Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

NJIT
New Jersey's Science &
Technology University

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

IP-BASED VIRTUAL PRIVATE NETWORKS AND PROPORTIONAL QUALITY OF SERVICE DIFFERENTIATION

by
Jingdi Zeng

IP-based virtual private networks (VPNs) have the potential of delivering cost-effective, secure, and private network-like services. Having surveyed current enabling techniques, an overall picture of IP VPN implementations is presented.

In order to provision the equivalent quality of service (QoS) of legacy connection-oriented layer 2 VPNs (e.g., Frame Relay and ATM), IP VPNs have to overcome the intrinsically best effort characteristics of the Internet. Subsequently, a hierarchical QoS guarantee framework for IP VPNs is proposed, stitching together development progresses from recent research and engineering work.

To differentiate IP VPN QoS, the proportional QoS differentiation model, whose QoS specification granularity compromises that of IntServ and DiffServ, emerges as a potential solution. The investigation of its claimed capability of providing the predictable and controllable QoS differentiation is then conducted.

With respect to the loss rate differentiation, the "packet shortage" phenomenon shown in two classical proportional loss rate (PLR) dropping schemes is studied. On the pursuit of a feasible solution, the potential of compromising the system resource, that is, the buffer, is ruled out; instead, an enhanced "debt-aware" mechanism is suggested to relieve the negative effects of "packet shortage." Simulation results show that "debt-aware" partially curbs the biased loss rate ratios, and improves the queueing delay performance as well.

With respect to the delay differentiation, the dynamic behavior of the average delay difference between successive classes is first analyzed, aiming to gain insights of system dynamics. Then, two classical delay differentiation mechanisms, that is,

proportional average delay (PAD) and waiting time priority (WTP), are simulated and discussed. Based on observations on their differentiation performances over both short and long time periods, a combined delay differentiation (CDD) scheme is introduced. Simulations are utilized to validate this method.

Both loss and delay differentiations are based on a series of differentiation parameters. Though previous work on the selection of delay differentiation parameters has been presented, that of loss differentiation parameters mostly relied on network operators' experience. A quantitative guideline, based on the principles of queueing and optimization, is then proposed to compute loss differentiation parameters. Aside from analysis, the new approach is substantiated by numerical results.

IP-BASED VIRTUAL PRIVATE NETWORKS AND PROPORTIONAL QUALITY OF SERVICE DIFFERENTIATION

by
Jingdi Zeng

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

January 2004

Copyright © 2004 by Jingdi Zeng

ALL RIGHTS RESERVED

APPROVAL PAGE

IP-BASED VIRTUAL PRIVATE NETWORKS AND PROPORTIONAL QUALITY OF SERVICE DIFFERENTIATION

Jingdi Zeng

Dr. Nirwan Ansari, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Jianguo Chen, Committee Member — Date
S.M.T.S., Agere Systems Inc. (formerly Lucent Bell Laboratories)

Dr. Edwin Hou, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Sirin Tekinay, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Lev Zakrevski, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Jingdi Zeng

Degree: Doctor of Philosophy

Date: January 2004

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, USA, 2004
- Master of Science in Computer Applications,
  Hunan University, Changsha, Hunan, China, 1998
- Bachelor of Engineering in Communications Engineering,
  Hunan University, Changsha, Hunan, China, 1995

Major: Electrical Engineering

Presentations and Publications:

Jingdi Zeng, Lev Zakrevski, and Nirwan Ansari, "Computing loss differentiation parameters for the proportional differentiation service model," *submitted to IEEE Communications Letters*, November 2003.

Jingdi Zeng and Nirwan Ansari, "Alleviating the "packet shortage" phenomenon in proportional differentiated services," *submitted to IEE Proceedings-Communications*, September 2003.

Jingdi Zeng and Nirwan Ansari, "On the performance of the proportional delay differentiation," *in Proc. IEEE High Performance Switching and Routing (HPSR) 2003*, Torino, Italy, June 2003.

Jingdi Zeng and Nirwan Ansari, "An enhanced dropping scheme for proportional differentiated service," *in Proc. IEEE International Conference on Communications (ICC) 2003*, Alaska, USA, May 2003.

Jingdi Zeng and Nirwan Ansari, "Toward IP virtual private network quality of service: a service provider perspective," *IEEE Communications Magazine*, vol. 41, no. 4, pp. 113-119, April 2003.

Jingdi Zeng and Nirwan Ansari, "Virtual queue occupancy and its applications on periodic bandwidth on demand schemes for IP/SONET," *IEICE Transactions on Communications*, vol. E85-B, no. 9, pp. 1749-1755, September 2002.

Jingdi Zeng and Nirwan Ansari, "Periodic bandwidth allocation based on virtual queue occupancy," *in Proc. IEEE International Conference on Communications (ICC) 2002*, New York, USA, April 2002.

To my parents, for believing in me and encouraging me.

给亲爱的爸爸和妈妈.

ACKNOWLEDGMENT

I am most appreciative of Prof. Nirwan Ansari, my doctoral advisor, who not only taught me the right attitude toward research and work, but also showed me by his example how powerful in one's life an understanding heart can be. Without him, this dissertation work could not have been accomplished.

I also wish to especially thank all other members of my dissertation committee, for the valuable inspiration from Prof. Hou and Prof. Tekinay in their courses, for the outstanding alumnus example set by Dr. Chen, for the excellent research collaboration with Prof. Zakrevski.

Special thanks go to Mr. Jeffrey Grundy, the Director of the Office for International Students and Faculty (OISF). Without him and Prof. Nirwan Ansari, I would not have had the experience in Germany that has enriched my life. Dr. Hermann Bischl in German Aerospace Center (DLR), Oberpfaffenhofen, also deserves my sincere appreciation, for his guidance and supervision.

Many thanks are extended to Dr. Ronald Kane, the Dean of Graduate Studies and the advisor of Graduate Student Association (GSA), for his warm-hearted support when I served as the treasurer of GSA.

As always, I feel extremely fortunate to have my mother who has the magic to lighten me whenever I feel down, and my father who gives me guidelines at every turning point of my life. They believe in me so much that I can never quit.

Finally, I wish to say thanks to my friends through all these years in New Jersey. Every minute we spent together in school, in New York City, and in other places was the source of much happiness and many memories. I am grateful to be among them.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

[1]The omitted values in the table do not affect the drawn conclusions.

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF SYMBOLS

AF:	assured forwarding
AH:	authentication header
ARP:	address resolution protocol
AToM:	any transport over MPLS
ATM:	asynchronous transfer mode
BE:	best effort
CA:	certificate authority
CAC:	call admission control
CAR:	committed access rate
CDF:	cumulative density function
CHAP:	challenge-handshake authentication protocol
CLI:	command line interface
COPS:	common open policy service
CORBA:	common object request broker architecture
CPE:	customer promise edge
CRL:	certificate revocation list
CR-LDP:	constraint-routing label distribution protocol
DES:	data encryption standard
3DES:	triple DES
DS:	differentiated service
EF:	explicit forwarding
ESP:	encapsulating security payload
FEC:	forwarding equivalence class
FIFO:	first in first out
GRE:	generic routing encapsulation

LIST OF SYMBOLS

(Continued)

GTS:	generic traffic shaping
HMAC:	hash-based message authentication code
IEEE:	Institute of Electrical and Electronics Engineers
IETF:	Internet engineering task force
IKE:	Internet key exchange
IP:	Internet protocol
IPSec:	IP security
ISAKMP:	Internet security association key management protocol
ISP:	Internet service provider
L2F:	layer 2 forwarding
L2TP:	layer 2 tunneling protocol
L2TPv3:	L2TP version 3
LAN:	local area network
LAC:	L2TP access concentrator
LDAP:	light directory access protocol
LDP:	label distribution protocol
LLA:	logical layer architecture
LNS:	L2TP network server
MD5:	message digest 5
MIB:	management information base
MPLS:	multi-protocol label switching
NMS:	network management system
OSS:	operation support system
PAP:	password authentication protocol

LIST OF SYMBOLS

(Continued)

PBN:	policy based network
PDF:	probability density function
PDP:	policy decision point
PE:	provider edge
PEP:	policy enforcement point
PHB:	per hop behavior
PKI:	public key infrastructure
POP:	point of presence
PPP:	point-to-point protocol
PPTP:	point-to-point tunneling protocol
QoS:	quality of service
RADIUS:	remote access dial-in user service
RAS:	remote access server
RSVP:	resource reservation protocol
SA:	security association
SHA-1:	secure hash algorithm 1
SLA:	service level agreement
SMI:	structure of management information
SMS:	service management system
SNMP:	simple network management protocol
SP:	service provider
SPI:	security parameter index
SSH:	secure shell protocol
SSL:	secure sockets layer

LIST OF SYMBOLS

(Continued)

SSP:	secure shell protocol
TINA:	telecommunications information networking architecture
TMN:	telecommunication management network
ToS:	type of service
VoIP:	voice over IP
VPN:	virtual private network
WAN:	wide area network

CHAPTER 1

INTRODUCTION

Electronic commerce has obtained its means of spreading out to every corner of the world via the Internet, which is reaching more homes and offices than ever. Aside from in-house functionalities such as database management and Intranet firewall, companies are starting to rely prominently on the Internet to bring together remote employees, branch offices, and customers.

Virtual private networks (VPNs), as interpreted by the name, aim to deliver information among multiple parties over a shared infrastructure (e.g., the Internet) with the private network-like manner: the same policies of the security, reliability, manageability, and quality of service (QoS). They become an effective solution for today's e-business applications, making access to the network worldwide available while protecting the information that flows across it.

1.1 IP-based Virtual Private Networks

Propelled by industry vendors, standard bodies, and research communities, the migration to todays' VPNs can be traced back to frame relay and asynchronous transfer mode (ATM) [1]. Frame relay's success came with the price of variable QoS performances and the complexity of integrating with customer devices. ATM, with QoS and the high speed provisioning, was designed to provide a full range of multimedia services. Unfortunately, ATM is simply too complex to be widely used for enterprise network applications. Internet protocol (IP)-based VPNs shed the light in the end, showing advantages in cost, flexibility (leased line and frame relay networks require capacity specifications in advance), security, and the ability to exploit existing frame relay/ATM investments.

Regarding application scenarios, VPNs can be broadly classified into remote access VPNs and LAN-to-LAN VPNs. As depicted in Fig. 1.1, dial-up or broadband access enables a remote user to connect through its local Internet service provider (ISP) point of presence (POP) to the headquarters' LAN. Another type of the remote access VPN allows Ethernet users to equally connect to the headquarters' LAN. The LAN-to-LAN VPN involves Intranet VPNs that bring together geographically separated branch offices, while Extranet VPNs enable business partners and external vendors to access specific portions of the headquarters' network.

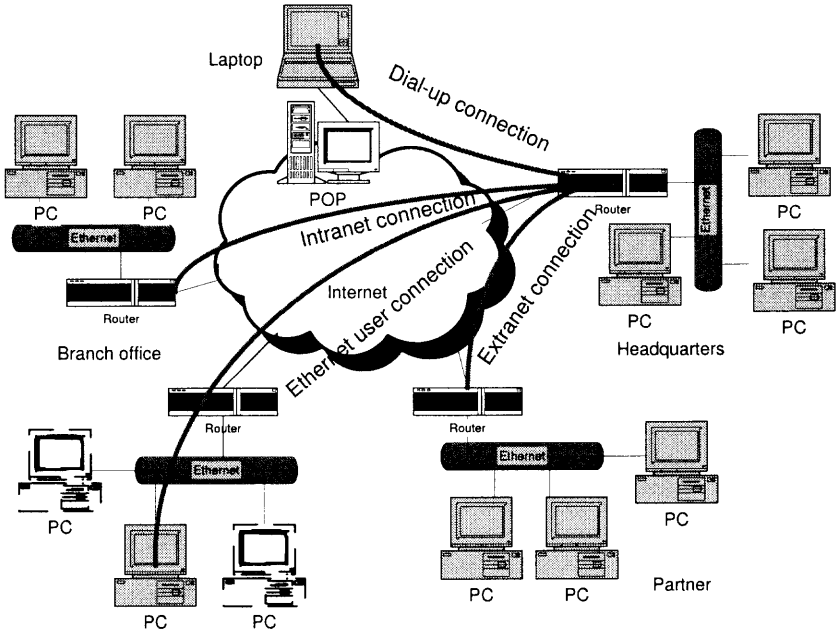


Figure 1.1 VPN application scenarios.

Compared to layer 2 strategies (e.g., frame relay and ATM), IP VPNs inherit the flexibility and simplicity of connectionless IP networks. Utilizing IP VPNs can save users more than 50 percent of the connectivity cost over the corresponding frame relay deployment [2]. However, the Internet is a two-edged sword: its ubiquitous feature offers VPNs more potential to grow, and yet, it is not the "right" network to support QoS, owing to its intrinsically best effort characteristics. What are called for, therefore, are standards-based, appropriate QoS mechanisms for IP VPNs. The rest

of the chapter justifies the rationale of adopting the proportional QoS differentiation for IP VPN QoS, and surveys existing proportional QoS differentiation mechanisms.

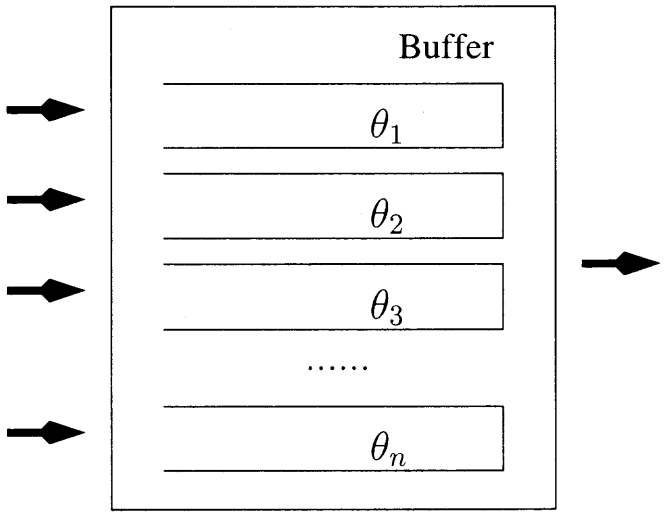


Figure 1.2 Proportional QoS differentiation.

1.2 Proportional Quality of Service Differentiation and IP VPNs

There are broadly two categories of decision-making methods for Internet QoS operations. Preset parameter schemes require *a priori* knowledge about the characteristics of all traffic streams, such as the sustained rate, peak rate, or burst length, and how they interact with each other. The measurement-based scheme makes on-line decisions by measured data, not by the statistical analysis and estimation. With the booming of Internet services and changing of users' surfing patterns, nevertheless, a traffic model, which is used by preset parameter schemes, requires significant preciseness and may not be effective. Therefore, with more certainty, measurement-based approaches become rather appealing.

Compared to IntServ, DiffServ defines a relative QoS architecture for scalable service differentiation in the Internet. It achieves scalability by implementing the classification function only at network boundary nodes, and by applying per hop behaviors (PHBs) to traffic aggregates which have been marked using the differentiated

services (DS) field in IP headers [3]. The shift from the individual packet flow-oriented, absolute QoS IntServ to traffic aggregate-oriented, relative QoS DiffServ model has had notable effects on QoS mechanisms.

To provision QoS, a buffer/queue handles delay and delay jitter by scheduling strategies, and guarantees packet loss by dropping mechanisms. The dropping mechanism basically defines when and which packets will be dropped; the scheduling strategy decides which packet will be served next. Several measurement-based scheduling and dropping strategies were proposed [4, 5, 6, 7, 8] for the IntServ model which represents QoS metrics as explicit loss ratio, delay, etc. Being measurement-based and traffic aggregate-oriented, moreover, a tailored relative service model, called proportional differentiated model, was suggested [9] for controllable, predictable, and relatively differentiated services.

As illustrated in Fig. 1.2, the proportional differentiation model groups the network traffic into n classes whose services are ordered, such that class i is better or at least no worse than class $i-1$ for $1 < i \leq n$, in terms of per-hop QoS metrics such as queueing delay and packet loss. By denoting the per hop local QoS metrics and the differentiation parameters as Θ_i and θ_i, respectively, the proportional differentiation strategy is stated as

$$\frac{\Theta_i}{\Theta_j} = \frac{\theta_i}{\theta_j}, 1 \leq i, j \leq n, \tag{1.1}$$

There were other differentiation mechanisms in the literature. The capacity based differentiation, for example, wighted fair queueing (WFQ), supplies classes with bandwidth shares relative to their traffic loads; its relative QoS differentiation varies with the traffic load. Priority mechanisms differentiate services consistently, but QoS differences between classes cannot be easily adjusted once the priority is set up. Considering these limitations, the ultimate goal [9] of the proportional differentiation

model is two-fold: predictability ensures that the QoS differentiation is consistent; controllability guarantees the adjustable spacing between service classes.

For the QoS architecture of IP VPNs [10], the proportional differentiation service model emerges as a potential solution. First, naive or casual VPN users could use more service definition "accuracy" than expedited forwarding (EF), assured forwarding (AF), and best effort (BE) classes specified by DiffServ, as long as extra in-house expertise is not required. With its differentiation factors, the proportional model compromises the accuracy and simplicity of IntServ and DiffServ QoS differentiations. Second, given that the proportional relationship is enforced, all VPN subscribers are put into a "self-maintenance" status. In other words, when the number of subscribers increases and their performances have to degrade due to the limited SP resources, they will be penalized fairly. Once VPN SPs expand their physical capacities, these downgraded performances shall be able to resume by themselves, without another around of resource provisioning and parameter setup. Third, the interactive [11] framework was proposed for proportional QoS differentiation, where SPs search for appropriate classes to meet subscribers' QoS requirements, and then retain the QoS differentiation. This may relief the dilemma of the need for service differentiation and the preference on the "flat rate" charge between SPs and customers.

1.3 Proportional Loss Differentiation

From the loss perspective, the proportional differentiation model is specified as follows: per-hop packet losses of all classes are proportional to the corresponding differentiation parameters chosen by network operators, such that

$$\frac{\bar{l}_i}{\bar{l}_j} = \frac{\sigma_i}{\sigma_j}, \quad 1 \leq i, j \leq n, \tag{1.2}$$

where $\bar{l}_i$ is the average loss rate of class i, and $\sigma_i, i = 1, 2, ..., n$, are differentiation parameters in terms of the packet loss rate, ordered as $\sigma_1 > \sigma_2 > \sigma_3 > ... > \sigma_n > 0$.

The typical loss schemes proposed under the term of the proportional differentiation, if not the first of its kind, are proportional loss rate (PLR) mechanisms called PLR(∞) and PLR(M). These were proposed [12] to closely approximate the differentiation parameters in terms of packet loss. In PLR(∞), the loss rate estimation l_i is the long-term fraction of packets from class i that have been dropped, being measured by counters for the arrivals and drops in all classes. Denote A_i, D_i, and $B(t)$ as the counter of packet arrivals of class i, the counter of packet drops from class i, and the set of backlogged classes at time t, respectively. Whenever the buffer overflows, PLR(∞) drops a packet from the class whose index is determined from

$$\min_{i \in B(t)} \left(\frac{D_i}{\sigma_i A_i}\right), \ i = 1, 2, ..., n. \tag{1.3}$$

In PLR(M), the loss rate of class i is estimated by the fraction of dropped packets from class i in the last M arrivals. A cyclic queue with M entries, called Loss History Table (LHT), records the number of arrivals $A_i(M)$ and the number of drops $D_i(M)$ from class i in the last M arrivals. PLR(M) and PLR(∞) have the same dropping strategy except with different parameter values.

Claiming that the loss rate estimator influences the short-term as well as long-term differentiations, the average drop distance (ADD) mechanism was suggested [13]. The ADD estimator calculates an average drop distance for each class. The drop distance is the number of transferred packets between two lost ones. By denoting the estimated ADD as $\bar{d}_i$ and the estimated loss rate as $\bar{l}_i = \frac{1}{\bar{d}_i}$, the estimated loss rate ratio between class i and j, i.e., $\frac{\bar{l}_i}{\bar{l}_j}$, is required to approximate the targeted loss rate ratio $\frac{\sigma_i}{\sigma_j}$. When dropping the packet, ADD adopts the same mechanism as those of PLRs.

Introducing an error threshold, a counter resetting mechanism was introduced [14] to target for loss differentiation that is adaptive to load fluctuations. The value of

$\frac{l_i}{l_j} - \frac{\sigma_i}{\sigma_j}$ is calculated on packet arrival basis, and all counters are reset once this value is less than the error threshold.

1.4 Proportional Delay Differentiation

On the delay differentiation [15] of proportional differentiation, per-hop average packet delays of all classes are proportional to the corresponding differentiation parameters, such that

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{\delta_i}{\delta_j}, \quad 1 \leq i, j \leq n, \tag{1.4}$$

where $\bar{d}_i$ is the average delay for class i, and $\delta_i, i = 1, 2, ..., n$, are differentiation parameters in terms of the packet delay, ordered as $\delta_1 > \delta_2 > \delta_3 > ... > \delta_n > 0$.

1.4.1 Properties of the Average Class Delay

Denote the average arrival rate of class i as λ_i, the average size (bytes) of class i packets as $\bar{L}_i$, and the average queue length (bytes) of the buffer as $\bar{Q}$. Without loss of generality, the value of δ_1 is set to 1, and (1.4) becomes

$$\bar{d}_i = \delta_i \bar{d}_1, \quad i = 2, 3, ..., n. \tag{1.5}$$

By adopting the conservation law [16] that constrains average class delays in work-conserving schedulers, the average queue length is an invariant with respect to the aggregated traffic load and the service rate. Independent of the scheduling discipline, this relationship is expressed as follows:

$$\sum_{i=1}^{n} \lambda_i \bar{L}_i \bar{d}_i = \bar{Q}. \tag{1.6}$$

Given that the scheduling mechanism fulfills the delay differentiation, from (1.5) and (1.6), the average delay of class i is

$$\bar{d}_i = \frac{\delta_i \bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k \bar{L}_k}, i = 1, 2, ..., n. \tag{1.7}$$

By assuming that all classes have the same packet size distribution, setting $\bar{L}_k = \bar{L} = 1$, and therefore measuring the queue length by average packet numbers, (1.7) is further simplified to

$$\bar{d}_i = \frac{\delta_i \bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k}, i = 1, 2, ..., n. \tag{1.8}$$

Based on (1.8), the behavior of average class delays have been presented [15], as other system parameters, such as the class traffic arrival, the class load distribution, and the delay differentiation parameter, vary.

Property 1.1: Increasing the arrival rate of a class, increases (in the sense of nondecreasing values) the average delay of all classes.

Property 1.2: Increasing the arrival of a higher class results in a larger increase in all average class delays than increasing the arrival of a lower class.

Property 1.3: Decreasing the delay differentiation parameter of a class increases the average delay of all other classes, and decreases the average delay of this class.

Property 1.4: When one or more users move to a higher class, the delay of all classes increases; when one or more users move to a lower class, the delay of all classes decrease.

Property 1.5: When a user switches from one class to another, it observes a consistent class ordering, that is, the higher class provides a lower delay.

Although it is not quantitatively expressed, how the average class delay varies under different conditions furnishes the general impression of the system dynamics; for instance, if the delay differentiation parameter of one class decreases, all other classes will experience bigger average delays.

1.4.2 Delay Differentiation Schemes

The following notation is used throughout the rest of the dissertation:

$B(t)$: the set of backlogged classes at time t.

$S_i(t)$: the set of class i packets that have departed before time t.

$\Phi(S_i(t))$: the number of packets in sequence $S_i(t)$.

d_i^m : the queueing delay of the m_{th} packet in sequence $S_i(t)$, $m = 1, 2, ..., \Phi(S_i(t))$.

$w_i(t)$: the waiting time of the packet at the head of class i at time t.

Three delay differentiation mechanisms were proposed [15] along with the proportional model itself. Each of them utilizes a certain delay related metric.

Proportional average delay (PAD) mechanism aims to equalize the normalized average delay among all classes. Its delay metric is the normalized average delay of class i at time t, that is,

$$\tilde{d}_i(t) = \frac{\sum_{m=1}^{\Phi(S_i(t))} d_i^m}{\delta_i \Phi(S_i(t))}. \tag{1.9}$$

PAD serves the packet from the class, say i, with the maximum normalized average delay, and stops the delay of this class from increasing. It essentially attempts to reduce the difference between $\tilde{d}_i(t), i = 1, 2, ..., n$. Over a long period, the normalized average delays are then expected to be almost the same. Simulation results show that PAD serves as a good differentiation scheduler over long time periods. Its performance, however, is not predictable over short time periods.

Waiting time priority (WTP) is a classical scheduler that assigns a packet the priority proportional to the packet's waiting time. The packet with the highest priority, that is, longest waiting time, gets served first. Different from PAD, the delay metric of WTP is the normalized waiting time of the packet at the head of class i at time t, that is,

$$\tilde{w}_i(t) = \frac{w_i(t)}{\delta_i}. \tag{1.10}$$

WTP tends to minimize the normalized waiting time difference of successively departing packets; thus, the queueing delay of successively departing packets will be proportional to the delay differentiation parameters. Simulation results illustrated [15]

that WTP performs well for providing differentiation in short time periods, but has difficulty to meet the average delay differentiation over long time periods.

Hybrid proportional delay (HPD) [15] compromises PAD and WTP, by adopting an HPD parameter. Its delay metric, called normalized hybrid delay, is defined as

$$\bar{h}_i(t) = \alpha \tilde{d}_i(t) + (1 - \alpha)\tilde{w}_i t. \tag{1.11}$$

Obviously, HPD becomes WTP when $\alpha = 0$, and turns into PAD when $\alpha = 1$. An empirical value of $\alpha = 0.875$ is chosen to balance both long time and short time period differentiations.

Weighted earliest due date (WEDD) is proposed [17] for real-time traffics with delay bounds. Its delay metric for the delay differentiation model is referred to as the deadline violation probability. With two counters D_i and L_i recording the deadline violating packets and the departure packets of class i, respectively, the deadline violation probability is

$$v_i(t) = \frac{D_i}{\delta_i L_i}. \tag{1.12}$$

Given delay bounds b_i for each class, a safety margin s_i, e.g., $s_i = \frac{b_i}{10}$, is chosen respectively. Packets arriving at time t are stamped a deadline tag $t + b_i$. Any packets exceeding their deadlines are removed right away. At the scheduling decision point, if there are more than one backlogged class with the deadline of its first packet smaller than $t + s_i$, the one having the maximum $v_i(t)$ will be served; otherwise, the one with the minimum tag $t + d_i$ will be scheduled.

