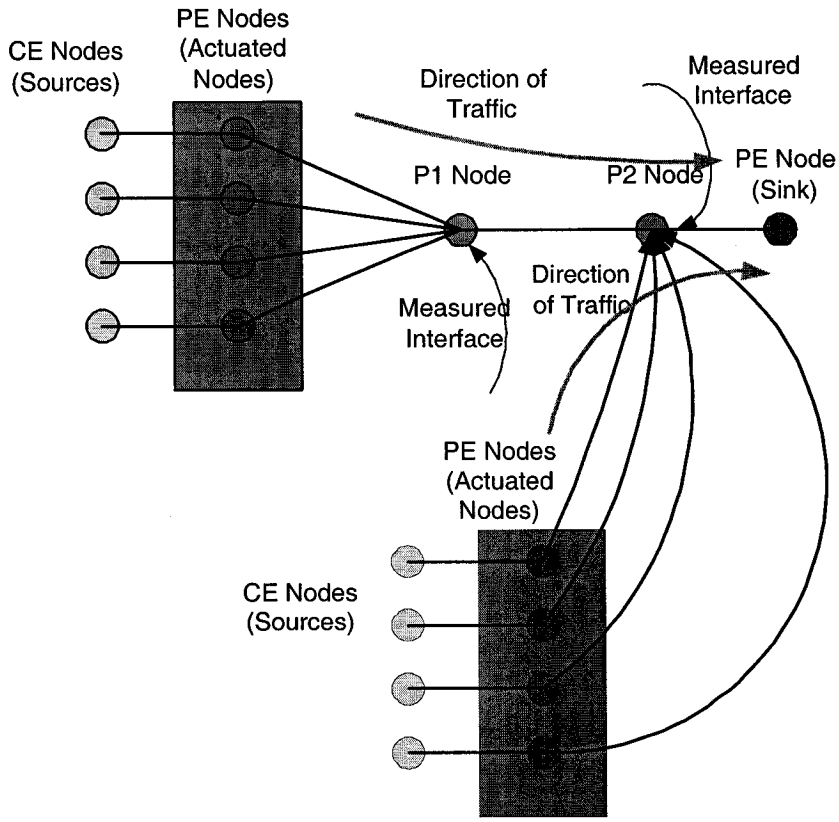

Figure 8.13: Concurrent Controller Example

8.3.2 Multiple Node Case

A more complex case, which involves concurrent CE traffic actuation, is presented in Figure 8.13. The topology contains P-nodes, in a feed-forward connection, with all the traffic from the CE nodes going to a sink node.

Similar to the single controlled node, a *proportional* reduction policy has been used for the simulations. Data from several runs, at different loss targets, and different buffer sizes have been collected and are shown in Figures 8.14 to 8.16.

The reconciling logic is implemented as a $min(r_0, r + min(\Delta r_k))$, where r_o denote the minimum rate that the source is allowed to send data, r is the current rate at which it sends data and $\Delta r_k \forall k$ represents the difference in actuation that is required during the current time step. This choice has the advantage that it has good computational complexity. In addition the choice is justified by the following argument: consider

that two nodes are sending actuation signals to a CE node akin to the case shown in Figure 8.13. If one of the nodes wants to increase the traffic from the CE node while the other wants to decrease it, then regardless of the current value of the source, the control action should be to decrease the current source rate. Hence, the $min(\Delta r_k)$ term in the above expression.

The simulations were run using the following parameters:

- $K_p = 0.02$
- $K_I = 0.0004$
- $delta = 120$ s

As in the single controller case, all links utilized for the tests have a capacity of $1Gbps$, buffers were fixed at $4ms$ each and the traffic trace was been obtained by mixing MPEG2 traffic with Ethernet traffic.

A major difference with the previous case, apart from the topology change, is that a very long delay (120s) has been chosen, to compensate for the time that is needed by the control system to change the parameters on the NEs.

Figure 8.14 shows the difference between the measured loss and the target loss at the P_1 node, while Figure 8.15 shows the same values for the P_2 node. To be able to draw some statistical conclusions, all of the error values are brought together in the empirical probability density function of Figure 8.16 and the empirical cumulative distribution function of Figure 8.17. The curves show that the control system is able to maintain the target loss very well as the expectation of the error is 0.0336 while the variance is 0.0193. Therefore, even with the very large total delay, of 2 mins, the system performed very well.

Simulations, with smaller values for δ, provide similar results. However, it remains an open question as to what is a practical range for the actuation updates on available NEs. This question warrants a statistical study in itself and it is crucial to practical implementation of the system.

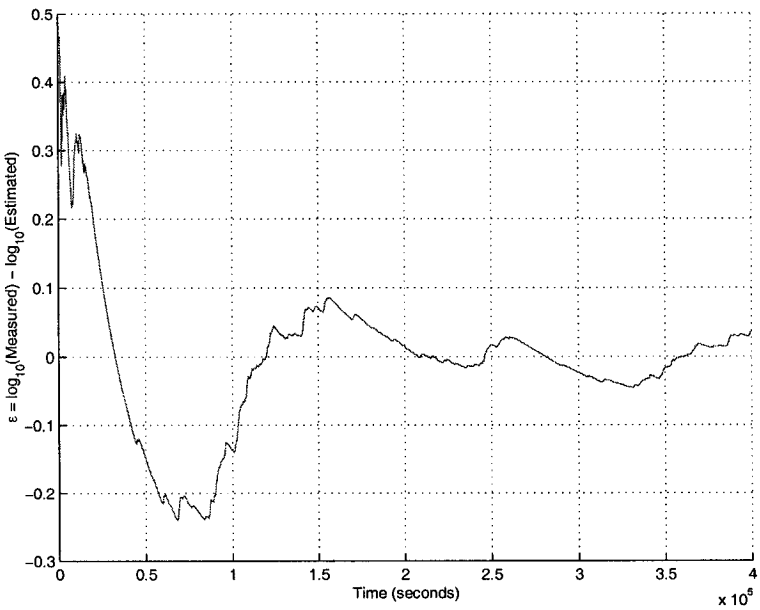

Figure 8.14: Time evolution of the difference between the measured loss and the target loss for node P1

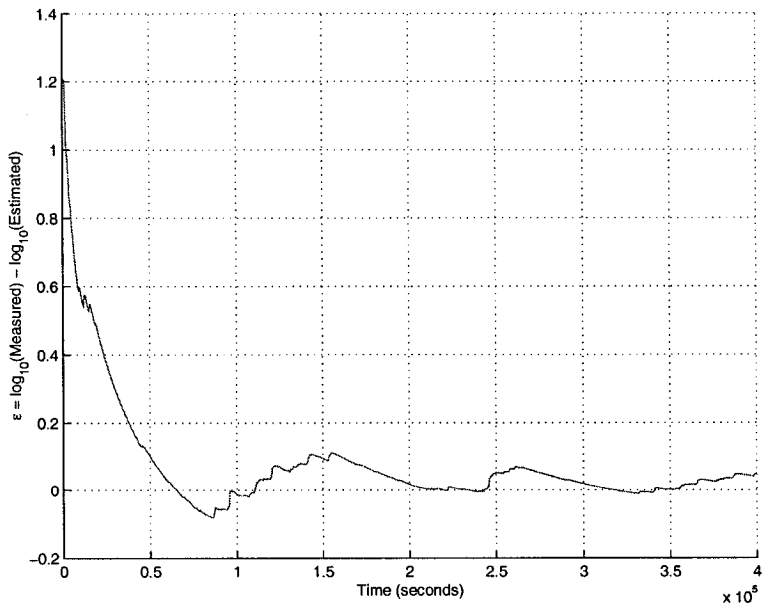

Figure 8.15: Time evolution of the difference between the measured loss and the target loss for node P2

