INFORMATION TO USERS

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

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

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

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

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

**Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

A User's Perspective on Connection Admission Control:
Integrating Traffic Shaping, Effective Bandwidths and Pricing.

by

Matthias Falkner

B.Sc., University of Bath (1991)

M.Sc., London School of Economics (1992)

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario

July 11, 2000

©Matthias Falkner, 2000

The author hereby grants to Carleton University permission to reproduce and to
distribute copies of this thesis document in whole or in part.

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

Our file *Notre référence*

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

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

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

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

0-612-57609-4

Canada

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

**A User's Perspective on Connection Admission Control: Integrating Traffic
Shaping, Effective Bandwidths and Pricing**

submitted by

Matthias Falkner (B.Sc., M.Sc.)
in partial fulfillment of the requirements for
the degree Doctor of Philosophy

Chair, Department of Systems and Computer Engineering

Thesis Supervisor

Thesis Supervisor

External Examiner

Carleton University

July 11, 2000

Abstract

Network architectures, such as DiffServ, IntServ or ATM, are moving towards the provisioning of Quality of Service (QoS) to accommodate the transmission needs by highly variable multi-modal traffic (e.g. voice, video or data). Provisioning QoS requires the user and the network to establish a traffic contract, where the negotiated QoS as well as the user's traffic characterization are fixed. Upon successful admission to the network, the user's packet stream is regulated (policed or shaped) to ensure compliance with the traffic contract. A leaky bucket algorithm (or its token bucket variant) or a $G/D/1/K$ queue are typically used for this purpose. Such traffic regulators provide a single interface between the network and the user.

In this thesis we develop a traffic shaping framework. Taking a user's perspective on connection admission control (CAC), we formulate an optimization problem to determine the parameters of a traffic shaper or policer. Under the assumption that the network sets prices for buffer and bandwidth and indicates their availability, the optimization problem determines the minimum cost settings for the traffic shaper which satisfy the desired QoS and guarantee access to the network. The solution to this optimization problem represents the resources required to satisfy the user's QoS demands. We then propose a pricing scheme, guaranteed service pricing, which charges the user according to this resource requirement.

Acknowledgements

I would like to thank both of my supervisors, Prof. M. Devetsikiotis and Prof. I. Lambadaris, for their outstanding support and guidance throughout the course of my thesis. I particularly appreciated the many helpful discussions in the early phase of my studies and the collaborative and professional approach that they adopted in the later stages. I would also like to thank both supervisors for fostering a productive research environment at the Broadband Networks Laboratory by bringing together numerous like-minded researchers from all over the world. Two of these deserve my particular appreciation: Sandra Tartarelli from the University of Pisa, without whom the chapter on effective bandwidth measurement would not have been possible; and Tarkan Taralp, with whom I enjoyed many brainstorming sessions on effective bandwidths.

This thesis would not have been possible without the financial support from the Natural Science and Engineering Research Council (NSERC), the Ontario Graduate Scholarship (OGS) and the financial support from the Department of Systems and Computer Engineering at Carleton University. Finally, I would like to thank my wife for her continuing support throughout the past years. Thank you for making this thesis possible.

To Julia and my parents

Contents

Acceptance Form	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	vi
List of Figures	xi
List of Tables	xiv
List of Acronyms	xv
1 Introduction	1
1.1 Background	1
1.2 Thesis Contributions	4
1.3 Thesis Organization	6
2 Quality of Service Provisioning under (σ, ρ)-constraints	8
2.1 (σ, ρ) -Constraints	9
2.2 Input-Output relations for the single node	10
2.3 End-to-end QoS provisioning	15
2.4 Connection Admission Control	18
2.5 Probabilistic bounds based on extremal traffic	19

2.6	Related Concepts	22
3	Quality of Service Provisioning under Effective Bandwidths	24
3.1	Effective Bandwidths	24
3.2	Input-Output relations for the single node	28
3.3	End-to-End QoS Provisioning	30
3.3.1	Large Buffer Asymptotics	31
3.3.2	Many Sources Asymptotics	34
3.3.3	End-to-end Provisioning Algorithm	36
3.4	Connection Admission Control	40
3.5	Other CAC Schemes	43
4	Traffic Regulation and QoS Provisioning in Practice	50
4.1	Traffic Shaping and Policing	50
4.1.1	The FCFS Queue	51
4.1.2	The Leaky-Bucket Algorithm	51
4.1.3	Window Algorithms	56
4.1.4	Effective Bandwidth Policing	57
4.1.5	Effectiveness of Policers	58
4.2	QoS Provisioning in Practice	60
4.2.1	Asynchronous Transfer Mode	60
4.2.2	Integrated Services	64
4.2.3	Differentiated Services	67
5	Pricing Concepts in Broadband Networks	71
5.1	Criteria for evaluating pricing schemes	73
5.2	Flat Pricing	77
5.3	Paris-Metro Pricing	80
5.4	Priority Pricing	82
5.5	Smart-Market Pricing	84
5.6	Edge Pricing	86

5.7	Expected Capacity Pricing	88
5.8	Responsive Pricing	90
5.9	Effective Bandwidth Pricing	92
5.10	Proportional Fairness Pricing	96
5.11	Concluding Discussion	98
6	Minimum Cost Traffic Shaping	103
6.1	Motivation	104
6.2	Minimum Cost Traffic Shaping Framework	108
6.2.1	QoS Constraints	110
6.2.2	Framework Formulation	112
6.2.3	CAC Constraints	115
6.3	Minimum Cost Traffic Shaping under (σ, ρ) -constraints	117
6.3.1	Framework Formulation	118
6.3.2	Example: Gaussian, Poisson and On-Off Sources	121
6.3.3	Example: Self-Similar Traffic Streams	126
6.4	Minimum Cost Traffic Shaping under Effective Bandwidths	135
6.4.1	Framework Formulation	135
6.4.2	Charging for Resources under Effective Bandwidths	138
6.4.3	Example: Gaussian Sources	141
6.5	QoS Pricing Scheme	145
6.5.1	Alternative Cost Functions for MCTS	148
6.5.2	Comparison with other Pricing Schemes	151
7	Measuring Effective Bandwidths	154
7.1	Effective Bandwidth Estimators	155
7.1.1	The direct estimator	156
7.1.2	The block estimator	157
7.1.3	The Kullback-Leibler Distance (KLD) estimator	158
7.1.4	The linear regression (LR) estimator	159
7.2	Traffic streams	160

7.3	Accuracy of measured effective bandwidths	162
7.3.1	Poisson traffic stream	162
7.3.2	Comparison with truncated Poisson process	166
7.3.3	On-off fluid traffic stream	167
7.3.4	FGN traffic stream	171
7.4	Effect of the trace length on measured effective bandwidths	176
7.4.1	Poisson traffic stream	177
7.4.2	On-off fluid traffic stream	177
7.4.3	FGN traffic stream	178
7.5	Summary of Findings	180
7.6	Empirical Effective Bandwidths	182
7.6.1	Properties of Empirical Effective Bandwidths	183
7.6.2	Behavior in the t space	184
7.6.3	Behavior with respect to the trace length	185
7.7	Experimental Results for CAC	186
7.7.1	Simulation Methodology	186
7.7.2	Results for the Poisson trace	187
7.7.3	Results for the on-off fluid trace	187
7.7.4	Results for the FGN trace	188
7.8	Minimum Cost Traffic Shaping and Measured Effective Bandwidths	190
8	Conclusions and Suggestions for Future Research	200
8.1	Conclusions	200
8.2	Suggestions for Future Research	202
A	Description of ATM Service Categories	204
A.1	Constant Bit Rate Service (CBR)	204
A.2	Real-time Variable Bit Rate Service (rt-VBR)	205
A.3	Non real-time Variable Bit Rate Service (nrt-VBR)	205
A.4	Unspecified Bit Rate Service (UBR)	205
A.5	Available Bit Rate Service (ABR)	206

B Brief Review of Kuhn-Tucker Conditions	207
C MCTS Problems in AMPL Format	210
D Table of Symbols	213
Bibliography	222

List of Figures

2.1	Network Model	11
2.2	Virtual Path model	15
3.1	Effective bandwidth, Decoupling bandwidth and output effective bandwidth for an on-off source with mean 1 and peak 10.	29
4.1	Traffic Shaper modeled as a G/D/1/B queue	51
4.2	Leaky bucket regulator without cell buffer	52
4.3	Leaky bucket regulator with cell buffer	53
5.1	A user's demand function for bandwidth	80
5.2	Conceptual representation of PMP	81
5.3	Conceptual representation of Smart-market pricing	85
5.4	Effective Bandwidth Pricing	93
6.1	Admissible region for effective bandwidth based CAC ($\alpha(.)$ here represents the aggregate effective bandwidth of all arrival streams).	107
6.2	Feasible region with loss, delay and jitter constraints.	111
6.3	Illustration of possible types of the optimal solution (D = delay allocation, L = loss allocation. P = peak rate allocation).	115
6.4	Graphical representation of the optimization problem	117
6.5	Modified model for a VP including a traffic regulator.	118
6.6	Plots for $\Lambda(\theta)$ and $\theta^2 \frac{\partial \alpha(\theta)}{\partial \theta}$ for an on-off source.	123
6.7	Simulated and Analytical loss probability	130

6.8	Contour Plot of the Feasible Region	131
6.9	Value of the Cost Function restricted to the Feasible Region	131
6.10	Virtual dedicated queue model.	139
6.11	Effect of κ on the shape of the loss constraint.	140
6.12	Exponential or hyperbolic cost functions.	151
7.1	Measured effective bandwidths for the Poisson trace using the direct and the block estimators. Note that the θ - axis is in log scale ranging from 0.00001 to 5 with a multiplicative step of 1.5.	163
7.2	Analytical and measured effective bandwidths for the Poisson trace for the four different methods.	163
7.3	Analytical and measured effective bandwidths for the Poisson trace with 95% CIs	165
7.4	Comparison of the measured EB against the analytical EB for both the Poisson and the truncated Poisson processes.	167
7.5	Analytical and measured effective bandwidths for the on-off fluid trace using the direct and the block estimators. Note that the θ - axis is in log scale ranging from 0.00001 to 5 with a multiplicative step of 1.5.	168
7.6	Squared error of the effective bandwidth measurements for the on-off fluid trace using the direct and the block estimators. Note that the θ - axis is in log from 0.00001 to 5 with a multiplicative step of 1.5.	169
7.7	Analytical and measured effective bandwidths for the on-off fluid trace . . .	170
7.8	Analytical and measured effective bandwidths for the on-off fluid trace with 95% CIs	171
7.9	Analytical and measured effective bandwidth for the FGN trace using the direct estimator	172
7.10	Squared Error between the analytical and the measured effective bandwidths for the FGN trace.	173
7.11	Comparison between the RMD and the FFT generators for $t = 1$ slot and $N = 131.072$	174

7.12 Comparison between the RMD and the FFT generators for $t = 32$ slots and $N = 131.072$	174
7.13 Comparison of the measured effective bandwidth for different trace lengths for the Poisson trace using the block and the KLD estimator.	178
7.14 Comparison of the measured effective bandwidth for different trace lengths for the on-off trace using the block and the KLD estimator.	179
7.15 Estimates of the effective bandwidth for the FGN trace taken over traces 1048576 and 512 samples long, at $t = 1$ slot.	179
7.16 Estimates of the effective bandwidth for the FGN trace taken over traces 1048576 and 512 samples long, at $t = 16$ slots.	180
7.17 Relevant time scale for the FGN trace for queuing performance	191
7.18 Illustration of the effective bandwidth of two on-off sources with different burstiness but the same mean and peak values	192
7.19 Graph for the fitted EEB for the Poisson case.	195
7.20 Graphs for the interpolated EEB for the Poisson case.	195
7.21 Graph for the fitted EEB for the on-off case.	195
7.22 Graphs for the interpolated EEB for the on-off case.	196
7.23 Graph for the fitted EEB for the Gaussian case.	196
7.24 Graphs for the interpolated EEB for the Gaussian case.	196
7.25 Graph for the fitted EEB for the Pareto case.	197
7.26 Graphs for the interpolated EEB for the Pareto case.	197

List of Tables

4.1	QoS parameters by ATM service class	62
4.2	Traffic descriptors by ATM service class	63
5.1	Summary of pricing scheme features.	99
6.1	Minimization results without constraints on service and buffer	132
6.2	Minimization results for different feasible regions spliced in the service-dimension	133
6.3	Minimization results for different feasible regions spliced in the buffer-dimension	134
6.4	Minimization results for input streams with different Hurst parameters for a target loss probability of 1.00×10^{-2} . The simulated loss probability is shown for comparison.	134
6.5	MCTS parameters under Gaussian traffic for EB based CAC	143
6.6	MCTS parameters for on-off traffic under EB based CAC	145
7.1	Behavior of an FGN trace for large values of $\theta.N = 512$	175
7.2	Behavior of an FGN trace for large values of $\theta.N = 131.072$	175
7.3	Behavior of an FGN trace for large values of $\theta.N = 1.048.576$	176
7.4	Simulation results for the Poisson trace (top: empirical effective bandwidth; bottom: analytical effective bandwidth)	188
7.5	Simulation results for the on-off Fluid trace (top: empirical effective band- width; bottom: analytical effective bandwidth)	189
7.6	Simulation results for the FGN trace (empirical effective bandwidth only) .	190
7.7	Computation times for the curve-fitting and interpolation methods.	199

List of Acronyms

ABR	Available Bit Rate
AF	Assured Forwarding
ARIMA	Auto-regressive Moving Average
ATM	Asynchronous Transfer Mode
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDVT	Cell Delay Variation Tolerance
CLP	Cell Loss Probability
CLP	Cell Loss Priority (in the context of ATM)
CLR	Cell Loss Ratio
CS	Class Selector
DS	Differentiated Services
DSCP	Differentiated Services Codepoint
EB	Effective Bandwidth
EEB	Empirical Effective Bandwidth
EF	Expedited Forwarding
FBM	Fractional Brownian Motion
FCFS	First Come First Serve
FGN	Fractional Gaussian Noise
GCRA	Generic Cell Rate Algorithm
GDP	Gross Domestic Product
GFR	Generic Frame Rate

IETF	Internet Engineering Task Force
ITU	International Telecommunications Union
IP	Internet Protocol
LRD	Long-range Dependent
MAP	Markov Arrival Process
MBAC	Measurement based admission control
MBP	Minimum bandwidth property
MCTS	Minimum cost traffic shaping
NNI	Network-network Interface
nrt-VBR	Non real-time VBR
OAM	Operations, Administration and Maintenance
PFP	Proportional Fairness Pricing
PGPS	Packetized General Processor Sharing
PHB	Per Hop Behavior
PMP	Paris-Metro Pricing
PSN	Processor Sharing Node
QoS	Quality of Service
RCBR	Renegotiated CBR
RED	Random Early Detection
RIO	RED with in and out
RSVP	Resource Reservation Protocol
rt-VBR	Real-time VBR
SCFQ	Self-clocked fair queuing
SF	Switching Fabric
SLA	Service Level Agreement
SRP	Spatial Renewal Process
TSPEC	Traffic Specification
UBR	Unspecified Bit Rate
UNI	User-network Interface

VBR	Variable Bit Rate
VC	Virtual Circuit
VN	Virtual Network
VP	Virtual Path

Chapter 1

Introduction

1.1 Background

Communication networks are already an essential part of many aspects of our lives. They enable us to obtain information from a variety of sources located almost anywhere in the world. The majority of information carried by networks is still in the form of data. However, future network architectures will without a doubt be able to carry information in a multi-modal fashion. Voice, data or video will all be carried in the same manner. They may even be interchanged for one another in some degree. Much of the research in the past decade has focused on enabling such integrated services networks.

Quality of service (QoS) is of particular interest in integrated services networks. Each type of media (such as voice, video or data) imposes different service requirements on the network. For example, traffic generated by applications can be classified into real-time and non real-time traffic. Real time traffic requires the timely delivery of data across the network. Often, a degree of losses can be tolerated by the application. Non real-time traffic on the other hand typically places a higher emphasis on lossless transmission but tolerates some network delays.

To accommodate the highly variable QoS requirements of applications, network architectures such as Asynchronous Transfer Mode (ATM), Integrated Services (IntServ) or Differentiated Services (DiffServ) currently provide a set of service categories. In ATM for

example, users (i.e. applications) are given the choice between constant bit rate (CBR), real-time variable bit rate (rt-VBR), non real-time variable bit rate (nrt-VBR), unspecified bit rate (UBR) and available bit rate (ABR) categories. Such service categories allow the network to provision sufficient resources to satisfy the QoS requirements. The network faces a trade-off in this provisioning task. On one extreme, bandwidth and buffer resources could be exclusively reserved for each application flow, assuming the worst-case scenario of continuous peak rate transmission. While this would ensure a high QoS, network resources would end up being lightly utilized, in particular if the traffic were very bursty. On the other extreme, the network could multiplex all streams regardless of their QoS requirements. This would allow resources to be used more efficiently. However, packets may be lost or delayed to a degree which is intolerable for the user. Service categories help the network to deploy different strategies for resource provisioning such that QoS requirements can be met.

In addition to service categories, a traffic contract is negotiated between the user and the network. In this contract, the user characterizes the arrival stream for which a certain QoS is required. Without such a characterization QoS guarantees could never be made. Based on this characterization, the desired QoS, and the available network resources, the contract is negotiated. It specifies the details to which both the user and the network agree to abide. This contract negotiation is often established on a long-term basis. In this case the network ensures that its capacity suffices to fulfill all traffic contracts established. Alternatively, the contract may be the result of a connection admission control (CAC) procedure. A stream may dynamically request admission to the network given its characterization and a QoS requirement. The network determines whether sufficient resources are available to admit the stream to provide its QoS under the condition that the QoS of streams which have already been admitted is not degraded.

The theories of effective bandwidths or (σ, ρ) -constraints provide frameworks for the network to determine the resource requirement for a desired QoS. The effective bandwidth is a function of the log moment generating function of a stochastic process. It therefore includes all the moments of the process and provides a complete characterization. The effective bandwidth is always between the mean and peak rates of the arrival stream and in this way accounts for a stream's burstiness. Effective bandwidths can be used in a variety

of ways. Given the bandwidth capacity of the link and a QoS target θ , they can be used to determine the required buffer space. Alternatively, for a given bandwidth capacity, buffer space and effective bandwidths of all streams feeding the buffer, an upper bound on the QoS parameters (losses, delays) can be obtained. The associated CAC procedure typically compares the aggregate effective bandwidth evaluated with parameter θ against the available bandwidth capacity. If the bandwidth capacity exceeds the aggregate effective bandwidth, further streams could be admitted without jeopardizing the QoS of the admitted streams. Otherwise, new flows have to be rejected.