1.5 Dissertation Overview

In Chapter 2, a general infrastructure of IP-based VPN implementation is presented, by identifying four deployment building blocks. Current enabling techniques for each block/functionality, though still evolving, are discussed and compared.

A hierarchical QoS-assurance network architecture, from the service provider (SP) perspective, is then proposed in Chapter 3 to address the IP VPN QoS issue. Moreover, recent development progresses from research and engineering work are surveyed to complete the whole picture.

With respect to the loss rate differentiation, Chapter 4 studies the "packet shortage" phenomenon shown in classical loss differentiation schemes. The possibility of compromising the system resource, that is, buffer size, to relieve the problem is first ruled out, according to analysis and simulation results. The "debt-aware" scheme is then suggested based on the idea of gaining more "operation space" and screening appropriate packets to enforce the required loss rate differentiation. In the end of the chapter, simulations are used to demonstrate the regained loss rate differentiation and reduced individual packet delay.

With respect to the delay differentiation, Chapter 5 first derives properties of the average delay difference; it furnishes the system dynamics when system parameters, such as the average class arrival and delay differentiation parameter, vary. Next, delay differentiation performances of two classical differentiation mechanisms, namely the proportional delay differentiation (PAD) and waiting time priority (WTP), over both short and long time periods are investigated. The combined delay differentiation (CDD) is then proposed to provide a certain degree of differentiation performance over both short and long time periods. Simulations are utilized to validate the method.

Both loss and delay differentiation are based on a series of differentiation parameters. Owing to the fact that there is no quantitative guideline for the computation of loss differentiation parameters, Chapter 6 proposes a new approach based on the principles of queueing and optimization. Analysis and numerical results are presented to describe and substantiate the method.

Observations and contributions of the dissertation are concluded in Chapter 7.

CHAPTER 2

GENERIC IP VPN DEPLOYMENT INFRASTRUCTURE

This chapter provides an overall picture of IP-based VPN implementations, along with further clarifications. It illustrates the macroscopic deployment framework from the ISP point of view, and helps subscribers obtain a better understanding of their VPN service criteria.

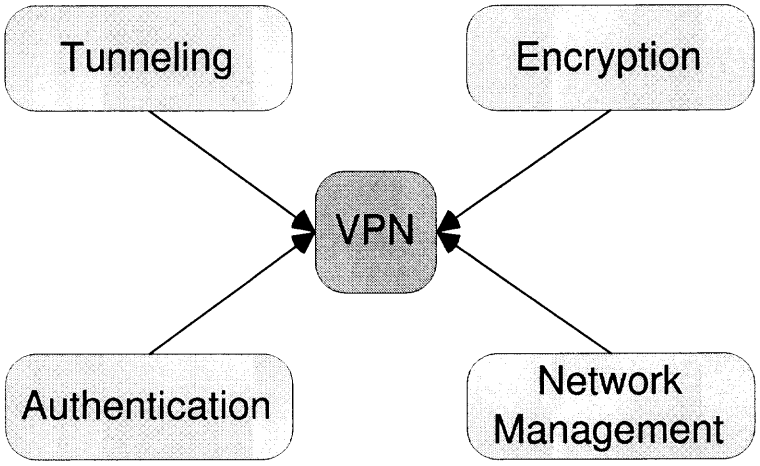


Figure 2.1 VPN implementation building blocks.

2.1 A Portrayal of VPNs

As illustrated in Fig. 2.1, VPN implementations involve tunneling/encapsulation, authentication, encryption, and network management.

Tunneling/encapsulation protocols encapsulate packets with extra headers and logically separate them from other traffic. Packets with different headers go through different virtual paths or routes, just like dedicated lines. The essential difference between VPN tunnels and real dedicated lines is that these "dedicated" paths are actually sharing a common link, or say, network pipe. Owing to the sensitivity of e-business information, all entities in a VPN have to go through the authentication

process which verifies and restricts the network access to validated users or devices. Even with authentication, however, plain packets being transferred over VPNs are open to attacks. Encryption protocols, therefore, are adopted to protect packets from illegal examination and manipulation. The network management infrastructure is then required for billing, resource management, service level agreement (SLA) enforcement, and other management related issues. The following sections itemize major techniques for functionalities mentioned above, respectively.

2.2 Tunneling

A tunnel is a specific pathway where packets encapsulated with extra headers are delivered. The destination strips the encapsulation header of the packets, and processes them as if they were received on a local interface.

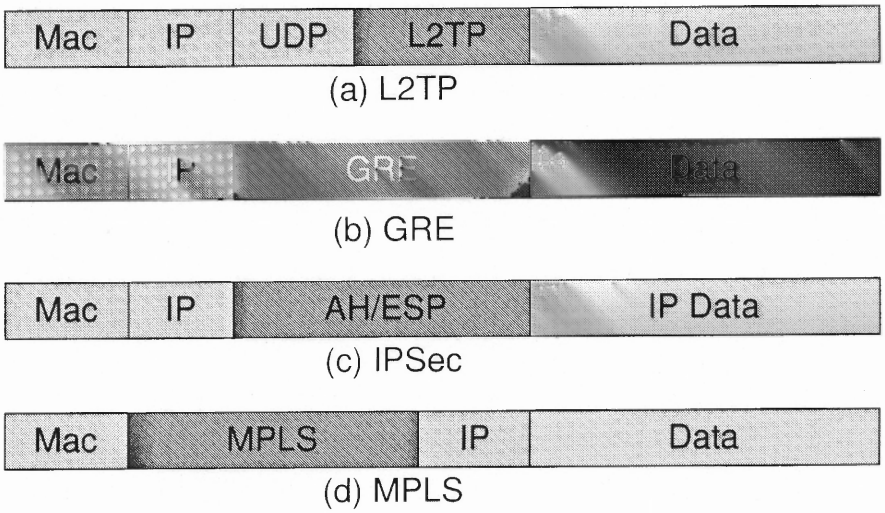


Figure 2.2 Tunneling/encapsulation protocols.

2.2.1 Tunneling Techniques

Fig. 2.2 depicts structures of an IP packet encapsulated by different protocols, i.e., layer 2 tunneling protocol (L2TP), generic routing encapsulation (GRE), IP security (IPSec), and multi-protocol label switching (MPLS).

L2TP [18] merges the best features of the point-to-point tunneling protocol (PPTP) and layer 2 forwarding (L2F) protocol. The two peers of an L2TP tunnel are the L2TP access concentrator (LAC), a device that physically terminates remote connection requests, and the L2TP network server (LNS) that terminates and authenticates PPP streams. Corresponding to the tunnel initiation point, the client-initiated tunnel requires end users to support L2TP, while the LAC-initiated tunnel relies on LAC's L2TP functionality.

GRE [19, 20] provides a common mechanism to place packets of any protocol, for example, address resolution protocol (ARP), Novell IPX, AppleTalk, etc, into any other types of protocols. When the protocol of the traffic is not compatible to that of the transport network, the GRE header is inserted as a "cushion" in between.

A packet encapsulated by the tunnel mode of IPSec [21] has an "outer" IP header (that specifies the IPSec processing destination) and an "inner" IP header that specifies the packet's ultimate destination. The security protocol header (will be covered in Section 2.4) resides between these two headers, and carries security parameter index (SPI) used by the receiving peer to select a security association (SA) under which received packets will be processed. SA, a set of security parameters for IPSec tunnel authentication and encryption, is managed by Internet key exchange (IKE) [22, 23]; IKE is formally known as Internet security association key management protocol (ISAKMP/Oakley).

MPLS [24] works on a label-based paradigm, tagging packets as they enter the provider network and expediting the forwarding through the IP core. Defined as a short, fixed length identifier to identify a forwarding equivalence class (FEC), the label is inserted between the data link layer header and the network layer header. Though functioning in a fairly different manner from IP forwarding, MPLS offers the equivalent security as that of frame relay or ATM.

Table 2.1 Comparison of Tunneling/Encapsulation Techniques.

	L2TP	GRE	IPSec	MPLS
Multiplexing	uses the session ID and tunnel ID.	uses the key field.	uses SPI.	uses the MPLS label.
Signaling	exchanges control messages.	shares a similar signaling mechanism as mobile-IP.	relies on IKE.	features label distribution protocol (LDP).
Data security	hides attribute value pairs (PAD), but has no data confidentiality.	can have authentication, but no data confidentiality.	provides authentication and data confidentiality.	is equivalent to layer 2 VPNs, but has no data confidentiality.
Multi-protocol support	inherits the multi-protocol capability from the point-to-point protocol (PPP).	supports any network layer protocols.	only supports IP.	supports multiple network layer protocols.
Frame sequencing	has a specific field to record the frame sequence.	has a specific field to record the frame sequence.	can extend the "sequence number" field for in-order frame delivery.	proposes a couple of Internet drafts to enhance the feature.
Tunnel maintenance	exchanges "keep-alive" messages among peers.	relies on external routing protocols.	relies on IKE to send out "hello" messages periodically.	employs LDP to detect bad label bindings.
QoS capability	does not manipulate the IP header, thereby is open for QoS operation enhancement.	needs to copy the type of service (TOS) information to the encapsulated header.	needs to copy the TOS information to the encapsulated header.	adopts a three-bit experimental field to define QoS criteria.

2.2.2 Comparison and Discussion

A comparison of all tunneling techniques mentioned above is shown in Table 2.1. The comparison metrics include the *multiplexing capability* [25] that enables one VPN peer to support multiple customers, the *signaling capability* that allows VPN tunnels to automatically exchange the configuration information, the *innate data security* characteristics, the *multi-protocol support capacity* that accommodates diverse enterprise network protocols, the *frame sequencing ability*, the *tunnel maintenance capability* that guards the connectivity of VPN peers (i.e., connectivity loss check and explicit failure indication) and takes appropriate actions (such as back up or tear down) if there has been a failure, and the *QoS compatibility* that provides tunnels the potential of further QoS operations.

A multiplicity of application scenarios means that there is no single ideal solution for VPN tunneling/encapsulation. Industry vendors often apply a combination of L2TP and IPSec to remote access VPNs because L2TP supports dial-up connections. For LAN-to-LAN VPNs, IPSec uses the tunnel mode to incorporate its own encryption functionality. Since the IPSec suite only works for IP traffic, another combination of GRE and IPSec is adopted for LAN-to-LAN VPNs to handle non-IP traffics. For the scalability concern, furthermore, IETF is advocating MPLS; the unique packet forwarding mechanism of MPLS highlights another realm of LAN-to-LAN VPN technology.

2.3 Authentication

The authentication process, whereby a remote or mobile entity is identified prior to accessing networks and network services, can be deployed both at the user and the device levels (the packet level authentication is done by IPSec for more stringent security). With different preferences, industry vendors adopt diverse authentication techniques such as manual key, token cards, challenge responses, digital certificates,

biometrics, smart cards, RSA SecurID, Kerberos, and light directory access protocol (LDAP).

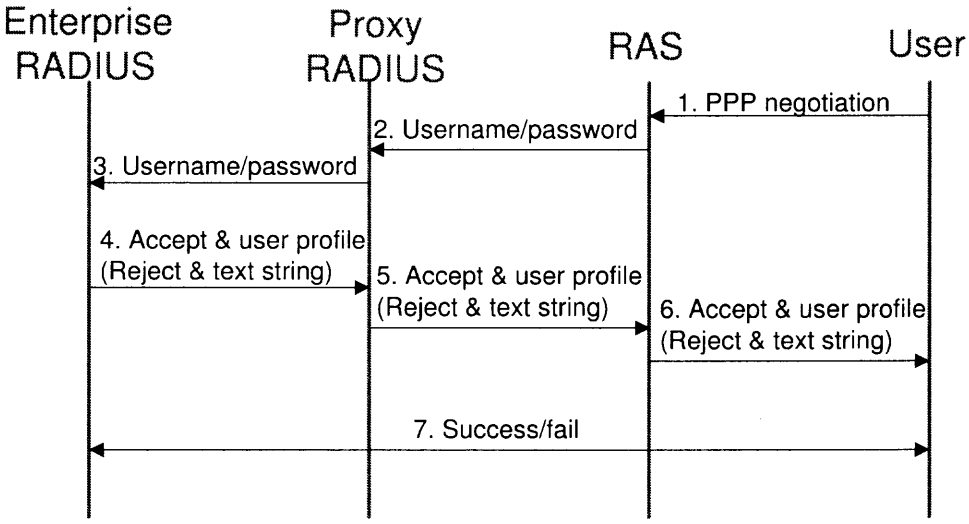


Figure 2.3 User authentication procedure.

2.3.1 User Authentication

For user authentication, remote access dial-in user service (RADIUS) [26], a challenge response technique, provides the industry-standard, client/server-based solution. A typical RADIUS authentication involves the remote user, ISP remote access server (RAS), the proxy RADIUS server, and the enterprise RADIUS server.

As illustrated in Figure 2.3, the authentication process [27] has several steps. The PPP negotiation starts when an end user dials in one of the ISP RASs. This RAS passes the authentication information, such as the username and password, to the ISP's proxy RADIUS server. Parsing the authentication information, the proxy RADIUS server performs a translation to determine the IP address of the end user's enterprise RADIUS server and passes on the user information to the enterprise RADIUS server. From the enterprise RADIUS server, either an "accept" or "reject" response is issued to the ISP's proxy server that in turn forwards this very response to the ISP's RAS. Along with the "accept" and "reject" response, there is the user

profile or the text string indicating the rejection reason obtained from the enterprise RADIUS server, respectively. The ISP RAS then accordingly allows the end user to access the enterprise network or terminates the connection.

To exchange messages between two authentication peers, two protocols are available: challenge-handshake authentication protocol (CHAP) encrypts usernames and passwords; password authentication protocol (PAP) exchanges plain passwords. In practical implementations, the enterprise RADIUS server can also be outsourced to VPN service providers and thus resides in the provider network; this is called the "internal" RADIUS authentication.

2.3.2 Device Authentication

The device authentication takes place when a VPN device is added or powered up. The pre-shared key [28] technique, a popular device authentication solution, uses unique, group, and wildcard keys distributed through a secured out-of-band channel. Unique and group pre-shared keys are tied to a specific IP address and a group name identity, respectively. Wildcard pre-shared keys, however, are the same for all devices in the network. The former two do not scale well because each device has to store all other keys, and the latter will no longer be safe even when one device is compromised.

Digital certificate, another technique which scales better, allows any device to authenticate any other device but does not have the security drawback of wildcard keys. It utilizes the unique information on the device that is validated by a trusted third-party known as certificate authority (CA). When the device using the digital certificate receives a tunnel establishment request, it checks the peer certificate against certificate revocation list (CRL). Should a hacker compromises or steals a device with a digital certificate, the network administrator is able to revoke the digital certificate and notify other devices by broadcasting a new CRL that contains a CA-signed list of revoked certificates.

Incorporating IKE and the digital certificate, a typical device authentication process [29] over IPSec tunnels can be demonstrated as follows. First, all participating IPSec peers recognize one CA as the authenticating authority, each IPSec peer has its own digital certificate issued and validated by this CA, and then each peer's certificate is ready to be used to encapsulate that peer's public key. Next, it takes a device four steps to sign on with a CA. First, a VPN client (either software or hardware) generates a public/private key pair. This client signs its outbound data with its private key. CA then uses this client's public key to validate that these data were originated by the VPN client. Second, the VPN client requests the CA's public key to validate inbound data from CA. Third, the VPN client sends an enrollment request to CA, while CA binds the VPN client's personal certificate with its public key, and then signs the personal certificate. Fourth, the VPN client receives the signed personal certificate and validates this certificate by decrypting the signed personal certificate with its private key. Note that the distribution of public keys is handled by the IKE protocol; the success of CAs depends on the deployment of public-key infrastructure (PKI) [30, 31].

With emerging new threats, firewall products alone are probably not sufficient to ensure the e-business safety. The scalability and effectiveness of authentication techniques, therefore, are under intensive considerations. For instance, as compared to pre-shared keys, digital certificate enhances the network scalability. The combination of multiple authentication techniques, such as the token card and password, has been suggested for a more effective authentication. As a price to pay, however, the additional administrative burden is significant when the size of VPN increases or a strong device authentication is adopted.

2.4 Encryption

With only tunneling and authentication, the data integrity of VPN services still remains an issue. This is where IPSec, along with other cryptographic protocols for the network management such as secure shell protocol (SSH) and secure sockets layer (SSL), comes into play. The IPSec suite provides the authentication header (AH) and encapsulating security payload (ESP) protocols, which can be used either separately or collaboratively to ensure data integrity and confidentiality.

AH [32] provides the connectionless data integrity and data origin authentication. The connectionless data integrity ensures that the original packets are not modified during the transit from the source to the destination, and the data origin authentication verifies the source of data. Being inserted between the IP header and the payload, the AH field contains the cryptographic checksum of the packet content and part of the IP header itself. It uses cryptographic algorithms, such as the hash-based message authentication code (HMAC) coupled with the message digest 5 (MD5) hash function or the secure hash algorithm 1 (SHA-1), to calculate the checksum (a hash function is a one-way mathematical function that takes a variable-length message and produces a unique fixed-length value). By calculating the cryptographic checksum of the received message and comparing it with the received value, the receiver can verify that the message has not been altered in transit.

ESP [33] provides the data confidentiality, authentication, and anti-replay capability. The confidentiality is achieved by the encryption process that takes a message, referred to as the clear text, and passes it through a mathematical algorithm to produce what is known as the cipher-text. Encryption algorithms, such as data encryption standard (DES) and triple DES (3DES), rely on a value, that is, the key, to encrypt and decrypt data. The secure distribution of the key is managed by IKE.

The major implementation concern for encryption techniques is the processing speed; it is driven by other high-speed network devices. Although the remote user

can use the software-based encryption for the sake of low cost, the VPN gateway or remote access server probably requires a hardware boost.

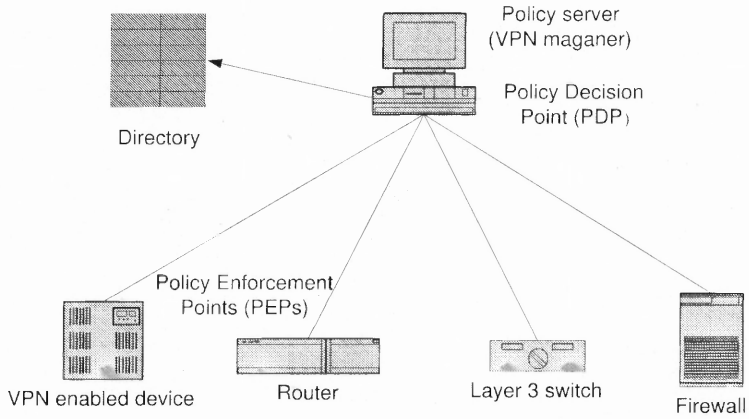


Figure 2.4 Policy-based network model.

2.5 Network Management Infrastructures

Following appropriate tunneling, authentication and encryption techniques, a network infrastructure is desired to manage VPN services. A couple of existing structures, for example, policy-based network (PBN), telecommunications management network (TMN), etc., are tailored for VPN applications. Omitting management details on provisioning, billing, SLA, fault management, and resilience, this subsection centers on different management infrastructures.

Originally designed [34] for security management purposes, particularly for access control, PBN has been adapted to monitor and manage VPNs, based on policies that define how and when to handle network applications. As indicated in Fig. 2.4, most vendors' policy management products [35] consist of a directory, a policy server known as the policy decision point (PDP), and VPN devices referred to as policy enforcement points (PEPs).

The directory stores global settings, coordinates and synchronizes multiple policy servers, and provides information about users, file servers, and other resources where

the policy server wants to apply policy. All interfaces on VPN devices are assigned a series of rules, which are defined in the policy server. For instance, there may be a policy called "the traffic goes from gateway A to gateway B uses the GRE tunneling and DES encryption." The policy server assigns policies to VPN device interfaces by using command line interface (CLI) commands, simple network management protocol (SNMP), or common open policy service (COPS) protocol. As the policy enforcement indicates, VPN devices ensure the given policy is carried out via specific hardware and software functionalities, such as packet filtering, bandwidth reservation, and traffic prioritization.

SNMP, a protocol supporting the communication between the policy server and VPN device interfaces, defines a means of monitoring and pushing the configuration or policy information among network entities. SNMP messages are instances of different object types defined either by Internet-standard management information base (MIB) or Internet-standard structure of management information (SMI).

Although industry vendors have developed diverse proprietary MIBs such as IPSec MIB, L2TP MIB, and VPN MIB, an accurate and clear policy definition can be a problem for large-scaled PBNs with heterogeneous VPN devices. As a matter of fact, breaking the service functionality into device-specific functions outlined in related MIBs is time-consuming and error-prone. The ongoing work in standard forums and research communities focuses on the element management problem, i.e., the policy specifications for managing multiple devices and supporting the end-to-end QoS [36].

As depicted in Fig. 2.5, the logical layer architecture (LLA) of TMN [37] defines "logical layers" (i.e., groups of management functions) and describes the relationship between these layers. The element management layer controls and coordinates a subset of network elements on both an individual and a collective basis, whilst maintaining statistical, log and other data. The network management layer is respon-

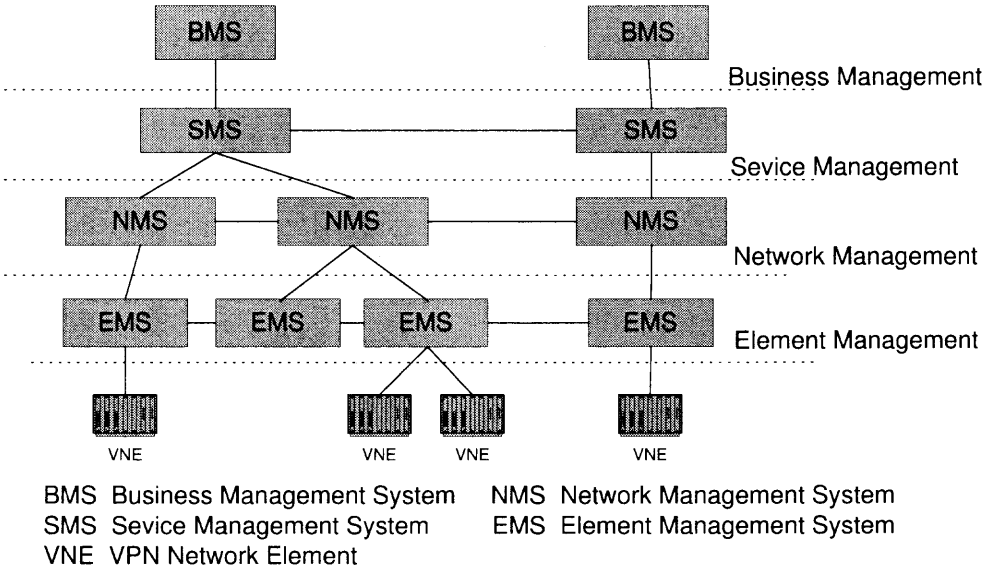


Figure 2.5 General TMN infrastructure.

sible for the management of the network supported by element management layers. The complete visibility of the whole VPN network and, as an objective, the technology independent view will be provided to the service management layer; in turn, the service management layer negotiates contractual agreements with VPN customers. The functionality of the business management layer is to optimize the investment and usage of new resources, while that of the service or network management layer is to maximize the utilization of existing resources.

In a nutshell, by defining the functionality of each layer and interfaces between components in the same layer as well as successive layers, the TMN system provides a wide variety of management areas including the planning, installation, operations, administration, maintenance and provisioning of the network and services. It is therefore a more comprehensive infrastructure for VPN services. The ongoing effort on the TMN infrastructure, in addition to the relationship between management systems in the same layer, is to define service layer management specifications and interface points between different TMNs.

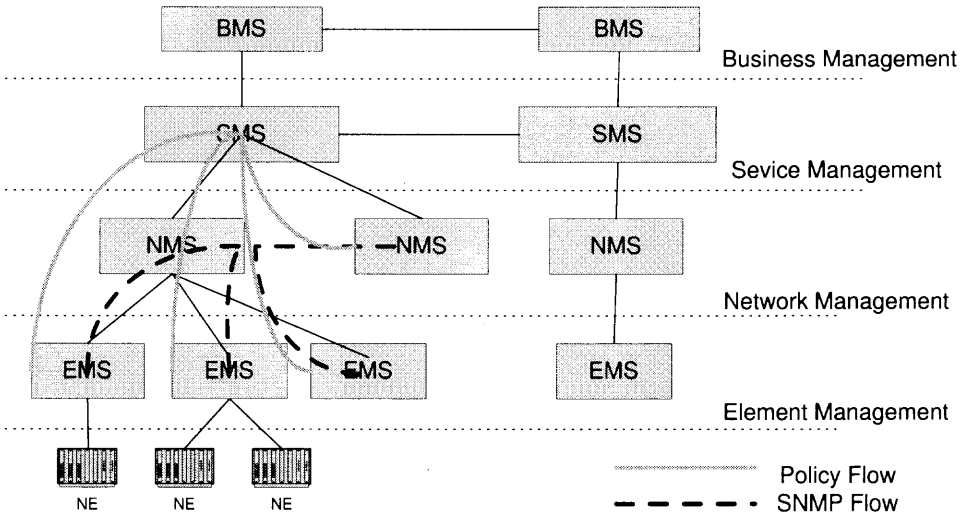


Figure 2.6 Hybrid VPN management infrastructure.

Given the DiffServ service model, a hybrid VPN management architecture [38] that brings together the advantages of several network infrastructures is depicted in Fig. 2.6. Lines expanding through the network element management and network management layers represent SNMP message flows. SNMP allows the monitoring of network elements and pushes the configuration information into all kinds of network devices; this solves the heterogeneous hardware problem and in some degree enforces the configuration consistency. SNMP MIBs are defined to represent the device management information. The device driver then translates user requests and pseudo policies into device-specific rules and accordingly configures VPN and DiffServ-aware routers.

The consistency among network configurations and running services is enforced through the centralized software agent in service management system (SMS); this agent is essentially a policy server. Before defining policies for VPN services, SMS needs to check the resource availability, by considering a collection of databases managed by network management system (NMS). For instance, the SLA database contains the user's identification, the maximum amount of traffic for a tunnel, and the boundary of a VPN; the connection database keeps a list of currently active VPN connections.