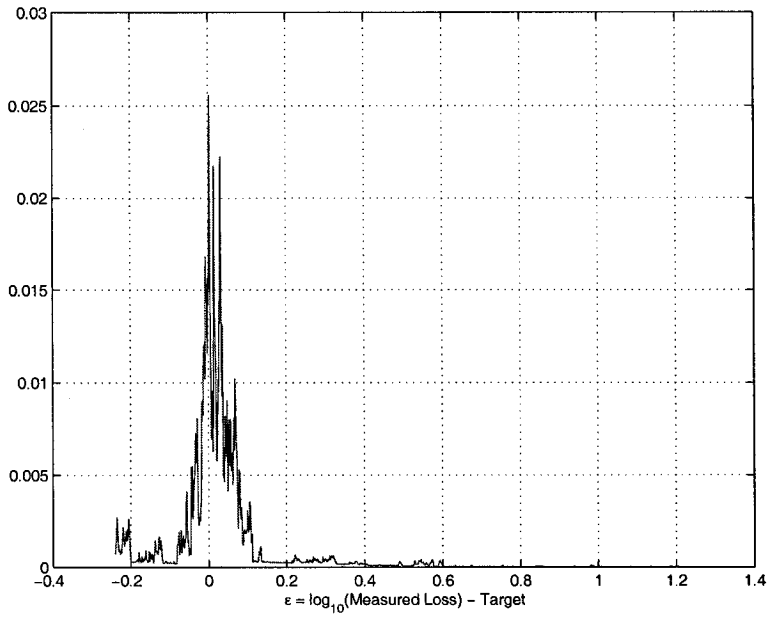

Figure 8.16: Empirical PDF of the control error at *both* nodes P1 and P2

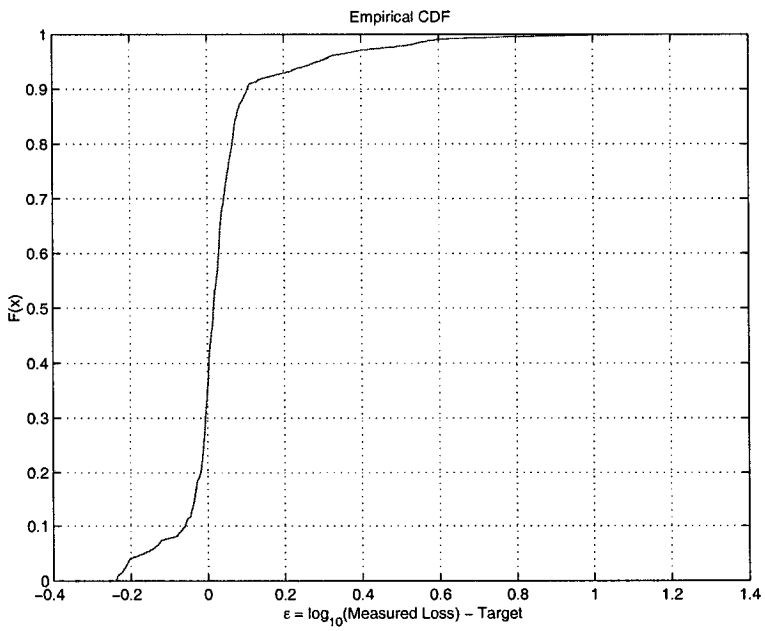

Figure 8.17: Empirical CDF of the control error at *both* nodes P1 and P2

8.4 Summary

This chapter presents the dynamic control system that maintains preset COS targets after service instantiation.

The system is based on concurrent control that actuates the traffic that is allowed in the network by the CE. There are two uses for this system: as a reactive control system when the network is heavily loaded or as a controller for the MO resizing mechanisms.

Each controller is an instance of a PI controller that is designed using the root-locus method from the description of the multiplexers. Based on the values that are chosen for the K_I and K_p constants the control system can be tuned to different behaviors as the previous section shows. However, regardless of the chosen mode of operation the method is effective in maintaining the network operation within a prescribed loss range.

Chapter 9

Concluding Remarks

This dissertation proposes a service control framework that uses on out-of-the-box architecture. There are a number of reasons why an OTB architecture is more suitable for service control. For example, such a model has the inherent advantage of better knowledge than any ITB controller. In the OTB case, the controller can make near optimal decisions, with respect to the network state, about the service setup based on the information known up to that point in time. The proposed architecture integrates two control mechanisms that complement each other: a *proactive* control approach, which takes place when a new service is instantiated in the network and a *reactive* control system that strives to maintain the loss targets that are assigned to each COS.

In this dissertation, we consider packet-switched networks that support several COS. Each VPN service instance is considered to follow the pipe model. Therefore for each service instance, between each pair of CE points that is instantiated, the service has an amount of reserved bandwidth for a subset of the COS offered by the network. At the center of the *proactive* control system lays a topology instantiation module together with the service admission control. To maximize the network utilization our SAC utilizes a measurement based service admission system. Each P-node egress interface is under the monitoring of a measurement agent that feeds data to the SAC module.

The SAC is based on the Large Deviations Theory results derived in Chapter 5. In the chapter we obtain loss approximations for aggregated traffic that is fed to a finite buffer multiplexer. The tightness of the results is checked against trace based simulations in Chapter 6. Large deviations based results can be estimated in two ways: by direct estimation of the moment generation function or by adopting a model and estimating

the model parameters. The chosen model is the Gaussian Noise Model (GNM). The simulation results presented in Chapter 6 show, that the model based aggregate traffic approximation is the better performing estimator. Hence, this estimator is subsequently used as the basis for the service admission control in Chapter 7.

To make the admission process more precise a new model correction is proposed. The model correction starts from the simple observation that in the loss estimation equation, the *variance* of the traffic can be considered to be unknown and hence obtained from the estimator equation, if we know the loss. Hence, by measuring the loss we can correct the traffic model that is assumed by the system. The correction mechanism, as demonstrated by simulations, is very effective in correcting the estimation errors that were observed in Chapter 6.

Similar estimators have been proposed in the past. Our work is novel in two important ways:

- The work presented in this dissertation uses *aggregate* traffic estimators with *finite* buffers, rather than individual flow estimators. This use of estimation is important for our framework as we monitor only the egress interfaces on the core nodes of the network.

- It compares the performance of model and modeless estimators and shows that under the right conditions, the model based estimator is a better choice, from both computational complexity standpoint and from an estimator error standpoint.

The work presented in Chapter 6 also contains several important conclusions for both network planning and service control. It is shown through simulations for links with large number of multiplexed sources, that the amount of buffer that is necessary to attain a given loss for fixed link utilization decreases with the increase of the link speed. The main practical implication of this conclusion is as follows. Consider the problem of planning a new network. Given two points in the network, n_1 and n_2 the capacity that is needed for the predicted traffic is c. In this case the best solution, from a delay perspective, is to use the largest available line card, with capacity c_m and therefore install a number of cards equal to $\frac{c}{c_e}$. Furthermore, if the network is planned using a cost optimization process, it is possible to introduce the advantage of lower delays as

a profit enhancer, and hence mitigate the higher costs for the interfaces with greater rates.

The final part of the dissertation presents the reactive control system. The reactive control system takes to types of actions:

- moves parts of services within the provider network, if there is spare network capacity.
- actuates the network bound traffic at the PE nodes when the network is heavily loaded or when a MO resizing policy is used.

Of the two types of control actions, only the last method is treated in the dissertation. In our solution, a number of controllers equal to the product of the number of provider network links multiplied with the number of interface queues, work concurrently at actuating the CE traffic. The actuation signals that are generated by each controller are reconciled and the CE traffic is throttled. The control system presented in Chapter 8 uses the long term loss as a target signal. Precisely designing a control system for network systems is made more challenging by the fact that most of the dynamical systems that are found in networks are non-linear. This problem can be circumvented, to some degree, by linearizing the input-state-output equations that describe the scheduling systems. By doing this, parameters for the control system can be obtained and the system can be controlled.

The validation of the reactive control architecture shows that the system is effective in maintaining prescribed loss targets. However, for any control system there are many parameter possibilities that in turn affect the type of error curve (difference between the measured loss and the target loss) that is observed. This curve can have a small amplitude, but have a difference function that varies rapidly between the positive and negative part of the y axes, or it can have larger amplitudes with smaller difference variations. The type of error curve directly affects the *number* of actuation commands. Having a large actuation frequency is undesirable, and, therefore, some parameter tuning is required in order to find the right balance between making the sum of the squared error as small as possible and having a small number of actuation commands.

The work presented in this dissertation can be expanded in several directions. The most promising direction of research would be to examine the reactive control system presented in Chapter 8 by considering the non-linearity of the scheduling elements in the design of the control system. In addition, the stability and controllability of the system can be studied.

A different, but related research direction can be undertaken by exploring the effects of the service setups and service tear-downs on the control system. It is important to check if the control system itself is stable under these load changes.

Appendix A

Analysis of High Resolution MPEG2 Streams

This chapter details the statistical characteristics of the MPEG2 traffic traces that were used in this work. The traces were obtained from MPEG2 encoded Region 1 (NTSC) DVD Movies with a resolution of 720 × 480 bits per frame, at 30 frames per second. These traces present interest because of the relatively high resolution, which is about four times the one of the available MPEG1 [86] and MPEG4 [40] movies, and because of the move towards higher quality TV imaging.

A.1 First and Second Order Properties

As link capacities increase, it becomes feasible to transmit higher resolution video streams on data networks. It is then important to understand the behavior of such sources. Since there is already a large body of digital video material at high MPEG2 resolutions available on DVD, and since there is web available software for streaming DVD movies across IP networks it was natural to ask what are the statistical properties of such sources. To study the question a number of movies have been decoded and the lengths of each of the frames recorded. The decoded sequence has then been deleted and a time series composed of the frame lengths obtained.

The set of numbers thus obtained can be viewed as the realization of a stochastic process. As such the first and second order characteristics can be studied.

The minimum, mean and maximum frame length statistics are available in Table A.1. These values are calculated as follows:

- The sequence length is calculated by multiplying the number of frames in each movie with the inter-frame time which is $\frac{1}{30}$ of a second.

- The minimum frame length is found through simple search in the sequence:

$$\hat{x}_{min} = min(\hat{x}_k : 0 \leq k \leq n), \tag{A.1}$$

- The maximum frame length is defined in a similar fashion with the previous characteristic:

$$\hat{x}_{max} = max(\hat{x}_k : 0 \leq k \leq n) \tag{A.2}$$

- The arithmetic sample mean $\bar{X}$ is calculated as the arithmetic mean of the frame sizes across a sequence:

$$\bar{x} = \frac{1}{n+1} \sum_{k=0}^{n} \hat{x}_k \tag{A.3}$$

- The peak per mean ratio gives a rough idea about the burstiness of the sequence. It is calculated as:

$$ptm := \frac{\hat{x}_{max}}{\hat{x}} \tag{A.4}$$

The items presents in Table A.1 are defined to be the following:

- The standard deviation of the sequence is calculated with the formula:

$$\hat{\sigma}_n = (\frac{1}{n} \sum_{k=1}^{n} (\hat{x}_k - \bar{x})^2)^{\frac{1}{2}} \tag{A.5}$$

- The Coefficient of variation is simply the ratio between the standard deviation and the sample mean:

$$\xi = \frac{\hat{\sigma}}{\bar{x}} \tag{A.6}$$

- The mean and the maximum rates are obtained by multiplying the mean and maximum frame sizes with 240 (8 bits per byte and 30 frames per second) and by scaling the results to Mbps.