A (σ, ρ) -constraint provides an upper bound on the burstiness of the arrival stream in a given time interval. If the stream feeds a single server queue with service rate ρ , then σ is the upper bound on the queue occupancy. If a stream is (σ, ρ) -constrained, then the number of arrivals in any intervals can be bounded by these two parameters. The network can perform CAC based on such constraints. New flows can be admitted if the sum of all σ 's is less than the buffer size and if the sum of all ρ 's is less than the bandwidth capacity. In these cases, no losses will occur inside the network and an upper bound on the end-to-end delay can be derived.

Upon successful admission / negotiation of a traffic contract, each stream is subsequently policed or shaped. The primary motivation for policing is to ensure that the values negotiated in the traffic contract are not violated. This is important since a malicious user who sends traffic in excess of its characterization impacts on the QoS of other users. The primary motivation for shaping (or re-shaping) is also to maintain the traffic contract. However, shaping is typically performed by the network to keep its contractual obligation. Due to the multiplexing or scheduling inside the network, the characterization of a stream may not be preserved inside the network. Multiplexing many streams together may increase the burstiness of any one stream. Similarly, the operation of some packet scheduling mechanisms may alter the original characterization of the packet flow. Traffic shaping inside the network is thus essential to maintain the streams characterization, and to preserve the QoS guarantees for all admitted flows.

The functions and mechanisms for integrated services networks are designed to provide QoS under the assumption of scarce resources. Empirical utilization statistics have shown

that new applications and an exponentially increasing user base place higher and higher demands on resources. In the case of the Internet, network services have so far either been free or charged for using a flat pricing schemes. However, in addition to providing the technical facilities to support QoS, alternative pricing schemes are currently being investigated. Charging users by traffic volume or by QoS not only allows network providers to recover the costs for providing the infrastructure, but it could also be used to alter users' transmission behavior by giving them monetary incentives not to transmit during congestion periods.

In the QoS provisioning approach described above a number of deficiencies can be identified: first, the procedure to establish the traffic contract is inadequately specified. Some contracts are established by a network operator and a human user for long time periods. These require no technical specification, but experience to characterize the traffic profile and QoS demands. However, such long-term contracts are also static and inflexible, and so specifications allow for dynamically established contracts and connections. In this case, a procedure to establish the traffic contract should be specified (but is generally not). Second, network architectures currently provide a limited set of QoS categories. These are often designed with specific applications in mind, often before these applications become available. The definitions may be overlapping or even conflicting. For example, the CBR and rt-VBR categories in ATM may be configured to provide the same QoS. More importantly, the service categories may be inappropriate for new applications. Finally, the issue of charging is often treated separately from the provision of QoS. Mismatches between the pricing schemes and the service categories can lead to undesirable inconsistencies.

In this thesis we propose a solution to these problems by integrating effective bandwidths, traffic shaping and pricing.

1.2 Thesis Contributions

The main contribution of this thesis is the concept of *minimum cost traffic shaping* (MCTS). We formulate an optimization problem by taking a user's perspective on connection admission control. The user wishes to gain access to the network, obtain a QoS guarantee (in terms of losses, bounded delays or bounds on the delay variation), and minimize the cost

of the connection. This optimization problem assumes that the network provides the user with a cost function, indicating the prices for buffer and bandwidth resources. Furthermore, the network is assumed to reveal the available resource capacities. If effective bandwidths are used to characterize the traffic stream and the QoS constraints imposed by the user, this optimization problem is shown to have a standard form. The solution to this optimization problem characterizes the minimum cost resources required to gain admission to the network and obtain the desired QoS.

We arrive at the MCTS framework by taking a user's point of view in the contract establishment process. Moreover, we designed the framework by integrating traffic shaping, effective bandwidths and pricing. This approach has several advantages:

- It determines the resources required to obtain a specified QoS level. Specifically, given prices for buffer and bandwidth, MCTS determines the buffer / bandwidth pair which minimizes a given cost function. The network can use this pair for dynamic resource planning, such as the dynamic allocation of resources to virtual networks.
- It forms the basis of a dynamic traffic contract. Given the resource prices and their availability by the network, the user can determine the resource requirements for the desired QoS level using MCTS. These parameters then characterize the requirements to the network. The traffic contract could be restricted to the MCTS solution. Furthermore, this solution provides the desired parameterization for the traffic shaper / policer. The computation requirements of MCTS are in the order of seconds, such that an on-line implementation seems feasible.
- The MCTS parameters form the basis for charging. The network can derive precise demand functions for buffer / bandwidth resources with MCTS.

A further contribution of this thesis is the proposal of a pricing scheme. Under the proposed pricing scheme the user is charged according to the resources required to satisfy the QoS demands. In combination with MCTS, the user can easily foresee the charges incurred by a flow. The pricing scheme allows the network to recover costs for provisioning QoS.

The final contribution in this thesis is the concept of empirical effective bandwidths. We

stress the implications of finite time traffic traces on the effective bandwidth. In particular, we show that the estimated effective bandwidth is always bounded above by the observed peak rate, whereas its analytical counterpart may reach infinity. We stress the importance of this finding for CAC. Furthermore, we show that this property also holds for self-similar traffic streams.

1.3 Thesis Organization

Chapters 2 and 3 provide background on (σ, ρ) -constraints and effective bandwidths. We illustrate the properties of effective bandwidths and (σ, ρ) -constraints as well as results which allow the dimensioning of network resources such that QoS guarantees can be made. In both chapters we provide the necessary results on which the proposed MCTS framework is based.

Chapter 4 then examines schemes for traffic shaping and policing. As indicated above, traffic regulators play an important role in the QoS provisioning framework. They are essential for the enforcement of the traffic contract. We also summarize the three network architectures currently intended for QoS provisioning: ATM, IntServ and DiffServ. The proposed MCTS framework is applicable in all three architectures.

In chapter 5 we then review a number of pricing concepts currently proposed for broadband networks. This chapter gives the reader a background on proposed broadband pricing schemes. We provide a brief description and evaluate each scheme according to numerous criteria. The pricing scheme suggested in this thesis in subsequent chapters will be compared to the schemes reviewed here according to the same criteria.

In chapter 6 we introduce MCTS and discuss the framework in details. We demonstrate how it can be applied to networks employing (σ, ρ) -constraints for provisioning QoS, as well as for networks using effective bandwidths. We show that the use of effective bandwidths leads to a standard form for the optimization problem faced by the user. We derive a general analytical solution which is independent of the underlying traffic type. Finally, we introduce guaranteed service pricing based on traffic contracts and compare its features against the pricing schemes reviewed in the previous chapter.

Since effective bandwidths are used extensively in the MCTS scheme, we analyze methods for measuring and estimating effective bandwidths in chapter 7. We derive properties for such estimated effective bandwidths and contrast these to their analytical counterpart. We also demonstrate the applicability of the proposed MCTS scheme under measured effective bandwidths.

Chapter 8 concludes this thesis and suggests a number of items for future research.

Appendix A describes the QoS categories provided by ATM. Appendix C gives an example of a MCTS problem in AMPL format. Finally, appendix D lists all the mathematical symbols used in this thesis by chapter, along with a short description.

Chapter 2

Quality of Service Provisioning under (σ, ρ) -constraints

In the next two chapters we review existing Connection Admission Control (CAC) algorithms which may be used to provide users with Quality of Service (QoS) guarantees. In this chapter we focus on (σ, ρ) -constrained CAC. The case of effective bandwidth based CAC is treated in the next chapter. The minimum cost traffic shaping scheme proposed subsequently in this thesis assumes that CAC is based on either (σ, ρ) -constraints or effective bandwidths and makes extensive use of the results presented in these chapters.

This chapter is organized as follows. Section 2.1 starts off with definitions and properties of (σ, ρ) -constraints. Sections 2.2 and 2.3 then establish how a FCFS queue impacts these traffic characterizations and then provide results how these concepts can be used to provide sufficient resources such that QoS guarantees can be given to a user. We treat the single node case in section 2.2 and the end-to-end case in section 2.3. We outline the corresponding CAC algorithm in section 2.4. Sections 2.5 and 2.6 conclude this chapter with a discussion on probabilistic bounds and with a review of related concepts.

2.1 (σ, ρ) -Constraints

The notion of a (σ, ρ) -constraint [31] provides an upper bound for the number of arrivals in a given interval of a particular arrival process. Let A_t denote the number of cell arrivals in slot t in a discrete time model¹. A given arrival process A_t is (σ, ρ) -constrained in the time interval $(t_1, t_2]: t_1, t_2 \in \mathbb{Z}^+$ if

$$\sum_{t=t_1}^{t_2} A_t \leq \rho(t_2 - t_1 + 1) + \sigma \quad (2.1)$$

Intuitively, ρ and σ can be interpreted as an arrival rate and a maximum burst size of the given arrival process. An important observation to make is that any arrival process A_t is bounded above by an infinite family of (σ, ρ) -constraints. This point is particularly relevant for the proposed minimum cost traffic shaping scheme. Henceforth we restrict ourselves to the set of tight (σ, ρ) -constraints. For a given rate ρ , the (σ, ρ) -constraint is said to be *tight* if there is no σ' such that

$$\sum_{t=t_1}^{t_2} A_t \leq \rho(t_2 - t_1 + 1) + \sigma' < \rho(t_2 - t_1 + 1) + \sigma$$

The following theorem establishes the additivity property of (σ, ρ) -constraints.

Theorem 2.1 (Additivity Property of (σ, ρ) -constraints) *Consider a set of J arrival processes $\{A_j\}: j = 1 \dots J$, each of which is (σ_j, ρ_j) -constrained. Then the aggregate arrival process is (σ, ρ) -constrained, where*

$$\sigma = \sum_{j=1}^J \sigma_j \quad (2.2)$$

$$\rho = \sum_{j=1}^J \rho_j \quad (2.3)$$

Proof: Consider the aggregate number of arrivals from all J streams in an interval $[t_1, t_2]$:

$$\sum_{j=1}^J \sum_{t=t_1}^{t_2} A_{tj} \leq \sum_{j=1}^J \{\rho_j(t_2 - t_1 + 1) + \sigma_j\}$$

¹We restrict ourselves to a slotted model in this section.

$$= \sum_{j=1}^J \rho_j (t_2 - t_1 + 1) + \sum_{j=1}^J \sigma_j$$

This is by definition a $(\sum_{j=1}^J \sigma_j, \sum_{j=1}^J \rho_j)$ -constraint. $\square$

We continue in the following section to derive results when (σ, ρ) -constrained streams are fed into a first-come-first-serve(FCFS) queue, which may be part of a processor sharing node (PSN). We then establish QoS results for such a single queue before applying the results to a number of such queues arranged in sequence.

2.2 Input-Output relations for the single node

The results in this section allow us to size a single network resource such that the cell delay and the cell delay variation are bounded and no cells are lost in the FCFS queue. The following theorem establishes the input-output relationship when an arbitrary discrete time arrival stream is fed into a $G/D/1$ queue.

Theorem 2.2 (Departure constraint from a $G/D/1$ queue) *Consider an arbitrary arrival process in discrete time $\{A_t\}$ being fed into a $G/D/1$ queue with constant service rate ρ . The departure process from such a queue is $(1, \rho)$ -constrained.*

Proof (sketch): Consider an arbitrary time interval $[t_1, t_2]$. The length of this interval is $(t_2 - t_1 + 1)$ slots. The maximum number of cells which can be served in this interval is thus $\rho(t_2 - t_1 + 1)$. However, the number of departures may differ by 1, since a cell may just depart the queue when the time interval starts. Hence the maximum number of departures is bound above by

$$(t_2 - t_1 + 1)\rho + 1$$

which is by definition a $(1, \rho)$ -constraint. $\square$

The proof is taken from Kesidis [93].

FCFS queues are typically embedded in a PSN architecture as depicted in Figure 2.1. In this figure, part of an end-to-end virtual path (VP) is depicted by the dotted line. The

Figure 2.1: Network Model

VP traverses two output buffer switches, consisting of a switching fabric and several output ports. Each output port has a scheduler serving a number of FCFS queues. Each queue provides a different QoS. One of the queues is typically a best effort queue which is served with lowest priority. Thus, instead of having a dedicated server for each FCFS queue of the PSN, each queue is allocated a share of the bandwidth capacity C of the respective output link. We subsequently assume that the bandwidth capacity is normalized to $C = 1$ and that each of the N queues at each port is given a share $\rho_n < 1$ of this bandwidth, with $\sum_{n=1}^N \rho_n = 1$.

The operation of the scheduler impacts the bandwidth allocation to a particular queue. For example, the scheduler may be *work-conserving* or not. If the scheduler is work-conserving and a QoS queue is idle, then one of the remaining non-empty QoS queues may be serviced pre-maturely. Queue n may thus obtain a higher bandwidth share than ρ_n . If the scheduler is not work-conserving and a QoS queue is idle when it is supposed to receive service, then the best-effort queue is served. In this case, each QoS queue n is never served at a rate higher than the allocated bandwidth ρ_n . Examples of work-conserving schedulers are packetized general processor sharing (PGPS, also known as weighted fair queuing) [126], self-clocked fair queuing (SCFQ) [64] or virtual clock [165]. Examples of non work-conserving schedulers are idling virtual clock (IVC) [77] or idling round-robin.

The operation of such schedulers is typically analyzed by comparing the departure times $\{d_i^n\}$ of the cells from QoS queue n with the departure time of the cloned process fed into an independently operating virtual queue with the same bandwidth allocation ρ_n . The departure times from the virtual queue $\{f_i^n\}$ (virtual finishing times) can then be compared to the departure times from the QoS queue $\{d_i^n\}$. Note that the virtual queue is allocated a bandwidth capacity of ρ_n and is not subject to variations induced by the scheduler. Hence the departure times from the virtual queue are characterized by

$$f_i^n = \max\{f_{i-1}^n, a_i^n\} + \frac{1}{\rho_n}; \forall i > 0 \quad (2.4)$$

with $f_0^n = 0$. Here $\{a_i^n\}$ denotes the arrival times process at queue n . Due to the operations of the scheduler at the PSN, the actual departure times $\{d_i^n\}$ may differ. How well the scheduler serves the n^{th} QoS queue is captured by the *minimum bandwidth property* (MBP) with parameter μ_n , defined by Kesidis [93, 77] as

$$\mu_n \geq d_i^n - f_i^n; \forall i > 0 \quad (2.5)$$

The overall minimum bandwidth property for the scheduler of a particular output port, μ , is then defined as

$$\mu = \max_{1 \leq n \leq N} \{\mu_n\} \quad (2.6)$$

The MBP thus provides an upper bound for the difference between the delay experienced by a cell when it is serviced by a scheduler and the delay the cell would experience in a similar dedicated FCFS queue.

To simplify the derivations of the QoS constraints in this context, we restrict ourselves to PSN with schedulers which have a minimum bandwidth property $\mu = 0$. This is the case for the PGPS or virtual clock schedulers. For a discussion on the minimum bandwidth property of other schedulers, see [93] or references therein.

The operation of schedulers may also lead to cells being clumped together, thus modifying the burstiness of the departure process. This clumping effect can be analyzed by comparing the input process into a FCFS queue of a PSN with its output process. The

following theorem determines the cell clumping effect of a FCFS queue under the above assumptions:

Theorem 2.3 (Cell clumping theorem(Theorem 4.2.1 in [93])) *Assume that a (σ, ρ) -constrained arrival process is fed into a FCFS queue with allocated bandwidth share ρ . Assume further that the FCFS queue is part of a PSN and that ρ is only a fraction of the available bandwidth allocation. Consider an arbitrary time interval between the departures of cells i and j , $1 \leq j < i$, i.e. the interval $[d_j, d_i]$. Then the output process is $(\sigma + 1, \rho)$ -constrained.*

For the proof of this theorem we require the following lemma:

Lemma 2.1 $f_i \geq f_j + \frac{i-j}{\rho}$

Proof:

$$\begin{aligned} f_i &= \max\{f_{i-1}, a_i\} + \frac{1}{\rho} \\ &= \max\{\max\{f_{i-2}, a_{i-1}\} + \frac{1}{\rho}, a_i\} + \frac{1}{\rho} \end{aligned}$$

Now, if $a_i \geq \max\{f_{i-2}, a_{i-1}\} + \frac{1}{\rho}$, then

$$\begin{aligned} f_i &= a_i + \frac{1}{\rho} \\ &> \max\{f_{i-2}, a_{i-1}\} + \frac{1}{\rho} + \frac{1}{\rho} \\ &= f_{i-1} + \frac{1}{\rho} \end{aligned}$$

else $a_i < \max\{f_{i-2}, a_{i-1}\} + \frac{1}{\rho}$, so

$$\begin{aligned} f_i &= \max\{f_{i-2}, a_{i-1}\} + \frac{1}{\rho} + \frac{1}{\rho} \\ &= f_{i-1} + \frac{1}{\rho} \end{aligned}$$

Applying this argument recursively yields the desired result. $\square$

We are now able to prove theorem 2.3.

Proof: (Cell clumping theorem) Consider again the time interval of length $d_i - d_j + 1$.

$$\begin{aligned} d_i - d_j + 1 &\geq a_i - d_j + 1 \quad (\text{since } d_i \geq a_i; \forall i) \\ &\geq a_i - f_j + 1 \quad (\text{since } f_j \leq d_j \text{ by the MBP}) \\ &= a_i - f_i + \frac{i - j}{\rho} + 1 \quad (\text{by Lemma 2.1}) \end{aligned}$$

Now consider the interval $[f_i - a_i)$. An upper bound for the delay through a FCFS queue with service rate ρ and buffer size σ is given by $\frac{\sigma+1}{\rho}$. However, since the FCFS queue is part of a PSN, a cell may have to wait at most $1 - \frac{1}{\rho}$ time units before being served again. This is the case when all other $N - 1$ QoS queues in the output port are non-empty. Hence

$$f_i - a_i \leq \frac{\sigma + 1}{\rho} + 1 - \frac{1}{\rho} = 1 + \frac{\sigma}{\rho}$$

and so $a_i - f_i \geq -1 - \frac{\sigma}{\rho}$. Substituting this into the above equation we get

$$d_i - d_j + 1 \geq \frac{i - j - \sigma}{\rho}$$

The number of departures in this interval is

$$\begin{aligned} \lceil (d_i - d_j + 1)\rho \rceil &\geq i - j - \sigma \\ \lfloor (d_i - d_j + 1)\rho + \sigma + 1 \rfloor &\geq i - j + 1 \end{aligned}$$

Note that $\rho < 1$ and hence $(d_i - d_j + 1)\rho$ is a real number. The number of departures may be $\lceil (d_i - d_j + 1)\rho + \sigma \rceil = \lfloor (d_i - d_j + 1)\rho + \sigma + 1 \rfloor$ due to the cut-through nature of the PSN (i.e. we allow $d_i = a_i$). The departure process is thus $(\sigma + 1, \rho)$ -constrained by definition. $\square$

We now proceed to establish results for QoS provisioning under (σ, ρ) -constraints for a VP traversing several PSNs.