The business management layer handles the negotiation between customers and SPs, such as the SLA establishment. Also, the billing information coming from the service management layer is collected here and sent to customers. In a multi-ISP scenario, the business management system carries out the SLA setup between SPs.

2.6 Future Trend

As a flourishing technology, VPN has been experiencing intensive changes. In March 2002, Cisco announced [39] its unified VPN suite for IP and MPLS backbones. This comprehensive delivery included new VPN provisioning tools and new protocols such as any transport over MPLS (AToM) and layer 2 tunneling protocol version 3 (L2TPv3). In January 2002, Aleron [40] became the first major Internet backbone provider that fully implemented MPLS technology across its entire core network; the network "switches" IP packets directly over optical networks to provide customers with decreased network latency and router hops. MPLS, associated with resource reservation and traffic engineering technologies, delivers highly configurable VPNs as well as QoS-defined applications. As a general trend, MPLS, DiffServ, and IPSec, will probably become major players in the VPN product market, where MPLS and DiffServ relieve the scalability bottleneck and thus enable the end-to-end QoS across the IP core, and IPSec secures e-business information down to the packet level.

2.7 Chapter Summary

This chapter illustrated the implementation of Internet-based VPN services, highlighting its capability of constructing a secure network infrastructure. For ISPs seeking for new avenues, the chapter provided a light but comprehensive VPN deployment framework; for enterprises seeking for supreme e-business services, the discussion on various enabling VPN techniques furnished a different view of the service criteria.

CHAPTER 3

TOWARD IP VPN QUALITY OF SERVICE: A SERVICE PROVIDER PERSPECTIVE

VPNs complement classical enterprise wide area network (WAN) infrastructures, aiming to accommodate mushrooming telecommuters, road warriors, and business partners dispersed around the world. They carve public WAN links out of the rest of the network, and thus connect sites through WANs or provide the remote access to enterprise networks, all in a private network-like manner, that is, the same policies of security, manageability, QoS, etc. The VPN hype will continue in years to come, owing to the rising desire for economical, reliable, and secure communications. Cahners In-Stat Group estimated that VPN services would hold a $23.7 billion strong share of the $104.4 billion worldwide IP service revenues in 2005.

A downside shared by legacy layer 2 VPN strategies, such as Frame Relay and ATM virtual networks, is the connection-oriented characteristic; in the network core, the mesh of the permanent virtual circuits required by provisioning redundancy becomes costly and does not scale well. For a bigger market share, a scalable and cheaper VPN solution is sought; this is where the Internet, with the global reachability and cost effectiveness, comes into play. Enabling a low-cost, secure IP solution to replace expensive, dedicated WANs, IP VPNs can be broadly classified into three categories: remote-access VPNs connect remote users to the enterprise LAN; Intranet VPNs connect branch offices and home offices within the enterprise WAN; Extranet VPNs supply business partners limited access to the enterprise LAN.

There are two typical VPN deployment strategies. First, taking control of their VPN services, enterprises adopt and manage their own VPN-enabled customer premise edge (CPE) devices. Second, enterprises outsource part or all of their VPNs to an SP; the VPN management complexity is then shifted to service provider edge (PE)

devices. The second strategy, an SP perspective solution that will be addressed in this chapter, becomes fairly popular. It gives SPs a foothold in enterprise networks for new revenues, and minimizes/eliminates enterprises' in-house need for the network management expertise.

The chapter assumes the following premises. First, peer to peer VPNs, all of whose routers have the capability to forward the VPN traffic to appropriate destinations, are addressed. Overlay VPNs, the alternative implementations that only take VPN tunnel endpoints into consideration, have no control on the intermediate routers; they cannot deliver end-to-end QoS, and therefore are of no interest here. Second, the term of VPN SP is used in the rest of the chapter to represent an Internet SP which provisions VPN services. Third, technical approaches for IP VPNs discussed in the chapter utilize IP-over-IP [41], IPSec [32, 33], and GRE [19, 20] protocols. Fourth, the end-to-end QoS in the chapter means the QoS enforcement between SP PE devices. The last mile from the subscriber edge to the SP edge is under the control of the subscriber.

Utilizing various enabling techniques on VPN tunneling, encryption, authentication, and network management detailed in Chapter 1, the typical IP VPN deployment architecture is depicted in Fig. 3.1. It illustrates one (or multiple) SP network(s) gluing together enterprise networks, SP edge routers, and core routers to accommodate overlapping VPNs. In the rest of the chapter, SP edge routers and core routers are referred to as provider edge (PE) routers and provider (P) routers, respectively.

3.1 IP VPN QoS Issue

The QoS guarantee is the capability of a network infrastructure to deliver different levels of services. Its metrics include but are not limited to packet loss, delay, delay jitter, bandwidth guarantee, and throughput. In addition to the information security, VPN services have various QoS requirements. For instance, an executive video

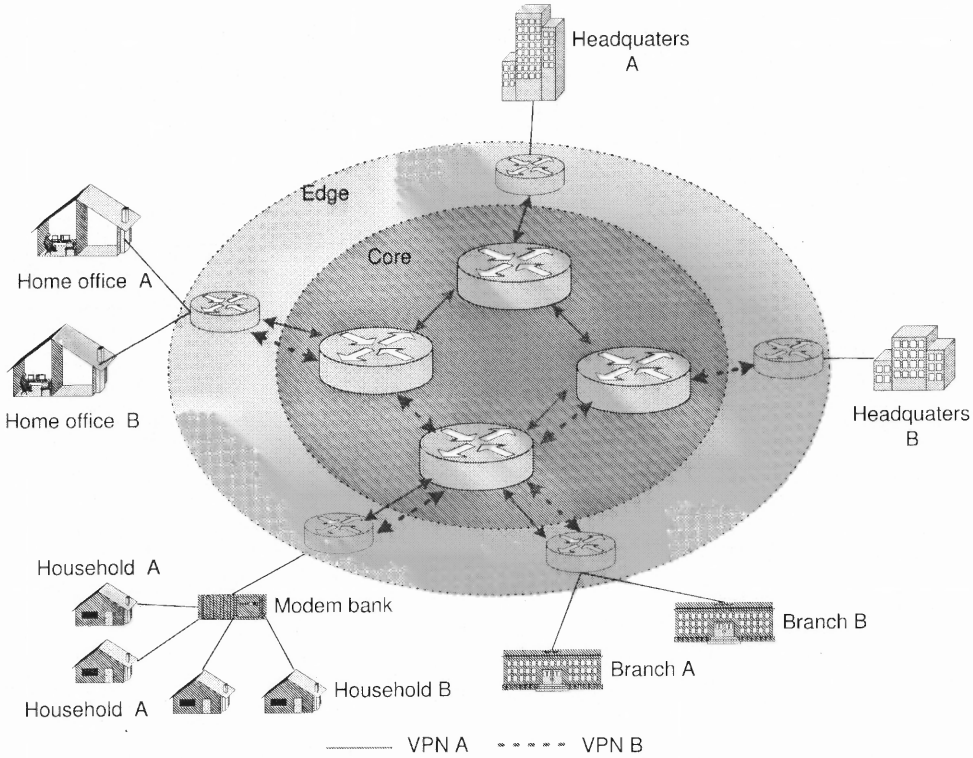


Figure 3.1 General IP-based VPN architecture.

conferencing may need stringent QoS as well as security requirements, whereas a secure database transaction may tolerate a certain QoS downgrade when the network resource is in short supply. In general, the VPN QoS can be delivered on the VPN subscriber and/or application type basis, as illustrated in Fig. 3.2; the whole issue can be viewed as handling multiple traffic classes/aggregates with different QoS criteria.

To yield the equivalent end-to-end QoS of connection-oriented layer 2 VPNs, IP VPNs fulfill the QoS control in a hierarchical manner. First, following the service level agreement (SLA) with subscribers, VPN SPs identify a route (or routes) capable of offering the required QoS and provision appropriate resources (e.g., bandwidth). Second, VPN QoS parameters are pushed down to router interfaces along the identified routes, by utilizing a certain centralized or signaling-based mechanism. QoS is then enforced by queueing and scheduling mechanisms in the routers. Bearing in mind

this hierarchical framework, the rest of the chapter will provide a glimpse into QoS enabling technologies of IP VPNs.

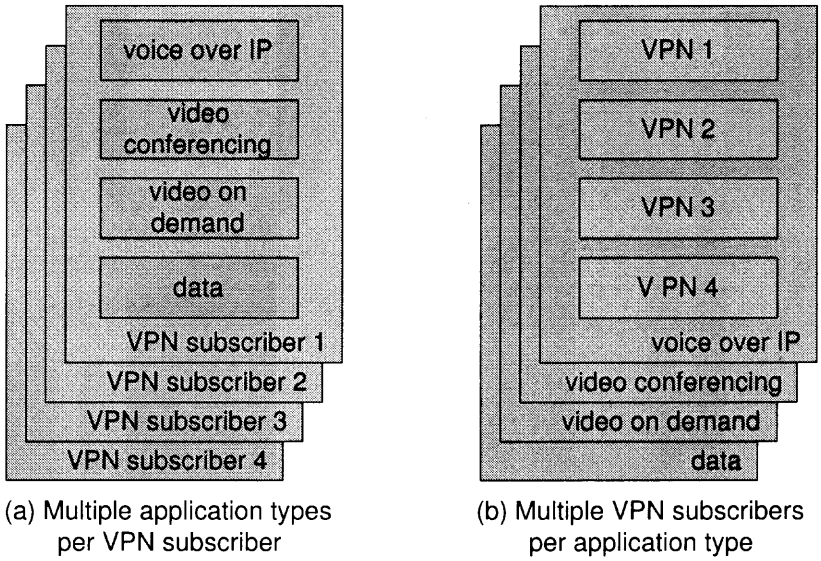


Figure 3.2 Two examples of how VPN SPs deliver QoS.

3.2 IP QoS Architectures

IP VPNs may adopt a number of IP QoS architectures whose differences, in terms of SLA policies, are exemplified in Fig. 3.2. Different architectures often use different mechanisms to establish network routes and enforce QoS guarantees.

3.2.1 Integrated Services

IntServ, along with the resource reservation protocol (RSVP) [42], provides hard, end-to-end, fine-grained service guarantees; all routers in the network participate in the RSVP signaling to reserve, tear-down and manage appropriate resources. The RSVP signaling often implies a per-flow resource allocation identified by a five-tuple (transport protocol, source address and port, destination address and port).

IntServ/RSVP leads to a severe scalability difficulty because it is impossible for a core router to maintain the state of all application flows routed through it. However,

it may be implemented on a limited scale, for instance, in an enterprise network, or in the core network where RSVP is under the control of a network management system to set up QoS-capable routes for traffic aggregates. The MPLS working group, likewise, proposed to use an extended version of RSVP [43] to set up explicit routes in the core network.

3.2.2 Differentiated Services

DiffServ defines three types of PHBs: EF, AF, and BE; they specify in which manner packets will be forwarded. With certain specifications in the packet header, customers indicate which type of service they require for an application. The philosophy of "move the complexity toward the edge" has led to a widely accepted concept that the DiffServ architecture should be implemented in the core, pushing IntServ to the edge.

The DiffServ infrastructure has been rather favored in IP VPN implementations owing to the following facts: DiffServ handles traffic aggregates, and is thus capable of differentiating QoS on per VPN basis or on per application basis within a VPN; DiffServ QoS operations become fairly straightforward when handling VPN traffic with explicit destinations; the scalability advantage of DiffServ benefits multi-SP VPN deployments.

As will be noticed, the majority of strategies in this chapter are based on DiffServ, taking the mainstream technologies into consideration.

3.3 VPN Network Perspective

Requiring the comprehensive information of a network, QoS operations at the VPN network level include resource provisioning, admission control, and routing. They can also be referred to as control plane functionalities.

An SLA between a VPN subscriber and its SP is a fundamental component. In addition to the charging and compensation matters in the event of an agreement violation, SLA defines conventional specifications such as the service availability and offered service (e.g., bandwidth, latency, packet loss, hop-count, and cost); other VPN-specific criteria, such as VPN tunnel start time, duration, and redundancy, are also included. VPN SPs, therefore, are challenged to provide services that meet this quantifiable commitment (i.e., SLA).

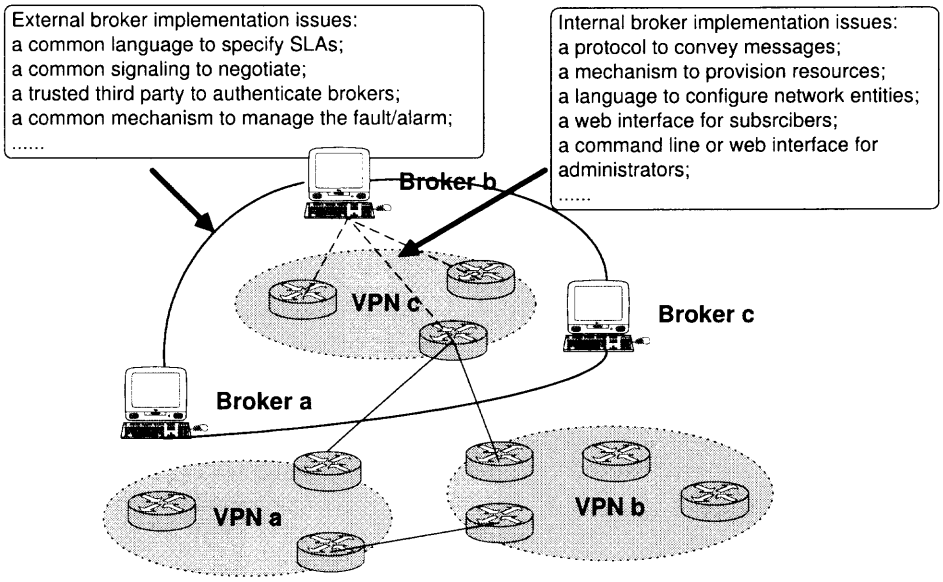


Figure 3.3 Implementation of the VPN service broker infrastructure.

3.3.1 Management Infrastructure

While the time-consuming, prone-to-error manual/static resource provisioning is still in practice, notable efforts have been made to bring more automation and intelligence into VPN network operations.

An automated software agent, namely the VPN service broker, has been under intensive discussion for VPN QoS management. It monitors and enforces the service as specified in SLA, by carrying out the functionality of a system administrator, such

as dynamic service configuration, VPN tunnel admission, and capacity provisioning. This concept can be implemented as an internal entity that does inter-domain resource allocation and pushes the configuration information down to routers within an SP domain. It can also be adapted as an external entity that handles VPN SLA requests and agreements with peer external brokers of adjacent VPN SPs.

There are full-fledged standards available for the intra-domain service broker implementation, such as the policy controlled network structure [44], SNMP [45], COPS [46], and LDAP. When several SPs collectively provide VPN services, however, the inter-domain broker implementation poses a new challenge. Heterogeneous operation support systems (OSSs) of different domains demand a means of exchanging accounting, billing, or resource provisioning information. For this inter-domain federation, therefore, an open and standardized framework as well as interfaces between OSSs is under intensive investigation. The general view of the service broker system is depicted in Fig. 3.3, taking both the current status and future expectation into account.

As for today, although there is no complete standard suite available, a large amount of work has been done by project groups to tackle the inter-domain federation issue. A generic and high-level inter-domain prototype system and a general QoS-enabled VPN management system [38] were developed in the charging and accounting technologies for the Internet (CATI) [47] project. Adopting the generic network model [48], a TMN [37] compatible infrastructure was suggested; it utilizes the cross domain VPN manager to handle the end-to-end VPN service activation and provisioning. Aligning with the telecommunications information networking architecture (TINA) [49] principle, a software platform [50] for VPN connection management, VPN service management, and SLA monitoring was developed; it is based on the common object request broker architecture (CORBA), a *de facto* middleware standard for interface and service definitions.

Industry vendors, furthermore, have already put their proprietary broker products into practice; for instance, Alcatel has implemented the VPN bandwidth broker solution and the dynamic call admission control (CAC) module.

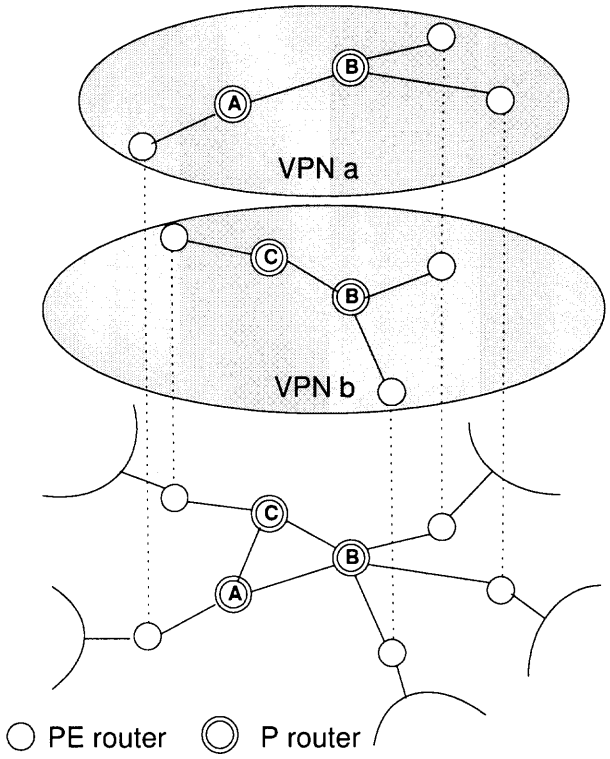


Figure 3.4 An example of VPN resource provisioning, where QoS-capable routes (topologies) of VPN A and B have been reserved from the original network, respectively.

3.3.2 Resource Provisioning

VPN resource provisioning can be viewed as searching for the cheapest network route or topology that satisfies a subscriber's QoS constraints. By generalizing the whole network into a weighted directed graph, searching for one or multiple sub-graphs (i.e., the topology of a QoS-warrant VPN) with the least cost improves the network resource utilization. The cost of a route can be defined as a function of the hop-count, residual bandwidth, VPN redundancy, and other QoS associated parameters. Fig. 3.4

illustrates an example of VPN resource provisioning, where resources for two VPNs are stipulated. With the knowledge of individual VPNs, the problem is modeled as the optimization of an objective function with particular constraints. It can become an NP problem and certain heuristic approximations (e.g., relaxing certain constraints) will have to be entailed to make the problem tractable.

Searching routes for VPNs can be deployed either in a centralized or a distributed way. A typical example of the first case is the service broker that is in charge of admitting, setting up, and tearing down VPN connections. In the previous broker implementations, often time network routes are determined without involving any routing intelligence. This is the very reason that centralized databases have to be consulted for the tunnel management. However, VPN SPs shall endeavor to accommodate more automation into their network infrastructures, targeting more diverse and flexible services, such as short-lived or highly dynamic VPNs. QoS (constraint-based) routing [51], with routers themselves searching for eligible network routes with sufficient resources to meet the QoS requirements in a distributed manner, can be a potential complementary of the VPN router functionality. Its general goals are two-fold: every admitted VPN connection has its QoS requirements satisfied; the total cost of all connections on a path is minimized.

The VPN resource provisioning and utilization optimization have been undergoing intensive study, taking SLAs, VPN topologies, VPN policies, and available resources into consideration. VServ [52], a comprehensive architecture, presented a set of automated functionalities to support intra- and inter-VPN resource provisioning. It utilizes a VPN description language to translate high-level customer criteria into lower level specifications, constructs a search space according to VPN requirements, and then looks for the optimal topology to complete the resource allocation.

3.3.3 Ongoing Issues

A VPN may geographically extend over multiple autonomous domains, or functionally, multiple SPs. The VPN QoS issue, as discussed above, has to deal with the federation among independent management entities. While a generic management infrastructure is under intensive pursuit, how to accommodate different IP QoS architectures (e.g., IntServ and DiffServ) is also on the agenda.

As an example, the service broker can aggregate per-VPN IntServ messages at PE routers, leaving the core network (often time a DiffServ domain) no hassle to process IntServ messages. Aiming to eventually fix the problem, standard bodies have been working on the issue of handling RSVP signaling in a DiffServ domain that is either RSVP-aware or RSVP-unaware. For a seamless inter-operation, the follow-up standardization work, such as mapping IntServ service specifications into DiffServ PHBs, defining a certain functionality for the IntServ signaling to deliver the aggregate traffic control, and designing a dynamic mechanism for DiffServ resource provisioning, is required [53].

Originally as a software module in VPN PE nodes/devices, a virtual router was proposed to handle control plane operations on a per VPN basis, thereby restricting the effect of a single misbehaving VPN. Each virtual router is expected to partition individualized service definition of bandwidth, priority, and security on either per subscriber or per traffic aggregate basis. Attributes that distinguish VPNs from each other could be topology, duration, and the service they carry. PE routers then maintain separated routing tables and make forwarding decisions for each distinctive VPN, respectively. To match packets to the corresponding VPN routing table, PE routers could use a certain tag, such as VPN ID [54] with a global significance. Other issues being addressed include the scalability of the number of routing instances, the processing power, and the separation between different VPN routing instances.

A typical industry implementation of the virtual router concept is the IPSX service processing switch family released [55] by CoSine Communications. Although it delivers finer-grained control over the routing topology, nevertheless, the virtual router implementation consumes extra bandwidth and router resources; it may not be cost-effective for simple VPN topologies.

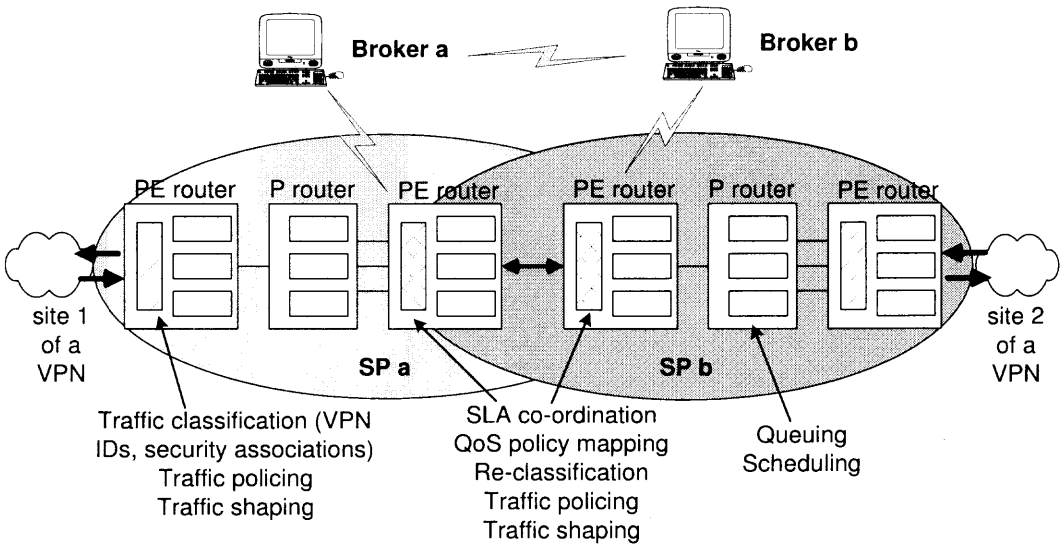


Figure 3.5 VPN data flow across multiple SP domains between sites 1 and 2.

3.4 VPN Node Perspective

VPN SP proprietary routers must act in concert with the network level operations to complete the end-to-end QoS enforcement. Therefore, the data plane operations in VPN nodes, which involve shaping, policing, queueing, and scheduling, must be configured according to QoS parameters determined by network level operations. Looking at a VPN tunnel as just another type of link, many existing QoS mechanisms can be applied to VPN traffic with VPN-specific parameters; so are the techniques adopted for IntServ and DiffServ. In association with the VPN data flow illustrated in Fig. 3.5, VPN router implementations from industry vendors are selectively touched on in the following sections.

3.4.1 Classification

Classification at the SP edge is the foundation of all other QoS operations in VPN routers. Its purpose is to subject the traffic for future specific treatments; for instance, a smaller delay for video conferencing applications, a lower dropping probability for mission-critical services, or a faster forwarding for "golden" VPN subscribers. The edge router groups the incoming VPN traffic according to predefined criteria from SLA and/or a policy server, such as the IP address and application type. It then marks packets, ensuring that the classification will be honored all the way to the other end of the VPN tunnel. An implementation example is Cisco's committed access rate (CAR), one of whose features is to partition the VPN traffic into multiple priority levels or service classes.

3.4.2 Conditioning

To enforce subscribers to follow their SLAs, traffic conditioning (shaping/policing) takes place on boundary nodes between VPN subscribers and SPs. As implied by their names, traffic shaping queues the bursty traffic and smooths the stream to a certain degree; traffic policing simply drops the excess traffic, and lost data have to be retransmitted. Depending on the application type of VPN traffic, these two mechanisms can be correspondingly deployed. Note that an SP may need to condition the traffic leaving its core too, depending on the SLA negotiation at that boundary. One example of industry implementations is Cisco's generic traffic shaping (GTS). It regulates the data sending rate and drops the last packet in the queue once the queue is full.

3.4.3 Queueing and Scheduling

In the network core, the SLA conformable VPN traffic classes/aggregates are placed into different queues that are either logically or physically separated. Scheduling

strategies then determine the transmission order of enqueued packets, using priorities assigned to the packets by diverse schemes. A number of scheduling mechanisms adopted by industry vendors, such as Cisco, Alcatel, and Nortel, are selectively listed below. Note that certain industry implementations may be slightly different from their academic counterparts.

Class-based weighted fair queueing (CBWFQ): It extends weighted fair queueing (WFQ), by supporting user-defined VPN classes, for example, a mission-critical application class. Traffic belonging to a certain class is then assigned an appropriate bandwidth, buffer length or drop policy.

Low latency queueing (LLQ): Serving packets based on the weights, CBWFQ grants no class of packets a strict priority. This could introduce delay, especially delay jitter to voice applications. By adding a priority queue to CBWFQ, therefore, LLQ is designed to provide the explicit priority to delay-sensitive voice applications.

Hierarchical class based queueing (HCBQ): HCBQ divides the traffic into classes and their sub-classes as well. One sub-class can take the bandwidth from other sub-classes of the same class. Different scheduling methods can be accordingly adopted.