Each of the figures from Figure A.1 to Figure A.4 depicts the following information:

- The first subfigure draws the normal probability plot. This helps identify how close both the frame lengths and the GOP frames are to the Gaussian distribution.

- The autocorrelation function for both the frames and the GOP. This function is calculated as:

$$\hat{R}(k) = \frac{1}{(N-k)\hat{S}_x} \sum_{j=0}^{N-k-1} (x(j) - \bar{x})(\hat{x}(j+k) - \bar{x}), \forall k \in [0..N-1] \qquad (A.7)$$

The estimator is unbiased as it can be easily seen:

$$\begin{aligned}
\mathbb{E}[\hat{R}(k)] &= \mathbb{E}[\frac{1}{(N-k)S_X} \sum_{j=0}^{N-k-1} (X(j) - \bar{X})(X(j+k) - \bar{X})] \\
&= \frac{1}{(N-k)S_x} \mathbb{E}[\sum_{j=0}^{N-k-1} (X(j) - \bar{X})X((j+k) - \bar{X})] \\
&= \frac{1}{(N-k)S_x} \sum_{j=0}^{N-k-1} \mathbb{E}[(X(j) - \bar{X})(X(j+k) - \bar{X})] \\
&= \frac{N-k}{N-k}\frac{1}{S_x}\mathbb{E}[(X(j) - \bar{X})(X(j+k) - \bar{X})] \\
&= \frac{\mathbb{E}[(X(j) - \bar{X})(X(j+k) - \bar{X})]}{S_x} \\
&= R(k)
\end{aligned}$$

The high spikes in the autocorrelation function of the frames correspond to the I frames that are a lot larger then the P and the B frames. The autocorrelation of the GOP is "smoother" as the jump in sequential sizes of pictures is a lot smaller.

- The distributions of the frame lengths and the GOP lengths.

A couple of comments regarding Table A.1 are required. First, it is interesting to note that both the average and the peak rates of the traces are large. Although the resolution is only about 4 times greater than that of the MPEG1 traces presented in [86] and the MPEG4 traces from [40], the average rates are at least 10 times larger. Because of this, the number of sources that can be multiplexed on a given link decreases by the same factor. Since the statistical properties of the traffic mix depends heavily on the number of multiplexes sources, it is of interest to see if the behavior of the MPEG2 mix is similar to that of the MPEG1 mixes.

Trace Name	SeqLength	MinF(KB)	MaxF(KB)	MeanF(KB)	Max/Mean
Annie Hall	01h:14m:25s	0.360	138.908	29.396	4.73
Austin Powers I	01h:11m:48s	2.240	94.272	20.706	4.55
Austin Powers II	01h:15m:59s	0.512	94.016	24.766	3.80
Big Lebowski	01h:33m:30s	0.360	132.120	20.054	6.59
The Big Blue	02h:14m:49s	0.270	118.660	23.993	4.95
Blade Runner	00h:50m:48s	0.360	120.304	24.576	4.90
Bugs Life	01h:15m:51s	0.360	175.896	28.134	6.25
Carlitos Way	01h:55m:08s	0.360	98.712	33.908	2.91
The English Patient	01h:08m:50s	0.512	119.488	30.079	3.97
Final Fantasy	01h:03m:58s	0.270	146.142	20.507	7.13
Happy Texas	01h:18m:44s	0.270	151.214	29.725	5.09
Heat	02h:16m:18s	0.360	132.196	25.043	5.28
A Life Less Ordinary	01h:22m:55s	0.360	110.108	24.649	4.47
Lock Stock	01h:25m:40s	0.360	120.720	23.031	5.24
Manhattan	01h:16m:53s	0.270	140.914	26.744	5.27
About Mary	01h:35m:10s	0.360	157.964	36.800	4.29
Matrix	01h:48m:59s	0.360	133.240	24.409	5.46
Nurse Betty	01h:27m:55s	0.512	92.416	26.106	3.54
OBrother Where Are	01h:26m:01s	0.272	129.824	25.069	5.18
Out of Sight	01h:38m:11s	0.960	133.696	35.344	3.78
Point Break	01h:37m:34s	0.360	155.668	34.865	4.46
Red October	01h:48m:06s	0.512	97.024	31.389	3.09
Run Lola Run	01h:04m:04s	0.270	143.962	22.443	6.41
Shakespeare in Love	00h:44m:37s	0.360	162.100	29.981	5.41
Shrek	01h:12m:06s	0.270	148.521	36.721	4.04
Snatch	01h:22m:07s	0.270	140.802	19.715	7.14
South Park	01h:04m:56s	0.360	206.392	26.421	7.81
Space Balls	01h:16m:51s	0.512	93.568	22.780	4.11
Strange Days	01h:56m:03s	0.360	121.136	31.858	3.80
Starship Troopers	01h:41m:08s	0.270	140.308	18.104	7.75
The Abyss	02h:16m:44s	0.360	116.644	21.869	5.33
Three Kings	01h:31m:47s	0.360	109.588	27.541	3.98
Total Recall	01h:30m:25s	0.360	117.172	23.344	5.02
A Fish Called Wanda	01h:26m:35s	0.512	123.985	24.498	5.06

Table A.1: Trace Frame Statistics

Trace Name	SampleStdDev	Coeff	MaxRate(Mbps)	MeanRate(Mbps)
Annie Hall	19569.25	0.665711	33.338	7.055
Austin Powers I	12043.76	0.581656	22.625	4.969
Austin Powers II	12420.87	0.501529	22.564	5.944
Big Lebowski	15878.24	0.791774	31.709	4.813
The Big Blue	16048.48	0.668882	28.478	5.758
Blade Runner	13694.44	0.557228	28.873	5.898
Bugs Life	19663.25	0.698914	42.215	6.752
Carlitos Way	10193.15	0.300612	23.691	8.138
The English Patient	14702.68	0.488802	28.677	7.219
Final Fantasy	16132.47	0.786681	35.074	4.922
Happy Texas	23341.44	0.785246	36.291	7.134
Heat	12586.81	0.502608	31.727	6.010
A Life Less Ordinary	23939.08	0.971199	26.426	5.916
Lock Stock	9843.65	0.427409	28.973	5.527
Manhattan	16081.98	0.601330	33.819	6.419
About Mary	27130.01	0.737228	37.911	8.832
Matrix	12935.78	0.529959	31.978	5.858
Nurse Betty	10633.64	0.407326	22.180	6.265
OBrother Where Are	24643.20	0.983015	31.158	6.017
Out of Sight	19089.74	0.540113	32.087	8.483
Point Break	19288.05	0.553221	37.360	8.368
Red October	12633.72	0.402489	23.286	7.533
Run Lola Run	14388.03	0.641092	34.551	5.386
Shakespeare in Love	20784.95	0.693271	38.904	7.195
Shrek	32392.84	0.882134	35.645	8.813
Snatch	14086.11	0.714487	33.792	4.732
South Park	30575.14	1.157229	49.534	6.341
Space Balls	10886.85	0.477913	22.456	5.467
Strange Days	14266.93	0.447829	29.073	7.646
Starship Troopers	12645.86	0.698512	33.674	4.345
The Abyss	11949.46	0.546411	27.995	5.249
Three Kings	11615.44	0.421751	26.301	6.610
Total Recall	11070.65	0.474240	28.121	5.603
A Fish Called Wanda	13662.86	0.557713	29.756	5.880