Figure 2.2: Virtual Path model

2.3 End-to-end QoS provisioning

In this section we establish the resource requirements for a VP such that QoS guarantees can be given to the user. As represented in Figure 2.1, the VP is assumed to traverse H PSNs from source to destination. We therefore model the VP as a series of H FCFS queues in sequence, as illustrated in Figure 2.2. Recall that each FCFS queue is part of a PSN and is thus affected by the operation of the scheduler.

We assume that J sources, each of which is (σ_j, ρ_j) -constrained, $j = 1 \dots J$, are fed into the VP. We further assume that the VP has been allocated a bandwidth of C at each node and that the schedulers have a MBP $\mu = 0$. By theorem 2.1, the aggregate arrival process is (σ, ρ) -constrained, where σ and ρ are determined by equations (2.2) and (2.3) respectively.

The following theorem allows us to determine the buffer size for a single node such that no cells are lost.

Theorem 2.4 (Single node buffer size) *If a (σ, ρ) -constrained arrival process is fed into a $G/D/1/\sigma$ FCFS queue with deterministic service rate ρ , then no cells are lost due to buffer overflow (assuming that the queue is initially empty).*

Proof: We denote the occupancy of the buffer in the FCFS queue at time t by Q_t and the aggregate number of arrivals by A_t . Note that in each slot, Q_t is governed by the Lindley equation. Consider an arbitrary time interval $[1, t_1)$. Hence,

$$\begin{aligned} Q_{t_1} &= \left\lceil \max_{0 \leq t < t_1} \left\{ \sum_{t=1}^{t_1} A_t - (t+1)\rho \right\} \right\rceil \\ &\leq \left\lceil \max_{0 \leq t < t_1} \left\{ (t+1)\rho + \sigma - (t+1)\rho \right\} \right\rceil \end{aligned}$$

$$= \sigma$$

Since this results holds for all t_1 , the proof is complete. $\square$.

This theorem, in conjunction with the cell clumping theorem 2.3 allows us to size the buffers in each of the FCFS queues along the VP such that *no cells are lost*. This is the case if the buffer at node h is sized according to

$$B_h \geq \sigma + h: \quad h = 1 \dots H \quad (2.7)$$

For a proof of this equation see [93]. Note that the additional h -term is a result from the operation of the scheduler at each output port. If all queues at the first PSN are non-empty and if the FCFS queue of interest has just been served, then the cell at the head of the queue is required to wait $1 - \frac{1}{\rho}$ slots before it can enter service. During this period, the cell remains in the buffer, which accounts for the additional buffer unit required. Similarly for queues $h = 2 \dots H$.

To derive an expression for the end-to-end cell delay we again use equations (2.4) and (2.5) from above.

Theorem 2.5 (Maximum end-to-end cell delay) *The maximum end-to-end delay for cell i , denoted by δ_i , along the given VP is bounded above by*

$$\delta_i \leq \max_{0 < i} \{f_{i-1}, a_i\} - a_i + \frac{H}{C} + \Pi \quad (2.8)$$

where Π denotes to total end-to-end propagation and switching delay.

Proof: Let d_i^h denote the departure time of cell i from node h , $h = 1 \dots H$. Note that $d_i^h \leq f_i^h$ by the MBP. Also, let $a_i^h = d_i^{h-1} + \Pi_{h-1,h}$ denote the arrival time of cell i at node h , $h = 2 \dots H$. Here $\Pi_{h-1,h}$ represents the propagation and switching delay between nodes $h-1$ and h only. We assume $\Pi_{0,1} = 0$. Using (2.4) we get

$$d_i^H \leq \max_{0 < i} \{f_{i-1}^H, a_i^H\} + \frac{1}{C}$$

$$\begin{aligned}
&\leq \max_{0 \leq i} \{f_{i-1}^H \cdot f_i^{H-1} + \Pi_{H-1,H}\} + \frac{1}{C} \quad (\text{since } f_i^H \geq d_i^H) \\
&\leq \max_{0 \leq i} \{f_{i-1}^H \cdot \max_{0 \leq i} \{f_{i-1}^{H-1} \cdot a_i^{H-1}\} + \frac{1}{C} + \Pi_{H-1,H}\} + \frac{1}{C}
\end{aligned}$$

but

$$f_{i-1}^H \geq f_{i-1}^{H-1} + \frac{1}{C} + \Pi_{H-1,H}$$

so

$$d_i^H \leq \max_{0 \leq i} \{f_{i-1}^H \cdot a_i^{H-1} + \frac{1}{C} + \Pi_{H-1,H}\} + \frac{1}{C}$$

Continuing recursively we get

$$d_i^H \leq \max_{0 \leq i} \{f_{i-1}^H \cdot a_i^1 + \frac{H-1}{C} + \Pi\} + \frac{1}{C}$$

so the maximum end-to-end delay for cell i , δ_i , is

$$\begin{aligned}
\delta_i = d_i^H - a_i^1 &\leq \max_{0 \leq i} \{f_{i-1}^H \cdot a_i^1 + \frac{H}{C} + \Pi\} - a_i^1 \\
&= \max_{0 \leq i} \{f_{i-1}^H - a_i^1 \cdot \frac{H}{C} + \Pi\} \quad \square
\end{aligned}$$

This upper bound on the end-to-end cell delay also allows us to derive an upper bound for the end-to-end cell delay variation. Note that the delay that cell i experiences through the VP is bounded *below* by

$$\delta_i \geq \Pi + \frac{H}{C}$$

This is the case when the cell is served immediately at every PSN. Hence the cell delay δ_i is in the range

$$[\Pi + \frac{H}{C}, \max_{0 \leq i} \{f_{i-1}^H - a_i^1 + \frac{1}{C} \cdot \frac{H}{C} + \Pi\}]$$

The cell delay variation γ_i is thus bounded above by

$$\begin{aligned}
\gamma_i &\leq \max_{0 \leq l} \{f_{i-1}^H - a_i^l + \frac{1}{C} - \frac{H}{C} - \Pi\} \\
&= \max_{0 \leq l} \{f_{i-1}^H - a_i^l - \frac{H-1}{C} - \Pi\}
\end{aligned} \tag{2.9}$$

Note that both the cell delay and the cell delay variation for cell i are functions of the departure time of all cells $j < i$ of the virtual queue. However, as suggested in [93, page 74] and [105] a sequence of FCFS queues with deterministic service rate C can be modeled by a single queue with adjusted service rate HC in this case. For a given arrival process, the problem of determining values for the cell delay and the cell delay variation are thus simplified. An off-line approach to determine the bandwidth requirement to meet such a delay bound for real-time VBR traffic is discussed in deVeciana and Kesidis [35]. Their approach is based on simulation studies of statistical multiplexing at a single queue. Also, QoS requirements for (σ, ρ) -constrained VBR traffic are derived.

The book by Kesidis [93] extends the results presented in the last two sections. In particular, MBP results are given for schedulers where $\mu > 0$. Also, delay bounds are established for arbitrary VPs. The assumption of J (σ, ρ) -constrained VCs being bundled end-to-end into a VP is relaxed and VCs are allowed to enter or leave at intermediary nodes.

2.4 Connection Admission Control

In the previous sections we assumed that the network uses the results established above to determine the buffer sizes at each node to provide QoS guarantees. The bandwidth capacity of the network was assumed to be fixed. However, over short time scales (in the order of magnitude of less than days), even the buffer sizes are fixed. The network can then use the above results to formulate a CAC policy based on (σ, ρ) -constraints. Such a policy restricts access to newly arriving VCs in such a way that the QoS guarantees provided by a VP are maintained. We assume here that each PSN maintains state information for each allocated VP, indicating the amount of spare buffer and bandwidth capacities available (assuming J sources are already admitted). Under these assumptions and by the theorems derived in the last sections, a newly arriving $(\tilde{\sigma}, \tilde{\rho})$ -constrained VC can be admitted to the VP if

$$\tilde{\rho} \leq C - \sum_{j=1}^J \rho_j \quad (2.10)$$

$$\tilde{\sigma} \leq \min\{B_h - \sum_{j=1}^J \sigma_j - h\}: h = 1 \dots H \quad (2.11)$$

are met. The state information at each PSN then has to be updated to reflect the reduced availability of spare capacity. All carried streams still experience no cell losses and a maximum end-to-end cell delay as in (2.8).

The cost of such a CAC algorithm is a loss of statistical multiplexing gain. In this scheme, resources are effectively reserved for each stream. The network cannot overbook resources to increase resource utilization.

2.5 Probabilistic bounds based on extremal traffic

The results presented above are extended by Kesidis and Konstantopoulos in [96, 95, 49]. They determine resource requirements such that probabilistic QoS guarantees can be given. Kesidis and Konstantopoulos particularly focus on worst-case scenarios and thus determine an upper bound for the amount of bandwidth that is required to obtain probabilistic guarantees (for a given buffer size). Alternatively, they address the question of which traffic stream will achieve the least guarantees for given buffer and bandwidth resources.

Consider a single FCFS queue with a given buffer size B . Assume that a continuous, stationary and ergodic arrival process $\{A\}$ feeds this queue and that this arrival process is both (σ, ρ) and $(0, \pi)$ -constrained. Here, ρ typically represents the mean arrival rate, σ the maximum burst size and π the peak rate of the arrival stream. Let Q_t denote the queue occupancy at time t , governed by the following equation

$$Q_t = \sup_{-\infty < s \leq t} \{A(s, t] - C(t - s)\} \quad (2.12)$$

where C denotes the service rate of the queue, restricted to the range $\rho < C < \pi$. The following theorem provides an upper bound on the queue occupancy and the cell delay:

Theorem 2.6 (Theorem 1, [95]) *For any $(\sigma, \rho) \wedge (0, \pi)$ -constrained arrival process feeding a queue with service rate C governed by (2.12)*

$$P(Q \geq B) \leq \frac{\sigma - \frac{\pi - \rho}{\pi - C} B}{\frac{C}{\rho} \sigma - B} \quad (2.13)$$

Proof: see [95, Pages 8ff.]

Theorem 2.7 (Theorem 3, [95]) *For the arrival process in theorem 2.6, the most conservative probability that the cell delay W through the queue exceeds $\delta = \frac{B}{C}$ is given by*

$$P(W \geq \delta) \leq \frac{\rho(\sigma - \frac{\pi - \rho}{\pi - C} B)}{C(\frac{C}{\rho} \sigma - B)} \quad (2.14)$$

Proof: see [95, Pages 10-13]

The derivation of theorem 2.7 uses the concept of *extremal shaped traffic*. Note that there is a set $\mathcal{A}$ of arrival processes which are $(\sigma, \rho) \wedge (0, \pi)$ -constrained. The worst case arrival process $A^* \in \mathcal{A}$ is the arrival process where $P(Q > B)$ is maximized, i.e.

$$A^* = \arg \sup_{A \in \mathcal{A}} \{P(Q \geq B)\}$$

Note that Q is a function of the constraints of the arrival process σ, ρ , and π as well as the service rate C used in theorem 2.6. A sample characterization of such a process is described in [95].

In the multiclass case discussed in [96], a number J of $(\sigma, \rho) \wedge (0, \pi)$ -constrained arrival processes are multiplexed into a shared queue. The assumption that for each stream $0 < \rho_j < \pi_j$ and $\sigma_j > 0$ for all $j = 1 \dots J$ is retained. To illustrate the results in this case, we assume that one arrival stream is isolated and indexed by 1, whereas all other $J - 1$ streams are bundled together and indexed by 2. Note that the aggregated $J - 1$ arrival streams are still $(\sigma_2, \rho_2) \wedge (0, \pi_2)$ -constrained by theorem 2.1. We further assume that the bandwidth C we seek to allocate to the shared queue satisfies $\rho_1 + \rho_2 < C < \pi_1 + \pi_2$. The following theorem establishes a probabilistic delay bound for the isolated arrival stream through the *shared queue*:

Theorem 2.8 (Theorem 1, [96]) *Under the assumptions for the arrival streams stated in the last paragraph and assuming that furthermore $\pi_2 > C$ and that $(\pi_2 - \rho_2)B > \sigma_2(\pi_2 - C)$, then*

$$P(W_1 > \delta) \leq \frac{\frac{C-\rho_2}{C-\rho}(\sigma_1 - B) + (\frac{\rho_1}{C-\rho} + \frac{\pi_2-C}{\pi_2-\rho_2})\sigma_2}{\frac{C-\rho_2}{C-\rho}\sigma_1 + \frac{\rho_1}{C-\rho}(\sigma_2 - B)} \quad (2.15)$$

where $\rho = \rho_1 + \rho_2$ and $\delta = \frac{B}{C}$.

Proof: see [96, Pages 5ff.]

Note that since in general the maximum queuing delay $\delta = \frac{B}{C}$, equation (2.15) also provides an upper bound on the cell loss rate for a fixed buffer size B .

The desired service rate for the shared queue can be found by the following algorithm for a given upper bound on the loss probability ϵ :

- For each stream $j: j = 1 \dots J$ with QoS requirement $P(W > \delta_j) < \epsilon_j$, solve for C_j

$$\epsilon_j = \frac{\frac{C_j-\tilde{\rho}}{C_j-\rho}(\sigma_j - \delta_j C_j) + (\frac{\rho_j}{C_j-\rho} + \frac{\tilde{\pi}-C_j}{\tilde{\pi}-\tilde{\rho}})\tilde{\sigma}}{\frac{C_j-\tilde{\rho}}{C_j-\rho}\sigma_j + \frac{\rho_j}{C_j-\rho}(\tilde{\sigma} - \delta_j C_j)}$$

where the aggregate arrivals of the stream not under consideration are $(\tilde{\sigma}, \tilde{\rho}) \wedge (0, \tilde{\pi})$ -constrained.

- Set $C = \max_{j=1 \dots J} C_j$.

The results presented in this section are significant because they are based on a few realistic assumptions. The service rate C of a queue is indeed required to be greater than the mean rate ρ of the aggregate arrival stream for stability. It also makes sense to restrict it to be less than the aggregate peak rate, $C \leq \pi$. No assumptions are made about the underlying stochastic model of the arrivals. What is more important however is that the delay and loss bounds presented above provide probabilistic QoS guarantees for individual streams, even if they are multiplexed into a shared queue! Furthermore, the results are based on conservative assumptions, by construction of the extremal shaped arrival process. The resulting resource allocations are thus still conservative.

Note that the above QoS bounds are for FCFS queues which are not part of a PSN architecture. However, the results can be extended to such architectures using the minimum bandwidth property as shown in the preceeding sections.

2.6 Related Concepts

The concept of an *arrival curve* is a generalization of (σ, ρ) -constraints [32]. An arrival curve is a non-decreasing function of time, $a(t)$, such that

$$A[t_1, t_2] \leq a(t_2 - t_1)$$

Note that a (σ, ρ) -constraint is a special case of such an arrival curve, namely $a(t_2 - t_1) = \rho(t_2 - t_1) + \sigma$. To derive delay or loss guarantees, a *service curve* is typically associated with an arrival curve (see also [126]). A service curve $s(t)$ is provided by a queue and gives a lower bound on the number of departures up to time t , assuming that the workload in the queue is zero at time 0. In the particular case where $s(t) = \max\{\rho t - \sigma_{svc}, 0\}$; $\sigma_{svc} > 0$; $\rho > 0$, the queue guarantees a service of (σ_{svc}, ρ) . End-to-end delay bounds can be derived using those two concepts in a manner similar to section 2.3. A CAC scheme for VBR traffic based on service curves is suggested in [137].

Also related to arrival and service curves is the concept of *burstiness curves* [103, 105]. A burstiness curve $b(C)$ characterizes the backlog in a FCFS queue with infinite buffer capacity fed by arrival stream A when it is served at a fixed rate C . It is defined in [105] as

$$b(C) = \sup_{t_1 \leq t \leq t_2} \sum_{t=t_1}^{t_2} [A_t - C] \quad (2.16)$$

where C represents the service rate and A_t represents the arrival at time $t \in [t_1, t_2]$ in a slotted model. Note that the formulation of such a function places particular emphasis on the buffer / bandwidth tradeoff. Any bounded arrival stream can be associated with such a burstiness curve. Similarly, a specific burstiness curve can be associated with a set of arrival streams. Again, the concept can be used for resource allocation subject to delay constraints and zero cell loss guarantees. These results are extended for probabilistic cell loss rates in [163] using the concept of a *loss function*. In [104] these concepts are embedded

into a general framework for service provisioning, where the optimal resource allocation in the network is determined by solving an optimization problem.

Chang [11] describes a filtering theory for deterministic traffic regulation and service guarantees based on the above concepts. This theory uses the $(\min, +)$ -algebra, where the usual multiplication operation is replaced by addition, and where the usual addition operation is replaced by the min-operator. A series of properties for this algebra are discussed and are subsequently applied to traffic regulation. This theory facilitates the proofs for some of the theorems described above and in the associated references. It also allows the systematic design of traffic regulators, for example how a leaky-bucket can be realized using a linear system.

Chapter 3

Quality of Service Provisioning under Effective Bandwidths

In this chapter we review effective bandwidth based Connection Admission Control (CAC), which may be used as an alternative algorithm to provide users with Quality of Service (QoS) guarantees. The chapter is organized in the same way as the last chapter. Section 3.1 starts off with definitions and properties of effective bandwidths. In section 3.2 we then provide results for the single node case with input and output relationships. In section 3.3 we present algorithms for QoS provisioning for a series of nodes in tandem, focusing on both asymptotic results for large buffer sizes and many sources. In section 3.4 we again illustrate how the results can be used for CAC. Section 3.5 summarizes this chapter with a review of other CAC schemes.

3.1 Effective Bandwidths

The concept of effective bandwidths has been widely studied in the last decade since it promises a resource provisioning approach analogous to circuit-switched networks [88, 85, 97, 68, 57, 48]. The effective bandwidth of a *stationary* source generating a workload $A[0, t]$ in the interval $[0, t]$ is defined as [88]

$$\alpha^A(\theta, t) := \frac{1}{\theta t} \log E[e^{\theta A(0, t)}] \quad (3.1)$$

where $0 < \theta, t < \infty$. The parameter θ is a network design parameter representing the QoS of the source. Alternatively, in the case where many sources are multiplexed over a single link, θ provides an indication of the degree of statistical multiplexing. In [57] it is described as a parameter measuring the susceptibility of the link to bursty traffic. The parameter t indicates the relevant time scale of the traffic. When the source is fed into a single-server queue with infinite buffer capacity, t is typically the most likely duration of the busy-period before the buffer reaches a threshold B [145].