Modified deficit round robin (MDRR): Regular deficit round robin (DRR), in a round robin manner, provides every queue equal scheduling opportunities. As an approximation of LLQ, MDRR has one of its queues defined as the priority queue, thereby providing low delay and jitter to delay-sensitive applications such as voice over IP (VoIP).

3.4.4 Congestion Management

Congestion avoidance recognizes and acts upon the congestion so as to relieve or eliminate its negative effects on QoS guarantees. Among a variety of strategies, two

Table 3.1 A Comparison of IP VPNs and MPLS VPNs.

Forwarding speed	MPLS VPNs tend to have a faster forwarding speed than IP VPNs, by avoiding the IP header look-up and using the information in MPLS labels instead.
Traffic engineering signaling	Except the centralized management architecture, IP VPN implementations work on the adaptation of IntServ signaling in the DiffServ domain. RSVP-TE, a candidate signaling for MPLS VPNs, has been under development by the IETF MPLS working group (note that the MPLS working group has decided to stop implementing constraint-routing label distribution protocol (CR-LDP)).
Scalability	PE routers of IP VPNs maintain a full mesh of tunnels among all sites of a particular VPN, and P routers hold the information for all accommodated VPNs. Nevertheless, no single router in the MPLS VPN backbone has to maintain the routing information for all supported VPNs [56]. By using route reflectors in MPLS VPNs, the scalability hazard of maintaining a full mesh of inter-site VPN connectivity is also eliminated.
IP address space	IP VPN traffic needs globally unique IP address to cross the IP core, whereas MPLS VPN subscribers can use globally unique address space, private IP address space, or even overlapping address space.
Security	IP VPNs can support strict information confidentiality by configuring IPSec security associations in PE routers among VPN sites. MPLS VPNs, by itself, provide equivalent security to layer 2 VPNs, but have no direct support for authentication and confidentiality. In addition to SP PE routers, therefore, the intermediate routers belonging to different MPLS administrative domains must be trusted.
Multi-provider environment	While a notable amount of work on the inter-domain federation has been done for IP VPNs, the same issue in MPLS VPNs has not yet created a firm basis owing to the lack of inter-operable standards.

classical congestion avoidance mechanisms adopted by leading VPN industry vendors are briefly described as below:

Random early detection (RED): The average queue size is calculated to compare with two thresholds, one minimum queue size and another maximum queue size. Below the minimum limit, no packet is marked; above the maximum threshold, every packet is marked. In between these two, packets are marked with a probability that is a function of the average queue size. The packets are then randomly dropped at the moment of congestion, attempting to avoid the global synchronization when multiple TCP streams change their rates [57].

Weighted RED (WRED): Combining the RED mechanism and different classification scenarios, it provides the preferential traffic handling and thus differentiated performances for service classes, by selectively discarding lower priority traffic at the moment of congestion. As in RED, network engineers have the flexibility to configure the minimum and maximum queue length thresholds as well as drop probabilities of each service class.

3.5 MPLS-based VPNs

Envisioning a backbone that supports QoS, MPLS entails significant changes in existing IP network architectures. As a hybrid of the Layer 3 and Layer 2 structures, it forwards layer 3 packets like a layer 2 switch, thereby taking advantages of layer 3 routing intelligence and layer 2 fast forwarding capabilities.

As one of the technical approaches for IP-based VPN implementations, MPLS is more than another innovative paradigm owing to its unique characteristics. First, MPLS-enabled routers or switches attach labels to packets according to FEC, and then forward packets based on the MPLS label instead of conventional IP address look-up. Second, instead of routing the packets through the network, MPLS passes on packets to the destination by swapping or peeling away their labels hop by hop.

Third, forwarding packets based on the labels and distributing the labels with routing protocols, MPLS-enabled devices separate these two functionalities. It introduces more implementation flexibility as compared to IP routers that couple forwarding decisions with the generation of routing information [58].

Although MPLS does address the QoS issue, its original motivation was more on improving the Internet scalability through better traffic engineering. Nevertheless, this does not hinder MPLS based VPNs to phenomenally gaining momentum. For instance, in June 2002, AT&T announced MPLS based IP VPN services in Australia. The QoS issue of MPLS VPNs, however, needs to be investigated from another, if not totally different, angle, and thus is beyond the scope of this chapter. As a matter of fact, since SPs will probably prefer retaining existing enterprise subscribers and gradually attracting new ones, both types of VPNs will exist alongside one another in years to come. To furnish a general rather than an exhaustive comparison, the differences between IP VPNs addressed in this chapter and MPLS VPNs are listed in Table 3.1.

3.6 Chapter Summary

Although several technologies for delivering IP VPNs are still in the "cloud," this booming service is adapting and gaining ground at a surprising speed as standard bodies, industry vendors, and research communities are pushing one another ahead. This chapter presented a QoS guarantee framework for IP VPNs. QoS operations from the VPN network perspective determine the QoS configuration parameters; routers at the node level are then configured in concert to enforce the end-to-end QoS. Diverse VPN QoS enabling technologies as well as development progresses from recent research and engineering work had been addressed, to complete the whole picture of the IP VPN QoS issue.

CHAPTER 4

"PACKET SHORTAGE" PHENOMENON AND "DEBT-AWARE" ENHANCEMENT

This chapter looks into the loss aspect of the proportional differentiation model. With respect to the PLR dropping mechanism, the "packet shortage" phenomenon is investigated. The failure of using the buffer resource to relieve the "packet shortage" phenomenon is implied by the difficulty of obtaining the close form expression, and is further verified by simulation results. Subsequently, the "debt-aware" enhancement is proposed; its merits are illustrated by analysis and simulations.

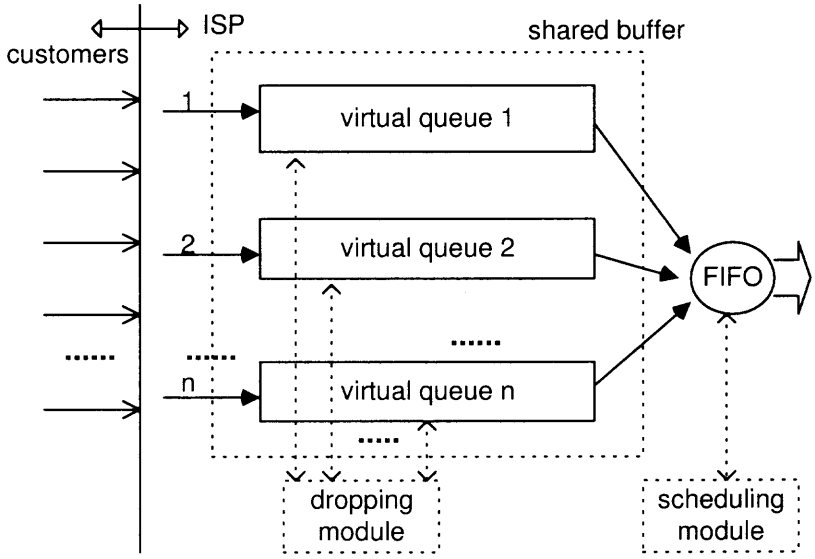


Figure 4.1 System model.

4.1 System and Traffic Models

A buffer/queue unit residing at the ISP network edge is assumed to support n classes of services, one for each class selection PHB. As shown in Fig. 4.1, the shared buffer consists of n logical queues, each associated with a class, respectively. The

scheduling module is assumed first in first out (FIFO). ISP is then challenged to provide differentiated losses among classes.

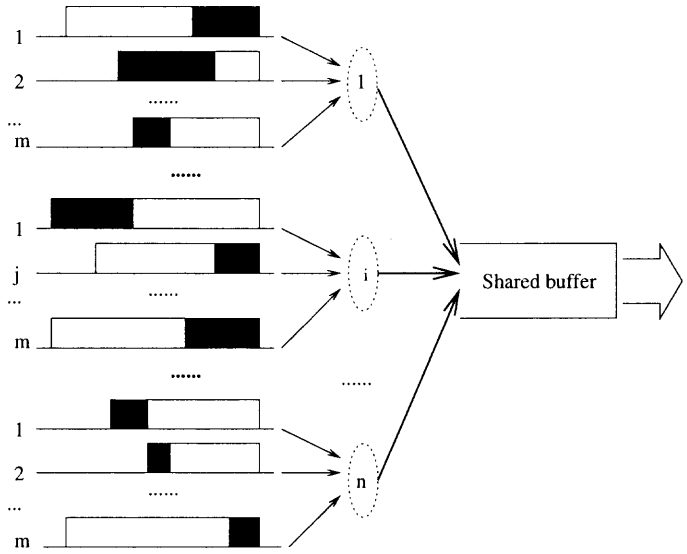


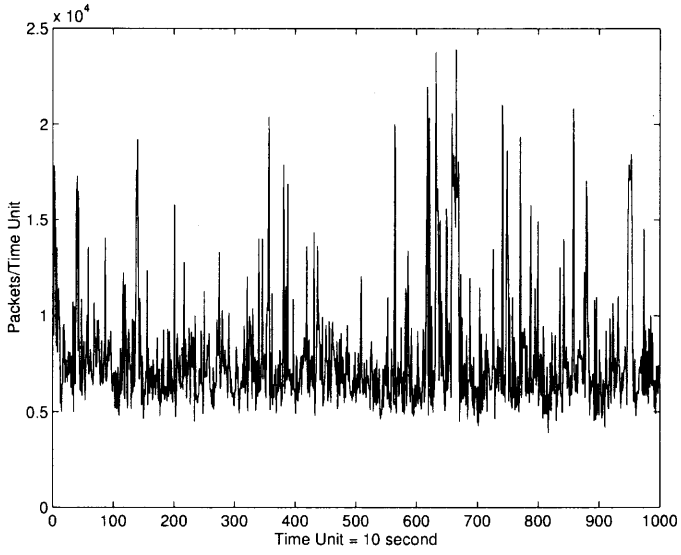
Figure 4.2 N classes of self-similar traffics are superpositioned from m ON-OFF sources, respectively.

One of the popular network traffic models, which is claimed to be simple, accurate, and realistic, is the one with the self-similarity (or long-range dependency) characteristic. According to empirical studies and mathematical results [59], the superposition of multiple Pareto distributed ON-OFF sources is adopted to produce such kind of traffic. As depicted in Fig. 4.2. The j_{th} ON-OFF source in class i is defined by a scale parameter $\alpha_{i,j}$, a lower cut-off of ON periods $b_{on_{i,j}}$, and a lower cut-off of OFF periods $b_{off_{i,j}}$, where $i = 1, 2, ..., n, j = 1, 2, ..., m$. Therefore, the probability density function (PDF) of an ON period $x_{on_{i,j}}$ follows:

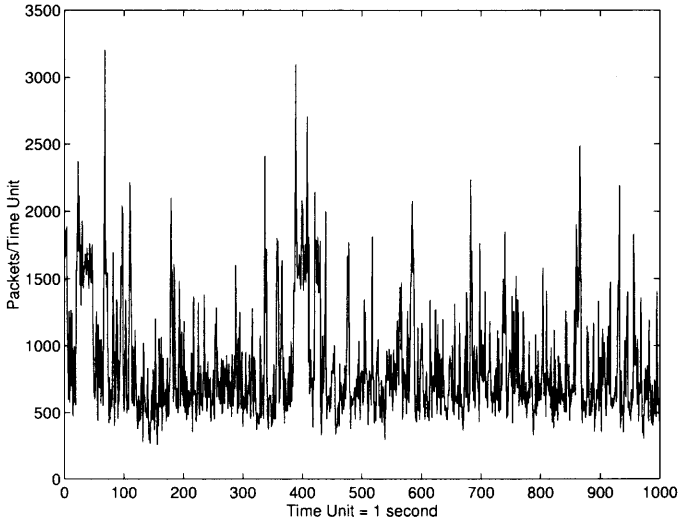
$$f_{X_{on}}(x_{on_{i,j}}) = \frac{\alpha_{i,j}(b_{on_{i,j}})^{\alpha_{i,j}}}{(x_{on_{i,j}})^{\alpha_{i,j}+1}}, \quad x_{on_{i,j}} \geq b_{on_{i,j}}, \tag{4.1}$$

and that of an OFF period $x_{off_{i,j}}$ is expressed as

$$f_{X_{off}}(x_{off_{i,j}}) = \frac{\alpha_{i,j}(b_{off_{i,j}})^{\alpha_{i,j}}}{(x_{off_{i,j}})^{\alpha_{i,j}+1}}, \quad x_{off_{i,j}} \geq b_{off_{i,j}}. \tag{4.2}$$



(a) 10-second interval.



(b) 1-second interval.

Figure 4.3 Self-similar traffic traces at different time intervals.

Several parameter assumptions are applied to simulations. First, three service classes are considered. Second, homogeneous ON-OFF sources are adopted to generate traffic aggregates, whose feasibility has already been proved [59]. Suggested by earlier empirical studies [60], typical scale parameters $\alpha = 1.1, 1.2, 1.3$, and 1.8 are selectively used. Third, the lower cut-offs of ON and OFF time periods for three classes are $0.5ms, 1ms, 1, 5ms$, and $1.61ms, 2.9ms, 4.85ms$, respectively. Traces of one resulting traffic class are illustrated in Fig. 4.3, where its self-similar characteristic at different time scales easily passes the "virtual" test. Additionally, sample points in the simulations are measured every $100K$ packet arrivals. All simulations assume a buffer length of $B = 2K$ *packets*. All packets have a constant length of $1K$ *bits*. The traffic load, ρ, defined as the ratio of the average arrival to the service rate, is specified in each figure.

4.2 "Packet Shortage" Phenomenon

Two customized proportional loss rate (PLR) schemes, namely PLR(∞) and PLR(M), were proposed [12] to closely approximate the differentiation parameters in terms of the packet loss. In PLR(∞), the loss rate estimation l_i is the long-term fraction of packets from class i that have been dropped, being measured by counters for the arrivals and drops in each class. Denote A_i, D_i, and $B(t)$ as the counter of packet arrivals of class i, the counter of packet drops from class i, and the set of backlogged classes at time t, respectively. Whenever the buffer overflows, PLR(∞) drops a packet from the class whose index is determined from

$$\min_{i \in B(t)} \left(\frac{D_i}{\sigma_i A_i}\right), \ i = 1, 2, ..., n. \tag{4.3}$$

In PLR(M), the loss rate of class i is estimated by the fraction of dropped packets from class i in the last M arrivals; it has the same dropping strategy as PLR(∞).

As illustrated in (1.2), the PLR dropping policy intends to equalize the differentiated loss rates of classes by penalizing a backlogged class with the minimal differentiated loss rate. To enforce the proportion as specified in (1.2), this backlogged class, nevertheless, may not necessarily be the one which needs to be disciplined. The reason is as follows: a class with a smaller differentiated loss rate $\frac{l_i}{\sigma_i}$ is supposed to be dropped more often than others; however, this class will probably not be backlogged as often as others if its traffic load is too light. Therefore, the "packet shortage" phenomenon happens when the dropping module cannot push out packets from the designated class, because this class is not backlogged at the moment of overflow. Likely, a loss occurs to a class whichever happens to be backlogged at the time of dropping, but this class is not necessarily the one with the minimal differentiated loss rate among all classes.

As shown in Fig. 4.4(a), with the normalized traffic load distribution of three classes $(L_1, L_2, L_3) = (56\%, 30\%, 14\%)$, PLR($\infty$) closely approximates the loss differentiation to the parameters $\sigma_1 : \sigma_2 : \sigma_3 = 4 : 2 : 1$. Given a 10% QoS deviation defined in SLA, however, the agreement can be violated under certain circumstances. For instance, if a class with an aggressive loss rate ratio to others has a relatively light traffic load, it probably will not be backlogged as often as others and will suffer from "packet shortage." Not surprisingly, when picking up another load distribution $(L_1, L_2, L_3) = (14\%, 30\%, 56\%)$, the ratios $\frac{l_1}{l_2}$ and $\frac{l_1}{l_3}$ of PLR(∞) exhibit a 12.5% deviation from $\frac{\sigma_1}{\sigma_2}$ and $\frac{\sigma_1}{\sigma_3}$, respectively, as illustrated in Fig. 4.4(b). The PLR dropping mechanism, therefore, can be further enhanced to curb the "packet shortage" problem.

To help the dropping module find a more eligible packet, holding more packets in the buffer could be a potential solution. We then hope to find a buffer bound that is long enough to accommodate packets from all classes with a certain probability, if not a deterministic value.

Lemma 4.1: Assume that a traffic class is aggregated by n ON-OFF sources with scale parameters $\alpha_i, i = 1, 2, ...n$, respectively. The ON and OFF periods of source i follow Pareto distribution with lower cut-offs of b_{on_i} and b_{off_i}, $i = 1, 2, ..., n$, respectively. The time period Δt, in which at least one ON period from *any* class will be accommodated in the buffer, is expressed as

$$\Delta t \geq \min_i (b_{on_i} + b_{off_i}), \quad i = 1, 2, ..., n.$$

Proof: For every ON-OFF source, its Pareto distributed ON and OFF periods x_{on_i} and x_{off_i} have their mean values drawn from (4.1) and (4.2) as follows.

$$\begin{aligned} E(x_{on_i}) &= \frac{\alpha_i}{\alpha_i - 1} b_{on_i}, \quad x_{on_i} \geq b_{on_i}, \\ E(x_{off_i}) &= \frac{\alpha_i}{\alpha_i - 1} b_{off_i}, \quad x_{off_i} \geq b_{off_i}, \qquad i = 1, 2, ..., n. \end{aligned}$$

Since ON and OFF periods are alternate, we consider one pair of ON and OFF periods as a single unit. Moreover, the length of ON and OFF periods are independent, and the length of a pair of successive ON and OFF periods $z_i = x_{on_i} + x_{off_i}, i = 1, 2, ..., n$, has its mean value:

$$\begin{aligned} E(z_i) &= E(x_{on_i} + x_{off_i}) \\ &= E(x_{on_i}) + E(x_{off_i}) \\ &= \frac{\alpha_i}{\alpha_i - 1}(b_{on_i} + b_{off_i}), \; z_i \geq (b_{on_i} + b_{off_i}), \; i = 1, 2, ...n. \end{aligned}$$

Therefore, for n ON-OFF sources, each of which has a successive ON-OFF pair with the length of $z_i, i = 1, 2, ..., n$, the time period for a buffer to accommodate at least one ON period from *any* source is determined by

$$\Delta t \geq \min_i (z_i),$$

that is,

$$\Delta t \geq \min_i(b_{on_i} + b_{off_i}), \quad i = 1, 2, ..., n. \tag{4.4}$$

Lemma 4.2: Assume there are n traffic classes, each superpositioned by m ON-OFF sources as shown in Fig. 4.2. Denote $l_{i,j}$ and $r_{i,j}$ as the load and peak rate of the j_{th} ON-OFF source in class i, respectively; likewise, $b_{on_{i,j}}$ and $b_{off_{i,j}}$ the ON and OFF periods of the j_{th} ON-OFF source in class i, respectively. The buffer length B, which is sufficient to hold at least one ON period (i.e., one burst) from *every* class, is then determined by

$$B \geq \max_i \min_j (b_{on_{i,j}} + b_{off_{i,j}}) \times \sum_i r_{i,J} l_{i,J},$$
$$i = 1, 2, ..., n, \; j = 1, 2, ..., m, \; J = \arg\min_j (b_{on_{i,j}} + b_{off_{i,j}}).$$

Proof: In class i which is aggregated by m ON-OFF sources, according to Lemma 4.1, the time period Δt_i for a buffer to accommodate at least one pair of ON and OFF periods follows

$$\Delta t_i \geq \min_j (b_{on_{i,j}} + b_{off_{i,j}}), \; i = 1, 2, ..., n, \; j = 1, 2, ..., m.$$

Among n classes, however, to see at least one pair of ON and OFF periods from *every* class, the corresponding time period Δt shall satisfy:

$$\Delta t \geq \max_i \min_j (b_{on_{i,j}} + b_{off_{i,j}}), \; i = 1, 2, ..., n, \; j = 1, 2, ..., m. \tag{4.5}$$

Furthermore, for each pair of ON and OFF periods in the j_{th} source of class i, the average traffic arrival is determined by $r_{i,j} \times l_{i,j}$. The total traffic arrival T of n classes during period Δt turns out to be

$$T = \sum_i r_{i,J} \times l_{i,J},$$
$$i = 1, 2, ..., n, \; j = 1, 2, ..., m, \; J = \arg\min_j (b_{on_{i,j}} + b_{off_{i,j}}). \tag{4.6}$$

From (4.5) and (4.6), therefore, the buffer size B which is able to accommodate at least one ON period from *every* class is

$$B = \Delta t \times T \geq \max_i \min_j (b_{on_{i,j}} + b_{off_{i,j}}) \times \sum_i r_{i,J} l_{i,J},$$
$$i = 1, 2, ..., n,\ j = 1, 2, ..., m,\ J = \arg\min_j (b_{on_{i,j}} + b_{off_{i,j}}). \tag{4.7}$$

For the j_{th} source in class i, the traffic load and the length of a pair of ON and OFF periods are $l_{i,j} = \frac{x_{on_{i,j}}}{x_{on_{i,j}} + x_{off_{i,j}}}$ and $z_{i,j} = x_{on_{i,j}} + x_{off_{i,j}}$, respectively. Thus, buffer size B is generalized from (4.7) as

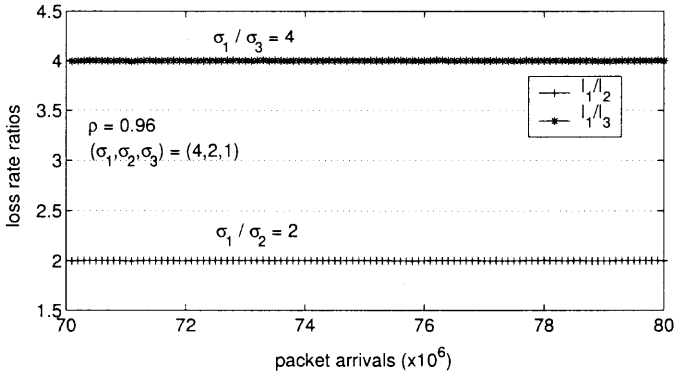
$$B(z_{i,j}, l_{i,j}) = \max_i \min_j z_{i,j} \times \sum_i r_{i,J} l_{i,J},\ \ J = \arg\min_j z_{i,j}. \tag{4.8}$$

Given an upper bound x, the probability of at least one ON period from *every* class accommodated in the buffer $P(B \leq x)$ is drawn from

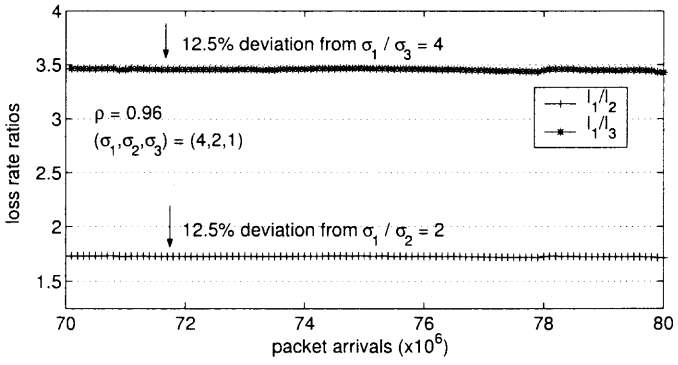
$$P(B \leq x) = \int_0^x f_B(x) dx, \tag{4.9}$$

where f_B is PDF of $B(z_{i,j}, l_{i,j})$. When planning the buffer bound within an available resource range, therefore, the bigger this probability value, the less chance buffer encounters the "packet shortage" problem. However, the characteristic function of the Pareto distribution is not integrable in a closed algebraic form; inversion methods of obtaining $f_z(x)$ and $f_l(x)$, that is, PDFs of $z_{i,j}$ and $l_{i,j}$, are not immediately applicable [61, 62]. This in turn rules out an explicit expression of $f_B(x)$ for network operators to estimate a straightforward buffer bound.

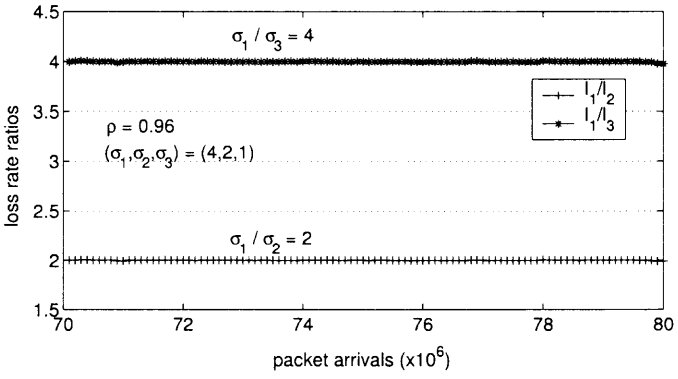
Using the same traffic load distribution which induces the rate ratio deviation in Fig. 4.4(b), the relationship of different buffer sizes and enforced loss rate ratios is illustrated in Fig. 4.5. Since the system service rate is $12K$ *packets/second* and all packets have the length of $1K$ *bits*, a buffer size ranging from 250 *packets* to $8K$ *packets* is utilized, by considering reasonable queueing delay constraints. As



(a) PLR(∞) with a normalized load $(L_1, L_2, L_3) = (56\%, 30\%, 14\%)$.

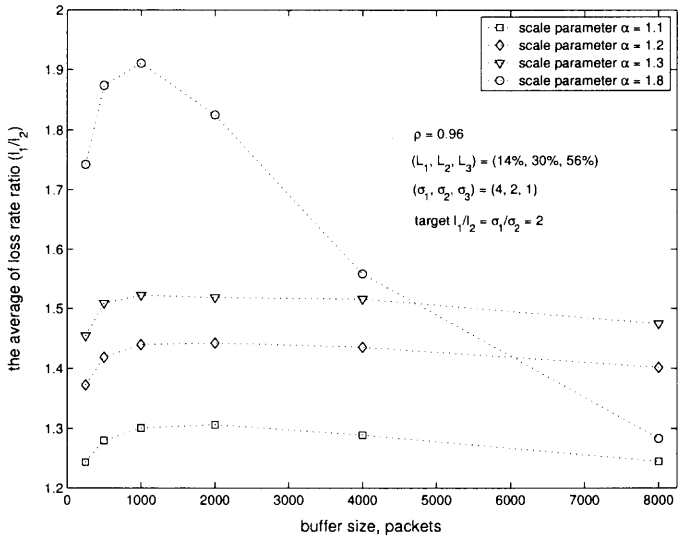


(b) PLR(∞) with a normalized load $(L_1, L_2, L_3) = (14\%, 30\%, 56\%)$.