Table A.2: Trace Frame Rates

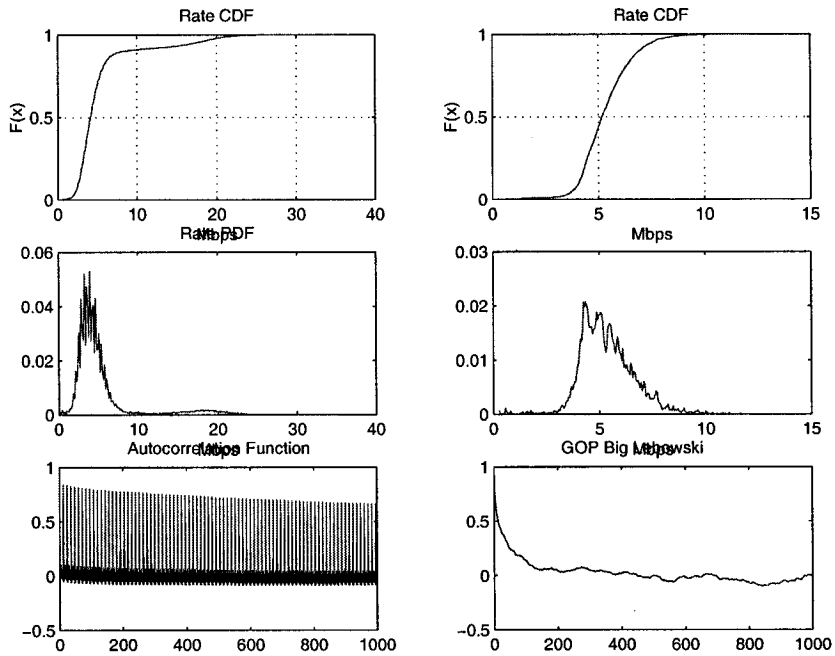

Figure A.1: Frame and GOP statistics for **The Big Lebowski**

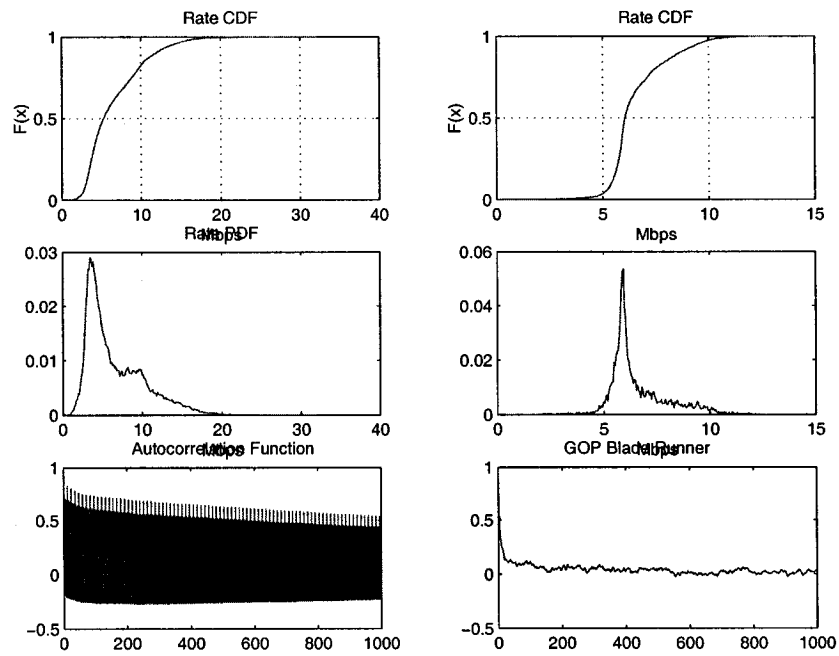

Figure A.2: Frame and GOP statistics for **Blade Runner**

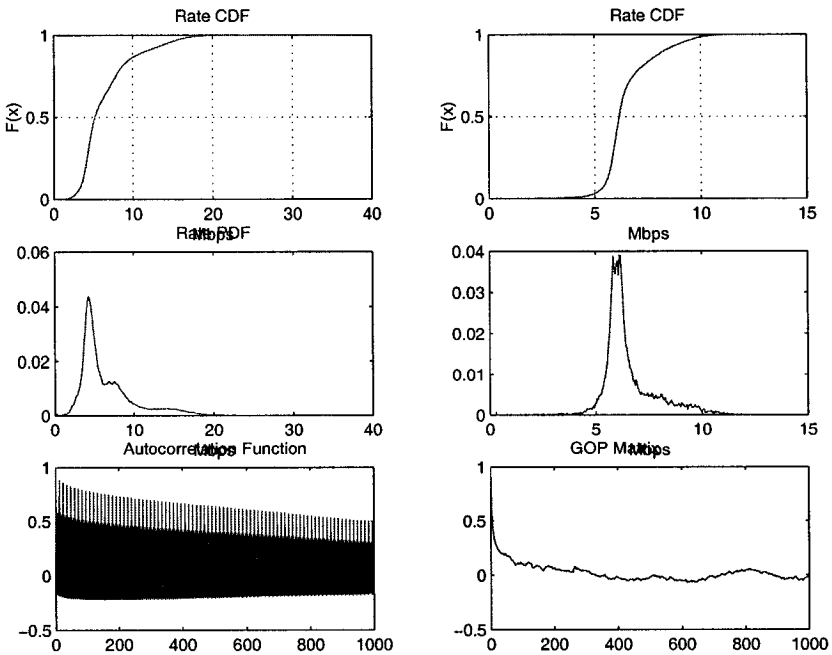

Figure A.3: Frame and GOP statistics for **Matrix**

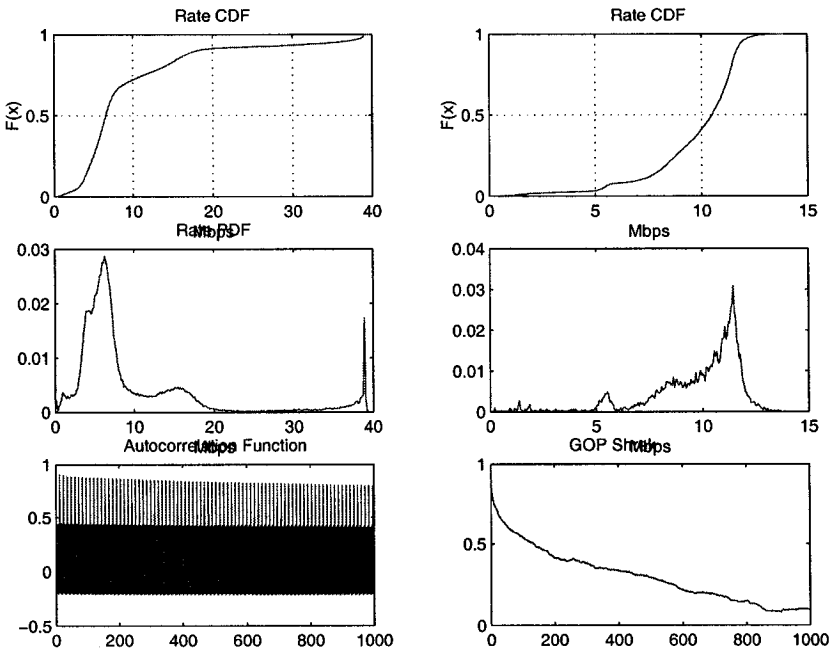

Figure A.4: Frame and GOP statistics for **Shrek**

A.2 Deterministic Smoothing of MPEG2 Traces

MPEG2 traffic can be smoothed by using a receive side buffer. To determine the amount of bandwidth that must to be allocated for a specific receiver buffer we consider that the playback process is known and it is represented by the MPEG2 data stream. This problem can be modeled as a reverse *deterministic bandwidth* allocation problem and can be solved in this framework. The *deterministic effective bandwidth* is defined to be the minimum egress capacity that has to be allocated to the traffic stream such that the maximum delay does not surpass a given value D_{max}. The amount of data $W(t)$ served by a server with egress capacity e_b is $W(t) \leq e_b(t-T)$ where T denotes the latency of the server. In probabilistic terms, this is equivalent with saying that $\mathbb{P}(\mathbb{Q} > c \times D_{max}) = 0$.

Here the *deterministic bandwidth* is the amount of bandwidth that has to be allocated between the sender and receiver, while the amount of buffer space represents the *receiver* side buffering.

The solution for this problem is given in two main steps:

- calculating the concave hull of the arrival curve. This is done by obtaining first the minimal arrival curve and then by applying one of the convex hull algorithms for the minimal arrival curve together with the time axes. Note that the concavity of the arrival curve is considered versus the y axis.

- Calculate the deterministic effective bandwidth for the arrival hull. The way to solve this is by finding the incline of the curve that is tangential to the arrival curve, given a specific amount of delay, which is represented by the intersection of the time axes by the effective bandwidth bound process.

In general, $e_b = p$ then $D = \frac{b_p}{p}$ while if $e_b = r$ then $D = \frac{b_r}{r}$. Given D we obtain e_b to be:

$$e_b = \begin{cases} r & : \quad D \geq \frac{b_r}{r} \\ p & : \quad D \leq \frac{b_p}{p} \\ \frac{pb_r - rb_p}{(b_r - b_p) + D(p-r)} & : \quad \text{otherwise} \end{cases}$$

Equation A.2 provides the *lossless effective bandwidth (LEB)* formula.

Figure A.5 show that for the frame determination the curves have three distinct regions. When choosing the optimal operating point on the curve the amount of buffer

Trace Name	Min(KB)	Max(KB)	Mean(KB)	Max/Mean
Annie Hall	11.424	552.152	352.759	1.57
Austin Powers I	67.843	548.609	248.479	2.21
Austin Powers II	36.352	622.848	297.208	2.10
Big Lebowski	11.948	543.708	240.661	2.26
The Big Blue	10.260	648.858	287.917	2.25
Blade Runner	12.140	578.772	294.913	1.96
Bugs Life	11.250	675.928	337.604	2.00
Carlitos Way	20.240	620.480	406.904	1.52
The English Patient	16.576	503.168	360.950	1.39
Final Fantasy	10.260	632.110	246.078	2.57
Happy Texas	16.958	640.098	356.717	1.79
Heat	11.160	563.248	300.532	1.87
A Life Less Ordinary	11.520	502.512	295.808	1.70
Lock Stock	11.460	647.316	276.374	2.34
Manhattan	10.261	608.349	320.931	1.90
About Mary	4.580	603.508	441.605	1.37
Matrix	11.160	555.172	292.922	1.90
Nurse Betty	36.352	563.329	313.271	1.80
OBrother Where Are	92.416	611.992	300.832	2.03
Out of Sight	20.224	575.232	424.127	1.36
Point Break	12.000	601.132	418.385	1.44
Red October	36.352	598.657	376.674	1.59
Run Lola Run	10.260	601.627	269.326	2.23
Shakespeare in Love	11.404	627.888	359.777	1.75
Shrek	3.240	625.256	440.659	1.42
Snatch	10.260	584.550	236.587	2.47
South Park	11.404	687.376	317.070	2.17
Space Balls	42.306	595.648	273.363	2.18
Strange Days	18.480	560.012	382.304	1.46
Starship Troopers	10.440	635.672	217.247	2.93
The Abyss	11.520	540.820	262.439	2.06
Three Kings	11.160	564.696	330.495	1.71
Total Recall	12.140	569.424	280.127	2.03
A Fish Called Wanda	16.576	499.649	293.970	1.70