Some of the properties presented for (σ, ρ) -constraints also apply to effective bandwidths. For example, the next theorem establishes the additivity property.

Theorem 3.1 (Additivity Property of effective bandwidths) *Consider a set of J independent arrival processes $A_j(\cdot); j = 1 \dots J$ with effective bandwidths $\alpha_j^A(\theta, t)$. Then the effective bandwidth $\alpha^A(\theta, t)$ of the aggregate arrival process is*

$$\alpha^A(\theta, t) = \sum_{j=1}^J \alpha_j^A(\theta, t) \quad (3.2)$$

Proof: The proof consists of simply applying the definition to the aggregate arrival process:

$$\begin{aligned} \sum_{j=1}^J \alpha_j^A(\theta, t) &= \sum_{j=1}^J \frac{1}{\theta t} \log E[e^{\theta A_j(0, t)}] \\ &= \frac{1}{\theta t} \log \prod_{j=1}^J E[e^{\theta A_j(0, t)}] \\ &= \frac{1}{\theta t} \log E\left[\prod_{j=1}^J e^{\theta A_j(0, t)}\right] \text{ (by independence)} \\ &= \frac{1}{\theta t} \log E[e^{\theta \sum_{j=1}^J A_j(0, t)}] \end{aligned}$$

which is by definition the effective bandwidth of the aggregate process. $\square$

Furthermore, by the following two theorems, the effective bandwidth always lies between the mean rate and the peak rate of the process.

Theorem 3.2 (Lower bound of effective bandwidths) *The effective bandwidth $\alpha^A(\theta, t)$ is bounded below by the mean rate μ , i.e.*

$$\lim_{\theta \rightarrow 0} \alpha^A(\theta, t) = \frac{1}{t} E[A(0, t)] \quad (3.3)$$

for fixed values of $t < \infty$ and as $\theta \rightarrow 0$.

Proof: Since the limit as $\theta \rightarrow 0$ for the numerator and the denominator equals 0, l'Hôpital's rule holds and so then

$$\begin{aligned} \lim_{\theta \rightarrow 0} \frac{1}{\theta t} \log E[e^{\theta A(0, t)}] &= \frac{\lim_{\theta \rightarrow 0} \frac{d}{d\theta} \log E[e^{\theta A(0, t)}]}{\lim_{\theta \rightarrow 0} \frac{d}{d\theta} \theta t} \\ &= \frac{1}{t} \lim_{\theta \rightarrow 0} \frac{E[A(0, t) e^{\theta A(0, t)}]}{E[e^{\theta A(0, t)}]} \\ &= \frac{E[A(0, t)]}{t} = \mu. \quad \square \end{aligned}$$

Theorem 3.3 (Upper bound of effective bandwidths) *The effective bandwidth $\alpha^A(\theta, t)$ is bounded above by the essential supremum $\sup_x \{P[A(0, t) > x] > 0\}/t$, i.e.*

$$\lim_{\theta \rightarrow \infty} \alpha^A(\theta, t) = \frac{\sup_x \{P[A(0, t) > x] > 0\}}{t} \quad (3.4)$$

for fixed values of $t < \infty$ and as $\theta \rightarrow \infty$.

Proof: See [88]. $\square$

For a stationary process with *independent increments*, the effective bandwidth reduces to a function of θ only, as shown in the following theorem. For the purpose of presentation, we restrict ourselves to a discrete time process with independent increments.

Theorem 3.4 (Independent increment effective bandwidth) *Consider a stationary discrete time arrival process $A_t; t = 1, 2, \dots$. If this process has independent increments, then the effective bandwidth $\alpha^A(\theta, t) = \alpha^A(\theta)$ for all t .*

Proof (sketch): Consider again the total number of arrivals in an interval $(0, t_1)$, where $t_1 \in \{1, 2, \dots\}$. By definition,

$$\begin{aligned}\alpha^A(\theta, t_1) &= \frac{1}{\theta t_1} \log E[e^{\theta \cdot A(0, t_1)}] \\ &= \frac{1}{\theta t_1} \log E[e^{\theta \sum_{t=1}^{t_1} A_t}] \\ &= \frac{1}{\theta t_1} \log E\left[\prod_{t=1}^{t_1} e^{\theta A_t}\right]\end{aligned}$$

Since there are independent increments, A_{t_1} is independent of A_{t_2} for all $t_1 \neq t_2$, so

$$\begin{aligned}\alpha^A(\theta, t_1) &= \frac{1}{\theta t_1} \log \prod_{t=1}^{t_1} E[e^{\theta A_t}] \\ &= \frac{1}{\theta t_1} \log E[e^{\theta A_1}]^{t_1} \quad (\text{by stationarity}) \\ &= \frac{1}{\theta} \log E[e^{\theta A_1}]\end{aligned}$$

The result follows by definition of the effective bandwidth. $\square$

Approximate effective bandwidths can be formulated by considering the expansion of $\alpha^A(\theta)$ in powers of θ . This yields the following expression:

$$\alpha^A(\theta) \approx \mu + \theta \frac{\sigma^2}{2} + o(\theta) \quad (3.5)$$

where μ and σ^2 represent the expected value and the variance of the arrival process respectively. For stationary sources [28, 21] feeding a buffer of size B served at a rate C the variance can then be related to the index of dispersion for intervals, given by

$$\varphi = \lim_{t_1 \rightarrow \infty} \frac{1}{t_1} \text{Var}\left(\sum_{t=1}^{t_1} A_t\right) \quad (3.6)$$

The relation between the effective bandwidth and the index of dispersion is then

$$\alpha^A(\theta) \approx \mu + \theta \frac{\varphi}{2B} \quad (3.7)$$

Note that the arrival process can indeed be auto-correlated. The index of dispersion for intervals captures this dependence of the arrival process in time. The results reported in [21] show the accuracy of such an approximation under these assumptions. The importance of the index of dispersion for intervals for the characterization of aggregate arrival processes was already pointed out by Sriram and Whitt [149].

3.2 Input-Output relations for the single node

In this section we present results for the effective bandwidth of the departure process from a FCFS queue. First, we consider the case of a single stream before discussing the case of aggregate streams multiplexed together into the queue. The results presented here are used in the next section to quantify the QoS provided by the network. To simplify the presentation of the results, we restrict ourselves to arrival processes with independent increments.

To determine the effective bandwidth of the output process of a FCFS, the concept of *decoupling bandwidths* is required, defined by deVeciana et al. [34] as

$$\beta^A(\theta) = \arg \sup_{\alpha} [\theta \alpha - \Lambda_A^*(\alpha)]. \quad (3.8)$$

Here,

$$\Lambda_A^*(\alpha) = \sup_{\theta} [\theta \alpha - \theta \alpha^A(\theta)] = \frac{d}{d\omega} \Lambda_A(\omega) \big|_{\omega=\alpha} \quad (3.9)$$

is the Legendre transform of the cumulant generating function $\Lambda_A(\theta) = \theta \alpha^A(\theta)$. The decoupling bandwidth represents rate of the arrival process so that *large* amounts of workload arrive at the queue in such a way that the output process is affected by the queue. It always exceeds the effective bandwidth of a source but is bounded above by $\beta^A(\theta) \leq 2\alpha^A(\theta) - \alpha^A(0)$. Consider now an arrival process with effective bandwidth $\alpha^A(\theta)$ feeding a single server queue with fixed service rate C . The queue modifies the process such that the effective bandwidth of the output process $\alpha^D(\theta, C)$ is given by

$$\alpha^D(\theta, C) = \begin{cases} \alpha^A(\theta) & \text{if } \beta^A(\theta) < C \\ C - \frac{1}{\theta} \Lambda_A^*(C) & \text{otherwise} \end{cases} \quad (3.10)$$

Figure 3.1: Effective bandwidth. Decoupling bandwidth and output effective bandwidth for an on-off source with mean 1 and peak 10.

Figure 3.1 illustrates the effective bandwidth, the decoupling bandwidth and the output effective bandwidth for an on-off source with mean rate 1 and peak rate 10, and service rate 8. Note that the decoupling bandwidth serves as a condition to determine whether the arrival stream is sufficiently serviced such that its effective bandwidth is unaffected by the queue. Both the decoupling bandwidth and the effective bandwidth of the output process can be modified to account for J streams multiplexed together into the queue. In this case, $\beta_j^A(\theta) = \arg \sup_{\alpha} [\theta \alpha - \Lambda_j^*(\alpha)]$ and

$$\alpha^D(\theta, C) = \begin{cases} \sum_{j=1}^J \alpha_j^A(\theta) & \text{if } \sum_{j=1}^J \beta_j^A(\theta) < C \\ C - \frac{1}{\theta} \inf_{\sum_{j=1}^J \alpha_j = C} \sum_{j=1}^J \Lambda_j^*(\alpha_j) & \text{otherwise} \end{cases} \quad (3.11)$$

Note that the decoupling bandwidth of the aggregate arrival process provides a condition determining whether the effective bandwidth of the *aggregate* output process remains unaffected by the queue. For service provisioning of individual streams however, a condition is required to determine when the output effective bandwidth of an *individual* stream remains the same. Such a condition is given by

$$\beta_i^A(\theta) + \sum_{j=1; i \neq j}^J \alpha_j^A(0) < C \quad (3.12)$$

where i indicates the stream of interest.

The extension of the above results to a PSN architecture has been studied in [36]. Here, instead of multiplexing the arrivals streams into a single shared buffer they are segregated into a number N of parallel buffers in such a way that only statistically identical streams share one buffer. This ensures that the QoS for the aggregate process also carries over to the individual processes. The N buffers *share* the total link bandwidth in a work-conserving manner by giving each buffer a fraction $\rho_n; n = 1 \dots N$ of the capacity C . The effective bandwidth of the departure process from a particular buffer is thus dependent on the bandwidth allocations $\rho_n; n = 1 \dots N$. It is given by

$$\alpha_n^D(\theta, C) = \begin{cases} \alpha_n^A(\theta) & \text{if } \beta_n^A(\theta) < \rho_n C \text{ and } E[A_n] \leq \rho_n C \\ \rho_n C - \frac{1}{\theta} \Lambda_n^*(\rho_n C) & \text{if } \beta_n^A(\theta) \geq \rho_n C \text{ and } E[A_n] \leq \rho_n C \\ \rho_n C & \text{if } E[A_n] > \rho_n C \end{cases} \quad (3.13)$$

where A_n denotes the aggregate arrivals into buffer n , $\alpha_n^A(\theta)$ and $\beta_n^A(\theta)$ denote the effective and decoupling bandwidths for buffer n respectively and where $\Lambda_n^*(.)$ denotes the Legendre transform of the aggregate arrival process into buffer n . Note that if the stability condition for buffer n , given by $E[A_n] < \rho_n C$, is satisfied then we have just the output effective bandwidth result from above. If buffer n receives insufficient service such that this condition is violated, the output effective bandwidth is constant and equal to the allocated service rate. However, in this case the buffer occupancy no longer has a finite mean.

For a further extension on the effective bandwidth of the departure process from a queue with stochastic service capacities the reader is referred to [13, 12].

3.3 End-to-End QoS Provisioning

The attraction of effective bandwidths is that they can be used to provision the network's resources such that QoS guarantees can be given. Typically, such guarantees are expressed as upper bounds on the loss probability or on the maximum end-to-end delay. These results are presented in this section. Two asymptotic regimes have emerged in this context: the large-buffer asymptotic and the many-sources asymptotic. These regimes establish the resource requirements at a particular node such that the desired QoS can be guaranteed.

The model of a VP consisting of H independent PSNs along the end-to-end path is retained. Before describing the end-to-end resource provisioning algorithm, we outline the resource requirements at a single node to guarantee QoS.

3.3.1 Large Buffer Asymptotics

Consider again a stationary arrival process $\{A_t\}$ with independent increments feeding a single server queue with fixed service rate C . The workload process in the interval $(0, t)$ is denoted by $\{W(0, t) = W_t\}$. As usual, stability conditions are imposed on the queue of the form $E[A(0, t) - Ct] < 0$ and $Ee^{\theta W} < \infty$. The effective bandwidth of the arrival process can be related to the probability of the buffer exceeding a threshold B as follows. Consider first the case of a finite time T :

$$\begin{aligned}
 P(Q_T > B) &= P(\sup_{0 < t \leq T} W_t > B) \\
 &\leq \sum_{t=1}^T P(W_t > B) \quad (\text{since } \{\sup_{0 < t \leq T} W_t > B\} \subset \bigcup_{t=1}^T \{W_t > B\}) \\
 &\leq T \sup_{0 < t \leq T} P(W_t > B) \\
 &\leq T \sup_{0 < t \leq T} e^{-\theta B} Ee^{\theta W_t} \quad (\text{from the Chernoff bound})
 \end{aligned}$$

and so taking the logarithm and letting B go to infinity we get:

$$\begin{aligned}
 \lim_{B \rightarrow \infty} \frac{1}{B} \log P(\sup_{0 < t \leq T} W_t > B) &\leq \lim_{B \rightarrow \infty} \frac{1}{B} \log [T e^{-\theta B} \sup_{0 < t \leq T} Ee^{\theta W_t}] \\
 &= -\theta \quad (\text{since } Ee^{\theta W_t} < \infty \text{ by assumption}) \quad (3.14)
 \end{aligned}$$

For the case $T \rightarrow \infty$ we can write

$$\begin{aligned}
 \sum_{t=T}^{\infty} P(W_t > B) &\leq e^{-\theta B} \sum_{t=T}^{\infty} Ee^{\theta W_t} \quad (\text{from the Chernoff bound}) \\
 &= e^{-\theta B} \sum_{t=T}^{\infty} e^{\theta t \alpha^W(\theta, t)}
 \end{aligned}$$

However, $\lim_{t \rightarrow \infty} \theta \alpha^W(\theta, t) < 0$ by the stability assumptions. Hence, for any particular t we can find a positive bound ϵ such that $-\epsilon > \theta \alpha^W(\theta, \infty)$ and so

$$\begin{aligned} e^{-\theta B} \sum_{t=T}^{\infty} e^{\theta t \alpha^W(\theta, t)} &\leq e^{-\theta B} \sum_{t=T}^{\infty} e^{-t\epsilon} \\ &< e^{-\theta B} \frac{1}{1 - e^{-\epsilon}} \end{aligned}$$

Taking the logarithm and the limit again provides

$$\begin{aligned} \lim_{B \rightarrow \infty} \frac{1}{B} \log P(\sup_{0 < t \leq T} W_t > B) &\leq \lim_{B \rightarrow \infty} \frac{1}{B} \log[e^{-\theta B} \frac{1}{1 - e^{-\epsilon}}] \\ &= -\theta \end{aligned}$$

Thus for all $\theta > 0$, we have [37, 41, 10]

$$\alpha^A(\theta, \infty) < C \iff \lim_{B \rightarrow \infty} \frac{1}{B} \log P(Q > B) \leq -\theta \quad (3.15)$$

A similar lower bound can be derived so that the inequality in (3.15) holds with equality (see [8, 143]). Equation (3.15) is referred to as the *large buffer asymptotic*. For stationary sources we thus have an exponential bound on the probability that the workload process exceeds a given threshold B :

$$P(Q > B) \leq P(W > B) \approx \kappa e^{-\theta B} \quad (3.16)$$

for some constant κ . This equation can also be used to obtain a bound on the delay through the queue as follows:

$$P(\mathcal{D} > \delta) = P\left(\frac{W}{C} > \delta\right) \approx \kappa e^{-\theta \delta C} \quad (3.17)$$

where $\mathcal{D}$ represents the worst-case delay and δ represents the corresponding threshold. A similar constraint for the delay jitter can be derived. The jitter $\mathcal{J} = |\mathcal{D} - E[\mathcal{D}]|$ is bounded above by the maximum transmission time for a node [37], since

$$P(|\mathcal{D} - E[\mathcal{D}]| > \gamma) < P(\mathcal{D} > \gamma)$$

so

$$P(\mathcal{J} > \gamma) < P\left(\frac{W}{C} > \gamma\right) \approx \kappa e^{-\theta \gamma C} \quad (3.18)$$

Equations (3.16, 3.17, 3.18) are the *effective bandwidth approximations* for resource provisioning. The probability of an infinite buffer exceeding threshold B is typically taken as an upper bound for the probability of a *finite* buffer exceeding its given capacity B . Note that the strictest θ -bound is given by θ^* for which

$$\alpha^\lambda(\theta^*, \infty) = C \quad (3.19)$$

Since $\alpha^\lambda(\theta, \cdot)$ is an increasing function in θ , we typically restrict ourselves to equation (3.19) when provisioning resources using the effective bandwidth approximation.

Further details on resource provisioning using effective bandwidths can be found in [143, 8, 83, 12, 35] and in the references therein.

Consider now the case of a PSN using a general processor sharing (GPS) scheduler, where N FCFS queues share a single link with capacity C and are thus only given a fraction $\rho_n; n = 1 \dots N$, of the total bandwidth capacity. DeVeciana and Kesidis [36] derived resource provisioning results for this case. To illustrate the result we denote the effective bandwidths of the *aggregate* arrival streams into their respective buffers by $\alpha_n^\lambda(\theta); n = 1 \dots N$. In this case, for the buffer of interest $i = 1 \dots N$ and for $\theta > 0$

$$\alpha_i^\lambda(\theta) + \min[(1 - \rho_i)C, \sum_{n=1; n \neq i}^N \alpha_n^\lambda(\theta)] \leq C \implies \lim_{B \rightarrow \infty} \frac{1}{B} \log P(Q_i > B) \leq -\theta \quad (3.20)$$

For details on this equation, the reader is referred to [36, 37].

Choudhury et al. [14] examined the accuracy of the above results. They show that for a single source, the effective bandwidth approximation with $\kappa = 1$ is reasonable yet still conservative. Moreover, if the exact value of κ is determined, the effective bandwidth approximation is surprisingly accurate even for small values of the buffer threshold B . However, these results do not carry over to the case where J streams are multiplexed. In this case, the accuracy of the effective bandwidth approximation deteriorates significantly as the following occur:

- the number of multiplexed sources increases
- the link utilization decreases
- the buffer size decreases
- the sources are significantly more or less bursty than a Poisson source.