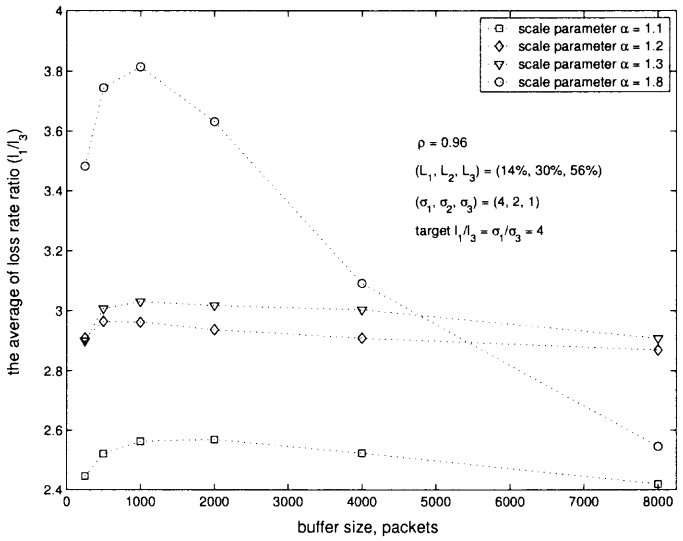


(c) "Debt-aware" with a normalized load $(L_1, L_2, L_3) = (14\%, 30\%, 56\%)$.

Figure 4.4 "packet shortage" phenomenon: (a) with an appropriate traffic load distri-bution, PLR(∞) approximates the targeted differentiation ratios well; (b) "packet shortage" caused by another traffic load distribution, how-ever, induces an about 12.5% deviation to both rate ratios of PLR(∞); (c) alleviating the "packet shortage" problem, "debt-aware" closely approximates the required rate ratios.



(a) the average of loss rate ratio $\frac{l_1}{l_2}$ vs. buffer size.



(b) the average of loss rate ratio $\frac{l_1}{l_3}$ vs. buffer size.

Figure 4.5 Trend of enforced loss rate ratios over enlarging queue sizes.

illustrated in Fig. 4.5, the improvement of the rate ratio deviation gained by a dramatically expanding buffer size gets saturated soon.

The reason that all traces start decreasing after a certain point is explained by the following. When the buffer size increases, there are more packets backlogged at the moment of overflow. The dropping module may then have more opportunity to drop from a designated, light-loaded class, say, class 1. However, this advantage no longer applies when the buffer size is sufficiently large to help the dropping module locate all possible packets from class 1. Next, the class, which has the differentiated loss rate value next to that of class 1, will have to be dropped. With the loss rate of class 1 remaining the same and those of other classes increasing, their rate ratios decrease as observed in Fig. 4.5. Up to this point, there is no further possibility to maintain the required ratios among classes as defined by differentiation parameters. In other words, the buffer resource can no longer be considered as a means of improving the loss differentiation.

4.3 Enhanced "Debt-aware" Dropping Scheme

From the previous discussion and simulations, we have learned that the buffer resource is not reliable for relieving the "packet shortage" phenomenon. Furthermore, features an enhanced dropping method shall have are three-fold: closely approximating loss differentiation parameters by relieving the "packet shortage" phenomenon; dropping packets whenever it is necessary; and still being based on simple on-line measurements.

An enhanced proportional dropping method [63] with a "drop debt" memory, referred to as "debt-aware," is therefore suggested. Instead of only considering backlogged classes, this method monitors all classes. It first sorts out ratios $\frac{D_i}{\delta_i A_i}$, $i = 1, 2, ..., n$, in an ascending order. A new array $H[n]$ is then introduced to hold the sorted values. Each element of this array is a structure variable with two members: the "*value*" field records the loss rate ratio, and the "*index*" field records the corresponding

class index. When the buffer overflows, the dropping module scans through array $H[n]$ until it hits the first backlogged class, say, $H[k].index, 0 \leq k \leq n-1$; this is also the class with the minimal value of $\frac{D_i}{\sigma_i A_i}$ among all backlogged classes. Another array $Q[n]$ is adopted to record the "debt" of each class. Once a packet from class $H[k].index$ is pushed out, $Q[H[i].index], 0 \leq i < k$, will be increased by one. If the buffer does not overflow, the dropping module pays back the "debt" registered in $Q[n]$ in a round robin manner, before accepting an incoming packet. The pseudo code is listed in Fig. 4.6.

Before looking into simulation results, essential characteristics and advantages of "debt-aware" are summarized as follows: first, it expands the reach of the dropping module to incoming packets, and thus partially curbs the adverse effect of the traffic load on the system performance. Second, a dropping takes place when there is a "debt." This "debt" memory is exactly the effort to immediately identify packets in the buffer which will eventually be pushed out. Dropping these packets at an earlier stage can not only avoid causing the loss of other packets, but can also improve the queueing delay performance. Third, taking the cheap memory and the fast access speed of digital circuits into consideration, the complexity of the system does not significantly increase, with an extra register for each of the limited number of service classes.

With the same load distribution which induces a 12.5% performance deviation of PLR(∞) in Fig. 4.4(b), "debt-aware" curbs the rate ratios back to their criteria, as shown in Fig. 4.4(c). In addition, "debt-aware" is able to achieve the equivalent performance of PLR(∞) under normal load distribution. One may argue that "debt-aware" drops packets too aggressively; this is not completely true. Since both PLR(∞) and "debt-aware" aim to curb loss rate ratios even over short time periods, PLR(∞) will eventually push out whatever is supposed to be dropped. Therefore, "debt-aware" does not over-drop, but just do so at an earlier stage. For the policing purpose,

D_i: the number of packets dropped from class i.

A_i: the number of packets arrived to class i.

δ_i: the loss differentiation parameter of class i.

d_i: the "drop debt" carried by class i.

A class i packet arrives, A_i++;

if (the buffer overflows)

{ sort $\frac{D_i}{\delta_i A_i}$, $i = 1, 2, ..., n$, in an ascending order;

find an eligible class j, $j = \arg\min_{i \in B(t)}(\frac{D_i}{\delta_i A_i})$, where $i = 1, 2, ..., n$, and $B(t)$ is the set of backlogged classes;

update the "drop debt" counters, that is, d_k++, where $k = 1, 2, ..., j-1$;

if (i != j) drop the packet at the tail of class j;

else block the incoming packet;

D_j++; }

$else$

{ loop through "drop debt" counters, and pick up class k which is backlogged;

d_k--;

drop the packet at the tail of class k, D_k++;

Accept the incoming packet; }

Figure 4.6 Pseudo code of "debt-aware."

PLR(∞) may not be strict enough because a biased ratio can go on for an unknown period of time until it is regulated in one of the succeeding overflow moments or SPs select another service class for the subscriber, whichever comes first.

Another enhancement of "debt-aware" is that it improves the performance of packet queueing delay, by distinguishing certain packets at an earlier stage. Since PLR(∞) only drops at the moments of overflow, certain packets may stay in the buffer and delay other packets during the interval of two successive overflows, until they either are finally dropped or leave the queue. Fig. 4.7 demonstrates three snapshots of a buffer, where every packet is marked with its class index. In the snapshot shown in Fig. 4.7(a), the arriving and serving processes are keeping a dynamic balance, whereby the buffer is full but does not overflow. Then a burst comes and an overflow is about to happen, as shown in Fig. 4.7(b). Denote the service time of one packet as d. Assume that the serving process of the last packet has finished and class 1 has the minimal value of $\frac{D_i}{\sigma_i A_i}, i = 1, 2, ..., n$. Under this circumstance, the dropping module of PLR(∞) will drop the tail packet of class 1; the "debt-aware" case, however, could have blocked out this very packet upon its arrival, if class 1 carries a "debt." The consequence of the PLR(∞) scenario, as demonstrated in Fig. 4.7(b), is that all packets behind the discarded one are penalized with an extra delay of d. Before the overflow is over, the same situation could happen again. As illustrated in Fig. 4.7(c), the tail packet of class 3 happens to be the next one eligible to be dropped, and therefore packets lining up behind this one suffer from another queueing delay of d in the PLR(∞) case.

To further illustrate the regained delay, Fig. 4.8(a) plots the individual packet queueing delay in a very short but typical period. Both traces are quite thick due to the very high sample density. With its sample trace completely under that of PLR(∞), "debt-aware" exhibits much less queueing delay than PLR(∞) does, confirming to the previous explanation. Another two traces with sparse samples are

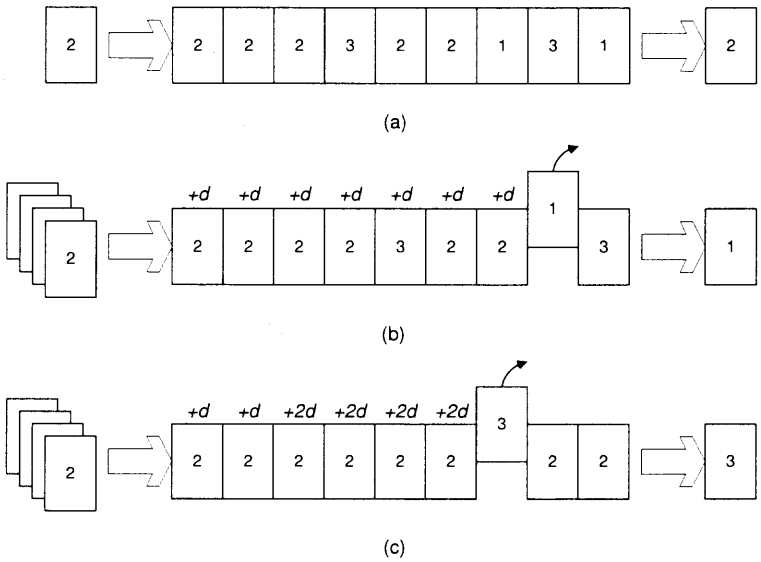
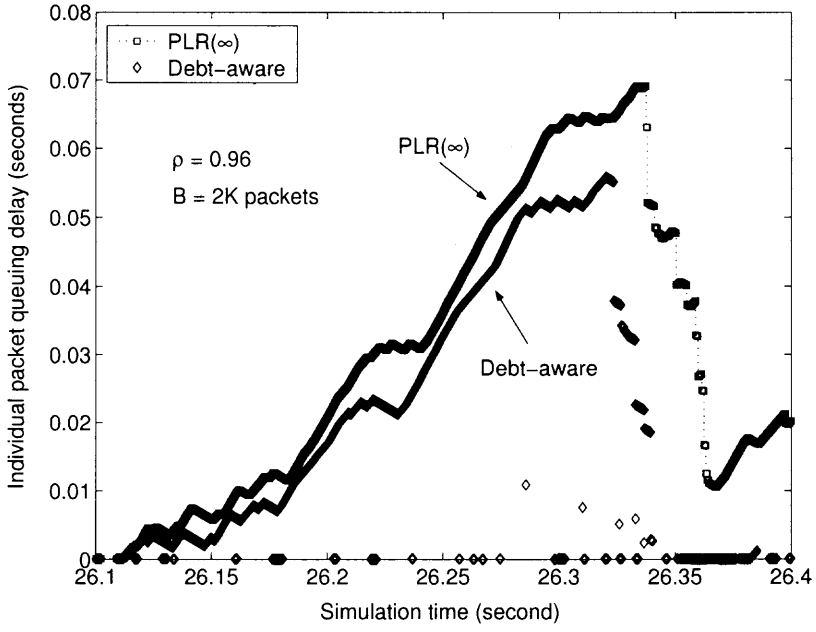
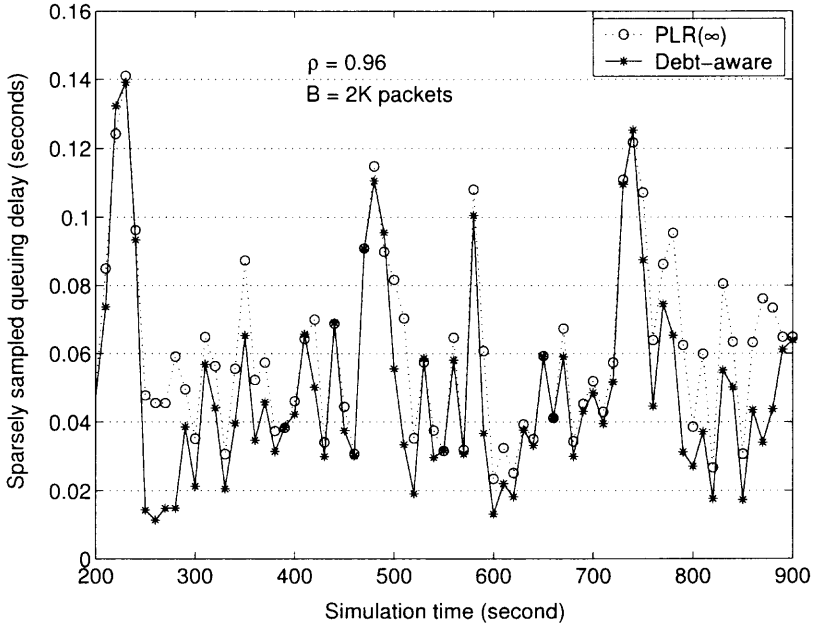


Figure 4.7 Snapshots demonstrating the excess queueing delay which can be regained by "debt-aware."

presented in Fig. 4.8(b), both of which contain 70 samples in a 700-second simulation period. Frequently, "debt-aware" exhibits smaller queueing delay as compared to PLR(∞), except at very few sample points. It may appear that the decreased queueing delay is resulted by a more aggressive dropping, instead of the early stage dropping feature of "debt-aware." However, the existence of exceptional sample points in Fig. 4.8(b), although very few, is exactly a good counterevidence: if the performance was simply gained by the aggressive dropping, all queueing delay values of "debt-aware" must have been lower than or at least equal to those of PLR(∞). The possible reason of these few exceptional values, nevertheless, can be explained as the following. Since "debt-aware" does the early dropping in a round robin manner which treats all classes equally, it may not follow the dynamic dropping order among classes as PLR(∞) does. Assume that "debt-aware" picks up packet A and PLR(∞) chooses packet B at the same overflow instance. If packet A is behind packet B in the queue, all packets between packet A and B will experience one more measure of queueing delay (i.e., d) in "debt-aware"; this contributes to a longer delay experienced by certain sample points in "debt-aware."



(a)



(b)

Figure 4.8 Packet queueing delay with different sample density for PLR(∞) and "debt-aware": (a) demonstrates individual packet delay in a very short but typical time period, where the trace of "debt-aware" is below that of PLR(∞), and shows smaller queueing delay; (b) sparsely plots 70 samples in a 700-second simulation period, where "debt-aware" fre-quently exhibits smaller queueing delay than PLR(∞) does.

4.4 Chapter Summary

This chapter addressed one aspect of the proportional differentiation: loss differentiation. The "packet shortage" phenomenon, caused by a lightly-loaded service class with aggressive loss rate ratios to other classes, had been investigated. We have demonstrated by analysis and simulations that this "packet shortage" problem cannot be curbed with an explicit buffer bound or be eliminated by simply enlarging the buffer size. Referred to as "debt-aware," an enhanced measurement-based dropping method was then suggested and evaluated. By simply adding one register/counter to each service class and blocking packets at an earlier stage, "debt-aware" partially curbs the "packet shortage" phenomenon to closely approximate loss differentiation parameters, and improves the queueing delay performance.

CHAPTER 5

PROPERTIES OF THE AVERAGE DELAY DIFFERENCE AND THE COMBINED DELAY DIFFERENTIATION SCHEME

Continuing from the previous work, this chapter first derives properties of the average delay difference among classes. Simulations and discussion have applied to two delay differentiation mechanisms PAD and WTP, examining the consistency of their differentiation guarantees. A combined delay differentiation scheme is then suggested to compromise these two mechanisms, aiming for a better differentiation performance over both short and long time periods.

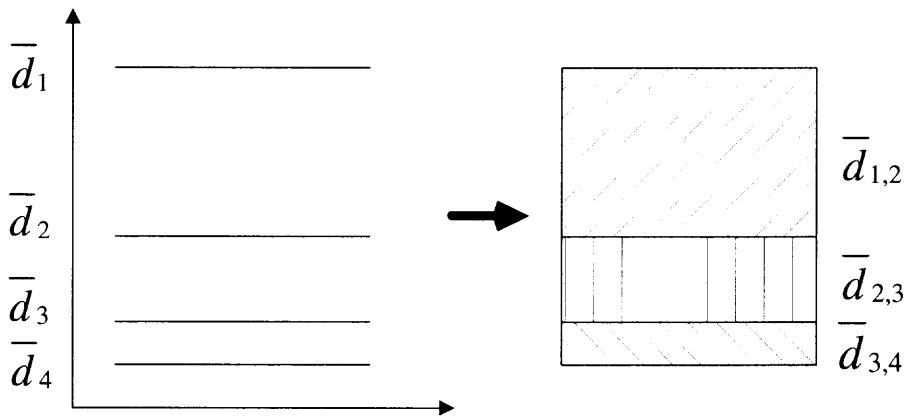


Figure 5.1 Differences of average class delay.

5.1 Properties of the Average Delay Difference

To understand the system dynamics, the average delay difference of successive classes is of interest. While the delay differentiation parameters δ_1 and δ_n, together with the available system sources, determine the delay range, the "gaps" between delays of successive classes are depicted in Fig. 5.1. To which direction these "gaps," that is, $\bar{d}_{1,2}$, $\bar{d}_{2,3}$, and $\bar{d}_{3,4}$ will move and whether or not they will expand help furnish the system behavior.

Adopting the conservation law of the mean waiting time (Appendix B), the sum of class waiting times weighed by the mean number of class arrivals is an invariant with respect to the scheduling mechanism. Therefore, in a scheduling system [15] with n classes, where class i has the arrival rate λ_i, average packet length $\bar{L}_i$, and average delay $\bar{d}_i$, we have:

$$\sum_{i=1}^{n} \lambda_i \bar{L}_i \bar{d}_i = \bar{Q}. \tag{5.1}$$

Assume that the delay differentiation defined by the proportional parameters is met. From (1.5), the average delay difference between any successive classes follow

$$\bar{d}_i - \bar{d}_{i+1} = (1 - \delta_{i+1}) \left(\prod_{k=1}^{i} \delta_k \right) \bar{d}_1, \quad i = 1, 2, ...n - 1. \tag{5.2}$$

From (5.1) and (5.2), the average delay differences between successive classes $\bar{d}_{i,i+1}$ are

$$\bar{d}_{i,i+1} = \left(\prod_{k=1}^{i} \delta_k \right) \frac{(1 - \delta_{i+1})\, \bar{Q}}{\sum\limits_{k=1}^{n} \lambda_k \bar{d}_k}, i = 1, 2, ..., n - 1. \tag{5.3}$$

The average packet length is set to one unit [15], and the normalized average queue length $\bar{Q}$ is then measured in average packet units.

As justified before, the average delay difference [64] is expected to reveal further information of the system behavior, although its practical meaning can vary with different application scenarios. For instance, the parameter change that causes a "gap" to move up as well as "expand" itself is probably worthy of attention; it can potentially violate QoS if the applications have certain delay bounds specified.

The following two properties reflect the delay difference behavior with respect to varying arrival rates.

Property 5.1: When $\lambda_i, i = 1, 2, ..., n$, increases, all average delay differences and the system delay range ΔG increase.

Proof: It has been proved [15] that increasing the arrival rate of a class will increase the average delays of all classes. When d_1 in (5.2) increases, all average delay differences between successive classes increase. The delay range of the system $\Delta G = \bar{d}_{1,2} + \bar{d}_{2,3} + ... + \bar{d}_{n-1,n} = \bar{d}_1 - \bar{d}_n$, furthermore, increases too.

Property 5.2: Increasing the arrival rate of a higher class introduces a bigger increase on all average delay differences, thereby resulting in a bigger delay range ΔG.

Proof: Refer to Appendix A.

Provided that SPs upgrade or downgrade the QoS levels of subscribers by switching them to a higher class or lower class, we have the following properties.

Property 5.3: When one or multiple subscribers move to a higher class, all average delay differences increase, and so does the system delay range; otherwise, both metrics decrease.

Proof: Refer to Appendix A.

When the traffic arrivals and the load distribution are relatively stable, varying delay differentiation parameters introduce the following three properties.

Property 5.4: Increasing the delay differentiation parameter δ_i, the delay differences $\bar{d}_{m,m+1}, m = 1, 2, ..., i$, increase, and the delay differences $\bar{d}_{m,m+1}, m = i+1, i+2, ..., n-1$, decrease.

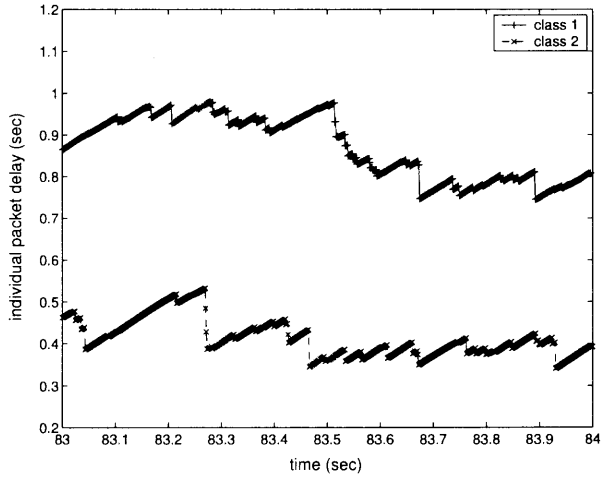
Proof: Refer to Appendix A.

Property 5.5: Increasing δ_1, the system delay range ΔG decreases. Increasing δ_n, the system delay range ΔG increases.

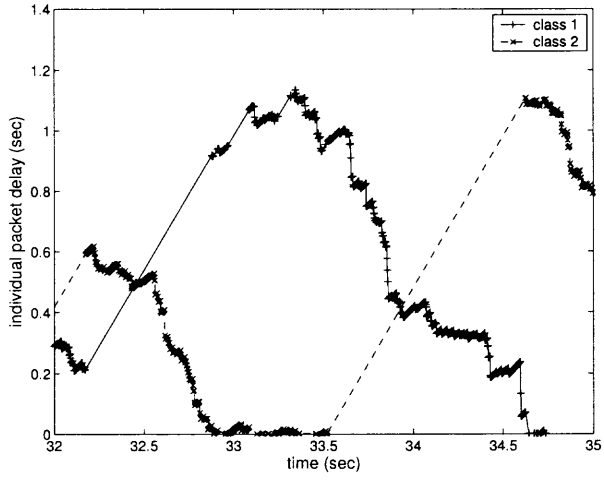
Proof: This is actually the special case of property 5.4. When δ_1 increases, all delay differences $\bar{d}_{i,i+1}, i = 1, 2, ..., n-1$, decrease. Therefore, the system delay range $\Delta G = \sum_{i=1}^{n-1} \bar{d}_{i,i+1}$ becomes smaller. On the contrary, when δ_n increases, the system delay range expands itself.

Property 5.6: Decreasing the delay differentiation parameter of a class decreases the delay difference between this class and the next higher class, and increases the delay difference between this class and the next lower class.

Proof: Refer to Appendix A.



(a)



(b)

Figure 5.2 Two scenarios of PAD (utilization factor $\rho = 0.97$, load distribution $(L_1, L_2) = (50\%, 50\%)$, targeted average delay ratio $\frac{d_1}{d_2} = 2$).

5.2 Discussion on PAD

The general idea of the PAD differentiation is to serve the class with the maximum normalized average delay; once it is no longer the maximum one, another class with the new maximum value will be served. Given that the scheduler keeps doing the same, the long-term normalized average delays of all classes are expected to be equalized (i.e., $\frac{d_i}{\delta_i} = \frac{d_j}{\delta_j}$), and thus average delays of classes eventually will be proportional to the differentiation parameters (i.e., $\frac{d_i}{d_j} = \frac{\delta_i}{\delta_j}$).

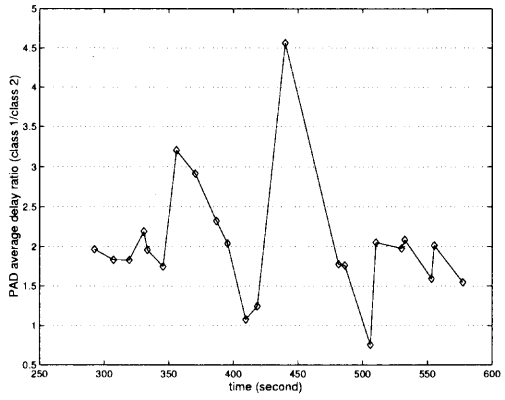
Simulations are applied to reveal more about this mechanism. To highlight the relationship between classes, only two traces of self-similar traffics are injected into the queue. The traffic loads and load distributions of classes are specified in the figures.

Two operation scenarios are identified. Fig. 5.2(a) is the stable and desirable scenario, where the class delays oscillate around a certain value and the targeted ratio is closely approximated. This situation, where two classes get to be served in turn, and with one normalized average delay value catching up with the other alternately, happens with "comparable" normalized average delays among classes. Nevertheless, Fig. 5.2(b) shows a distorted delay differentiation where the higher class falsely has lower delays. The cause of the distortion is explained below.

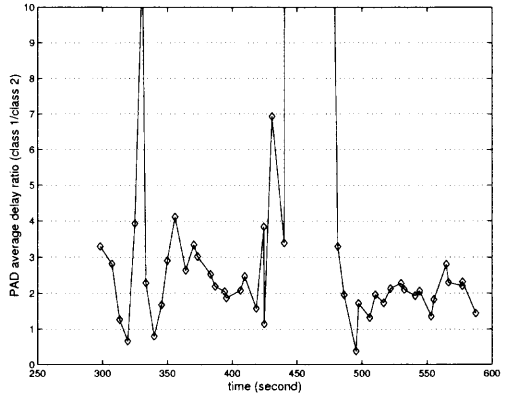
PAD schedules packets from the class with the maximum normalized average delay, and therefore stops the average delay of the class from increasing. There is an intuition that the maximum normalized average delay will keep decreasing with packets being scheduled. However, this is not always true.