Table A.3: Trace Group of Pictures (GOP) Statistics

Trace Name	Mean Rate(Mbps)	Max Rate(Mbps)
Annie Hall	7.055	11.04
Austin Powers I	4.970	10.97
Austin Powers II	5.944	12.46
Big Lebowski	4.813	10.87
The Big Blue	5.758	12.98
Blade Runner	5.898	11.58
Bugs Life	6.752	13.52
Carlitos Way	8.138	12.41
The English Patient	7.219	10.06
Final Fantasy	4.922	12.64
Happy Texas	7.134	12.80
Heat	6.011	11.26
A Life Less Ordinary	5.916	10.05
Lock Stock	5.527	12.95
Manhattan	6.419	12.17
About Mary	8.832	12.07
Matrix	5.858	11.10
Nurse Betty	6.265	11.27
OBrother Where Are	6.017	12.24
Out of Sight	8.483	11.50
Point Break	8.368	12.02
Red October	7.533	11.97
Run Lola Run	5.387	12.03
Shakespeare in Love	7.196	12.56
Shrek	8.813	12.51
Snatch	4.732	11.69
South Park	6.341	13.75
Space Balls	5.467	11.91
Strange Days	7.646	11.20
Starship Troopers	4.345	12.71
The Abyss	5.249	10.82
Three Kings	6.610	11.29
Total Recall	5.603	11.39
A Fish Called Wanda	5.879	9.99

Table A.4: Trace GOP Rate Statistics

Trace Name	Min(KB)	Max(MB)	Min(KB)(G)	Max(MB)(G)
Annie Hall	1.500	30.136	1.500	60.770
Austin Powers	1.500	89.897	1.500	90.906
Big Lebowski	1.500	37.265	1.500	46.906
The Big Blue	1.500	113.725	1.500	271.365
Blade Runner	1.500	24.530	1.500	35.745
Carlitos Way	1.500	43.137	1.500	62.580
The English Patient	1.500	33.892	1.500	47.527
Final Fantasy	1.500	37.455	1.500	50.295
Happy Texas	1.500	76.784	1.500	116.753
Heat	1.500	79.953	1.500	132.821
A Life Less Ordinary	1.500	22.236	1.500	71.946
Lock Stock	1.500	31.192	1.500	32.372
Manhattan	1.500	68.372	1.500	250.535
Nurse Betty	1.500	41.622	1.500	141.388
OBrother Where Are	1.500	49.241	1.500	65.488
Out of Sight	1.500	63.071	1.500	464.783
Point Break	1.500	27.232	1.500	47.156
Run Lola Run	1.500	64.701	1.500	109.474
Shrek	1.500	43.588	1.500	285.430
Snatch	1.500	40.090	1.500	69.967
Strange Days	1.500	27.910	1.500	46.517
Starship Troopers	1.500	47.039	1.500	110.832
Three Kings	1.500	53.968	1.500	138.889
Total Recall	1.500	57.387	1.500	103.162

Table A.5: Deterministic Effective Bandwidth Buffer Ranges

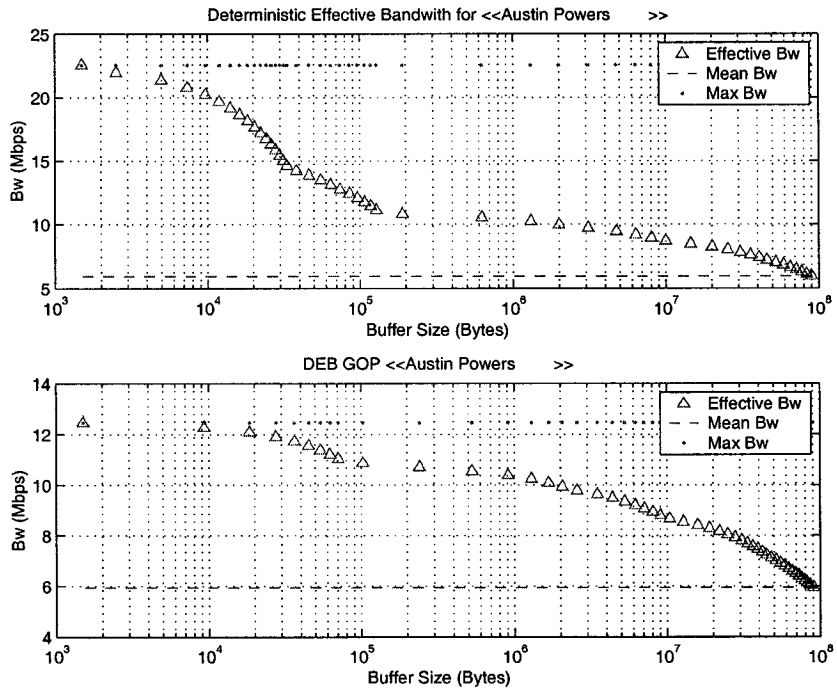

Figure A.5: Deterministic Effective Bandwidth for **Austin Powers II**

space has to be considered together with the amount of bandwidth that each trace consumes. Hence, the problem to optimize becomes:

$$\begin{aligned} \textit{minimize} \quad & f(b_{new}) + g(c_{new}) \\ s.t. \quad & U_c + c_{new} \leq C \end{aligned}$$

where U_c denotes the existing committed capacity

while C denotes the capacity of the class

$$U_b + b_{new} \leq B$$

where U_b denotes the existing committed buffer space

while B denotes the total buffer space allocated to the class

The functions f and g determine the importance of the bandwidth and the buffer in some monetary form: they represent the utility functions for these two variables. Utility functions are in general non-linear.

Table A.2 summarizes the extreme points on the deterministic bandwidth graphs. The analysis of the table reveals that the amount of buffering that is necessary in order to send at the average frame rate is rather large, in the range of 10 to 100 MB. Since this is receiver side buffering it can be easily allocated.

The amount of initial playback delay is also of importance when considering playback. For a given buffer size B that is filled at speed c this delay can be easily calculated to be $d = \frac{B}{c}$. The buffer fill up speed can be different from the rate that is allocated for the bulk of the trace. Considering an upper bound for d, which can be chosen based on the tolerance to wait of the average user, the maximum buffer space in the receiver is limited by the maximum transfer speed of the initial load.

A.3 Statistical Multiplexing Gains for MPEG2 Traffic

It this section we consider the effects of statistically multiplexing some of the traces described in this Chapter. Low resolution MPEG traces have been used in simulation of traffic for some time now (see for example [65, 94] and some of the references within). These studies prove that although it was not their main topic of research, the widely held perception that the statistical gain generated by multiplexing MPEG1 sources is large, is true. The question that we would like to answer in this section is how big the statistical multiplexing gain is for MPEG2 traces. The measure that we use for this is the peak-to-mean ratio. From this, the statistical multiplexing gain can be calculated as the ratio of the peak-to-mean measures for two different traffic mixes.

Figure A.6 shows the results of this investigation of two of the traffic traces chosen at random. It can be seen that as the number of sources increases that P2M ratio decreases. Nevertheless, the decrease is not towards 1, as expected, but to a slightly larger value.

The experiments were repeated for different start times for the sequences. It can be seen that if the start times are in a small interval, the P2M ratio does not decrease as much as if the start times are uniformly dispersed over a longer interval. Nevertheless, the multiplexing gain is significant in this case too.

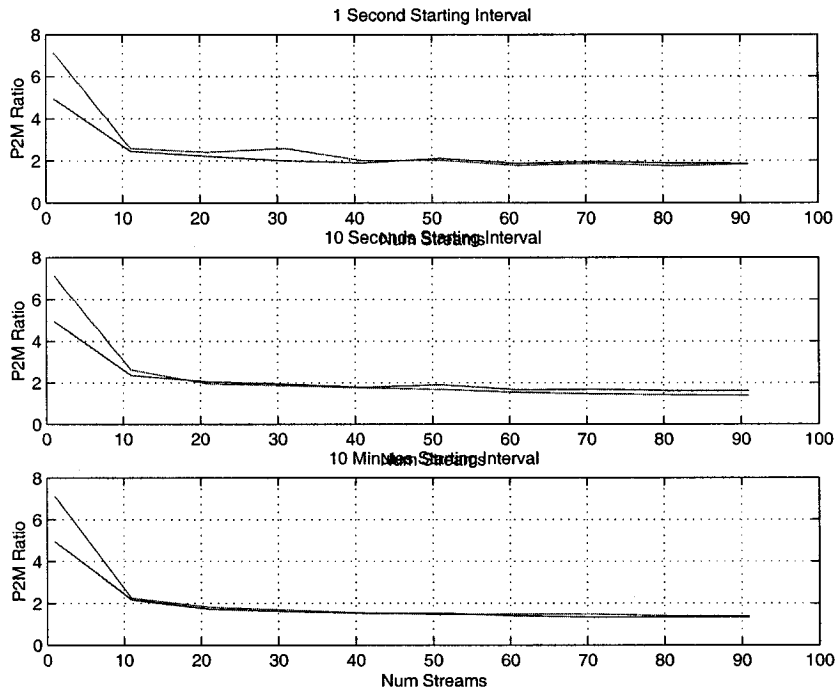

Figure A.6: Peak-to-Mean Ratio for 2 of the MPEG2 sequences

The benefits of the statistical multiplexing gain on the network are substantial. The small peak to mean ratio means that the network can work at a high utilization level. Considering that the NE buffer space is kept to a minimum in order to decrease the delays, and that $c = p_r$, the utilization becomes $\rho = \frac{1}{p2m}$. Furthermore, this also means that the benefit from smoothing is limited.