In particular, the constant in (3.16) is significantly different from $\kappa = 1$. For example, if a cell loss probability constraint of 10^{-9} is imposed, the effective bandwidth approximation is good for values $\kappa \geq 10^{-4}$, only moderately accurate in the range $10^{-4} < \kappa < 10^{-9}$, and poor otherwise. Choudhury et al. analyze a scaled arrival process, where the number of sources J increases while keeping the total aggregate arrival process fixed. While such a scaling is certainly not applicable for real traffic, it does provide insights into the approximation as J increases. In this case, the constant κ is found to take an exponential form $\kappa \sim \iota e^{-J\omega}$, such that a better approximation is given by

$$P(Q > B) \approx \iota e^{-J\omega} e^{-\theta B} \quad (3.21)$$

for some constant ι . The decay parameter ω here is found to decrease with the traffic intensity of the aggregate arrival process but increases as the burstiness of the sources deviates from an equivalent Poisson model.

3.3.2 Many Sources Asymptotics

The shortcomings of the effective bandwidth approximation based on the large buffer asymptotic motivated research in the asymptotic behavior of the arrival process as the number of sources *and* the buffer size increases. Consider again the case where J sources *share* a single buffer. The results are categorized as the *many-sources asymptotic*, which is founded on the following theorem:

Theorem 3.5 (Many Sources Asymptotic) *Suppose that an aggregate of J sources feeds a single server queue with service rate JC and buffer size JB . Denote the effective bandwidth of the aggregate arrivals by $\alpha^A(\theta, t) = \sum_{j=1}^J \alpha_j^A(\theta, t)$, where $\alpha_j^A(\cdot)$ denotes the effective*

bandwidth of the individual arrival processes. Assume further that the effective bandwidths of the individual source are differentiable and that their limits with respect to t and θ exist. Then

$$\lim_{J \rightarrow \infty} \frac{1}{J} \log P(Q > JB) = \sup_t \inf_{\theta} [\theta t \alpha^A(\theta, t) - \theta(B + tC)] \quad (3.22)$$

Proof: The proof is given in [29].

Thus the right hand side of equation (3.22) gives the exponential decay rate of the queue size. An analogous expression to (3.15) is

$$\sup_t \inf_{\theta} [\theta t \alpha^A(\theta, t) - \theta(B + tC)] < -B\vartheta \iff C \geq \sup_t \inf_{\theta} \left[\frac{B}{t} \left(\frac{\vartheta}{\theta} - 1 \right) + \alpha^A(\theta, t) \right] \quad (3.23)$$

So if the decay rate of the queue size is to be bounded above by $B\vartheta$ for a given B and ϑ , then equation (3.23) can be used to find the appropriate service rate C . Unfortunately, the right hand side of (3.23) indicates that in addition to the aggregate effective bandwidth function, the optimal values for θ and t are also required. This is referred to as the *operating point* of the queue. These parameters have been studied in [61, 145, 26], where the operating point is determined using numerical methods. Alternatively, for a given QoS requirement ϱ , the following can be set.

$$\varrho = - \sup_t \inf_{\theta} [\theta t \alpha^A(\theta, t) - \theta(B + tC)]$$

Taking the partial derivatives with respect to B and C we then obtain

$$\frac{\partial \varrho}{\partial B} = \theta \quad (3.24)$$

$$\frac{\partial \varrho}{\partial C} = \theta t \implies t = \frac{\partial B}{\partial C} \quad (3.25)$$

Hence, θ is the rate of change of the logarithm of the QoS requirement with respect to the buffer size (given C). Similarly, the product θt is the rate of change of the QoS requirement with respect to the service rate (given B). Both partial derivatives indicate how the

QoS requirement would change in response to a change in the resources from the current allocation (B, C) . As shown in equation (3.25), the time scale is also linked to the buffer bandwidth tradeoff for a fixed QoS requirement ϱ .

The many-sources asymptotic is related to the large buffer asymptotic by noting that

$$\lim_{B \rightarrow \infty} \frac{1}{B} \sup_t \inf_{\theta} [\theta t \alpha^A(\theta, t) - \theta(B + tC)] = -\theta \quad (3.26)$$

The proof for this equation is also given in [29]. In this way, the large buffer asymptotic is a special case of the many sources asymptotic.

The many-sources asymptotic can also be related to approximation (3.16). As in the case of Choudhury et al., Courcoubetis and Weber observe a relationship between the burstiness of the arrival process and the decay rate. In particular, for sources more bursty than Poisson.

$$\sup_t \inf_{\theta} [\theta t \alpha^A(\theta, t) - \theta(B + tC)] < B\theta$$

whereas for sources less bursty than Poisson

$$\sup_t \inf_{\theta} [\theta t \alpha^A(\theta, t) - \theta(B + tC)] > B\theta$$

The difference between the left hand side and the right hand side thus gives an indication of the burstiness of the arrival process. Courcoubetis and Weber also establish a precise relationship between κ and the number of multiplexed sources. In particular

$$\kappa = \iota e^{-J\omega} \approx e^{-J[I(B, C) - B\theta - \frac{g(B, C, J)}{J}]} \quad (3.27)$$

where $\lim_{B \rightarrow \infty} I(B, C) = \theta$ and where $g(B, C, J)/J \rightarrow 0$ as $J \rightarrow \infty$ for fixed B . The above equation emphasizes that the value of the statistical multiplexing factor κ not only depends on the number of sources J , but also on the provisioned resources B and C . For details, see [29].

3.3.3 End-to-end Provisioning Algorithm

The results presented above can be modified to obtain the resource requirements at each node of the end-to-end path. We restrict ourselves in this section to the case where the

path consists of H independent PSN nodes. We assume that the buffer size $B_{n,h}$ is fixed and that the link capacity C_h as well as the bandwidth allocations to the FCFS buffers $\rho_{n,h} = C_{n,h}/C_h$ are to be determined by the QoS requirements. This allows us to impose loss, delay and jitter constraints. Note that the derivations presented above provide QoS guarantees for the aggregate arrival stream serviced by a queue. However, if we restrict only statistically identical streams to share a particular queue, then the QoS for the aggregate carries over to a QoS for the individual streams. We thus assume that at each hop the virtual path is part of an N -queue PSN architecture, and that only $J_{n,h}: n = 1 \dots N; h = 1 \dots H$ statistically identical streams with similar QoS requirements are multiplexed together. We augment the usual notations introduced in previous sections by superscripts / subscripts (n, h) which index the n^{th} queue at the h^{th} hop of the end-to-end path. Finally, we restrict ourselves to the large buffer asymptotic in this part.

Consider first the case of a loss guarantee. Assume that a stream requires an end-to-end loss probability bounded above by $\epsilon_{\mathcal{L}} < \kappa H$. We need to allocate sufficient bandwidth $C_{n,h}^{\mathcal{L}}$ at each node to meet this requirement. We can make the following argument to ‘distribute’ such a requirement over the H queues [37]:

$$\begin{aligned}
 P(\text{Loss}_{\setminus P}) &= 1 - P(\neg \text{Loss}_{\setminus P}) \\
 &= \prod_{h=1}^H 1 - P(\neg \text{Loss}_h) \\
 \Rightarrow P(\text{Loss}_{\setminus P}) &\approx \sum_{h=1}^H P(\text{Loss}_h) \\
 &= HP(\text{Loss}_1)
 \end{aligned}$$

where the last equality holds if the queues have identical loss probabilities. We thus distribute the losses equally over all queues in the end-to-end path. The QoS requirement for queue n, h thus becomes

$$\theta_{h,n} = -\frac{\log(\epsilon_{\mathcal{L}}/(\kappa H))}{B_{n,h}} \quad (3.28)$$

An algorithm for bandwidth allocation for queue $n: n = 1 \dots N_h$ at hop $h: h = 1 \dots H$ along the end-to-end path then is

- Compute $\theta_{n,h} = -\frac{\log(\epsilon_{\mathcal{L}}/(\kappa H))}{B_{n,h}}$
- For each $j = 1 \dots J_h$ compute

$$\sum_{j=1}^{J_{n,h}} \alpha_{j,n,h}^A(\theta_{n,h}) = C_{n,h}$$

to ensure that the QoS requirement is met.

- Allocate bandwidth $C_{n,h}^{\mathcal{L}}$ and $C_h = \sum_{n=1}^{N_h} C_{n,h}^{\mathcal{L}}$ at PSN h
- Compute $\alpha_{j,n,h-1}^A(\cdot) = \alpha_{j,n,h}^D(\cdot)$ for resource allocation at the downstream queue.

Note that the decoupling bandwidth condition (3.12) is not used in this algorithm. We therefore have to compute the output effective bandwidth for resource allocation at the downstream node. This requires not only the effective bandwidth to be known for all θ , but also the decoupling bandwidth. We further note that the $\theta_{n,h}$ may differ, not only within a PSN to accommodate streams with different statistics, but also between nodes because streams may be multiplexed / demultiplexed at intermediary nodes along the end-to-end path. We could modify the algorithm to avoid computing the output effective bandwidth as follows:

- Compute $\theta_{n,h} = -\frac{\log(\epsilon_{\mathcal{L}}/(\kappa H))}{B_{n,h}}$
- For each $j = 1 \dots J_{n,h}$ compute

$$\begin{aligned} \sum_{j=1}^{J_{n,h}} \alpha_{j,n,h}(\theta_{n,h}) &= C_{n,h} \\ \beta_{j,n,h}^A(\theta_{n,h}) + \sum_{i=1; i \neq j}^{J_{n,h}} \alpha_{i,n,h}(0) &= C_{j,n,h} \end{aligned}$$

to ensure that the QoS requirement is met and that the output effective bandwidth remains unchanged by the operation of the queue.

- Allocate bandwidth $C_{n,h}^{\mathcal{L}} = \max[C_{n,h}, \{C_{j,n,h} : j = 1 \dots J_{n,h}\}]$

Fast simulation results for cell loss probabilities based on a similar algorithm were reported in [50].

Consider now a maximum end-to-end delay constraint of the form

$$P(\mathcal{D} > \delta) < \epsilon_{\mathcal{D}}$$

where δ represents the queuing and processing delay along the end-to-end path only (thus ignoring the link propagation delays and the switching delays). Again, we only allow streams with similar delay constraints to be multiplexed in a queue. We assume that each node is allowed an equally distributed portion of the total delay, so we impose

$$P(\mathcal{D}_{n,h} > \frac{\delta}{H}) < \epsilon_{\mathcal{D}} \quad (3.29)$$

Using the effective bandwidth approximation (3.17) we get

$$\theta_{n,h} C_{n,h}^{\mathcal{D}} = -\frac{\log(\epsilon_{\mathcal{D}}/\kappa)}{\delta/H}: \quad \epsilon_{\mathcal{D}} < \kappa \quad (3.30)$$

which can easily be solved since $\theta_{n,h} C_{n,h}^{\mathcal{D}} = \theta_{n,h} \sum_j \alpha_{j,n,h}(\theta_{n,h})$ (by equation (3.19) and the additivity property of effective bandwidths) is strictly increasing in $\theta > 0$. From the solution of this equation we immediately obtain the bandwidth requirement for queue n at hop h . Note that this QoS requirement is independent of the buffer size, as shown in the effective bandwidth approximation (3.17).

Finally consider a constraint for the delay jitter of the form

$$P(|\mathcal{D} - E[\mathcal{D}]| > \gamma) < \epsilon_{\mathcal{J}}$$

By an argument in [101] the end-to-end jitter is in the order of magnitude of the maximum transfer time of *one* queue. At each node, we can thus use the effective bandwidth approximation (3.18) to find

$$\theta_{n,h} C_{n,h}^{\mathcal{J}} = -\frac{\log(\epsilon_{\mathcal{J}}/\kappa)}{\gamma}: \quad \epsilon_{\mathcal{J}} < \kappa \quad (3.31)$$

and again solve for both $\theta_{n,h}$ and $C_{n,h}^{\mathcal{J}}$. Note that we have presented the bandwidth allocation requirements for the three QoS types separately. Should any combination of them have to be satisfied at the same time, the allocation to queue (n, h) should of course be the maximum of $\{C_{n,h}^{\mathcal{L}}, C_{n,h}^{\mathcal{D}}, C_{n,h}^{\mathcal{J}}\}$. The above resource allocations can also be adapted for the

case where the bandwidth allocation is fixed and the buffer size is variable. However, since the effective bandwidth approximations for delay and jitter are independent of the buffer size, for fixed bandwidth allocation and aggregate arrivals to the queue we are unable to provision for delay and jitter QoS constraints.

Finally, in the above presentation we have included the constant κ in the effective bandwidth approximation. Recall that with $\kappa = 1$, the resource allocations are conservative and thus allow the network to *guarantee* QoS.

3.4 Connection Admission Control

Resource provisioning is typically applicable over longer time periods. However, in time periods of order of magnitude of a connection, the buffer / bandwidth resources in the network are fixed. Rather than viewing the aggregate arrival process as fixed to determine the resource requirements, we assume that the resources are fixed and we view the number of arrival processes as variable. The network performs connection admission control to arriving streams to ensure that the QoS guaranteed to existing connections is not violated if an additional stream is accepted. To simplify the notation in this section, we drop the subscripts h introduced in the last section.

Considering the large buffer asymptotic, the CAC conditions are given by equations (3.15) and (3.20) for the FCFS queue and the PSN case respectively. In this case, θ , B and C are fixed. θ represents the QoS provided by the queue and can be directly related to a loss probability through equation (3.16). Similarly, θ can be related to a delay / jitter QoS through equations (3.17) and (3.18), given δ and γ . For the FCFS queue, a newly arriving stream with effective bandwidth $\alpha_{J+1}^A(\theta)$ can be admitted to the queue while preserving the QoS with parameter θ of the J existing calls if

$$\alpha_{J+1}^A(\theta) \leq C - \sum_{j=1}^J \alpha_j^A(\theta) \quad (3.32)$$

or equivalently for the PSN case a newly arriving stream with effective bandwidth $\alpha_{J_i-1,i}^A(\theta)$ can be admitted to queue i

$$\alpha_{i,J_i+1}^A(\theta) \leq C - \sum_{j=1}^{J_i} \alpha_{j,i}^A(\theta) - \min[(1 - \rho_i)C, \sum_{n=1:n \neq i}^N \alpha_n^A(\theta)] \quad (3.33)$$

where as above we denote the *aggregate* effective bandwidth into queue n by $\alpha_n^A(\cdot)$.

Note that the parameter κ could be incorporated in the CAC decision through a modified QoS parameter $\tilde{\theta}$ for all arrival streams. The parameter θ measures the decay rate of the logarithm of $P(Q > B)$ for the aggregate stream. Thus, if a newly arriving stream requires QoS $\epsilon_{\mathcal{L}} = e^{-\tilde{\theta}B}$, then

$$\begin{aligned} \epsilon_{\mathcal{L}} = e^{-\tilde{\theta}B} &\geq \kappa e^{-\theta B} \\ &= e^{-\theta B - \log \kappa} \\ &= e^{-B(\theta - (\log \kappa)/B)} \\ &\iff \tilde{\theta} \leq \theta - (\log \kappa)/B \\ &\iff \tilde{\theta} + (\log \kappa)/B \leq \theta \end{aligned}$$

So if the network guarantees a decay rate of θ , a newly arriving stream can be admitted if

$$\alpha_{J+1}^A(\tilde{\theta} + (\log \kappa)/B) \leq C - \sum_{j=1}^J \alpha_j^A(\tilde{\theta} + (\log \kappa)/B) \quad (3.34)$$

Since $(\log \kappa)/B \leq 0$, the modified effective bandwidth condition is smaller than under equation (3.32). More streams can be admitted because of statistical multiplexing in the network.

The CAC algorithm associated with the many-sources asymptotic is more complex. Equation (3.23) is hard to interpret but does indicate a clear relationship between the effective bandwidth of the aggregate arrival process and the resource capacity at the queue. Part of the complexity arises from the additional time parameter. Furthermore, the decay rate of the logarithm of the probability that the queue length exceeds a threshold involves a minimization in θ and a maximization in t . This was described as the *operating point* of the link earlier. Evaluating the impact of an additional stream on the operating point is hard.

However, if we classify the arrival streams into K classes and if each class has $j_k: k = 1 \dots K$ arrival streams, then an acceptance region can be defined. With this modification, the aggregate effective bandwidth can be written as

$$\alpha^A(\theta, t) = \sum_{k=1}^K j_k \alpha_k(\theta, t)$$

and the right hand side of equation (3.22) becomes

$$\sup_t \inf_{\theta} [\theta t \sum_{k=1}^K j_k \alpha_k(\theta, t) - \theta(B + tC)] \quad (3.35)$$

For a given QoS requirement $J\vartheta$, we can then find a set of vectors $\{j = (j_1 \dots j_K)\} \in Z_+^J$ such that for each j and for a given JC, JB we have

$$\log P(Q > JB) \leq -J\vartheta$$

or

$$\sup_t \inf_{\theta} [\theta t \sum_{k=1}^K j_k \alpha_k(\theta, t) - \theta(B + tC)] < -J\vartheta \quad (3.36)$$

We denote the set of these vectors j by $A(J\vartheta, JC, JB)$ to reflect the dependency on the QoS requirement and the resource capacity of the link. If this set is scaled by the number of streams J , then the following limit holds [29, 88]

$$\lim_{J \rightarrow \infty} \frac{A(J\vartheta, JC, JB)}{J} = \bigcap_{0 < t < \infty} A_t \quad (3.37)$$

where

$$A_t = \left\{ j : \inf_{\theta} \left[\theta t \sum_{k=1}^K j_k \alpha_k(\theta, t) - \theta(B + tC) \right] < -J\vartheta \right\} \quad (3.38)$$

For each operating point it then suffices to find the boundary of this acceptance region. If this boundary is differentiable, then it can be approximated by a tangent at the operating point (θ^*, t^*)

$$\sum_{k=1}^K j_k \alpha_k(\theta^*, t^*) = C + \frac{B}{t^*} - \frac{\vartheta}{\theta^* t^*} \quad (3.39)$$

For a given operating point, the right hand side of this equation is given and hence the boundary of the acceptance region can be calculated. The problem is then to determine the operating point of the link for a given buffer and bandwidth capacity. Courcoubetis et al. [25] have analyzed the values of (θ^*, t^*) under a variety of traffic mixes and resource capacities. They find that the operating point remains stable for small changes in the traffic mix¹. They propose to determine the operating points for a given bandwidth / buffer capacity off-line by solving equation (3.36), and thus determining the boundaries of the acceptance region for the K classes of traffic streams. The operating point of the link remains valid for periods where the traffic mix is stable. It may thus be changed few times during the day. Details are given in [25, 88, 61].

3.5 Other CAC Schemes

In this section we discuss a number of alternative CAC schemes. This review is not intended to be exhaustive, but summarizes some of the work in this context.