Lemma 5.1: Given that class k has the maximum normalized average delay at time t, that is,

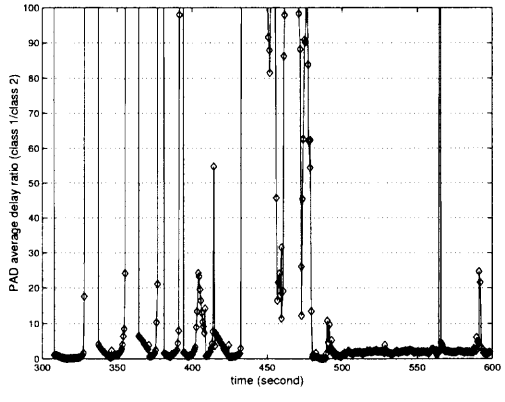
$$k = \arg\max_{i \in B(t)} \bar{d}_i(t). \tag{5.4}$$



(a) the average delay ratio per 20K packets.



(b) the average delay ratio per 10K packets.



(c) the average delay ratio per 1K packets.

Figure 5.3 Differentiation performances of PAD over different time periods (utilization factor $\rho = 0.85$, load distribution $(L_1, L_2) = (50\%, 50\%)$, targeted average delay ratio $\frac{d_1}{d_2} = 2$).

For the next packet departing at time t', to guarantee a decreasing normalized average delay, the delay of this packet shall not be bigger than the normalized average delay of the class it belongs to, that is,

$$d_i^{\Phi(S_i(t'))} < \frac{\sum_{m=1}^{\Phi(S_i(t))} d_i^m}{\delta_i \Phi(S_i(t))}. \tag{5.5}$$

Proof: Assume that at time t' a packet has departed; it has the queuing delay $d_i^{\Phi(S_i(t'))}$. From (1.9), the updated normalized average delay of the class turns to be

$$\bar{d}_i(t') = \frac{(\sum_{m=1}^{\Phi(S_i(t))} d_i^m) + d_i^{\Phi(S_i(t'))}}{\delta_i(\Phi(S_i(t)) + 1)} = \frac{\sum_{m=1}^{\Phi(S_i(t'))} d_i^m}{\delta_i \Phi(S_i(t'))}. \tag{5.6}$$

Provided that the normalized average delay decreases, that is,

$$\frac{(\sum_{m=1}^{\Phi(S_i(t))} d_i^m) + d_i^{\Phi(S_i(t'))}}{\delta_i(\Phi(S_i(t)) + 1)} = \bar{d}_i(t') \leq \bar{d}_i(t) = \frac{\sum_{m=1}^{\Phi(S_i(t))} d_i^m}{\delta_i \Phi(S_i(t))}, \tag{5.7}$$

the necessary condition shall be

$$d_i^{\Phi(S_i(t'))} < \frac{\sum_{m=1}^{\Phi(S_i(t))} d_i^m}{\delta_i \Phi(S_i(t))}. \tag{5.8}$$

However, it is possible for a class to accumulate a long waiting line of packets with long queuing delays. When this class gets served, its normalized average delay will not decrease until some "younger" packets, that is, a new burst of packets from the same class, participate to average down this value. In a multiple class environment, when this "increasing" duration gets longer, it puts another class on hold; this another class in turn passes on the negative effects to others.

What can be learned from Lemma 5.1 is that achieving the desired differentiation actually needs the packet delay information. The unawareness on the instantaneous packet waiting time, therefore, results in PAD's distorted delay differentiation at the

packet level. This in turn negatively affects PAD's differentiation performance over short time periods. As illustrated in Fig. 5.3, when the time period over which the delay ratio is enforced gets smaller, that is, from 20K packets to 1K packets, the achieved delay ratios deviate from the targeted values.

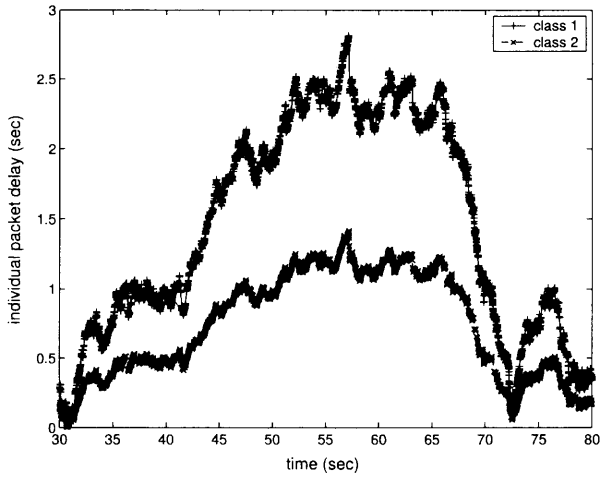
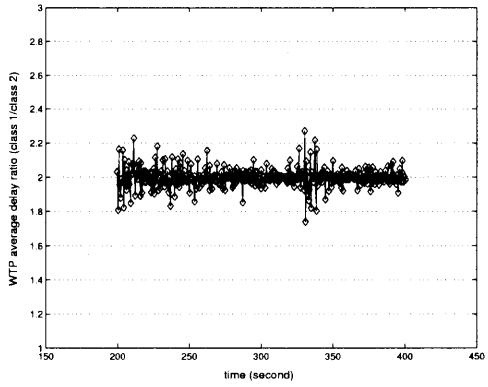


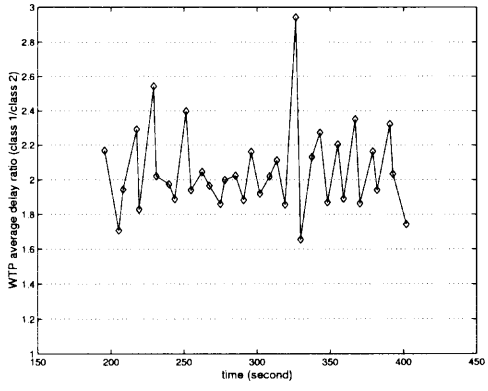
Figure 5.4 Differentiation performance of WTP at packet level (utilization factor $\rho = 0.97$, load distribution $(L_1, L_2) = (50\%, 50\%)$, targeted delay ratio $\frac{d_1}{d_2} = 2$).

5.3 Discussion on WTP

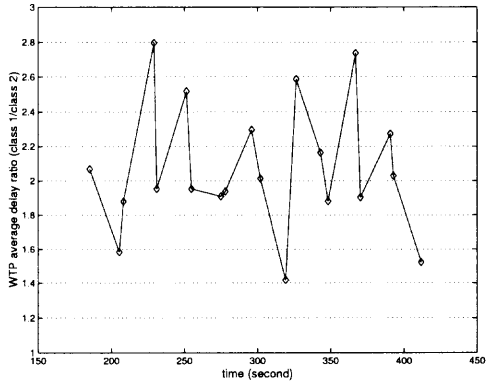
WTP overcomes the problem that PAD has no knowledge on individual packets' waiting times. This problem causes the delay differentiation distortion at the packet level. Equalizing the normalized waiting times of the packets waiting at the head of queues, WTP attempts to minimize the differences between the normalized waiting times of successively departing packets. It has been proved [15] that with Poisson arrivals, WTP converges to the original proportional delay differentiation model as the link utilization approaches 100%. In addition, empirical study showed that under the circumstance of self-similar traffic arrivals, WTP approximated PAD delay differentiation definition as the aggregated backlog (i.e., queue length) $\bar{Q}$ increases



(a) average delay ratio per 1K packets.



(b) average delay ratio per 10k packets.



(c) average delay ratio per 20K packets.

Figure 5.5 Differentiation performances of WTP over different time periods (utilization factor $\rho = 0.85$, load distribution $(L_1, L_2) = (50\%, 50\%)$, targeted delay ratio $\frac{d_1}{d_2} = 2$).

toward infinity. These investigations, nevertheless, imply that WTP may not enforce the differentiation over long long time periods.

In fact, should the packets from different classes be served in a strict alternative order, WTP is able to enforce the differentiation at the packet level as well as over longer time periods. Under other circumstances, which are unfortunately most of the situation, the obtained differentiation is no longer tractable when the time period gets longer. Obviously, the memoryless behavior on the packet delay negatively impacts the longer period differentiation enforcement.

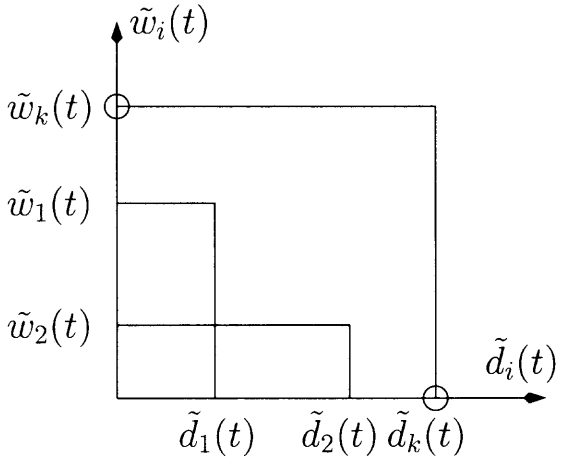
Using the same simulation parameters as those of PAD, the performance of WTP over different time periods have been investigated. As plotted in Fig. 5.4, WTP does provide higher class with lower delays even at the individual packet level, and therefore outperforms PAD in terms of the short time period delay differentiation. Nevertheless, as demonstrated in Fig. 5.5(a), (b), and (c), the differentiation guarantee of WTP weakens as the time period increases.

5.4 Combined Delay Differentiation (CDD)

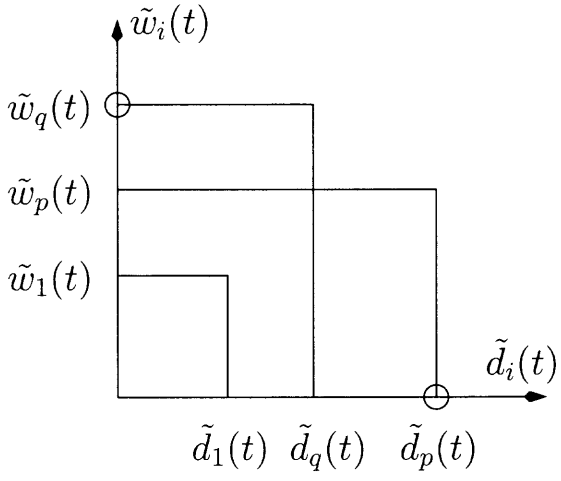
Concealing the instantaneous information of the packet waiting time inside the average class delay, PAD serves the class with the maximum normalized average delay. The normalized values are expected to be equalized (i.e., $\frac{d_i}{\delta_i} = \frac{d_j}{\delta_j}$), and the average delays are in turn proportional to differentiation parameters (i.e., $\frac{d_i}{d_j} = \frac{\delta_i}{\delta_j}$). The WTP scheme, on the other hand, serves the class with its packet waiting at the head of the queue having the maximum normalized delay. Minimizing the differences between the normalized waiting times of successively departing packets, WTP expects the class average delays to be eventually proportional to the differentiation parameters.

To summarize, PAD enforces the differentiation well at longer time periods, but tends to lose control when the time periods get shorter. WTP consistently provides

higher classes with lower delays at the packet level, but brings up differentiation uncertainty over longer time periods.



(a) class k has both the maximum values of $\tilde{d}_i(t)$ and $\tilde{w}_i(t)$.



(b) class p and class q have the maximum values of $\tilde{d}_i(t)$ and $\tilde{w}_i(t)$, respectively.

Figure 5.6 Selection strategies of CDD.

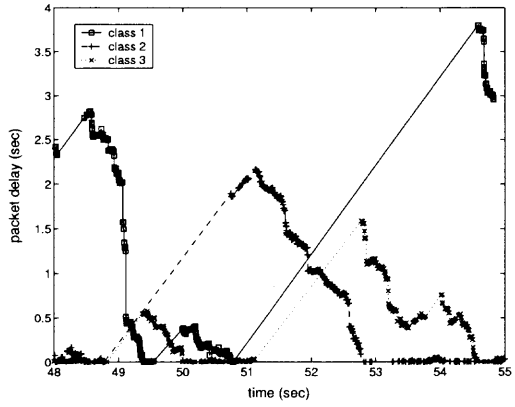
HPD [15], consequently, intends to balance the operation portion of PAD and WTP, by using an HPD parameter. When the HPD parameter approaches one, HPD becomes WTP; when the HPD parameter approaches zero, it becomes PAD.

Simulation results have shown that the empirical value of 0.875 brings the most "optimized" performance over both short and long time periods.

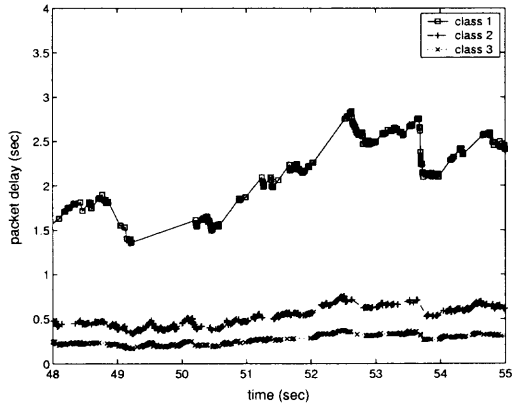
Rather than looking for an appropriate portion of the delay metrics used by PAD and WTP, nevertheless, the combined delay differentiation (CDD) approach is suggested to make scheduling decision based on both delay metrics. Note that the average class delays and instantaneous packet delays normalized by differentiation parameters at time t are denoted as $\tilde{d}_i(t)$ and $\tilde{w}_i(t)$, respectively. PAD serves the class with the maximum value of $\tilde{d}_i(t)$, and thus approximates differentiation over long time periods; WTP schedules packets from the class with the maximum value of $\tilde{w}_i(t)$, and therefore strictly enforce the differentiation at the packet level. Taking both metrics, that is, $\tilde{d}_i(t)$ and $\tilde{w}_i(t), i = 1, 2, ..., n$, into account, CDD first finds classes with the maximum values of $\tilde{d}_i(t)$ and $\tilde{w}_i(t)$. Then as illustrated in Fig. 5.6 (a), if one class, say, class k, has both maximum $\tilde{d}_i(t)$ and $\tilde{w}_i(t)$, CDD serves class k. Otherwise, assume that class p has maximum $\tilde{d}_i(t)$ and class q has maximum $\tilde{w}_i(t)$, as shown in Fig. 5.6(b). If the ratio of the average class delay of class q to that of class p, that is, $\frac{\tilde{d}_q(t)}{\tilde{d}_p(t)}$, is bigger than the ratio of the instantaneous packet delay of class p to that of class q, that is, $\frac{\tilde{w}_p(t)}{\tilde{w}_q(t)}$, class q will be served; otherwise, CDD serves class p. The intuition behind CDD is that scheduling a class with both maximum $\tilde{d}_i(t)$ and $\tilde{w}_i(t)$ is the most favored action, if such a class exists. Otherwise, between these two classes, the scheduler ensures that the selected class has at least one maximum metric, and has the other metric closer to the maximum value of its kind. Looking for a "middle ground" solution, CDD takes into account of both delay metrics, which enforce the differentiation either at the packet level or over the average class delay.

5.5 Simulation Results and Discussion

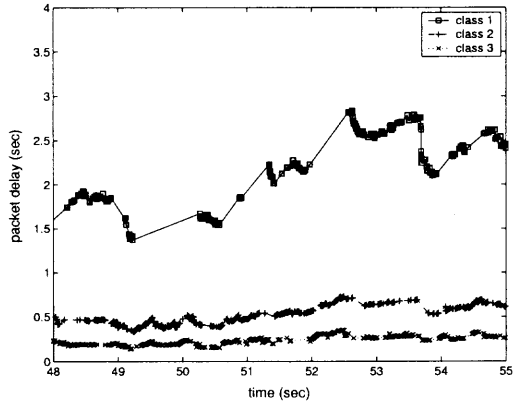
The capability of the three methods on providing higher classes with lower delays are plotted in Fig. 5.7. PAD shows distorted delay differentiation often. For instance,



(a) PAD.

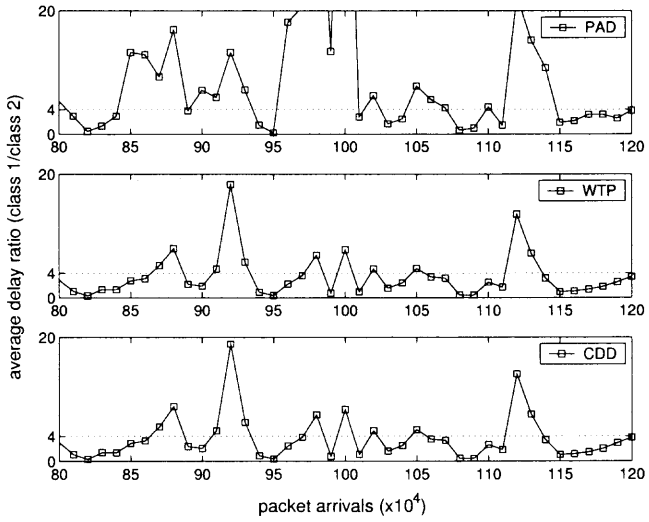


(b) WTP.

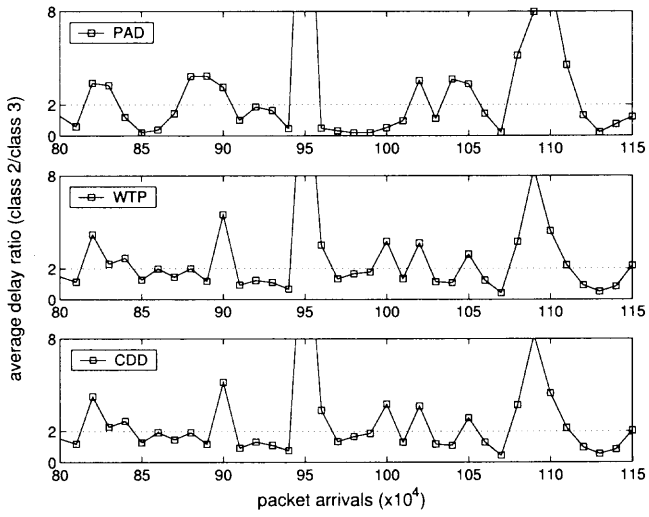


(c) CDD.

Figure 5.7 Packet level delay differentiation performances of PAD, WTP, and CDD (utilization factor $\rho = 0.94$, load distribution $(L_1, L_2, L_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$).



(a) the average delay ratio of class 1 to class 2.



(b) the average delay ratio of class 2 to class 3.

Figure 5.8 Delay differentiation performances of PAD, WTP, and CDD over a period of 10K packet arrivals (utilization factor $\rho = 0.94$, load distribution $(L_1, L_2, L_3)=(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, targeted delay ratios $\frac{d_1}{d_2} = 4$, $\frac{d_2}{d_3} = 2$).

at around the simulation time 51, class 2 falsely has higher delay than class 1. At the simulation time of 53, class 3 experiences higher delay than class 2. CDD, with higher classes consistently having lower delays, performs as well as WTP which is considered as the best for packet level differentiation. Showing the short time period performance, Fig. 5.8 illustrates two average delay ratios of all methods. There are many more samples of WTP and CDD that are around the targeted delay ratios, that is, 4 and 2, respectively, than those of PAD. Besides, PAD has more samples that are far from the desired delay ratio values, which is another sign of weak performance on short time period differentiation. Obviously, CDD outperforms PAD, and closely approximates WTP, with respect to the differentiation at the packet level and over short time periods.

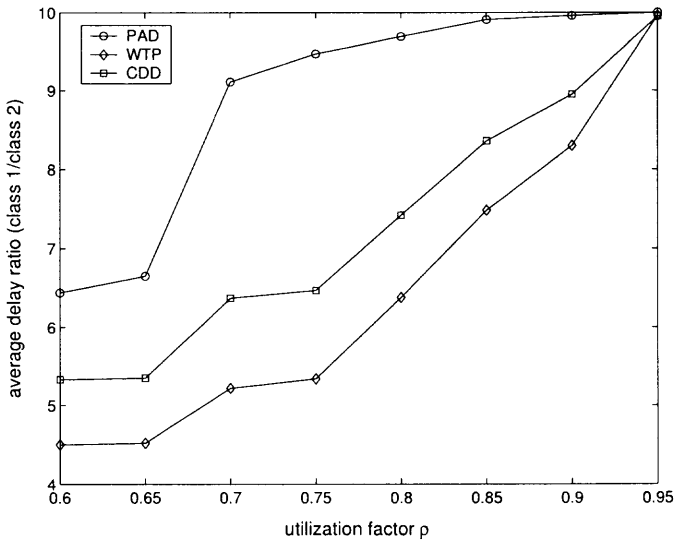


Figure 5.9 Performance comparison on the average delay ratios of PAD, WTP, and CDD (load distribution $(L_1, L_2) = (60\%, 40\%)$, targeted delay ratio $\frac{d_1}{d_2} = 10$).

The differentiation performances over long time periods are simulated with two classes. Under a preset system configuration, the targeted delay ratio is set as 10 so that all schemes need to "stretch" to meet the differentiation requirements. The

average delay ratios collected with respect to the increasing system utilization factor is then plotted in Fig. 5.9. When the system utilization factor ρ decreases, all mechanisms have a certain difficulty to enforce the differentiation. While PAD, as expected, performs the best, CDD shows better resistance on the decreasing utilization factor than WTP does.

Let's look at the differentiation performance over short time periods as a scale bar. PAD and WTP lie at both ends representing the worst and best possible performances, and CDD closely approximates WTP. Turning to the long time period performance, where PAD and WTP again locate at both ends but representing the best and worst possibilities, CDD is also somewhere in the middle. Depending on what is more concerned by SPs, that is, long-term average class delay, consistent better services to higher classes, or consistent better services to higher classes plus an acceptable level of average delay differentiation, they can accordingly deploy PAD, WTP, or CDD.

5.6 Chapter Summary

This chapter presented the properties of the average delay difference of the proportional delay differentiation, which furnished system dynamics from another aspect. The performances of two classic delay differentiation schemes PAD and WTP were investigated by analysis and simulations, to reveal some basic principles. A combined delay differentiation method was subsequently proposed, to find a "middle ground" between the short time and long time period differentiation performances. The chapter drew the conclusion that at the current stage, PAD and WTP are the best for long time and short time period differentiation, respectively; CDD closely approximates WTP and outperforms PAD in short time period delay differentiation, and surpasses WTP in long time period delay differentiation. For a system where both differentiations are important, CDD is a better choice than the other two.

CHAPTER 6

COMPUTATION OF LOSS DIFFERENTIATION PARAMETERS FOR PROPORTIONAL QOS DIFFERENTIATION

While a large amount of work [12, 15, 63, 65, 64] has been done on enforcing the QoS differentiation based on the differentiation parameters, how to select feasible parameters is also critical. Chances are that the QoS differentiation may not be fulfilled if the chosen parameters does not comply with the system condition, such as the traffic load and load distribution. The feasibility issue of the delay differentiation parameters were discussed [15] in detail, by taking the strict priority (SP) scheduler as the reference system to determine appropriate parameters. Investigated by simulations [12], nevertheless, that of the loss differentiation parameters was not more than an intuition based on network operators' experience. As to the best of our knowledge, no clear-cut ideas were suggested to select the loss differentiation parameters. A guideline on selecting these parameters, therefore, is called for. As a general goal, the indication on the loss differentiation parameters cannot be random, and it shall not be deterministic either. The intent of this chapter is to introduce a quantitative expression to compute the loss differentiation parameters.

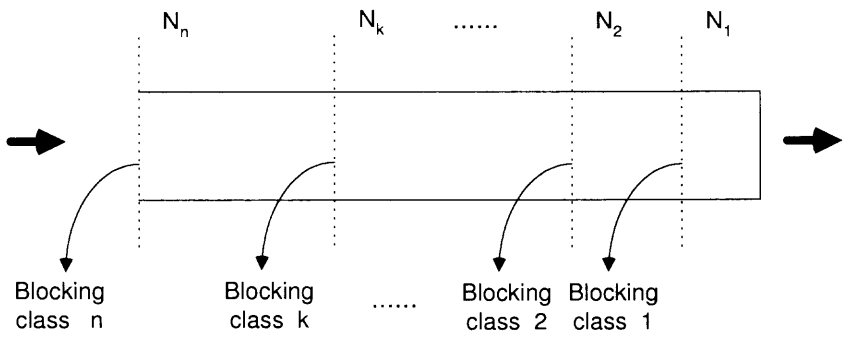


Figure 6.1 Queueing model.

6.1 System Model and the New Approach

Supporting n service classes, a buffer/queue accommodates a FIFO module that determines which class shall be served next, and a dropping module that decides when and which packets to be dropped. In a single server queue, as depicted in Fig. 6.1, several blocking thresholds $N_i, i = 1, 2, ..., n$, can be adopted to distinguish service priorities. In other words, when the buffer content reaches N_i, the dropping module starts blocking traffic from class i; the service priorities are then reflected by the value of these blocking thresholds. These thresholds can be a good reference to the differentiation parameters, in the sense of differentiating service priorities. Though a large amount of work has been done on analyzing various loss systems such as the reciprocity of blocking probabilities [66] and retry blocking probabilities [67], to the best of our knowledge, none has addressed the optimization problem discussed here. Various properties and analyses presented in this chapter are thus derived based on the fundamental results of the queueing theory [68].