A.4 Modeling of MPEG2 traffic

The plots of the autocorrelation function of any for the traces (see Figure A.1, for example) reveal that the function has some periodicity.

This type of autocorrelation function appears in all MPEG streams that have been encoded with a deterministic encoding pattern and is valid for the MPEG1 streams such as Star Wars [86]. The period of the pattern can be obtained from the autocorrelation function by counting the distance between spikes. This determination is automated by Algorithm 8.

Algorithm 8 Period Detection (Input: MPEG2 stream x)

1: compute $r_x(k)$ the autocorrelation function for x
2: sort the values in $r_x(k)$ in descending order by key $r_x(k)$ while maintaining the original index of the entries index of $a(j) \leftarrow index(sort(r_x(k))(j)$
3: $p = a(1) - a(0)$

Applying Algorithm 8 to the Star Wars sequence, which it is known to have been encoded with an IBBPBBPBBPBB [86] frame pattern, reveals that the sequence has a periodicity of 12.

Splitting the sequence into 12 subsequences and plotting the empirical cumulative distribution functions (ECDF) for all of them shows that the Levy distance between the P substreams is very small. The same observation can be made about the P and I frame substreams. Since the averages of the three substream types are different, this implies that the sequence is *not* stationary in either a strict or wide sense.

Applying Algorithm 8 to the MPEG2 streams of this chapter shows that all have been encoded with 12 intermediate frames between the I frames.

The ECDFs of the MPEG2 streams are qualitatively different from the ECDF of the Star Wars sequence. While the latter, as represented by Figure A.7, has three distinct ECDF curves, each one corresponding to each of the frame types, there is no such separation for any of the MPEG2 sequences. On the Star Wars sequence, the Kolmogorov-Smirnoff test shows that all the B-frame sequences have closely related empirical distributions. The same is valid for the P-frame subsequences[1].

The difference can be explained by the method of encoding. While the MPEG1 sequence has been encoded as one contiguous unit, from start to finish, DVD movies are usually collated from scenes, to allow indexed jumps. As scenes do not always finish at GOP intervals, and since a new scene has to start with an I frame, DVD MPEG2 sequences have a quantitatively different structure than the MPEG1 sequence described above. Although the autocorrelation function shows that all of MPEG2 sequences have some statistical regularity, it is of a different nature than the one that is experienced in the MPEG1 sequences. Therefore, to obtain a model that closely resembles this type of

[1]The KS test has run on all combinations of substreams. If the null hypothesis is false, a black square is displayed in the results matrix, while if the null hypothesis might be true then white square is displayed.

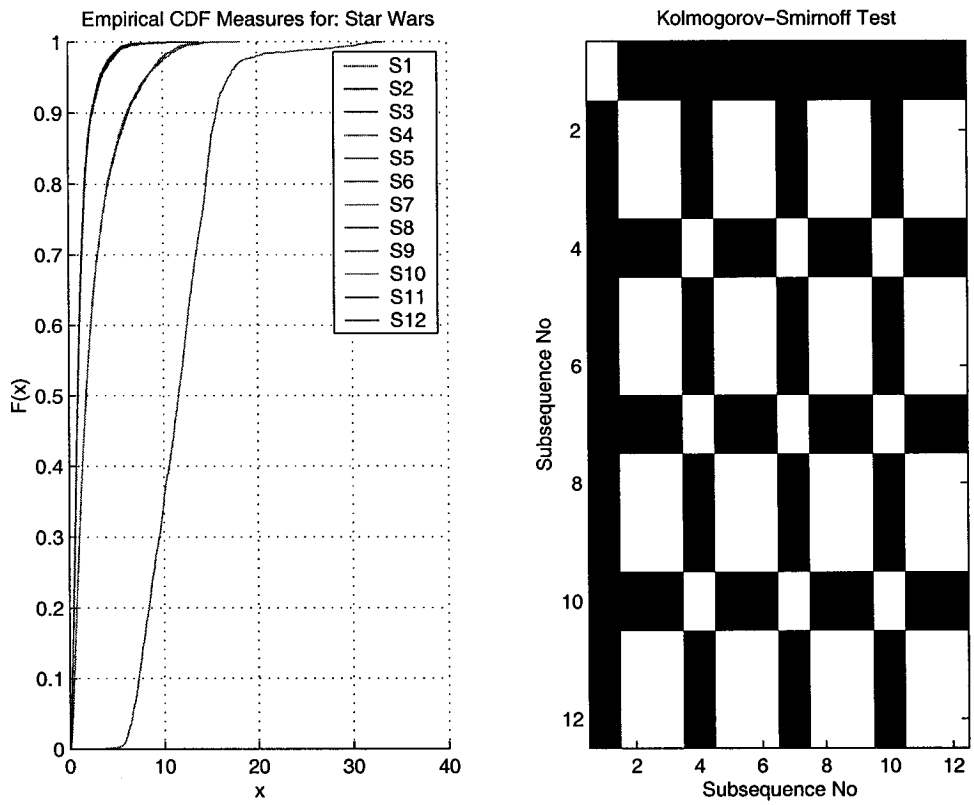

Figure A.7: Empirical CDF Functions for the Star Wars Sequence. All 12 subsequences are displayed, however due to the closeness of the distributions for each class of frames only three curves are visible

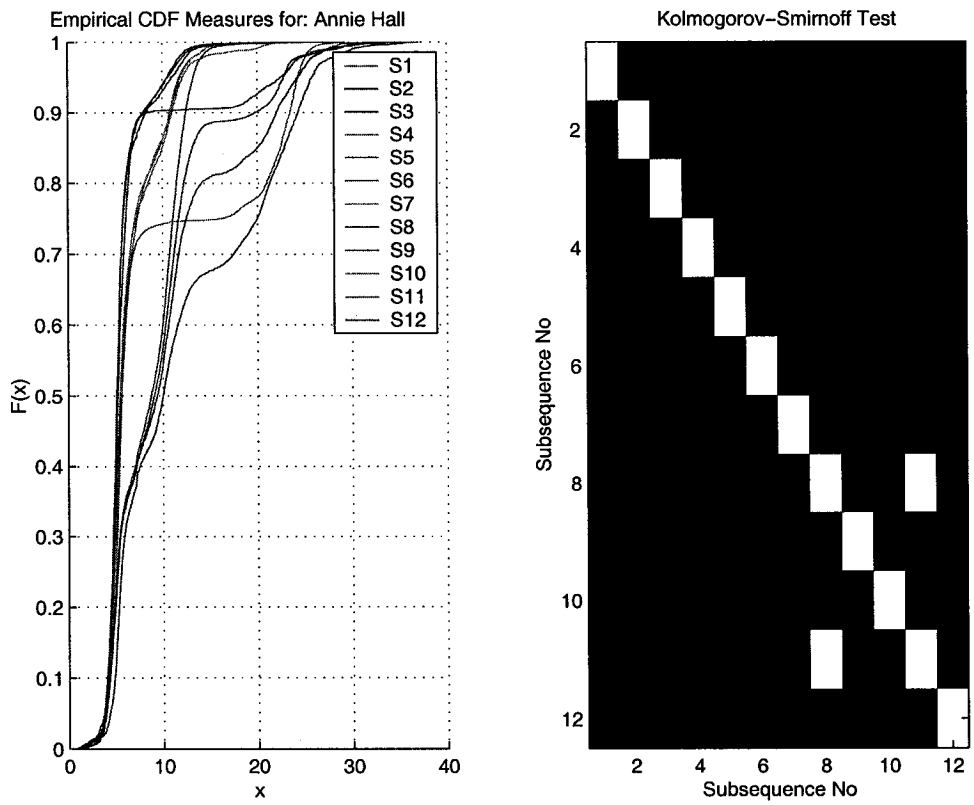

Figure A.8: Empirical CDF Functions for *Annie Hall.* The stream has been divided into 12 sub-streams and the empirical CDF of each is displayed. The right hand side figure displays the results of the Kolmogorov-Smirnoff test for pairs of the sub-streams.