The concept of fast buffer or bandwidth reservation was proposed by Tranchier et al. and Turner [152, 154]. Fast buffer reservation is intended for bursty ATM sources. Under this scheme, a source dynamically requests small increases / decreases in the allocated bandwidth. For example, before a burst is transmitted, a fast bandwidth reservation cell (in form of a OAM cell) is sent to request the required bandwidth for the burst. This cell propagates to the destination. Each node along the end-to-end path makes an accept / reject decision, and allocates the requested bandwidth if sufficient bandwidth is available. Similarly, at the end of a burst a reservation release cell is transmitted and each node releases the bandwidth reserved for a burst.

The fast buffer reservation scheme proposed by Turner operates in a similar manner, allocating buffer rather than bandwidth for particular VCs. A VC is considered to be either in an idle state or in an active state. When in the active state, buffer space is allocated to the VC according to the traffic contract. When the VC enters the idle state this buffer

¹Their experiments include multiplexing many MPEG and non-MPEG video streams as well as combining video streams with voice streams and Internet traffic over different link speeds and buffer sizes.

space is released again. State transitions per VC occur by specially marked cells at the beginning or end of a burst. Note that this has the advantage of avoiding the transmission of OAM cells at the beginning of a burst. The CAC algorithm proposed by Turner is based on this concept.

The CAC scheme proposed by Vesilo and Mark [155, 156] is related to effective bandwidth CAC. However, they compute the effective bandwidth at each node using either the generalized or burst peakedness, defined as the variance to mean ratio. Each node inside the network then reports to the CAC algorithm in edge device its bandwidth availability. Under this approach, no assumptions have to be made about the underlying traffic model. Furthermore, statistical multiplexing gains can be incorporated in this measure for the effective bandwidth. Analogous to effective bandwidth CAC, this approach assumes exponentially decaying overflow probabilities. The peakedness of a call can then be related to the required QoS θ of a source through

$$\theta = \frac{\mu - \lambda}{z(\theta)} \quad (3.40)$$

where $z(\theta)$ represents the peakedness, μ is the available bandwidth and λ the mean arrival rate. If the QoS parameter of the network exceeds this value for θ , the source can be admitted. Otherwise it is rejected.

Wischik [162] proposed an admission control scheme which allows bandwidth intense calls to book resources in advance. This scheme is intended for calls which can be planned and which require strict QoS guarantees from the network, such as video conferences. Allowing network operators to book resources in advance increases the call's priority and does not require over provisioning by the network. The proposed CAC algorithm provides statistical multiplexing at the call level. Wischik analyzes the call blocking probability using an effective bandwidth formulation at the call level. He particularly analyzes the timing requirements of the proposed CAC scheme. Booking ahead is shown to reduce call blocking if resources are reserved sufficiently in advance. However, booking too far in advance does not improve the blocking probability of the call. Surprisingly, such a scheme is shown to either increase or decrease utilization levels inside the network, depending on the traffic mix.

Connection admission control algorithms for self-similar sources are proposed by Fu and Latchman [55] and Giordano et al. [63]. In both papers the cell loss probability of single server queue with deterministic service rate is determined when fed with a self-similar source (modeled by an FGN process). Under the assumption that the user is able to characterize the mean, variance and Hurst parameter of the arrival stream, an upper bound for the CLP is determined (see equation 6.29). Based on this formula, the bandwidth required to achieve a desired CLP is determined. The corresponding CAC algorithm admits such a stream if sufficient bandwidth is available, if not the stream is rejected. Giordano et al. even consider the cases where J streams with identical Hurst parameter are multiplexed into a single queue, as well as the case of multiplexing J streams with different Hurst parameters.

A review of several CAC algorithms is given in Perros and Elsayed [131]. They categorize schemes into 5 groups: equivalent capacity, heavy traffic approximation, upper bounds on the CLP, fast buffer / bandwidth reservation and time windows. Representative algorithms of the first three groups are then evaluated numerically. Perros and Elsayed investigate the effect of the buffer size (small, large) on the number of admitted streams and the sensitivity of each CAC scheme to changes in the buffer size or to changes in the required CLP. They also evaluate the effect of changing the ratio of buffer size to mean burst size on the number of admitted streams. In all their experiments CAC schemes based on equivalent capacity outperform the other schemes.

A class of CAC algorithms is categorized under the term *measurement based admission control* (MBAC). In such schemes the network measures the available capacity. Arrival streams are often assumed to be bounded (e.g. (σ, ρ) -constrained) and can be admitted if the measured available capacity exceeds ρ . Such an approach is in contrast the CAC algorithms introduced so far, where sources have been assumed to accurately characterize their behavior.

An MBAC algorithm based on effective bandwidths is proposed by Courcoubetis et al. [30]. They measure the loss probability of a queue with buffer size B , service rate C and J admitted streams, which is denoted by $\phi(J, B, C)$. Admitting a new stream results in a loss probability $\phi(J(1 + \epsilon), B, C)$, which can be approximated as

$$\phi(J(1 + \epsilon), B, C) \approx \phi(J, B, \frac{C}{1 + \epsilon}) \quad (3.41)$$

A newly arriving stream can be admitted to the network if the resulting CLP does not violate the QoS requirements of existing calls and if it satisfies the QoS requirements of the new call. To measure $\phi(J, B, \frac{C}{1 + \epsilon})$ even for small loss probabilities, Courcoubetis et al. propose to measure the decay rate of a virtual queue with a smaller buffer size. The decay rate of the virtual queue can then be used to approximate the decay rate of the original queue by a simple extrapolation. Their MBAC algorithm thus proposes to continuously measure the decay rate of a virtual queue using the existing traffic flowing through the queue. This could be done by a special chip inside the switches. The estimated spare capacity can then be fed into the routing or CAC algorithms.

Floyd [53] and Ràcz [132] also propose an MBAC scheme based on equivalent bandwidths. However, their approach is based on Hoeffding bounds. Under this approach the equivalent capacity of admitted streams with peak rate $\pi_j; j = 1 \dots J$ is

$$C_H(m, \{\pi_j\}, \epsilon_{\mathcal{L}}) = m + \sqrt{\frac{\log(1/\epsilon_{\mathcal{L}}) \sum_{j=1}^J \pi_j^2}{2}} \quad (3.42)$$

where m is the measured mean rate of all admitted streams so far and $\epsilon_{\mathcal{L}}$ is an upper bound for the probability of the instantaneous arrival rate exceeding the allocated capacity. Note that the peak rates of the admitted streams are assumed to be enforced by a policer (e.g. a leaky bucket parameterized for peak-rate policing). A newly arriving stream with peak rate π_{J+1} can then be admitted with QoS $\epsilon_{\mathcal{L}}$ if

$$C_H(m, \{\pi_j\}, \epsilon_{\mathcal{L}}) + \pi_{J+1} < C \quad (3.43)$$

where C is the total capacity of the link. If the stream is admitted, then the measured mean rate will change accordingly. The mean rate of existing calls is assumed to be measured using an exponentially weighted moving average. Also, it is assumed that this measured mean rate is adjusted before the subsequent CAC decision has to be made.

The notion of *acceptance regions* is introduced in [58, 60, 65]. Again, the current network load is measured. Upon arrival of a call, this load is compared to a pre-computed acceptance

region. Calls are admitted if the load is less than a threshold derived from the acceptance region. These threshold values are determined using Bayesian decision theory. In this way, both call rejection and cell losses can be traded-off for each other. In [58] Gibbens and Kelly illustrate their MBAC scheme using K classes, each with $j_k: j_k = 1 \dots J_k: k = 1 \dots K$ streams. Arrivals are assumed to follow a Poisson distribution. Call holding times are exponentially distributed. Each call follows an on-off pattern. Based on these assumptions the acceptance region can be computed. If the currently measured load, measured over time periods of length τ , is below the pre-computed threshold, the call is admitted. Otherwise, the call is rejected and no other CAC decisions are made until a call which is currently in progress terminates. Gibbens and Kelly analyze a variety of acceptance regions, based on tangents at the peak / arbitrary points / origin or of slope 1.

Jamin et al. [79, 80] and Clark et al. [17] propose a measurement-based admission control algorithm for integrated services networks. Their scheme considers only (σ, ρ) -constrained streams and a network architecture as described in section 2.1. In particular, they assume that a reservation protocol is used to communicate the resource requirements of a stream to the network. They also assume that each newly arriving stream knows its values for (σ, ρ) . Instead of computing the available capacity by summing up the (σ_j, ρ_j) -constraints of the J existing calls as in equations 6.12 and 6.13, the MBAC algorithm *measures* the current (σ, ρ) constraint of all existing streams. A newly arriving stream can be admitted if the measured (σ, ρ) -constraints of the existing streams and the $(\sigma_{J+1}, \rho_{J+1})$ -constraint of the new stream does not violate the desired delay bound. The advantage of this approach is that the declared parameters for the mean or peak rate only affect the CAC decision. However, these are superseded by the measurement algorithm once the stream is admitted. In this way, it does not matter if the traffic descriptors are inaccurate. Jamin et al. focus on predictive services, where jitter can be tolerated (e.g. real-time applications which can adapt to cell delays). Their simulation studies demonstrate the MBAC scheme using on-off, Pareto on-off and fractional ARIMA processes as examples. In this way they show that the proposed MBAC scheme is also valid for LRD traffic.

Grossglauser and Tse [66, 67] study measurement based admission control under heavy traffic assumptions. In their study there is always a call waiting to be admitted to the

network, represented by a single link, bufferless model. Calls are assumed to have exponential holding times and their bandwidth requirements are modeled by a Markov fluid source. Under these assumptions the authors compute the maximum number of allowable calls such that a certain loss probability is achieved. In particular, they study the impact of estimation errors (which may lead to subsequent erroneous admission decisions), call dynamics (which determine the recovery times from estimation errors) and memory implications (estimators can be improved by including more memory at the cost of reduced ability to adapt to non-stationarities). They find that estimation errors are particularly harmful to the observed CLP if the CAC algorithm is memoryless and if the measured values are taken to be the true values (i.e. estimated parameters are real parameters). Furthermore, errors in estimating the mean arrival rate are more significant than measurement errors of the standard deviation. The call departure dynamics are also shown to be important and lead Grossglauser and Tse to define a critical time scale, as $\tilde{T}_h = T_h/\sqrt{n}$, where T_h is the average call holding time and where n system size defined as C/μ . The critical time scale is shown to decrease with shorter holding times or a bigger system. Finally, increasing the memory of the CAC algorithm is shown to be advantageous since bandwidth estimates are more accurate (due to a larger number of samples) and smoother. This gives the MBAC algorithm a better control over the worst estimation error over the critical time scale.

A survey of MBAC schemes is provided by Shiimoto et al. [142]. They classify CAC along three dimensions: whether buffering is considered or not, whether QoS is measured and compared to a threshold value or whether an effective bandwidth is compared to the available bandwidth, and whether traffic models are assumed or whether the traffic characteristics are measured. Using this taxonomy, they describe the details of 6 MBAC schemes, proposed in [60, 135, 140, 141, 151, 43]. The schemes are then compared according to their effective bandwidth or CLR calculation, their assumptions about buffering, the bandwidth efficiency, implementation complexity and the dependency on the traffic model. The overview concludes that MBAC schemes are attractive because of the difficulties related to traffic descriptors (especially with the emergence of new applications) and because of the distortions introduced inside the network. MBAC schemes based on effective bandwidths and where the buffering effect is not explicitly considered are particularly promising for

future high-speed networks.

Chapter 4

Traffic Regulation and QoS Provisioning in Practice

The last two chapters illustrated how resources can be provisioned to give users QoS guarantees. In this chapter we review traffic shaping and policing mechanisms which are intended to enforce the traffic contract between the user and the network. We focus on the $G/D/1/B$ FCFS queue and the leaky bucket regulators in sections 4.1.1 and 4.1.2. We particularly emphasize their performance. Section 4.1.3 then provides an overview of window-based algorithms which may be used for policing. We conclude this chapter with a brief summary of QoS-provisioning in broadband networks in practice, focusing on the Asynchronous Transfer Mode (ATM), Integrated Services (IntServ) and Differentiated Services (DiffServ) standards.

4.1 Traffic Shaping and Policing

In this section we describe traffic shaping and policing mechanisms. The two concepts are closely related. In traffic shaping, the arrival stream is passed through a device (such as a $G/D/1/B$ queuing system) which is parameterized such that the resulting output stream complies with the traffic contract. In contrast, an arrival stream is assumed to already comply with the traffic contract before it is passed into a policer. The policer is parameter-

Figure 4.1: Traffic Shaper modeled as a G/D/1/B queue

ized according to the contractual values. It verifies that the arrival stream indeed complies with the established contract. In case of violations, punitive actions (marking or losses) are taken. Otherwise the stream passes through the policer unchanged. In both cases, the output stream of the traffic shaper or policer satisfies the traffic contract. Shapers typically introduce artificial delays to smooth the arrival stream whereas policers are supposed to make instantaneous control decisions (allow cell to pass unmarked or mark / drop the cell). We use the generic term *traffic regulator* for both shapers and policers.

4.1.1 The FCFS Queue

A single server queue with deterministic service rate C is the simplest mechanism to regulate the user's traffic (see figure 4.1). Cells or packets are buffered in the queue and are delayed for at least $1/C$ time units. The output stream is bound above by the rate of the shaper C . The buffer thus absorbs any burstiness in the arrival stream, presenting to the network a 'smoother' stream. The burst reducing properties of such a queue have been proven in [103, 105, 106, 163].

Recall that the performance results derived in the last two chapters were based on such deterministic queues. We will thus not repeat the results for loss probabilities, delays or delay variations here.

4.1.2 The Leaky-Bucket Algorithm

The leaky bucket algorithm is a popular traffic regulator proposed for policing or shaping. An unbuffered and a buffered version of this regulator are typically considered. We discuss each of these in turn.

Figure 4.2: Leaky bucket regulator without cell buffer

The traditional leaky bucket algorithm keeps a counter which is incremented at a rate C' until a maximum B is reached. Every time a cell arrives the counter is decremented. If an arriving cell finds the counter at zero, it is either marked as a lower priority cell or dropped immediately. This operation is typically modeled using tokens¹, representing the counter. Figure 4.2 illustrates this version of the leaky bucket. Tokens arrive at the deterministic rate C' and are buffered in a token buffer of size B . Tokens arriving into a full token queue are lost. Any arriving cell has to consume a token. If no token is available, then the cell is marked or lost (to simplify the subsequent discussion, we only consider the case of losses henceforth). Otherwise, the cell is admitted into the network without further delay.

Because of the lack of queuing for arriving cells, the performance of the leaky bucket is characterized by its loss behavior. Cells are not delayed. It can be shown that the cell losses of the regulator correspond to those of a $G/D/1/B$ queuing system with deterministic service rate C . A simple analysis based on closed queuing networks is given in [136], where both the cell arrival process as well as the token arrival process are assumed to follow a Poisson distribution. The case of on-off fluid sources is derived by Butto et al. in [9] using Laplace transforms. For exponentially distributed on and off phases with parameters λ_1 and λ_2 respectively and with peak rate π , the probability of a cell / packet being lost by the leaky bucket is

¹These can be interpreted as permits to enter the network.

Figure 4.3: Leaky bucket regulator with cell buffer

$$P_{loss}^{LB} = \frac{\pi - C}{\pi} \frac{\frac{1}{\lambda_1(\pi - C)} - \frac{1}{\lambda_2 C}}{\frac{1}{\lambda_1(\pi - C)} e^{\frac{B}{\lambda_1(\pi - C) + \lambda_2 C}} - \frac{1}{\lambda_2 C}} \quad (4.1)$$

Note that this equation actually represents the fraction of *bits* lost by the leaky bucket out of the total number of bits presented to the leaky bucket. This is considered as the cell loss probability by Butto et al. See also [133, 134].

The second variation of the leaky bucket algorithm includes a cell buffer as depicted in figure 4.3. We will call it *buffered leaky bucket* from now on. The operation in the case when tokens are queued in the token buffer is identical to the unbuffered leaky bucket. However, in case no tokens are available when cells arrive into the leaky bucket, they are queued until tokens become available. The token buffer still has a finite capacity B^T . The cell buffer is modeled as either a finite buffer with capacity B^P or as an infinite buffer. Note that cell losses now occur only in the first case when arriving cells find the cell buffer full. Tokens are still arriving at fixed rate C and are lost when the token buffer occupancy is B^T upon arrival. The inclusion of a second buffer now implies that cells can also be delayed in the buffered leaky bucket.

Performance analysis of the buffered leaky bucket is again similar to a $G/D/1/B$ queue with fixed service rate C , representing the token arrival rate. However, here B is to be interpreted as to *combined* buffer size $B = B^T + B^P$. This is motivated by the observation

that at any point in time either the cell buffer or the token buffer can have a positive occupancy, but not both² [47, 5]. Let Q , Q^T and Q^P denote the occupancy of the combined buffer, the token buffer and the cell buffer respectively. If $Q^T > 0$ then $Q^P = 0$ and if $Q^P > 0$ then $Q^T = 0$. The occupancy of the combined buffer is given by $Q = Q^P - Q^T + b^T$. Note that $Q = 0$ if the token buffer is full and no cells are queued. Occupancies $Q < b^T$ represent the states of the leaky bucket where tokens but no cells queue. For $Q = b^T$, both cell and token buffers are empty. Occupancies $Q > b^T$ represent the states where cells queue but where the token buffer is empty.

Based on this analogy, Elwalid and Mitra [47] derived the distribution of the combined queue for general N -state Markov fluid sources. Given the queue size distribution Q , they derive the respective distributions for the token and cell buffer, throughput and loss statistics as well as a characterization of the output process of the buffered leaky bucket. They note in particular that the partitioning between the cell buffer and the token buffer characterizes the trade-off between the delay incurred in the leaky bucket and the burstiness of the output process. For example, if $B^T = B$ and $B^P = 0$, the maximum burst size of the output process is given by B^T and cells are not delayed in the leaky bucket. On the other hand, if $B^P = B$ and $B^T = 0$ the output stream is smoothed to single cell departures at most every $1/C$ time units. The cell delay is then bounded above by B^P/C . Results for Poisson cell arrivals and deterministic token arrivals are derived in [6]. Berger analyzes the buffered leaky bucket for Markov Arrival Processes (MAP). Performance results for discrete time Markov processes are given by Sidi [144]. They derive throughput, waiting time distribution and the inter-departure process using Laplace transforms for both cases of a finite and infinite cell buffer. The burst-reducing properties of this leaky bucket regulator have again been proven by [2, 100, 107, 105, 103, 32, 163]. These authors show that the output of a leaky bucket controller is (B^T, C) -constrained and therefore as B^T rises the burstiness of the output stream rises. Anantharam and Konstantopoulos particularly analyze the effect of the token buffer on the output process. They argue that the token buffer size should be minimized (obeying possible delay constraints) such that the output stream is smoothed. In this way, the leaky bucket absorbs the statistical fluctuations of the arrival streams as

²If both a token is available and a cell is queued then the cell departs the leaky bucket without delay.

much as possible.