Assume that each class has Poisson arrivals with the mean rate of $\lambda_i, i = 1, 2, ..., n$, and the system service rate is μ. From the state-transition diagram shown in Fig. 6.2, we obtain the probability that the system contains k members as

$$p_k = \begin{cases} \left(\frac{\lambda_1+...+\lambda_n}{\mu}\right)^k p_0, & 0 \le k \le N_1 \\ \left(\frac{\lambda_2+...+\lambda_n}{\mu}\right)^{k-N_1} \left(\frac{\lambda_1+...+\lambda_n}{\mu}\right)^{N_1} p_0, & N_1 \le k \le N_2 \\ \qquad\qquad ..... & \\ \left(\frac{\lambda_{i+2}+...+\lambda_n}{\mu}\right)^{k-N_{i+1}} \left(\frac{\lambda_{i+1}+...+\lambda_n}{\mu}\right)^{N_{i+1}-N_i} ... \left(\frac{\lambda_1+...+\lambda_n}{\mu}\right)^{N_1} p_0, & N_i \le k \le N_{i+1} \\ \qquad\qquad ..... & \\ \left(\frac{\lambda_n}{\mu}\right)^{k-N_{n-1}} \left(\frac{\lambda_{n-1}+\lambda_n}{\mu}\right)^{N_{n-1}-N_{n-2}} ... \left(\frac{\lambda_{i+1}+...+\lambda_n}{\mu}\right)^{N_{i+1}-N_i} ... \left(\frac{\lambda_2+...+\lambda_n}{\mu}\right)^{N_2-N_1} & \\ \left(\frac{\lambda_1+...+\lambda_n}{\mu}\right)^{N_1} p_0, & N_{n-1} \le k \le m. \end{cases} \tag{6.1}$$

Solving for p_0 from (6.1), we have

$$\begin{aligned} p_0 &= [\frac{1-\rho_{1,\ldots,n}^{N_1}}{1-\rho_{1,\ldots,n}} + \rho_{1,\ldots,n}^{N_1}\frac{1-\rho_{2,\ldots,n}^{N_2-N_1}}{1-\rho_{2,\ldots,n}} + \ldots + \rho_{1,\ldots,n}^{N_1}\rho_{2,\ldots,n}^{N_2-N_1}\ldots\rho_{i,\ldots,n}^{N_i-N_{i-1}}\frac{1-\rho_{i+1,\ldots,n}^{N_{i+1}-N_i}}{1-\rho_{i+1,\ldots,n}} \\ &+ \rho_{1,\ldots,n}^{N_1}\rho_{2,\ldots,n}^{N_2-N_1}\ldots\frac{1-\rho_n^{M-N_{n-1}+1}}{1-\rho_n}]^{-1}, \end{aligned} \tag{6.2}$$

where $\rho_{i,\ldots,n} = \frac{\lambda_i+\ldots+\lambda_n}{\mu}$, $\rho_i = \frac{\lambda_i}{\mu}$, $i = 1, 2, \ldots, n$. Subsequently, the blocking probability of every class r_i, $i = 1, 2, \ldots, n$, can be obtained.

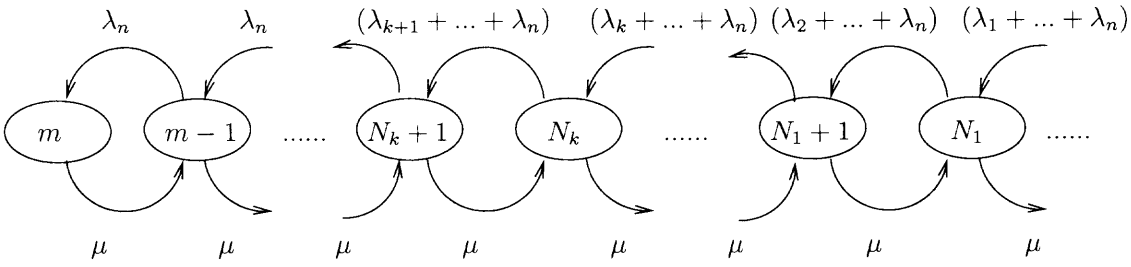


Figure 6.2 State-transition diagram.

However, to connect the blocking thresholds to loss differentiation parameters, a certain relationship in between is sought. With known blocking probabilities r_i, $i = 1, 2, \ldots, n$, and the buffer size m, an optimization problem is thus formed to minimize the sum of class blocking probabilities weighed by differentiation parameters, that is,

$$\min_{N_1, N_2, \ldots, N_{n-1}} (\sigma_1 r_1 + \sigma_2 r_2 + \ldots + \sigma_n r_n),$$

subject to the constraints

$$1 = \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0,$$

$$0 \leq N_1 \leq N_2 \leq \ldots \leq N_n = m,$$

$$\sigma_1, \sigma_2, \ldots, \sigma_n \textit{ are real numbers},$$

$$N_1, N_2, \ldots, N_n \textit{ are integers}.$$

The blocking thresholds $N_i, i = 1, 2, ..., n$, and the differentiation parameters $\sigma_i, i = 1, 2, ..., n$, are thus coupled together, by achieving the minimum system blocking probability.

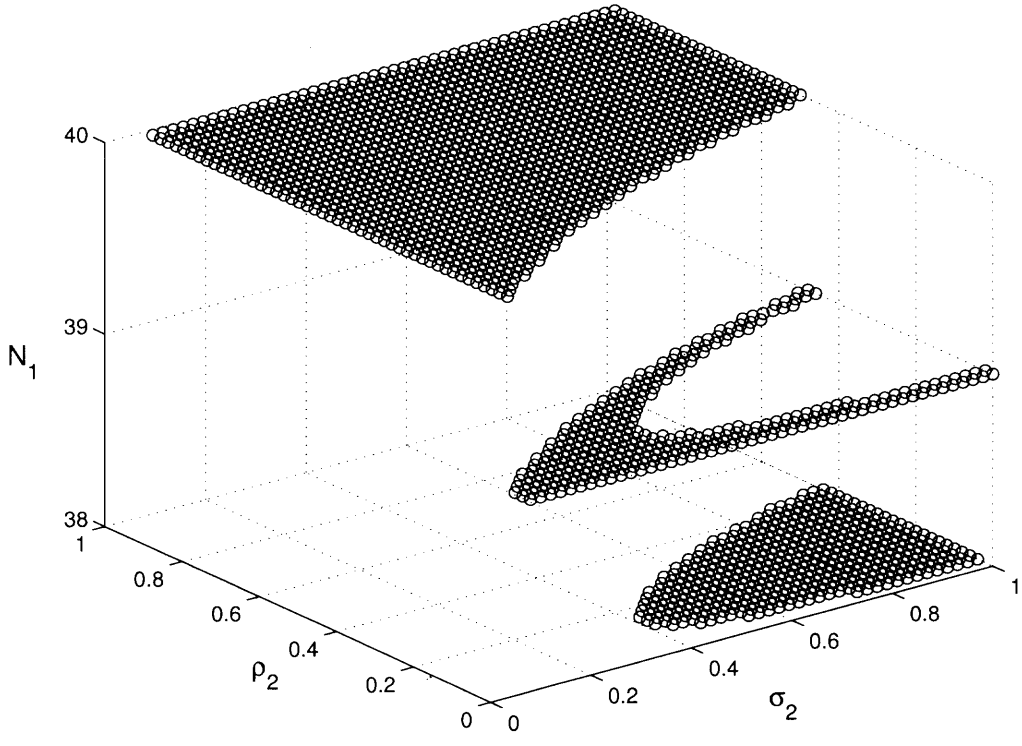


Figure 6.3 Relationship of N_1, σ_2, and ρ_2, where queue size $m = 40$.

6.2 Analysis Results

6.2.1 Two-class Scenario

The previously formed problem is first tackled for a two-class scenario. From (6.1) and (6.2), the blocking probabilities of class 1 and 2, that is, r_1 and r_2, are expressed as

$$r_1 = \frac{1-\rho_2^{M-N_1+1}}{1-\rho_2}\rho_{1,2}^{N_1}\left[\frac{1-\rho_{1,2}^{N_1}}{1-\rho_{1,2}} + \rho_{1,2}^{N_1}\frac{1-\rho_2^{M-N_1+1}}{1-\rho_2}\right]^{-1}$$

$$r_2 = \rho_2^{M-N_1}\rho_{1,2}^{N_1}\left[\frac{1-\rho_{1,2}^{N_1}}{1-\rho_{1,2}} + \rho_{1,2}^{N_1}\frac{1-\rho_2^{M-N_1 1}}{1-\rho_2}\right]^{-1}$$

The optimization problem, accordingly, is re-stated as the follows.

$$\min_{N_1,N_2}(\sigma_1 r_1 + \sigma_2 r_2),$$

subject to the constraints

$$1 = \sigma_1 \geq \sigma_2 > 0,$$

$$0 \leq N_1 \leq N_2 = m,$$

$$\sigma_1, \sigma_2 \; are \; real \; numbers,$$

$$N_1, N_2 \; are \; integers.$$

Since the algebraic solution of the objective function is not straightforward, numerical computation is used to solve the optimization problem. In a two-class system, there are overall six parameters: blocking thresholds N_1, N_2, utilization factors ρ_1, ρ_2, and differentiation parameters σ_1, σ_2. However, with acceptable assumptions, that is, $N_2 = m$, $\rho_{1,2} = \rho_1 + \rho_2 = 0.99 \rightarrow 1$, a tractable three-dimension figure of N_1, σ_2, and ρ_2 are plotted in Fig. 6.4.

The points where minimum objection function values are achieved are aggregated in three ladders in the figure. Since each ladder is associated with a blocking threshold value, two equations are formed as follows to illustrate the boundary conditions.

$$\begin{aligned} F_{\sigma_2,N_1=40}(\rho_2) &= F_{\sigma_2,N_1=39}(\rho_2) \\ F_{\sigma_2,N_1=39}(\rho_2) &= F_{\sigma_2,N_1=38}(\rho_2) \end{aligned} \tag{6.3}$$

Expanding (6.3), we have

$$\frac{\rho_{1,2} * [\frac{1-\rho_2^1}{1-\rho_2} + \alpha_2\rho_2^0]}{\frac{1-\rho_{1,2}^{40}}{1-\rho_{1,2}} + \rho_{1,2}^{40} * \frac{1-\rho_2^1}{1-\rho_2}} = \frac{\frac{1-\rho_2^2}{1-\rho_2} + \alpha_2\rho_2^1}{\frac{1-\rho_{1,2}^{39}}{1-\rho_{1,2}} + \rho_{1,2}^{39} * \frac{1-\rho_2^2}{1-\rho_2}}$$

$$\frac{\rho_{1,2} * [\frac{1-\rho_2^2}{1-\rho_2} + \alpha_2\rho_2^1]}{\frac{1-\rho_{1,2}^{39}}{1-\rho_{1,2}} + \rho_{1,2}^{39} * \frac{1-\rho_2^2}{1-\rho_2}} = \frac{\frac{1-\rho_2^3}{1-\rho_2} + \alpha_2\rho_2^2}{\frac{1-\rho_{1,2}^{38}}{1-\rho_{1,2}} + \rho_{1,2}^{38} * \frac{1-\rho_2^3}{1-\rho_2}},$$

and obtain the following two curves

$$G_{\sigma_2,N_1=(40,39)}(\rho_2) = \frac{-A_{40}(1+\rho_2) + A_{39}\rho_{12}}{(A_{40}+B_{40}) * \rho_2^1 - [A_{39}+B_{39}(1+\rho_2)] * \rho_{12}},$$

$$G_{\sigma_2,N_1=(39,38)}(\rho_2) = \frac{-A_{39}(1+\rho_2+\rho_2^2) + A_{38}\rho_{12}(1+\rho_2)}{[A_{39}+B_{39}(1+\rho_2)] * \rho_2^2 - [A_{38}+B_{38}(1+\rho_2+\rho_2^2)] * \rho_{12}\rho_2^1},$$

where $A_{40} = \frac{1-\rho_{1,2}^{40}}{1-\rho_{1,2}}$, $B_{40} = \rho_{1,2}^{40}$, $i = 1, 2, ..., n$, and $\rho_{1,2} = 0.99$. Together, these curves form a contour of ρ_2 and σ_2 as depicted in Fig. 6.4.

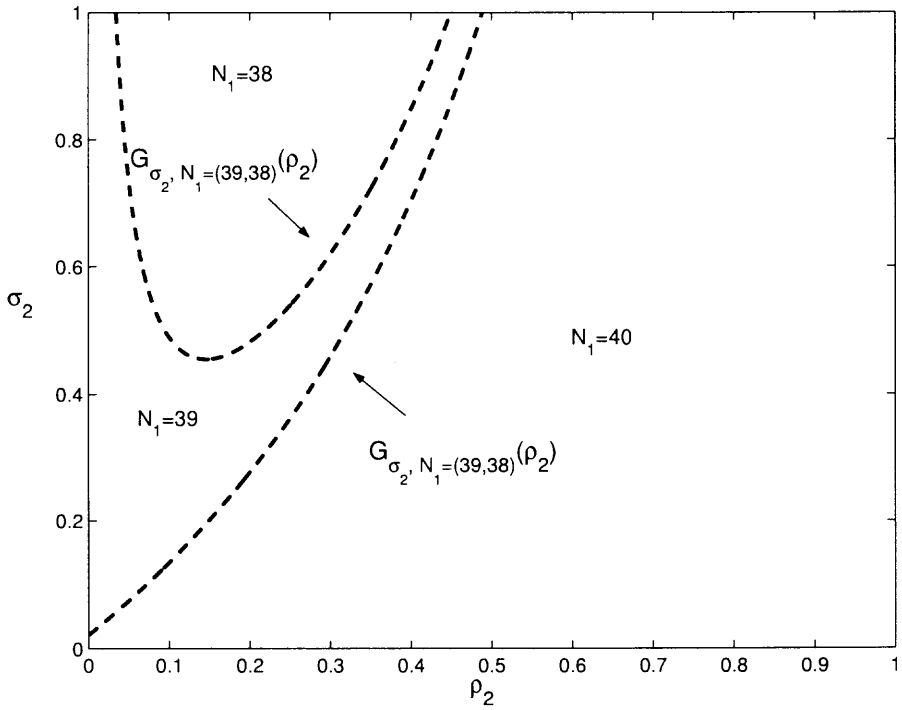


Figure 6.4 Contour of σ_2 and ρ_2 in previous 3-D figure.

To further incorporate the information of the buffer size m into the formulation, it is reasonable to investigate other boundary conditions, that is, other possible ladders in the 3-D figure. In fact, with a buffer size $m = 80$, a new 3-D figure includes four ladders. Counting the ladders in the direction of decreasing blocking threshold N_1, therefore, we induct the curve formula between ladder i and $i-1$ as

$$
\begin{aligned}
&G_{\sigma_2,N_2=(m-i,m-i-1)}(\rho_2) \\
&= \frac{-(A_{m-i}+B_{m-i}\sum_{k=0}^{i}\rho_2^k)*\sum_{k=0}^{i+1}\rho_2^k+[A_{m-i-1}+B_{m-i-1}\sum_{k=0}^{i+1}\rho_2^k]*\rho_{1,2}\sum_{k=0}^{i}\rho_2^k}{(A_{m-i}+B_{m-i}\sum_{k=0}^{i}\rho_2^k)*\rho_2^{i+1}-[A_{m-i-1}+B_{m-i-1}\sum_{k=0}^{i+1}\rho_2^k]*\rho_{1,2}\rho_2^i} \\
&= \frac{-A_{m-i}\sum_{k=0}^{i+1}\rho_2^k+A_{m-i-1}*\rho_{1,2}\sum_{k=0}^{i}\rho_2^k}{(A_{m-i}+B_{m-i}\sum_{k=0}^{i}\rho_2^k)*\rho_2^{i+1}-[A_{m-i-1}+B_{m-i-1}\sum_{k=0}^{i+1}\rho_2^k]*\rho_{1,2}*\rho_2^i} \qquad (6.4)
\end{aligned}
$$

A lemma is then drawn from (6.4).

Lemma 6.1: The general boundary equation as shown in (6.4) has a lower bound that is bigger than zero. Moreover, when the ladder index i increases, that is, the blocking threshold N_1 decreases, this lower bound increases.

Proof: Knowing that $1 < A_i < A_m$, $0 < B_m < B_i < 1$, $0 < \rho_2^{i+1} < \rho_2^i < 1$, and $\rho_2^m \leq \rho_2^{m-i} \leq (m-i+1)\rho_2^{m-i} \leq \sum_{k=0}^{m-i} \leq m-i+1 \leq m$, $i = 1, ..., n$, a lower bound of (6.4) is obtained as follows:

$$
\begin{aligned}
&G_{\sigma_2,N_2=(m-i,m-i-1)}(\rho_2) \\
&= \frac{A_{m-i}\sum_{k=0}^{i+1}\rho_2^k-A_{m-i-1}*\rho_{1,2}\sum_{k=0}^{i}\rho_2^k}{-(A_{m-i}+B_{m-i}\sum_{k=0}^{i}\rho_2^k)*\rho_2^{i+1}+[A_{m-i-1}+B_{m-i-1}\sum_{k=0}^{i+1}\rho_2^k]*\rho_{1,2}*\rho_2^i} \\
&\geq \frac{(A_{m-i}-A_{m-i-1}*\rho_{1,2})\sum_{k=0}^{i}\rho_2^k}{-(A_{m-i}+B_{m-i}\sum_{k=0}^{i}\rho_2^k)*\rho_2^{i+1}+[A_{m-i-1}+B_{m-i-1}\sum_{k=0}^{i+1}\rho_2^k]*\rho_{1,2}*\rho_2^{i+1}} \\
&= \frac{\sum_{k=0}^{i}\rho_2^k}{\rho_2^{i+1}[-(A_{m-i}+B_{m-i}\sum_{k=0}^{i}\rho_2^k)+(A_{m-i-1}+B_{m-i-1}\sum_{k=0}^{i+1}\rho_2^k)*\rho_{1,2}]} \\
&\geq \frac{\sum_{k=0}^{i}\rho_2^k}{\rho_2^{i+1}[-(1+B_m*\rho_2^m)+(A_m+m)*\rho_{1,2}]} > 0 \qquad (6.5)
\end{aligned}
$$

As observed in (6.5), when the search procedure continues, that is, the ladder index i increases, $\sum_{k=0}^{i}\rho_2^k$ increases and ρ_2^{i+1} decreases; subsequently, the value of this lower bound increases with respect to i. Since the differentiation parameter $\sigma_2 < 1$,

when $G_{\sigma_2,N_1=(m-i,m-i-1)}(\rho_2) > 1$, the curve will fall out of the contour; it is the stop point of the curve search. Therefore, for any buffer size m, the curves in the contour can be obtained by repeating the search procedure until the lower bound of the curve is bigger than one.

6.2.2 N-class Scenario

To directly induct the previous discussion to the n-class scenario is not a trivial task, owing to the sharp increase in the number of variables. Therefore, the previous analysis is practiced recursively. First, aggregate classes $1, 2, ...,$ and $n-1$ into one class. The contour showing the relationship between ρ_n, σ_n, and N_{n-1} consists of the following curves:

$$G_{\sigma_n,N_{n-1}=(m,m-1)}(\rho_n) = \frac{-A_m * \sum_{k=0}^{1} \rho_n^k + A_{m-1} * \rho_{1,...,n}}{(A_m + B_m) * \rho_n^1 - [A_{m-1} + B_{m-1} \sum_{k=0}^{1} \rho_n^k] * \rho_{1,...,n}},$$

$$G_{\sigma_n,N_{n-1}=(m-1,m-2)}(\rho_n) = \frac{-A_{m-1} * \sum_{k=0}^{2} \rho_n^k + A_{m-2} * \rho_{1,...,n} \sum_{k=0}^{1} \rho_n^k}{(A_{m-1} + B_{m-1} \sum_{k=0}^{1} \rho_n^k) * \rho_n^2 - [A_{m-2} + B_{m-2} \sum_{k=0}^{2} \rho_n^k] * \rho_{1,...,n}\rho_n},$$

.....,

$$G_{\sigma_n,N_{n-1}=(m-i,m-i-1)}(\rho_n) = \frac{-A_{m-i} * \sum_{k=0}^{i+1} \rho_n^k + A_{m-i-1} * \rho_{1,...,n} \sum_{k=0}^{i} \rho_n^k}{(A_{m-i} + B_{m-i} \sum_{k=0}^{i} \rho_n^k) * \rho_n^{i+1} - [A_{m-i-1} + B_{m-i-1} \sum_{k=0}^{i+1} \rho_n^k] * \rho_{1,...,n}\rho_n^i},$$

where $G_{\sigma_n,N_{n-1}=(m-i,m-i-1)}(\rho_n) > 1$. From this contour, with the known ρ_n and selected blocking threshold N_{n-1}, the value range for differentiation parameter σ_n is reached.

Next, class n is excluded from the queue. By aggregating classes $1, 2, ...,$ and $n-2$, the contour of class $n-1$ is solved as

$$G_{\sigma_{n-1},N_{n-2}=(N_{n-1},N_{n-1}-1)}(\rho_{n-1}) = \frac{-A_{N_{n-1}} * \sum_{k=0}^{1} \rho_{n-1}^{k} + A_{N_{n-1}-1} * \rho_{1,...,(n-1)}}{(A_{N_{n-1}} + B_{N_{n-1}}) * \rho_{n-1}^{1} - [A_{N_{n-1}-1} + B_{N_{n-1}-1} \sum_{k=0}^{1} \rho_{n-1}^{k}] * \rho_{1,...,(n-1)}},$$

$$.....,$$

$$G_{\sigma_{n-1},N_{n-2}=(N_{n-1}-i,N_{n-1}-i-1)}(\rho_{n-1}) = \frac{-A_{N_{n-1}} * \sum_{k=0}^{i+1} \rho_{n-1}^{k} + A_{N_{n-1}-i-1} * \rho_{1,...,(n-1)} \sum_{k=0}^{i} \rho_{n-1}^{k}}{(A_{N_{n-1}} + B_{N_{n-1}} \sum_{k=0}^{i} \rho_{n-1}^{k}) * \rho_{n-1}^{i+1} - [A_{N_{n-1}-i-1} + B_{N_{n-1}-i-1} \sum_{k=0}^{i+1} \rho_{n-1}^{k}] * \rho_{1,...,(n-1)} \rho_{n-1}^{i}},$$

where $G_{\sigma_{n-1},N_{n-2}=(N_{n-1}-i,N_{n-1}-i-1)}(\rho_{n-1}) > 1$. Note that in this iteration, the buffer size has been updated by the blocking threshold N_{n-1}, and $\rho_{1,...,n-1} \to 1 - \rho_n$. Again, with the known ρ_{n-1} and the chosen blocking threshold N_{n-2}, we obtain the value range for the differentiation parameter σ_{n-1}.

The parameter search procedure finishes after getting the contour of class 2, that is,

$$G_{\sigma_2,N_1=(N_2,N_2-1)}(\rho_2) = \frac{-A_{N_2} * \sum_{k=0}^{1} \rho_2^{k} + A_{N_2-1} * \rho_{1,2}}{(A_{N_2} + B_{N_2}) * \rho_2^{1} - [A_{N_2-1} + B_{N_2-1} \sum_{k=0}^{1} \rho_{n-1}^{k}] * \rho_{1,2}},$$

$$.....,$$

$$G_{\sigma_2,N_1=(N_2-i,N_2-i-1)}(\rho_2) = \frac{-A_{N_2-i} * \sum_{k=0}^{i+1} \rho_2^{k} + A_{N_2-i-1} * \rho_{1,2} \sum_{k=0}^{i} \rho_2^{k}}{(A_{N_2-1} + B_{N_2-1} \sum_{k=0}^{i} \rho_2^{k}) * \rho_2^{i+1} - [A_{N_2-i-1} + B_{N_2-i-1} \sum_{k=0}^{i+1} \rho_2^{k}] * \rho_{1,2} \rho_2^{i}},$$

where $G_{\sigma_2,N_1=(N_2-i,N_2-i-1)}(\rho_2) > 1$. Up to now, the differentiation parameters $\sigma_2, ..., \sigma_n$ and the blocking thresholds $N_1, N_2, ..., N_{n-1}$ shall be solved. Note that $\sigma_1 = 1$ and $N_n = m$.

Though based on Poisson arrivals, this solution can be experimentally adapted for the self-similar traffic model. From a reference system emulating the practical one, empirical data such as ρ_i, σ_i, N_i, and resulted values of $\sigma_1 r_1 + \sigma_2 r_2 + ...$, are collected. These data, being manipulated similarly as mentioned above, will then yield a series of differentiation parameters that minimize the system blocking probabilities.

6.3 Numerical Results

The previously stated search procedure is carried out on a three-class scenario. Assume the utilization factor for each class, that is, $\rho_i = \frac{\lambda_i}{\mu}, i = 1, 2, 3$ are given as $\rho_1 = 0.5$, $\rho_2 = 0.3$, and $\rho_3 = 0.19$. The buffer size is preset as $m = 30$. Following the previously described steps, we first choose $N_2 = 28$, and obtain $\sigma_3 \in [0.553564, 1]$. Next, we choose $N_1 = 26$, and have $\sigma_2 \in [0.699037, 1]$.

An exhaustive search method, which checks all the combination of variables, such as utilization factors ρ_i, differentiation parameters σ_i, and blocking thresholds N_i, is brought in as a reference to show the merits of this new approach. First, as observed from Table 6.1, the minimum blocking probabilities found by the new approach are close enough to those of exhaustive search, although differences between these two probabilities grow bigger with the increasing number of classes. This is resulted by the iterations that accumulate approximation errors. Since the number of classes in the DiffServ model is limited, nevertheless, this tendency shall have no considerably negative effects on the selection of differentiation parameters. Second, the new approach significantly shortens the search time, benefiting from all contours that reduce the search space. This new method, subsequently, foresees the potential of practical implementations.

6.4 Chapter Summary

This chapter presented a quantitative approach to select the loss differentiation parameters for the proportional differentiation service model. Guidelines based on the principles of queueing and optimization are proposed and validated. The intrinsic characteristic of the method also guarantees that the system blocking probability is minimized with respect to blocking thresholds.

Table 6.1 Performance Comparison between the Exhaustive Search and the New Approach[a].

	Search approach	2-class scenario	3-class scenario	4-class scenario	5-class scenario	6-class scenario
Minimum value	Exhaustive	0.032852017118	0.032852017118	0.032852017118	—	—
	New	0.032916639774	0.034071945419	0.035342797211	0.036707803154	0.037420184398
Simulation time (second)	Exhaustive	0.22	122.29	26277.72	—	—
	New	0.01	0.42	8.03	105.06	937.17

[a]The omitted values in the table do not affect the drawn conclusions.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

While E-commerce is getting increasingly important in today's world, the desire for secure IP-based VPNs is imminent. Standard bodies, research groups, and industry vendors are pushing one another ahead, proposing diverse implementation strategies and enabling technologies. Moreover, how to provide QoS to VPNs is under intensive discussion and investigations. While the application and deployment differences between IntServ and DiffServ are becoming clearer, several refinements on the DiffServ QoS model have been emerging. Among them, proportional QoS differentiation is attracting attention owing to its simplicity and improved QoS differentiation granularity. To achieve comparable performance to other alternatives, however, proportional QoS differentiation needs further enhancement. Based on a thorough study on the IP VPN QoS issue, this dissertation has addressed the topic of adopting the proportional QoS differentiation to provide QoS guarantees to IP VPNs. Original contributions of this dissertation include the following:

- An overall picture of IP-based VPN implementation, surveying various enabling techniques for each deployment building block.
- A hierarchical QoS guarantee framework for IP VPNs, from the service provider perspective, stitching together development progresses from the recent research and engineering work.
- The investigation on the proportional loss differentiation, where the "packet shortage" phenomenon has been discussed and the "debt-aware" enhancement was proposed to partially solve the problem.