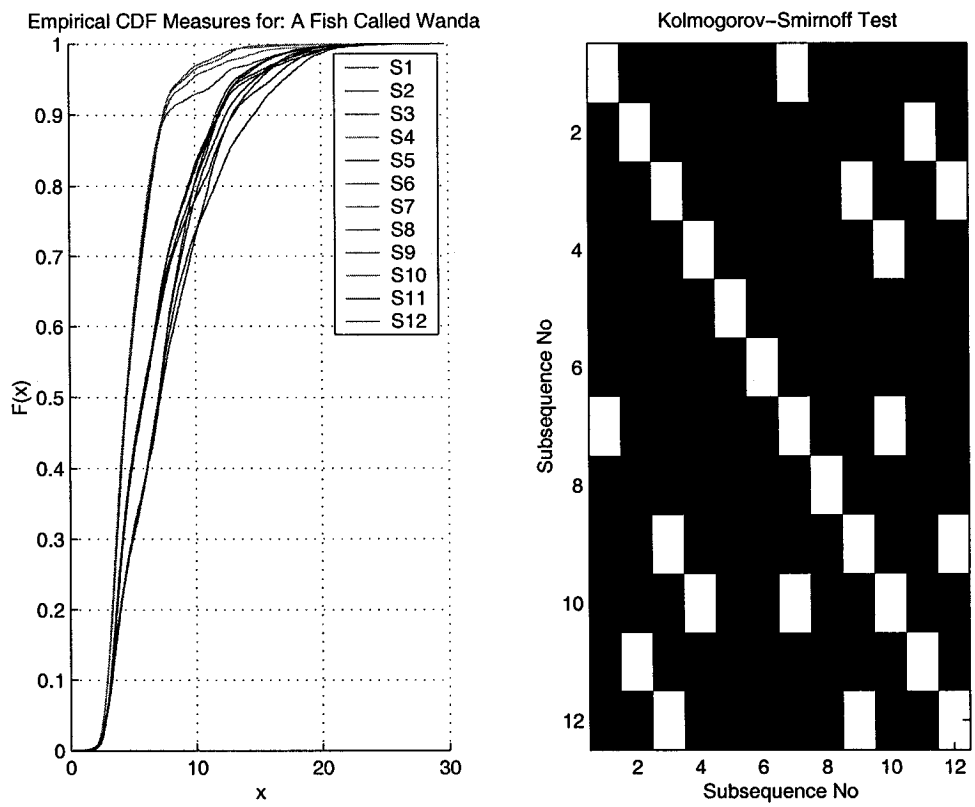

Figure A.9: Empirical CDF Functions for *A Fish Called Wanda.* The stream has been divided into 12 sub-streams and the empirical CDF of each is displayed. The right hand side figure displays the results of the Kolmogorov-Smirnoff test for pairs of the sub-streams.

sequences, a periodic model (for example a periodic Markov chain) has to be composed with statistical models for each of the sub-streams.

Bibliography

[1] D. Anick, D. Mitra, and M.M. Sodhi. Stochastic theory of a data handling system with multiple sources. *Bell Sys. Tech Journal*, 61(8), 1982.

[2] R.R. Bahadur and R.R. Rao. On deviations of the sample mean. *Ann. Of Mathematical Statistics*, (23):1015–1027, 1960.

[3] L. Benmohamed and S. M. Meerkov. Feedback Control of Congestion In Packet Switching Networks: The case of a single congested node. *IEEE/ACM Transactions in Networking*, 1:693–708, 1993.

[4] J. Beran. *Statistics for Long-Memory Processes.* Chapman-Hall, 1994.

[5] A.W. Berger and W. Witt. Effective bandwidths with priorities. *IEEE/ACM Transactions on Networking*, 6(4), 1998.

[6] D. Bertsimas, I. Paschiliadis, and J. Tsitsiklis. Large deviations analysis of the generalized processor sharing policy. *Queuing Systems*, 32(1), 1999.

[7] G. Bianchi and all. Throughput Analysis of End-to-End Measurement Based Admission Control. In *INFOCOM*, 2000.

[8] F. Blanchini, R.L. Cigno, and R. Tempo. Robust Rate Control for Integrated Service Packet Networks. *IEEE/ACM Transactions in Networking*, 5:644–652, 2002.

[9] A. Botvitch and N.G. Duffield. Large deviations, economies of scale and the shape of the loss curve in large multiplexers. *Queueing Systems*, (23):293–320, 1995.

[10] Lee Breslau and Sugih Jamin. Comments on the performance of measurement-based admission control algorithms. In *INFOCOM'2000*, 2000.

[11] M. Carugi and D. McDysan. Service requirements for Layer 3 Provider Provisioned Virtual Private Networks. Technical report, www.ietf.org, 2002.

[12] Coskun Cetinkaya and Edward W. Knightly. Egress admission control. In *INFOCOM'2000*, 2000.

[13] C.S. Chang. Sample path large deviations and intree networks. *Queueing Systems*, 20(1), 1995.

[14] C.S. Chang. *Performance Guarantees In Communication Networks.* Springer-Verlag, 2000.

[15] A. Charny and J.L. Boudec. Delay bounds in network with aggregated scheduling. In *QOFIS*, 2000.

[16] S. Chen. *Routing Support For Providing Guaranteed End-to-End Quality of Service.* PhD thesis, University of Illinois at Urbana-Champaign, 1999.

[17] I. Chlamtac, A. Farago, and T. Zhang. Optimizing the System of Virtual Paths. *IEEE/ACM Transactions in Networking*, 2:581–587, 1994.

[18] G.L. Choudhury, D. M. Lucantoni, and W. Whitt. Squeezing the most out of atm. *IEEE Transactions on Communications*, 44(2), 1996.

[19] Ming-Chwan Chow. *Understanding SONET/SDH.* Andan Publisher, 1995.

[20] The ATM Forum Technical Committee. LAN Emulation over ATM Version 1.0. Technical report, ATM Forum, http://www.atmforum.org, 1995.

[21] The ATM Forum Technical Committee. Private Network-Network Interface Specification 1.0. Technical report, ATM Forum, http://www.atmforum.org, 1999.

[22] The ATM Forum Technical Committee. Traffic Management Specification Version 4.1. Technical report, ATM Forum, http://www.atmforum.org, 1999.

[23] R. M. Corless. Material on the Lambert W Function. Technical report, http://www.apmaths.uwo.ca/ rcorless/frames/PAPERS/LambertW/, 2002.

[24] C. Courcoubetis, A. Dimakis, and G. Stamoulis. Traffic equivalence and substitution in a multiplexer with applications to dynamic available capacity estimation. *IEEE/ACM Transactions on Networking*, (2):217–231, 2002.

[25] C. Courcoubetis, F. Kelly, and R. Weber. Measurement based charges in communications networks. *Operations Research*, 48(1), 2000.

[26] C. Courcoubetis and R. Weber. Buffer Overflow asymptotics for a buffer handling many traffic sources. 1995.

[27] S. Crosby. Statistical properties of a near-optimal measurement-based cac algorithm. In *Proceedings of the IEEE ATM '97 Workshop, Lisboa, Portugal, 1997.*, 1997.

[28] R.L Cruz. A calculus for network delay, part i: Network elements in isolation. *IEEE Transactions in Information Theory*, 37:114–131, 1991.

[29] R.L Cruz. A calculus for network delay, part ii: Network analysis. *IEEE Transactions in Information Theory*, 37:132–141, 1991.

[30] G. de Veciana and G. Kesidis. Bandwidth allocation for multiple qualities of service using gps. *IEEE/ACM Transactions on Information Theory*, (1):268–278, 1996.

[31] A. Dembo and O. Zeitouni. *Large Deviation Techniques and Applications.* Springer-Verlag, 1997.

[32] A. Demers, S.Keshlav, and S.Shenker. Analysis and simulation of a fair queuing algorithm. *Journal of Interworking Research and Experience*, pages 3–26, 1990.

[33] N. Duffield and all. Entropy of atm traffic streams: a tool for estimating qos parameters. *IEEE Journal of Selected Areas in Communications*, 13(6):981–990, 1995.

[34] N. G. Duffield. Asymptotic sampling properties of effective bandwidth estimation for admission control. In *Infocom*, 1999.

[35] N. G. Duffield, Pawan Goyal, Albert Greenberg, Partho Mishra, K. K. Ramakrishnan, and Jacobus E. van der Merwe. A flexible model for resource management in virtual private networks. In *ACM SIGCOMM'99*, 1999.

[36] Z. Dziong, M. Jurda, and L.G. Mason. A framework for bandwidth management in atm networks-aggregate equivalent bandwidth estimation approach. *IEEE Transactions in networking*, 5(1), 1997.

[37] M. El-Darieby, J. Rolia, and D.C. Petriu. Performance Modeling for Virtual Network-based Service Provisioning . In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management IM'2001*, 2001.

[38] A. Elwalid and D. Mitra. Effective bandwidth of general Markovian traffic sources and admission control of high speed networks. *IEEE Transactions in Networking*, 3:329–343, 1993.

[39] A. Farago, S. Blaanjerg, L. Ast, G. Gordos, and T. Henk. A New Degree of Freedom in ATM Network Dimensioning: Optimizing the Logical Configuration. *IEEE/ACM Transactions in Networking*, 13:1199–1205, 1995.

[40] F.H.P. Fitzek and M. Reisslein. MPEG-4 and H.263 Video Traces for Network Performance. Technical report, University of Berlin, 2001.

[41] S. Floyd. Comments on measurement-based admissions control for controlled-load services. Technical report, Lawrence Livermore Laboratory, 1996.

[42] TeleManagement Forum. SLA Management Handbook. Technical report, www.tmforum.org, 2001.

[43] G.F. Franklin, J.D. Powell, and M.L. Workman. *Digital Control of Dynamic Systems.* Addison-Wesley, 1990.

[44] R.J. Gibbens and F.P. Kelly. A decision theoretic approach to call admission control in atm networks. *IEEE JSAC*, 13(1), 1995.