An analysis of the buffered leaky bucket based on effective bandwidths can be found in [33]. We present these results in detail since they can be used subsequently for our minimum cost traffic shaping framework. DeVeciana also establishes that the buffered leaky bucket may be modeled as a single server queue with deterministic service rate C . He characterizes the dynamics of the leaky bucket in discrete time by the following Lindley equation³

$$\tilde{Q}_{t+1} = \max[-B^T, \tilde{Q}_t + A_t - C] \quad (4.2)$$

The workload processes for the token and cell buffers can be derived from (4.2) as follows:

$$Q_{t+1}^T = \max[0, -\tilde{Q}_t] \quad (4.3)$$

$$Q_{t+1}^P = \max[0, \tilde{Q}_t] \quad (4.4)$$

Note that if we set $Q_t = \tilde{Q}_t + B^T$ then we have the usual Lindley equation for a single server queue with deterministic service rate⁴ C . It follows that if an arrival stream with effective bandwidth $\alpha^A(\theta)$ feeds the leaky bucket with parameters $(B = B^P + B^T, C)$, then the effective bandwidth of the output stream is also characterized by (3.10) (see [33] for the proof). We can thus also derive large buffer effective bandwidth approximations for the loss and delay. In particular,

$$P(Q^P > B^P) = P(Q > B^P + B^T) \quad (4.5)$$

$$= P(Q > B^P(1 + x)) \quad (4.6)$$

$$= P(Q > B) \quad (4.7)$$

where the token buffer size is expressed as a fraction $x \geq 1$ of the cell buffer size $B^T = xB^P$.

From our results in the previous chapter we thus have

³Which are equivalent to the dynamics of a random walk reflected at $-B^T$.

⁴Which is just the usual random walk reflected at 0.

$$P(Q^P > B^P) = P(Q > B^P(1+x)) \quad (4.8)$$

$$\leq e^{-\theta B^P(1+x)} \quad (4.9)$$

$$= e^{-\theta^P B^P} \quad (4.10)$$

where $\theta^P = \theta(1+x)$. Note that $\theta^P > \theta$, so the logarithm of the cell loss probability decays faster than the equivalent single server queue with buffer size B^P and service rate C . Alternatively, if a QoS requirement is set to ϵ_C , then the required buffer size for the cell buffer in the leaky bucket regulator is $B^P = -\frac{\log \epsilon_C}{\theta(1+x)} \leq -\frac{\log \epsilon_C}{\theta}$. The corresponding delay constraint for a given threshold δ can similarly be established as

$$P(D^P > \delta) = P(Q^P/C > \delta) = P(Q > \delta C(1+x)) \leq e^{-\theta \delta C(1+x)} \quad (4.11)$$

Note that the performance analysis for the buffered leaky bucket regulator based on effective bandwidths is more general than the analytical results summarized above. Equations (4.10) and (4.11) hold for any SRD arrival stream with effective bandwidth $\alpha(\theta)$.

4.1.3 Window Algorithms

Windowing algorithms traditionally used for flow or congestion control can also be used as policers. In this section we briefly review the sliding window, fixed window and exponentially weighted moving average algorithms from a policing perspective. Note that none of these algorithms uses buffers to make an accept / reject decision for cells, and hence no delays are incurred.

The sliding window policer is parameterized by a fixed window size W and a threshold B of allowed cells. This window slides over time, advancing each slot. The number of arrivals in a window is compared to the threshold B . Any cells exceeding this threshold are dropped. In this way, the sliding window policer ensures that the output stream is limited by a maximum burst size of B cells per time window W . Note that the windows are overlapping. The performance of a sliding window policer can be analyzed by a loss system without buffering and with B servers [136, 134].

The fixed window policer operates in a similar way. It too is parameterized by a window size W and a threshold B . However, in this case the windows are advanced every W slots such that two consecutive windows are non-overlapping. The loss probability can easily be computed by calculating the probability of more than B arrivals in an interval of length W (see also [134]).

The exponentially weighted moving average policer is again similar to the fixed window policer. The arrivals in non-overlapping windows of length W is considered. However, in this case an additional parameter ζ is needed to determine the number of conforming cells. This number is based on a recursive equation which takes the arrivals in previous windows into account through the weighting factor ζ . In particular, let S_t denotes the weighted number of arrivals for slot t , which can be defined by either of the following equations

$$S_t = (1 - \zeta)A_t + \zeta S_{t-1} \quad (4.12)$$

$$S_t = A_t + \zeta S_{t-1} \quad (4.13)$$

Cells are dropped if $S_t > B$. The performance of such a policer can again be derived by considering the number of arrivals in an interval of length W .

4.1.4 Effective Bandwidth Policing

A method of policing effective bandwidth traffic descriptors has been proposed by Kesidis [93]. Consider a user declaring effective bandwidth $\beta^A(\theta)$, where θ indicates the user's QoS requirements. The user's stream is shaped by a token bucket regulator with token arrival rate set to the peak rate of the arrival process. In this way, the token bucket regulator initially does not impact the user's arrival stream. The output stream of the token bucket regulator is then fed into the network, and also cloned into a virtual buffer policing device. The purpose of the virtual buffer policing device is to measure the decay rate of the output stream of the traffic regulator, and if a violation is detected, modify the token arrival rate of the regulator.

The virtual buffer policer serves the cloned output stream of the regulator at the user-declared effective bandwidth $\beta^A(\theta)$. The decay rate of the virtual buffer queue occupancy

is then measured (e.g. using the Kullback-Leibler approach described in chapter 7).

If the source is in compliance, then the measured decay rate $\hat{\theta}$ of the virtual buffer policer should exceed or equal the QoS parameter θ corresponding to the declared rate. θ corresponds to the declared effective bandwidth, and so with the virtual buffer being served at rate $\beta^A(\theta)$, the desired QoS should be observed. In this case, the arrival rate of tokens into the token bucket regulator remains at the source's peak rate.

If the source is in violation, then the measured decay rate $\hat{\theta}$ of the virtual buffer policer is less than the declared θ . This indicates that the output stream of the token bucket regulator served at the declared effective bandwidth $\beta^A(\theta)$ experiences more cell losses than corresponding to the declared θ , for example. However, this should not be the case, since the virtual buffer is served at the declared rate $\beta^A(\theta)$, and thus indicates that the source sends traffic in excess of its declared effective bandwidth.

To enforce compliance, the token arrival rate into the regulator is then changed from the peak rate to the declared effective bandwidth. Once $\hat{\theta} \geq \theta$, the token arrival rate is once again set to the peak rate of the arrival stream, rendering the regulator ineffective during periods of compliance.

Note that the policer takes some time before a violation is detected. Also, the policer may become active during periods of compliance if the estimation of $\hat{\theta}$ is erroneous. For details see once [93, pages 126 ff].

4.1.5 Effectiveness of Policers

The effectiveness of the leaky bucket and the three window-based policers was studied by Rathgeb [133] and Butto [9] (see also [38]). They studied how well each of these mechanisms ensures that a given parameter set in the traffic contract is not violated. Rathgeb compared these schemes using discrete time on-off sources according to the following criteria:

- ability to detect violations quickly
- proportional reaction to violations (i.e. the loss rate should be in proportion to the degree of violation)
- simplicity

While all of the above policing algorithms can easily be implemented in real-time (using counters), their ability to react proportionally and to detect violations quickly differed significantly. In particular, the sliding window and fixed window algorithms are less flexible than the leaky bucket or the exponentially weighted moving average algorithms. The former two algorithms are not flexible enough to handle short term burst. These studies particularly highlight the problem of policing traffic descriptors based on mean values (e.g. the mean arrival rate or the mean burst length). Such means are by definition calculated over long periods, typically the entire connection duration. Consequently, short term violations of such descriptors are hard to detect. For example, if the leaky bucket algorithm is parameterized by the mean rate of a bursty arrival process, then the token buffer has to be unrealistically large to allow conforming bursts to pass the policer unchanged. However, this then implies that a violation of the long term mean rate takes a long time to detect, since it is hard to distinguish from a burst. This problem is aggravated the more bursty the arrival process. Setting the parameters of such policers to acceptable values is thus hard. Recommendations for leaky bucket parameters for different traffic types are given in [134, 9].

Policing the peak rate is also problematic. Peak rate policing must be done over short time intervals (of the order of magnitude of the maximum burst size) and it is thus possible to easily detect peak-rate violations of the source. The parameters of the corresponding policer are typically set to the peak rate of the source. In case of a leaky bucket policer, the bucket size b is set to zero. However, peak rate policing becomes ineffective if the policer is far removed from the original source. For example, if the cells have to traverse a user's network before they are policed, the peak rate characteristics may change. This problem is studied by Guillemin and Roberts in [69] who derive alternative parameterizations for the policers (see [134] for a summary). Due to such limitations of policing algorithms, streams may have to be re-shaped inside the network to ensure their compliance with the traffic contract.

In light of the problems associated with policing the traffic contract, we focus in chapter 6 on traffic regulation, using either a leaky bucket or a FCFS queue. Thus, instead of allowing the user to declare the mean rate or the peak rate as traffic descriptors and subsequently

policing the arrival stream, we propose to use the parameterization of the traffic regulator as traffic descriptors for the traffic contract. The arrival stream is then passed through a regulator parameterized according to the contractual parameters. Our minimum cost traffic shaping framework determines these parameters such that the cost of the connection is minimized while given QoS constraints are satisfied.

4.2 QoS Provisioning in Practice

In this section we review the main QoS features of three prominent network architectures: Asynchronous Transfer Mode (ATM), Integrated Services (IntServ) and Differentiated Services (DiffServ). These architectures are all designed to enable QoS provisioning inside the network. To limit the length of this section, we only review those parts of the mechanisms and functions of each of these technologies which are specific to support QoS demands. In particular, we underline the use of traffic shaping and traffic contracts in each architecture.

4.2.1 Asynchronous Transfer Mode

Each call in an ATM network is classified into one of five service categories: constant bit rate (CBR), real-time variable bit rate (rt-VBR), non real-time variable bit rate (nrt-VBR), unspecified bit rate (UBR) and available bit rate (ABR). Appendix A provides a detailed description of each service class. Quality of Service in ATM networks centers around the concept of a *traffic contract*, which is negotiated between the user and the network. It is made up of the following elements:

- the QoS parameters
- the traffic descriptors for each service class, including the network's definition of a compliant connection
- conformance checking rules

The QoS parameters describe the end-to-end performance of the ATM connection. The traffic descriptors characterize the user's traffic stream. They are further categorized as

the source traffic descriptors (provided by the user), and the connection traffic descriptors (which include the source traffic descriptors, the cell delay variation tolerance (CDVT) and conformance definitions). The conformance checking rules indicate how conformance is determined.

Upon connection establishment, the network performs a CAC function based on this traffic contract. The user's stream is admitted if sufficient resources are available and if its admission does not impede on the QoS of already admitted streams. In this case, the required resources are allocated. Otherwise the stream is rejected. A successfully admitted stream is subsequently policed by the ATM usage parameter control (UPC) or network parameter control (NPC) function⁵. UPC is performed by the GCRA, a variant of the leaky bucket algorithm. Depending on the service class as well as the conformance definition in the traffic contract, one or more GCRA's may be used to detect contract violations. Cells which do not conform to the traffic contract are said to be non-conforming. They are either marked by setting the cell loss priority (CLP) bit in the ATM header to 1, or dropped immediately. For this reason each service class may have two cell streams, identified by the value of their CLP (CLP=0 and CLP=0+1). CAC and UPC/NPC functions differ by service category. Note that a traffic contract is specified for each direction.

The ATM Forum [54] currently defines the following indicators for QoS:

Maximum Cell Transfer Delay (MaxCTD) : transmission delay of a cell between the source and destination, including transmission, propagation and switching delays.
Upper bound on the end-to-end transfer delay.

Mean Cell Transfer Delay (MeanCTD) : average end-to-end transfer delay.

Cell Loss Ratio (CLR) : ratio of lost cells to the sum of all successfully transmitted cells (therefore excluding dropped and errored cells).

Cell Error Ratio (CER) : ratio of errored cells to the sum of transmitted cells (therefore including errored cells, but excluding dropped cells).

⁵UPC is typically performed across the UNI. NPC is typically performed across the NNI. UPC is the more general term and often used to refer to either UPC or NPC.

Table 4.1: QoS parameters by ATM service class

	CBR	rt-VBR	nrt-VBR	UBR	ABR
Max CTD	Yes	Yes	No	No	No
Mean CTD	No	No	Yes	No	No
CDV	Yes	Yes	No	No	No
CLR	Yes	Yes	Yes	No	Yes

Severely Errored Cell Block Ratio (SECBR) : number of cell blocks in which cells have been lost, misinserted, or errored as a ratio to the total number of transmitted cell blocks. A cell block is defined as a sequence of user cells, typically 28, 256, 512 or 1024 cells. If the number of errored, misinserted or lost cells exceeds a certain threshold, then the cell block is counted as severely errored.

Cell Misinsertion Rate (CMR) : number of misinserted cells over a specified period of time, expressed as a ratio to the total number of cells. A cell is misinserted if its header contains an error which is not detected and leads to a transmission along the wrong VC/VP.

In the ATM Traffic Management specification 4.0, only the first 4 QoS parameters are negotiated between the network and the user. The latter two are a function of the network's performance. Table 4.1 lists the QoS parameters required by each service class.

Each connection is typically characterized by a set of up to 4 source traffic descriptors. These are

Peak Cell Rate (PCR) : maximum number of cells which can be transmitted during a specific time interval.

Sustainable Cell Rate (SCR) : maximum average cell transmission rate ($< PCR$).

Maximum Burst Size (MBS) : maximum number of cells that can be transmitted at the peak rate.

Table 4.2: Traffic descriptors by ATM service class

	CBR	rt-VBR	nrt-VBR	UBR	ABR
PCR	Yes	Yes	Yes	Yes	Yes
SCR	n/a	Yes	Yes	n/a	n/a
MBS	n/a	Yes	Yes	n/a	n/a
MCR	No	No	No	No	Yes
CDVT(PCR)	Yes	Yes	Yes	Yes	Yes
CDVT(SCR)	n/a	Yes	Yes	n/a	n/a

Minimum Cell Rate (MCR) : minimum number of cells that the source is transmitting during any interval.

The connection traffic descriptors consist of these four parameters, and adds two further components: a CDVT and a definition of a compliant connection. The CDVT is typically determined by the network, specifying the maximum amount of time that the network is allowed to alter the jitter of a connection. This could happen as a result of traffic shaping, through multiplexing or by inserting OAM cells to perform management functions. The addition of this parameter prevents the network from increasing the jitter to unacceptable levels for the user.

The definition of a compliant connection includes a specification of what defines a conforming cell, when such cells are dropped or when cells are counted as delay etc. This definition is typically based on the GCRA and differs by service class. For some service classes different definitions may be made depending on the value of the CLP bit (i.e. for the cell stream $CLP=0$, or for the cell stream $CLP=0+1$). In all cases, cells which pass the GCRA are considered compliant. However, in some cases even cells which are tagged by the GCRA may be categorized as compliant. Table 4.2 lists the traffic descriptors for each service class.

Implementation of the CAC function is left to the hardware implementation. The specification allows for differentiation between the $CLP = 0$ and the $CLP = 0 + 1$ streams.

For details on QoS provisioning in ATM networks see [51, 111, 54].

4.2.2 Integrated Services

In contrast to ATM, the Integrated Services model (IntServ) provides QoS functions at the network layer. IntServ is an extension of the best-effort IP protocol. There are three conceptual similarities between IntServ and ATM: both architectures define a limited set of QoS categories, both rely on traffic contracts to define the QoS level and to obtain traffic characterizations from the user (and consequently require the users stream to be policed), and both IntServ and ATM rely on virtual connections between the sender and the receiver. Packets or cells always follow the path established by the CAC function.

In addition to the best effort service provided by the current Internet, two QoS categories are supported in IntServ:

- Guaranteed Quality of Service
- Controlled Load

The focus in these categories is on the timely delivery of traffic, i.e. on the packet delay. IntServ distinguishes real-time traffic from non real-time traffic. The latter is referred to as *elastic* traffic. Real-time traffic requires strict QoS guarantees from the network. The transmission rate of a flow is determined by the application, which is not considered to be adaptive. In contrast, elastic traffic is able to tolerate some delay variation. Applications are assumed to be adaptive, for example by allowing flow control mechanisms to vary the rate of the flow.

The traffic contract (called TSPEC) which is established between the user and the network in IntServ consists of the following parameters:

- token bucket rate
- token bucket size
- peak rate
- minimum policed unit

- maximum packet size

The first two parameters fully characterize a token bucket policer / shaper. IntServ assumes that each flow receiving QoS is policed or shaped. The network is also expected to re-shape flows in between routing or QoS domains. The minimum policed unit allows the network to calculate a reasonable estimate of the resources required to forward a packet. Packet sizes smaller than the minimum policed unit are rounded up to this parameter. The maximum packet size defines the largest packet size which can conform to the traffic contract. Packets of larger size may not receive the contracted QoS. Such a traffic contract has to be provided by each flow demanding either controlled load or guaranteed QoS. CAC is performed by the network prior to the data transmission. IntServ recommends the use of the RSVP protocol to establish a path between the source and the destination and reserve the required resources.

The goal of the controlled load service category is to provide better than best effort service. It is intended for adaptive applications which can tolerate some losses or delay. The user can expect a service level which is equivalent to a lightly utilized best-effort service. The specification [139] does not allow for precise bandwidth or delay guarantees. The traffic is characterized by the TSPEC and each flow is policed by a token bucket. The TSPEC is used by each node to ensure that sufficient resources are available to handle a flow. The details on resource provisioning are left open. Each router on the end-to-end path verifies the conformance of packets. Routers are expected to still forward non-conforming traffic on a best effort bases. However, they also have to ensure that this does not affect the QoS of any conformant traffic.

The guaranteed service, in contrast, provides both bandwidth guarantees and a bounded delay. No packet losses are incurred through buffer overflows. Such guarantees can be given by controlling the maximum queuing delay in the routers, which can be calculated using the TSPEC parameters (in particular the token-bucket buffer size and rate). Each router reserves the required buffer size and bandwidth to enable the QoS guarantees. In addition to the TSPEC, flows are characterized by the reservation characteristics, the RSPEC. The latter consists of the reserved bandwidth along the end-to-end path, as well as a slack

term. The reserved bandwidth is required to exceed the token bucket rate. The slack term represents the difference between the desired delay and the delay obtained with the reserved bandwidth. Each guaranteed service flow again goes through a CAC process where the required resources are determined and reserved if successful. Flows are again policed by a token bucket.