- The investigation on the proportional delay differentiation, where the differentiation consistency has been studied and a combined delay differentiation scheme was proposed to enforce the differentiation over both short and long time periods.

- A new quantitative guideline, based on the principles of queueing and optimization, to compute the loss differentiation parameters.

In addition, the dissertation has created the following future research opportunities:

- While the proportional QoS operations at the VPN network device level is rather clear, investigations on those at the VPN network level, if ever required, will help make the whole picture complete.

- A generic network infrastructure that accommodates all involved QoS operations and provides the proportional QoS differentiation to IP-based VPNs is sought.

- Though aiming to deliver relative QoS, the proportional QoS differentiation can have a stronger appeal by offering a certain degree of absolute QoS guarantees.

APPENDIX A

DERIVATION OF PROPERTIES OF THE AVERAGE DELAY DIFFERENCE

This appendix includes derivations for the properties presented in Chapter 5.

Property 5.2: Increasing the arrival rate of a bigger class introduces a larger increase on the average delay difference between successive classes, thereby resulting in a bigger delay range ΔG.

Proof: From (5.3), it can be derived that

$$\begin{aligned}
\frac{\partial \bar{d}_{i,i+1}}{\partial \lambda_j} &= \frac{\partial}{\partial \lambda_j}[(1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}] \\
&= [(1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)]'\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k} + (1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)[\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}]' \\
&= (1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)\frac{\left(\sum\limits_{k=1}^{n}\lambda_k\delta_k\right)\frac{\partial\bar{Q}}{\partial\lambda_j} - \bar{Q}\delta_j}{\left(\sum\limits_{k=1}^{n}\lambda_k\delta_k\right)^2} \qquad \text{(A.1)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \bar{d}_{i,i+1}}{\partial \lambda_k} &= \frac{\partial}{\partial \lambda_k}[(1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}] \\
&= [(1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)]'\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k} + (1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)[\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}]' \\
&= (1-\delta_{i+1})\left(\prod_{k=1}^{n}\delta_k\right)\frac{\left(\sum\limits_{k=1}^{n}\lambda_k\delta_k\right)\frac{\partial\bar{Q}}{\partial\lambda_k} - \bar{Q}\delta_k}{\left(\sum\limits_{k=1}^{n}\lambda_k\delta_k\right)^2} \qquad \text{(A.2)}
\end{aligned}$$

Considering the difference between (A.1) and (A.2), we have

$$\frac{\partial(\bar{d}_{i,i+1})}{\partial \lambda_j} - \frac{\partial(\bar{d}_{i,i+1})}{\partial \lambda_k} =$$
$$\frac{(\delta_i-\delta_{i+1})}{\left(\sum\limits_{k=1}^{n}\lambda_k\delta_k\right)^2}\left[\left(\sum_{k=1}^{n}\lambda_k\delta_k\right)\left(\frac{\partial\bar{Q}}{\partial\lambda_j} - \frac{\partial\bar{Q}}{\partial\lambda_k}\right) + \bar{Q}\left(\delta_k - \delta_j\right)\right]. \qquad \text{(A.3)}$$

Given that the inter-arrivals and packet lengths of all classes follow the same distributions, we have $\frac{\partial \bar{Q}}{\partial \lambda_j} = \frac{\partial \bar{Q}}{\partial \lambda_k}$. Since the delay differentiation model defines that $1 = \delta_1 > \delta_2 > \delta_3 > ... > \delta_n > 0$, if $j > k$, $\delta_j < \delta_k$, the following relationship holds:

$$\frac{\partial(\bar{d}_{i,i+1})}{\partial \lambda_j} - \frac{\partial(\bar{d}_{i,i+1})}{\partial \lambda_k} > 0. \tag{A.4}$$

This property can also be drawn from the individual class delay property [15]: increasing the rate of a higher class causes a larger increase in the average class delays than increasing the rate of a lower class. By increasing the rates of class k and j, two delay scalars $\{\bar{d}_i{}'\}$ and $\{\bar{d}_i{}''\}$ are resulted from $\bar{d}_i, i = 1, 2, ...n$, respectively. If $k < j$, we have $\bar{d}_1{}'' > \bar{d}_1{}'$. Property 5.2, therefore, holds from (5.2):

$$\begin{aligned} \bar{d}'_{i,i+1} &= (1 - \delta_{i+1}) \left(\prod_{k=1}^{n} \delta_k \right) \bar{d}'_1 \\ &< (1 - \delta_{i+1}) \left(\prod_{k=1}^{n} \delta_k \right) \bar{d}''_1 \qquad \text{(A.5)} \\ &= \bar{d}''_{i,i+1} \qquad \text{(A.6)} \end{aligned}$$

Property 5.3: When one or multiple subscribers move to a higher class, all average delay differences increase, so does the system delay range. Otherwise, both metrics decrease.

Proof: Considering the case of one customer, the act of moving to a higher class causes the arrival rate of this higher class increase and the rate of the previous class decrease. Denote the arrival of this customer as $\Delta\lambda_k$, and the resulting traffic loads become $\lambda_1, \lambda_2, ..., \lambda_k - \Delta\lambda_k, ..., \lambda_m + \Delta\lambda_k, ..., \lambda_n$. From (5.3), therefore, the updated average delay difference is

$$\bar{d}'_{i,i+1} = \left(\prod_{k=1}^{i} \delta_k \right) \frac{(1 - \delta_{i+1})\, \bar{Q}}{\sum_{k=1}^{n} \lambda_k \bar{\delta}_k + \delta\lambda_k(\delta_m - \delta_k)}, \tag{A.7}$$

where $i = 1, 2, ..., n-1$. Since $k < m$, $\delta_k > \delta_m$, it is straightforward that $\bar{d}'_{i,i+1} > \bar{d}_{i,i+1}$. The same conclusion holds for the multiple subscriber case. This property implies

that moving subscribers between classes, or say, between QoS levels, to achieve higher or lower delays may actually have the opposite effects.

Property 5.4: Increasing the delay differentiation parameter δ_i, the delay differences $\bar{d}_{m,m+1}, m = 1, 2, ..., i$, increase, and the delay differences $\bar{d}_{m,m+1}, m = i+1, i+2, ..., n-1$, decrease.

Proof: From (5.3), when the differentiation parameter varies, two scenarios apply. When $m \leq i$, we have

$$\begin{aligned}
\frac{\partial \bar{d}_{i,i+1}}{\partial \delta_m} &= \frac{\partial}{\partial \delta_m}[(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k}] \\
&= [(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)]'\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k} + [(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)][\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k}]' \\
&= [0+(1-\delta_{i+1})(\delta_1\delta_2\delta_3\delta_{m-1}\delta_{m+1}...\delta_i)]\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k} \\
&\quad +[(1-\delta_{i+1})(\delta_1\delta_2\delta_3\delta_{m-1}\delta_m\delta_{m+1}...\delta_i)]\frac{-\bar{Q}\lambda_m}{\left(\sum_{k=1}^{n}\lambda_k\delta_k\right)^2} \\
&= (1-\delta_{i+1})(\delta_1\delta_2\delta_3\delta_{m-1}\delta_{m+1}...\delta_i)\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k} \\
&\quad -(1-\delta_{i+1})(\delta_1\delta_2\delta_3\delta_{m-1}\delta_m\delta_{m+1}...\delta_i)\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k}\frac{\lambda_m}{\sum_{k=1}^{n}\lambda_k\delta_k} \\
&= (1-\delta_{i+1})(\delta_1\delta_2\delta_3\delta_{m-1}\delta_{m+1}...\delta_i)\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k}[1-\frac{\delta_m\lambda_m}{\sum_{k=1}^{n}\lambda_k\delta_k}] > 0. \qquad \text{(A.8)}
\end{aligned}$$

When $m > i$, we have

$$\begin{aligned}
\frac{\partial \bar{d}_{i,i+1}}{\partial \delta_m} &= \frac{\partial}{\partial \delta_m}[(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k}] \\
&= [(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)]'\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k} + [(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)][\frac{\bar{Q}}{\sum_{k=1}^{n}\lambda_k\delta_k}]'
\end{aligned}$$

$$= 0 \bullet \frac{\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k} + [(1-\delta_{i+1}) \left(\prod_{k=1}^{i} \delta_k\right)] \frac{-\bar{Q}\lambda_m}{\left(\sum_{k=1}^{n} \lambda_k \delta_k\right)^2}$$

$$= -[(1-\delta_{i+1}) \left(\prod_{k=1}^{i} \delta_k\right)] \frac{\bar{Q}\lambda_m}{\left(\sum_{k=1}^{n} \lambda_k \delta_k\right)^2} < 0. \tag{A.9}$$

Therefore,

$$\frac{\partial \bar{d}_{i,i+1}}{\partial \delta_m} = \left(\prod_{k=1}^{i} \delta_k\right) \frac{(1-\delta_{i+1})\,\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k} \left(1 - \frac{\delta_m \lambda_m}{\sum_{k=1}^{n} \lambda_k \delta_k}\right) > 0, \tag{A.10}$$

where $m \leq i$, and

$$\frac{\partial \bar{d}_{i,i+1}}{\partial \delta_m} = -\left(\prod_{k=1}^{i} \delta_k\right) \frac{(1-\delta_{i+1})\,\bar{Q}\lambda_m}{\left(\sum_{k=1}^{n} \lambda_k \delta_k\right)^2} < 0, \tag{A.11}$$

where $m > i$. From (A.10) and (A.11), property 5.4 is concluded.

Property 5.6: Decreasing the delay differentiation parameter of a class decreases the delay difference between this class and the next higher one, and increases the delay difference between this class and the next lower one.

Proof: The property is drawn from the following derivations:

$$\frac{\partial \bar{d}_{i,i+1}}{\partial \delta_i} = \frac{\partial}{\partial d_i}[(1-\delta_{i+1}) \left(\prod_{k=1}^{i} \delta_k\right) \frac{\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k}]$$

$$= [(1-\delta_{i+1}) \left(\prod_{k=1}^{i} \delta_k\right)]' \frac{\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k} + [(1-\delta_{i+1}) \left(\prod_{k=1}^{i} \delta_k\right)][\frac{\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k}]'$$

$$= (1-\delta_{i+1})(\delta_1\delta_2\delta_3 \ldots \delta_{i-2}\delta_{i-1}) \frac{\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k}$$

$$+(1-\delta_{i+1})(\delta_1\delta_2\delta_3 \ldots \delta_{i-1}\delta_i) \frac{-\bar{Q}\lambda_m}{\left(\sum_{k=1}^{n} \lambda_k \delta_k\right)^2}$$

$$= (1-\delta_{i+1})(\delta_1\delta_2\delta_3 \ldots \delta_{i-2}\delta_{i-1}) \frac{\bar{Q}}{\sum_{k=1}^{n} \lambda_k \delta_k}[1 - \frac{\delta_i \lambda_i}{\sum_{k=1}^{n} \lambda_k \delta_k}] > 0. \tag{A.12}$$

$$
\begin{aligned}
\frac{\partial \bar{d}_{i,i+1}}{\partial \delta_{i+1}} &= \frac{\partial}{\partial d_i}[(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}] \\
&= [(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)]'\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k} + [(1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)][\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}]' \\
&= [(1-\delta_{i+1})'\prod_{k=1}^{i}\delta_k + (1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)']\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k} \\
&\quad +[(1-\delta_{i+1})\prod_{k=1}^{i}\delta_k][\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k}]' \\
&= -\left(\prod_{k=1}^{i}\delta_k\right)\frac{\bar{Q}}{\sum\limits_{k=1}^{n}\lambda_k\delta_k} - (1-\delta_{i+1})\left(\prod_{k=1}^{i}\delta_k\right)\frac{\bar{Q}\lambda_m}{\left(\sum\limits_{k=1}^{n}\lambda_k\delta_k\right)^2} < 0. \qquad \text{(A.13)}
\end{aligned}
$$

The conclusion can also be reached from the individual class property [15], which states that decreasing the delay differentiation parameter of a class increases the average delay of all other classes, and decrease the average delay of that class. Accordingly, when δ_i increases, $\bar{d}_i$ decreases, $\bar{d}_{i-1}$ and $\bar{d}_{i+1}$increases, and thus $\bar{d}_{i,i+1}$ decreases and $\bar{d}_{i-1,i}$ increases.

APPENDIX B

CONSERVATION LAW OF THE MEAN WAITING TIME

This appendix derives the conservation law which was used in Chapter 5; the conservation law states that the sum of average waiting times weighed by delay differentiation parameters is an invariant.

Let's consider a singer-server queue with n types of classes. Class i arrives according to a general arrival process with rate $\lambda_i, i = 1, 2, ..., n$. The mean service time of class i is denoted by $E(S_i)$, the mean residual service time of class i is denoted by $E(R_i)$. Define $\rho_i = \lambda_i E(S_i)$. To ensure that the server can handle the amount of work offered per unit of time, we assume that $\sum_{i=1}^{n} \lambda_i E(S_i) < 1$. The service process is also general, and thus all classes are served according to FIFO or random rule in a non-preemptive manner.

For a work-conserving scheduling discipline P, denote $E(W(P))$ as the mean amount of work in the system, and denote $E(Q_i(P))$ as the mean number of class i packets waiting in the queue. The mean amount of the work in the system can be given by the sum of the mean amount of work in the queue and the mean amount work at the server, that is,

$$E(W(P)) = \sum_{i=1}^{n} E(Q_i(P)) \bullet E(S_i) + \sum_{i=1}^{n} \rho_i E(R_i). \tag{B.1}$$

Obviously, the total amount of work in the system does not depend on the service order of classes. The amount of work decreases with one unit per unit of time; when a new packet arrives, the amount of work increases by one unit of service time. Therefore, in (B.1), both $E(W(P))$ and $\sum_{i=1}^{n} \rho_i E(R_i)$ do not depend on the scheduling discipline P, so shall the mean amount of the work in the queue $\sum_{i=1}^{n} E(Q_i(P)) \bullet E(S_i)$.

From Little's law, the mean number of class i packets in the queue equals to the multiplication of the arrival rate λ_i and the mean waiting time $E(W_i(P))$, that is, $E(Q_i(P)) = \lambda_i \bullet E(W_i(P))$. Subsequently, the amount of work in the queue becomes

$$\begin{aligned}
\sum_{i=1}^{n} E(Q_i(P)) \bullet E(S_i) &= \sum_{i=1}^{n} \lambda_i \bullet E(W_i(P)) \bullet E(S_i) \\
&= \sum_{i=1}^{n} \lambda_i \bullet E(S_i) \bullet E(W_i(P)) \\
&= \sum_{i=1}^{n} \rho_i \bullet E(W_i(P)). \qquad \text{(B.2)}
\end{aligned}$$

Since the mean amount of the work in the queue is independent of the scheduling discipline P, the conservation law for mean waiting times is stated as:

$\sum_{i=1}^{n} \rho_i \bullet E(W_i(P))$ is an invariance with respect to the scheduling discipline P.

BIBLIOGRAPHY

[1] "Network-based IP VPNs: the ole of MPLS," white paper, Ennovate Networks, 1999. [Online]. Available: http://www.ennovatenetworks.com

[2] "A primer for implementing a cisco virtual private network," reference guide, Cisco, Aug. 2000. [Online]. Available: http://www.cisco.com/warp/public

[3] S. Blake, S. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, Dec. 1998.

[4] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 49–59, Mar. 1992.

[5] T. Yang, D. Tsang, and S. Li, "Cell scheduling and bandwidth allocation for a class of VBR video connections," in *Proc. IEEE Workshop in Visual Signal Processing and Communications*, New Jersey, USA, Sep. 1994, pp. 95–101.

[6] T. Yang, D. Tsang, and P. McCabe, "Cell scheduling and bandwidth allocation for heterogeneous VBR video conferencing traffic," in *Proc. IEEE 1995 Global Telecommunications Conference (GLOBECOM'95)*, Singapore, Nov. 1995, pp. 371–377.

[7] T. Yang and J. Pan, "A measurement-based loss scheduling scheme," in *Proc. IEEE Fifteenth Annul Joint Conference of the Computer and Communications Societies (INFOCOM'96)*, San Francisco, USA, Mar. 1996, pp. 1062–1071.

[8] H. J. Chao and H. Cheng, "A new QoS-guaranteed cell discarding strategy: self-calibrating pushout," in *Proc. IEEE 1994 Global Telecommunications Conference (GLOBECOM'94)*, San Francisco, USA, Dec. 1994, pp. 929–934.

[9] C. Dovrolis and P. Ramanathan, "A case for relative differentiated services and the proportional differentiation model," *IEEE Network*, vol. 13, no. 5, pp. 26–34, Sept./Oct. 1999.

[10] J. Zeng and N. Ansari, "Toward IP virtual private network quality of service: a service provider perspective," *IEEE Communications Magazine*, vol. 41, pp. 113–118, Apr. 2003.

[11] T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "Delay differentiation and adaption in core stateless networks," in *Proc. IEEE Nineteenth Annual Joint Conference of the Computer and Communications Societies (INFOCOM'00)*, California, USA, Mar. 2000, pp. 421–430.

[12] C. Dovrolis and P. Ramanathan, "Proportional differentiated services, part II: loss rate differentiation and packet dropping," in *Proc. IEEE Eighth International Workshop on Quality of Service (IWQoS'00)*, Philadelphia, USA, Jun. 2000, pp. 53–61.

[13] U. Bodin, A. Jonsson, and O. Schelen, "On creating proportional loss-rate differentiation: predictability and performance," in *Proc. IEEE ninth International Workshop on Quality of Service (IWQoS'01)*, Karlsruhe, Germany, Jun. 2001, pp. 372–388.

[14] Y. Chen, M. Hamdi, D. H. K. Tsang, and C. Qiao, "Proportional QoS provisioning: a uniform and practical solution," in *Proc. IEEE International Conferences on Communications (ICC'02)*, vol. 4, New York, USA, May 2002, pp. 2363–2367.

[15] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: delay differentiation and packet scheduling," *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 12–26, Feb. 2002.

[16] L. Kleinrock, *QUEUEING SYSTEMS, Volume II: Computer Applications.* New York: Wiley-Interscience, 1976.

[17] S. Bodamer, "A new scheduling mechanism to provide relative differentiation for real-time IP traffic," in *Proc. IEEE 2000 Global Telecommunications Conference (GLOBECOM'00)*, vol. 1, San Francisco, USA, Nov. 2000, pp. 646–650.

[18] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zron, and B. Palter, "Layer two tunneling protocol L2TP," IETF RFC 2661, Aug. 1999.

[19] S. Hanks, T. Li, D. Farinacci, and P. Traina, "Generic routing encapsulation (GRE)," IETF RFC 1701, Oct. 1994.

[20] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic routing encapsulation (GRE)," IETF RFC 2784, Mar. 2000.

[21] S. Kent and R. Atkinson, "Security architecture for the Internet protocol," IETF RFC 2401, Nov. 1998.

[22] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet security association and key management protocol (ISAKMP)," IETF RFC 2408, Nov. 1998.

[23] D. Harkins and S. Carrel, "The Internet key exchange (IKE)," IETF RFC 2409, Nov. 1998.

[24] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, Jan. 2001.

[25] B. Gleeson, A. Lin, J. Heinanen, G. Armitage, and A. Malis, "A framework for IP based virtual private networks," IETF RFC 2764, Feb. 2000.

[26] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote authentication dial in user service (RADIUS)," IETF RFC 2865, Jun. 2000.

[27] "ISPs simplify remote access for enterprises," white paper, Bay Networks, 1997. [Online]. Available: http://www.adimpleo.com/library/nortel/radius.pdf

[28] "SAFE VPN: IPSec virtual private networks in depth," white paper, Cisco, 2001. [Online]. Available: http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safe wp.htm

[29] "Case study for layer 3 authentication and encryption," white paper, Cisco, Aug. 2000. [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csvpnc/csvpnsg

[30] C. Adams and S. Farrell, "Internet X.509 public key infrastructure certificate management protocols," IETF RFC 2510, Mar. 1999.

[31] S. Chokhani and W. Ford, "Internet X.509 public key infrastructure certificate policy and certification practices framework," IETF RFC 2527, Mar. 1999.

[32] S. Kent and R. Atkinson, "IP authentication header," IETF RFC 2402, Nov. 1998.

[33] ——, "IP encapsulating security payload (ESP)," IETF RFC 2406, Nov. 1998.

[34] D. C. Blight and T. Hamada, "Policy-based networking architecture for QoS interworking in IP management-scalable architecture for large-scale enterprise-public interoperation," in *Proc. of the Sixth IFIP/IEEE International Symposium on Integrated Network Management*, Boston, USA, May 1999, pp. 813–826.

[35] S. J. Shepard, "Policy-based networks:hype and hope," *IT Professional*, vol. 2, no. 1, pp. 12–16, Jan. 2000.

[36] R. Rajan, A. Chiu, and S. Civanlar, "A policy based approach for QoS-on-demand over the Internet," in *Proc. of the Eighth International Workshop on Quality of Service (IWQoS'00)*, Pittsburgh, USA, Jun. 2000, pp. 167–169.

[37] *Principles for a telecommunications management network*, ITU-T Std. Rec. M3010, 1996.

[38] T. Braun, M. Guenter, and I. Khalil, "Management of quality of service enabled VPNs," *IEEE Communications Magazine*, vol. 39, no. 5, pp. 90–98, May 2001.

[39] [Online]. Available: http://newsroom.cisco.com/dlls/prod-030402.html

[40] [Online]. Available: http://www.businesswire.com

[41] W. Simpson, "IP in IP tunneling," IETF RFC 1853, Oct. 1995.

[42] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP) – version 1 functional specification," IETF RFC 2205, Sep. 1997.

[43] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions for RSVP for LSP tunnels," IETF RFC 3209, Dec. 2001.

[44] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, "Policy core information model – version 1 specification," IETF RFC 3060, Feb. 2001.

[45] B. Wijnen, D. Harrington, and R. Presuhn, "An architecture for describing SNMP management frameworks," IETF RFC 2571, Apr. 1999.

[46] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The COPS (common open policy service) protocol," IETF RFC 2748, Jan. 2000.

[47] Charging and accounting technologies for the Internet (CATI) project. [Online]. Available: http://www.tik.ee.ethz.ch/ cati/home.html

[48] Q. Kong, I. Rose, and D. Cameron, "Towards technology independent and automated service activation and provisioning," in *Proc. IEEE/IFIP Network Operation and Management Symposium*, Florence, Italy, Apr. 2002, pp. 931–933.

[49] TINA consortium. [Online]. Available: http://www.tinac.com

[50] F. D. Turck, S. Vanhastel, F. Vandermeulen, and P. Demeester, "Design and implementation of a generic connection management and service level agreement monitoring platform supporting the virtual private network service," in *Proc. IEEE/IFIP International Symposium on Integrated Network Management*, Seattle, USA, May 2001, pp. 153–166.

[51] S. Chen and K. Nahrstedt, "An overview of quality-of-service routing for the next generation high-speed networks: problems and solutions," *IEEE Network*, vol. 12, no. 6, pp. 64–79, Nov./Dec. 1998.

[52] R. Isaacs and I. Leslie, "Support for resource-assured and dynamic virtual private networks," *Journal on Selected Areas in Communications*, vol. 19, no. 3, pp. 460–472, Mar. 2001.

[53] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "A framework for integrated services operation over diffserv networks," IETF RFC 2998, Nov. 2000.

[54] B. Fox and B. Gleeson, "Virtual private network identifier," IETF RFC 2685, Sep. 1999.

[55] "Managed network-based site-to-site IP VPN services," application note, Cosine Communications, Aug. 2002. [Online]. Available: http://www.cosinecom.com/virtualipservices

[56] "RFC 2547bis: BGP/MPLS VPN fundamentals," white paper, Juniper Networks, 2001. [Online]. Available: http://www.juniper.net/techcenter/techpapers

[57] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993. [Online]. Available: http://www.icir.org/floyd/red.html

[58] "Next generation VPNs," white paper, Lucent Technologies, Nov. 2001. [Online]. Available: http://www.lucent.com/knowledge/documentdetail

[59] W. Willinger, M. S. Tappu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," *IEEE Transactions on Networking*, vol. 5, no. 1, pp. 71–86, Feb. 1997.

[60] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," in *Proc. ACM Conference on Communications, Architectures, Protocols and Applications (SIGCOMM'93)*, San Francisco, USA, Sep. 1993, pp. 183–193.

[61] H. L. Seal, *Survival probabilities: the goal of risk theory.* Chichester: Wiley, 1978.

[62] H. Buhlmann, *Mathematical methods in risky theory.* Heidelberg: Spring-Verlag Berlin, 1970.

[63] J. Zeng and N. Ansari, "An enhanced dropping scheme for proportional differentiated services," in *Proc. IEEE International Conferences on Communications (ICC'03)*, vol. 3, Alaska, USA, May 2003, pp. 1897–1901.

[64] ——, "On the performance of proportional delay differentiation," in *Proc. IEEE High Performance Switching and Routing (HPSR'03)*, Torino, Italy, Jun. 2003.

[65] Y. Chen, C. Qiao, M. Hamdi, and D. Tsang, "Proportional differentiation: a scalable QoS approach," *IEEE Communications Magazine*, vol. 41, pp. 52–58, Jun. 2003.

[66] J. T. Virtamo, "Reciprocity of blocking probabilities in multiservice loss systems," *IEEE Transactions on Communications*, vol. 36, no. 10, pp. 1174–1175, Oct. 1988.

[67] J. S. Kaufman, "Blocking in a completely shared resource environment with state dependent resource and residency requirements," in *Proc. IEEE Eleventh Annual Joint Conference of the Computer and Communications Societies (INFOCOM'92)*, Florence, Italy, May 1992, pp. 2224–2232.

[68] L. Kleinrock, *QUEUEING SYSTEMS, Volume I: Theory.* New York: Wiley-Interscience, 1975.