[45] B. Gleeson and all. *A Framework for IP Based Virtual Private Networks.* Internet RFC 2764, 2000.

[46] M. Grossglauser and N.C. Tse. A framework for robust measurement based admission control. *IEEE Transactions in Networking*, 7(3), 1999.

[47] R. Guerin and all. Equivalent capacity and its application to bandwidth allocation in high speed networks. *IEEE JSAC*, 9:968–981, 1991.

[48] A. Gupta, A. Kumar, J. Kleinberg, R. Rastogi, and B. Yener. Provisioning a virtual private network: A network design problem for multicommodity flow . In *ACM STOC 2001*, 2001.

[49] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. RFC 2597: Assured Forwarding PHB Group. Technical report, www.ietf.org, 1999.

[50] D. Heyman and T. Lakshman. What are the implications of long-range dependence for vbr traffic engineering. *IEEE/ACM Transactions in Networking*, 4:301–317, 1996.

[51] C. Hollot, V. Misra, D. Towsley, and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *INFOCOM 2001*, 2001.

[52] G. F. Italiano, R. Rastogi, and B. Yener. Restoration Algorithms for Virtual Private Networks in the Hose Model. In *Infocom 2002*, 2002.

[53] ITU-T. Terms and definitions related to quality of service and network performance including dependability. Technical report, www.itu.ch, 1994.

[54] ITU-T. E.410 - International Network Management - General Information. Technical report, www.itu.ch, 1998.

[55] ITU-T. E.416 - Network Management Principles and Practice for B-ISDN Traffic. Technical report, www.itu.ch, 2000.

[56] ITU-T. Network-based IP VPN over MPLS architecture. Technical report, www.itu.ch, 2000.

[57] ITU-T. Traffic control and congestion control in B-ISDN. Technical report, www.itu.ch, 2000.

[58] ITU-T. E.417 - Framework for Network Management of IP based Networks. Technical report, www.itu.ch, 2001.

[59] ITU-T. Y.1311 - Network-based IP VPN over MPLS architecture. Technical report, www.itu.ch, 2001.

[60] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated services packet networks. In *SIGCOMM'95*, pages 2–13, 1995.

[61] F. Kelly. Effective bandwidths at multi-class queues. *Queuing Systems*, 9:5–16, 1991.

[62] Frank Kelly. Notes on effective bandwidths. In *Stochastic Networks: Theory and Applications*, volume Stochastic Networks: Theory and Applications, pages 141–168. Oxford University Press, 1996.

[63] H. S. Kim and N. B. Shroff. Loss probability calculations and asymptotic analysis for finite buffer multiplexers. *IEEE/ACM Transactions in Networking*, 9:755–767, 2001.

[64] Y. Kim and San qi Li. Timescale of interest in traffic measurement for link bandwidth allocation. In *Infocom*, pages 738–748, 1996.

[65] E. W. Knightly and N. Shroff. Admission control for statistical qos: Theory and practice. *IEEE Network*, 2:20–29, 1999.

[66] A. Koralov and G. Ramamurthy. A Control-Theoretic Approach to the Design of an Explicit Rate Controller for ABR Service. *IEEE/ACM Transactions in Networking*, 7:741–753, 1999.

[67] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener. Algorithms for provisioning virtual private networks in the hose model . In *ACM SIGCOMM 2001*, 2001.

[68] K. Kumar, G. Margrave, and D. Mitra. Novel Techniques for Design and Control of Generalized Processor Sharing Schedulers for Multiple QoS Classes. In *INFOCOM 2000*, 2000.

[69] Krishanan Kumaran, Geoffrey E. Margrave, Debasis Mitra, and Keith R. Stanley. Novel techniques for the design and control of generalized processor sharing schedulers for multiple qos classes. In *INFOCOM'2000*, 2000.

[70] Yuan-Cheng Lai and Sheng-Fu Tsai. Some fair measurement-based admission controls. In *GLOBECOM*, pages 2566–2570, 2001.

[71] J.L. LeBoudec. *Network Calculus.* Springer-Verlag-LNCS 2050, 2001.

[72] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[73] N. Likhanov and R. R. Mazumdar. Cell loss asymptotics in buffer fed with a large number of independent stationary sources. In *INFOCOM 1998*, 1998.

[74] M. May, C. Diot, B. Lyles, and J. Bolot. Influence of Active Queue Management Parameters on Aggregate Traffic Performance. Technical report, N-3995-INRIA, 2000.

[75] D. Mitra, J.A. Morrison, and K.G. Ramakrishnam. ATM Network Design and Optimization a Multirate Loss Network Framework. *IEEE/ACM Transactions in Networking*, 4:531–543, 1996.

[76] D. Mitra, J.A. Morrison, and K.G. Ramakrishnan. Virtual private networks: Joint resource allocation and routing design. In *Infocom*, 1999.

[77] M. Montgomery and G. de Veciana. On the relevance of time scales in performance oriented traffic characterizations. In *Infocom*, 1996.

[78] J. Moy. *OSPF Version 2.* IETF RFC 2328, 1998.

[79] D. Nandita, J. Kuri, and H.S. Jamadagni. Optimal call admission control in generalized processor sharing (gps) schedulers. In *Infocom*, 2001.

[80] N. O'Connell. Large deviations for Queue Lengths at a multi-buffered resource. Technical report, HP Laboratories Bristol, 1996.

[81] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling tcp throughput: A simple model and its empirical validation. In *ACM Sigcomm*, 1998.

[82] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Transactions on Networking*, 1:137–150, 1993.

[83] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2:347–357, 1993.

[84] J. Qiu and E.W. Knightly. Measurement-based admission control with aggregate traffic envelopes. *IEEE Transactions in Networking*, 9(2), 2001.

[85] James Roberts and all. (eds). *Broadband Network Teletraffic - Performance Evaluation and Design of Broadband Multiservice Networks.* Springer-Verlag LNCS 1155, 1996.

[86] O. Rose. Statistical Properties of MPEG video traffic and their impact on traffic modeling in ATM systems. Technical report, University of Wuerzburg, 1995.

[87] E. Rosen and all. *MPLS Label Stack Encoding.* Internet RFC 3032, 2001.

[88] E. Rosen and Y. Rekhter. *BGP/MPLS VPNs.* www.ietf.org, 1999.

[89] B. Ryu and A. Elwalid. The importance of long-range dependence of vbr video traffic in atm traffic engineering: Myths and realities. *Computer Communications Rev.*, 26:3–14, 1996.

[90] A. Sang and San qi Li. A predictability analysis of network traffic. In *Infocom*, 2000.

[91] Y. Serberst and Sang gi Li. Unified measurement functions for traffic aggregation and link capacity assessment. In *INFOCOM*, pages 1522–1531, 1999.

[92] A.N. Shiryaev. *Probability.* Springer-Verlag, 2000.

[93] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis.* Chapman-Hall, 1995.

[94] V. Siris. *Performance Analysis and Pricing in Broadband Networks.* PhD thesis, University of Crete, 1997.

[95] T.E. Stern and A.I.Elwalid. Analysis of a separable markov modulated rate models for information handling systems. *Adv. Applied Prob.*, 23:105–139, 1992.

[96] W. Stevens. *TCP/IP Illustrated - vol 1.* Addison-Wesley, 1994.

[97] R. Szabo, P. Barta, F. Nemeth, J. Biro, and C-G Perntz. Call admission control in generalized processor sharing(gps) schedulers using non-rate proportional weighting of sessions. In *INFOCOM'2000*, 2000.

[98] D. Tse and M. Grosslauser. Measurement-based call admission control: Analysis and simulation. In *INFOCOM'97*, pages 981–989, 1997.

[99] D.N.C. Tse, R.G. Gallager, and J.N. Tsitsiklis. Statistical multiplexing of multiple time scale markov streams. *IEEE/ACM Transactions in Networking*, 1(6), 1995.

[100] B. Tsybakov and N. Georganas. On Self Similar Traffic in ATM Queues: Definitions, Overflow probability bound and cell delay distribution. *IEEE/ACM Transactions in Networking*, 5:397–409, 1997.

[101] B. Tsybakov and N. Georganas. Overflow Probability in an ATM Queue with Self-Similar Traffic . In *ICC'97 Montreal*, 1997.

[102] A. Weiss. A new technique of analyzing large traffic systems. *Advances in Applied Probability*, 18:506–532, 1986.

[103] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks . *IEEE/ACM Transactions in Networking*, 5:1374–1396, 1995.

[104] H. Zhang and D. Ferrari. Rate-controlled service disciplines. *Journal of High Speed Networks*, 3(4):389–412, 1994.

[105] Z.L. Zhang. Large deviations and the generalized processor sharing scheduling in a two queue system. *Queueing Systems*, 26(1), 1997.

[106] Z.L. Zhang. Large deviations and the generalized processor sharing scheduling in a multiple queue system. *Queueing Systems*, 28(1), 1998.

[107] Y. Zhao, S. Li, and S. Sigarto. A Linear Dynamic Model for Design of Stable Explicit-Rate ABR Control Schemes. Technical report, Department of ECE, University of Texas at Austin, 1996.

[108] Z.L.Zhang, Z.Liu, J. Kurose, and D. Towsley. Call Admission Control Schemes under the GPS scheduling discipline. Technical report, UM-CS-95-52 University of Massachusetts, 1995.