IntServ recommends the use of RSVP to reserve the required resources. RSVP is a signaling protocol which communicates the resource requirements along the end-to-end path and allows for reservations to be made. In contrast to virtual circuits, RSVP takes a connectionless (soft-state) approach. The reservations made by each flow have to be renewed periodically, otherwise a time-out occurs and routers make the reserved resources available again. Reservations are made using two message types: RESV and PATH messages. First, the sender transmits a PATH message, containing the user's TSPEC and an optional characterization of the flow called the ADSPEC. As this PATH message traverses the network to the destination each router registers the flow. The address of the upstream router is added to the PATH message. Upon arrival of the PATH message at the destination, each router is aware of its previous hop, thus establishing a virtual connection. Finally, each intermediary router optionally updates the information in the ADSPEC, which collects performance characteristics along the end-to-end path. In particular, the ADSPEC characterizes the minimum bandwidth on the path between source and destination. It indicates if all routers on the virtual connection are capable of making reservations. Furthermore, delay information is collected, such as the minimum path latency (sum of the individual link latencies) or parameters indicating the queuing delay⁶. Upon arrival of the PATH message at the destination, the receiver can calculate the resources required to achieve the desired QoS. The RESV message then propagates back from the receiver to the sender to finalize the appropriate reservations. RESV messages are designed to handle multicast messages in the form of different reservation styles. For example, reservations can be distinct for each sender or shared between a group of senders. The list of senders can be specified explicitly or through wildcards.

⁶In form of the *error terms* C and D in the specification. These terms characterize the deviation of the delay from the delay experienced by a fluid-flow model.

Details on IntServ can be found in [51, 159, 164, 139].

4.2.3 Differentiated Services

The Differentiated Services (DiffServ) framework proposed in [119, 7, 118, 98] specifies a set of tools or functional elements to augment the current IP architecture such that different types of services can be provided. This is primarily achieved by re-defining the TOS field in the IPv4 / IPv6 header to a differentiated services (DS) field. Packets are given different values (codepoints) in the DS field which subsequently determine how a packet is treated inside the network. A network may construct a number of services upon the values of the DS field.

The emphasis in the DiffServ architecture is to push much of the complexity associated with the provision of different services to the edge of the network. This allows the implementation of fast forwarding algorithms in the network interior. The scalability problems related to IntServ are addressed by allowing definitions for aggregate flows.

The architecture provides an integrated framework of a variety of technical aspects enabling service differentiation. Nodes are assumed to implement one of several scheduler (weighted fair queuing, class based queuing, weighted round robin, priority scheduling etc.). Similarly, queues are assumed to support QoS-enabling dropping mechanisms such as random early detection (RED) or RIO. Like the previous two architectures introduced in this section, DiffServ is fundamentally based on traffic contracts (called 'service level agreements', SLAs). These are established between users and the network, where the user may be a different network domain supporting the DiffServ architecture. To ensure compliance and enforce these SLAs, traffic classifiers and conditioners are placed at the boundary of each DiffServ domain. A classifier identifies packets pertaining to a specific SLA. It maps certain packets to a codepoint. A traffic conditioner may consist of several building blocks: a meter, packet marker, traffic shaper, and a policer or packet dropper. The meter measures the temporal characteristics of the identified flow. The marker is responsible for setting the codepoint (DSCP), either to all packets in the flow, or according to the state of the meter. The traffic shaper ensures compliance with the flow's SLA by introducing artificial delays if necessary. Finally, the policer / dropper is responsible for discarding some packets if these

are not conforming to the SLA. DiffServ relies on token buckets for shapers or policers.

The value of the codepoint subsequently defines how each packet is treated (forwarded) inside the network. Each DiffServ domain (DS domain) may define a number of services, called 'per-hop behaviors' (PHBs), based on the value of the DS field. The PHBs define the details on how a particular service is provided. This may be done by specifying the allocated resources or in terms of observable service indicators, such as delays or losses. For example, a simple PHB may guarantee a certain share of the total bandwidth capacity. Included in the PHB definition are also the actual values used for the DSCP. PHBs may be grouped together to form a PHB group, according to a prevalent behavior characteristic (e.g. all PHBs in the PHB group may have the same loss characteristics but varying prices or delay characteristics). The DiffServ standard proposes three PHB groups: class selector, assured forwarding and expedited forwarding.

The class selector (CS) PHB group consists of a total of 8 DSCPs. This PHB group is defined in [118] to provide backward compatibility with the TOS bits used in IPv4. The 8 PHBs allow a relative ordering between different packet flows. The forwarding behavior is described in relative terms. The lowest PHB defines a best-effort service. All other PHBs are intended to be defined to provide better than best-effort service. Furthermore, these should be defined by a network operator such that each PHB receives better service than the lower PHB (similar to introducing different levels of priority). In this way, a network operator can define different services in terms of delay and loss. This PHB group allows network operators to upgrade the boundary nodes to be DiffServ compatible, without having to change all the interior routers in the network. No traffic conditioning is specified.

The expedited forwarding (EF) PHB [78] defines a service comparable to the CBR service in ATM. For example, it may be used to emulate a leased line service. The EF PHB allows low loss, delay, and jitter by requiring the network to forward packets at the same rate as they arrive. This implies that the network has to provide sufficient resources to ensure that the maximum aggregate arrival rate of all EF flows in any node is less than the minimum departure rate. Buffering should be minimized inside the network to reduce delays. The provision of such a service requires tight traffic control at the edges. Packets are conditioned to conform to the corresponding SLA. This may be done by an

unbuffered token bucket. Packets which do not conform to the SLA inside the network are discarded immediately, since they are considered as an indication of a configuration error or a misbehaving source. Changing of the DSCP to a lower drop precedence value is not an option. In order to maintain this service, other PHBs may be preempted. Only one value of the DSCP is specified to indicate the EF PHB.

The assured forwarding (AF) PHB [74] consists of 12 different DSCPs, which are arranged into 4 classes with 3 drop precedence levels each. A network operator can thus define up to 12 different services. However, the specification provides little or no guidance on how such services are to be defined relative to each other. This implies that the services may be ill-defined or overlapping. Each of the four classes is allocated a minimum buffer space and bandwidth. Long term congestion is required to be minimized by packet dropping. Short term congestion may be handled through queuing. The amount of AF traffic entering the network is assumed to be controlled through a traffic conditioner. No node may reorder packets belonging to the same flow. An example of AF PHB services is given in terms of three delay classes (labeled 'gold', 'silver' and 'bronze'), each supporting three different precedence (e.g. loss) levels. Thus, a total of 9 PHBs can be defined. The drop precedence levels within a class could be determined by a token bucket. If the token buffer is full upon arrival of a packet, it is marked with the lowest drop precedence. If the token bucket is empty the packets are marked with the highest drop precedence. For intermediary token buffer occupancies, packets are marked with medium drop precedence.

Finally, Kilkki [98] defines a dynamic real-time / non real-time PHB consisting of two PHB classes with six PHBs each (i.e. using 12 DSCPs). This PHB group specifies the relationships between packets belonging to different PHBs. The network operator can thus provide two distinct delay levels, each providing further six levels of importance to distinguish flows in terms of loss or by price, for example. This PHB group emphasizes resource sharing in the network interior. The resource allocation is specified relative to other PHBs in the group. The edge nodes are simply required to measure the bit rates of the marked flows to determine such allocations.

An intermediary architecture implementing DiffServ is described in [120]. By using two bits in the IP packet headers, a premium service and best-effort service is defined. The

premium service emulates leased lines by specifying the desired peak rate per flow.

Chapter 5

Pricing Concepts in Broadband Networks

In this chapter we review and evaluate recent pricing concepts for broadband multiservice networks. Even in the late eighties, engineers paid little attention to pricing concepts in their research, focusing instead on technical issues. Walrand and Varaiya [158] claim that this was mainly due to the lack of economic background in the engineering community. However, by now a respectable number of researchers from both engineering and economics have investigated the role of pricing from two perspectives: recovering expansion charges and as a means for congestion control.

Network traffic has grown exponentially in recent years, and there seems to be no dispute that this trend will continue in the future. Spurred by almost insatiable user demand, engineers roll out applications with ever increasing resource requirements. The focus in recent years has shifted towards the provision of real-time applications requiring stricter quality of service (QoS) guarantees from the underlying network infrastructure. At the same time, the user base will increase in the future. These forecasts imply a scarcity of networking resources (buffer, bandwidth and processing). Financing future networks or expansions seems impossible without charging end users. Part, if not all, of the cost could be recovered by the introduction of usage-based charges. Pricing can thus be used as a means to resolve the allocation of these scarce resources to users.

Perhaps of more interest to engineers, pricing is now also considered as a new means to control congestion. Users are inherently price sensitive. Using prices, the network can send signals to the users, providing incentives which influence their behavior. Pricing thus becomes a means for traffic management and congestion control, as some of the schemes discussed below will show.

However, a number of economic, social and technical problems arise through the introduction of complex pricing schemes. On the economic side, if a network is already in place, the initial investment is typically considered a sunk cost, and therefore irrelevant. What does the demand function for transmission services look like? Is demand elastic or inelastic¹, and how does it change over time? Which pricing structures are preferred by users? How would users react to charges laid upon each packet?

A similar question arises from a social welfare point of view. High prices would prevent poor users from using the services, and thus make network services a luxury good, in extreme cases. On the other hand, providing free network services leads to the "tragedy of the commons"², where the network quickly becomes congested because of over-use. How should prices be set in light of such impacts on social welfare? Finally, several technical difficulties arise through the introduction of sophisticated pricing schemes. How can individual users be billed, and how are their charges accounted for? Associating a price with individual packets may lead to enormous accounting and billing overheads. Are the charges computed only at the user-network interface or is every network component charging a user? Who is paying for multicasting, and how is traffic charged for where the recipient is the beneficiary, as is the case for video-on-demand or similar information services. Can pricing be accommodated by existing network protocols, or do we have to develop and install an entirely new platform?

¹Demand is said to be inelastic if the quantity demanded does not change significantly as price changes, or if the demand curve has a negative slope less than -1. Similarly, demand is said to be elastic if users respond significantly to a small change in price. In this case, if the slope of the demand curve is between -1 and 0.

²Villagers share a common grazing ground (commons) for their cattle. With unlimited and free access to the commons, each villager will allow her cattle to graze as much as possible without considering the cost imposed on other villagers (who have less grass available for their cattle as a result). Therefore, the commons will be overgrazed.

A number of pricing schemes have recently been proposed, addressing such issues. The goal of this chapter is to review and evaluate these schemes using a variety of criteria, as introduced in section 5.1. In sections 5.2 to 5.10 we then discuss each of the schemes in detail. In particular, we review

- Flat pricing
- Priority pricing
- Paris-Metro pricing
- Smart-market pricing
- Edge pricing
- Expected capacity pricing
- Responsive pricing
- Effective bandwidth pricing
- Proportional Fairness pricing

In section 5.11 we summarize our evaluation and provide concluding remarks.

5.1 Criteria for evaluating pricing schemes

We evaluate the schemes reviewed in this chapter according to the following criteria:

- Compliance with existing technologies
- Measurement requirements for billing and accounting
- Support for congestion control or traffic management
- Provision of individual QoS guarantees
- Degree of network efficiency

- Degree of economic efficiency
- Impact on social fairness
- Pricing time frame.

We view *compliance with existing technologies* as an important indicator for the implementation complexity. Obviously, pricing schemes which are compatible with existing technologies are easier to adapt than those requiring significant changes in the underlying network technology. We view this criteria as stricter than an evaluation of the scheme's implementation complexity. A scheme may be easily implemented, but non-compliance with existing technologies such as ATM or IP precludes immediate deployment. The amount of *measurements required for billing and accounting* is a related criterion indicating the implementation complexity of the pricing schemes. Some schemes require no measurements at all, thus simplifying accounting and billing considerably. Other schemes require extensive measurements. Such schemes may not be easily scalable, in particular if measurements have to be taken for individual streams instead of aggregate streams. *Support for congestion control or traffic management* indicates the applicability of the pricing scheme for controlling network congestion by exploiting the price-sensitivity of users, thus inducing them to transmit less or during uncongested periods. We differentiate here between schemes which encourage the users to value their traffic such that a relative ordering can be established by the network, and schemes which are tightly related to connection admission control (CAC). Under the former category, users are still encouraged to transmit traffic at will, but a successful transmission may not be guaranteed (packets may be dropped during congestion periods). Under the latter category, the user's connection may be refused, thus forcing the user to delay his / her transmission. *Support for individual quality of service* is another category in our evaluation. We indicate the scheme's support for QoS guarantees to *individual* users, as opposed to raising the service level for a typical user. We do not differentiate here between strict or probabilistic guarantees. The category labeled *network efficiency* evaluates the expected utilization levels in networks. While high utilization levels are desirable for the network, they are not necessarily desirable for the user, as indicated by Odlyzco [122]. Low utilization levels imply availability of service. A highly utilized network on the other hand

may have to deny service to some customers. To take the user's point of view into account, we also evaluate the *economic efficiency* of the pricing scheme. This criterion indicates the scheme's focus on the overall utility level (or user benefit) of the user community³. The *social fairness* criterion indicates whether the pricing scheme prevents some users from accessing the network during congestion periods purely because of their inability to pay. Note that "fairness" is a vague concept and has many definitions. In proportional fairness pricing for example, a resource allocation is considered fair if it is in proportion to the charge. But this is just one interpretation. Fairness is in general hard to define, as pointed out by Clark [16]. Is it defined at the packet or the call level? How can a large file transfer be handled in a manner that is fair to all users? In light of these remarks we realize that our category only captures a very specific aspect of fairness. Another criterion in our comparison is the underlying pricing *time frame*. We view short time frames (minutes or less) as increasing the schemes flexibility to respond to congestion periods - network congestion occurs in short time frames. If a pricing scheme supports short time frames, the network operator can change prices in response to varying network conditions. However, as in the case of the criterion network efficiency, short term fluctuations may not be desirable from a user's point of view. Long pricing time frames (days or more) on the other hand do not enable the network to use pricing as a means for congestion control. We consider time frames in the order of hours as medium.

Our intent with this set of criteria is not to provide an absolute ranking of the pricing schemes reviewed here. Instead, the criteria are used to summarize the main features, allowing the reader to establish a final evaluation based on his / her preferences. Indeed, some of the criteria may be difficult to qualify, depending on the reader's point of view. For example, as indicated under the time frame or network utilization criteria, users and the network may have totally opposing views on the value of some criteria.

We conclude this section by contrasting costs and charges for networks from a traditional financial / economic perspective. MacKie-Mason and Varian [110] identify five types of *costs*

³The network is said to operate economically efficiently if the set of admitted users receiving a certain level of QoS value their services higher than users who have been denied service by the network for the same service level.

occurring in computer networks. First, there is a fixed cost of providing the networking infrastructure. A network should be provided if the sum of the user's willingness to pay for the network exceeds the cost of provisioning. Second, there is an incremental cost of connecting to the network, which is usually paid by the user in the form of a connection cost. This cost arises from bringing the network to the individual users. Third, there is the cost of expanding the network capacity. This cost should only be borne by the users who wish to use the network when it is congested. Users who are willing to defer their transmission during congestion times should not pay for the expansion of the network's capacity. Fourth, there is an incremental cost of sending an extra packet. This cost should be very low or zero without congestion, since bandwidth of a broadband network is typically a shared resource. The notion of an individual user's capacity on the backbone is hard to quantify in light of statistical multiplexing and huge capacities relative to user traffic. Finally, a social cost can be identified. The transmission of a packet invariably leads to a delay in other users' packets. This externality should be considered a social cost. A number of these costs are analyzed for the Internet in [112].

By charging users, the network operator attempts to cover part, if not all, of the above mentioned costs. Traditionally, these costs have been condensed into a single price, fixed over long periods of time. However, Walrand and Varayia [158] foresee the total charge to be split up into four categories: an access charge, a usage charge, a congestion charge and a quality of service charge. Fixed access charges are commonplace by now. The user is charged for the privilege of accessing a network. They are typically imposed to recover the fixed costs of running the network¹. Usage charges are imposed to recover the variable costs of running the network. The user pays for each packet or message transmitted over the network. Congestion charges reflect the additional cost of transmitting a message when the network is heavily loaded. Any transmission imposes negative externalities on other users, in the sense that a user's own traffic delays other users' traffic and thereby affects their flow, particularly during congestion periods. A congestion charge induces users to transmit during off-peak periods. Congestion charges should be imposed to finance future network

¹The initial investments into a network are considered as sunk costs and should thus be disregarded in a charging scheme.

expansions. Users who insist on transmitting traffic during congestion periods should pay a higher price for their transmission, which also represents the value they place on expanding the network's resources. Most of the schemes described in this chapter could be classified as congestion charges. Finally, quality of service (QoS) charges account for the fact that the user should pay to receive a certain QoS. A study of these pricing components is presented in [128].

Note that the pricing schemes presented next in this chapter may not necessarily fall under a single charging category. For example, any usage charge will also have an effect on network congestion for price sensitive users.

We now discuss each of the pricing schemes reviewed in this chapter in detail.

5.2 Flat Pricing

Under a flat pricing scheme [1, 52] the user is charged a fixed amount per time unit (e.g. month), irrespective of usage. With respect to our evaluation criteria, this pricing scheme has several advantages. Most importantly, it is simple and convenient. Flat pricing makes no assumptions about the underlying network technology and is already deployed. Since charges are unrelated to usage, no measurements are required for billing and accounting. This also leads to social fairness in the sense that no distinction is made between poor and rich users. Provided that the flat price can be paid, anybody can access the network while receiving the same service level.

However, this scheme does not allow the network to influence the users transmission behavior over short time frames (which are precisely the time frames when network congestion occurs). Users are deterred from being adaptive. They have no incentive to alter their transmission behavior to support the network operation. Flat pricing is therefore unsuitable for congestion control or traffic management. The underlying pricing time frame is assumed to be long-term. Over such time frames, the rapid changes in networking technologies such as fiber-optic transmission or wave-division multiplexing lead to increasing supply of resources (particularly bandwidth). Resources are seen to be abundant such that congestion is not an issue. Economies of scale can be realized by traffic aggregation and statistical multiplexing.

The price per unit of bandwidth decreases dramatically as the bandwidth capacity of the link increases (even though the multiplexing gain is reduced in the presence of long-range dependence). In fact, proponents of flat-pricing argue against the application of marginal cost pricing over long time frames. If network operators apply marginal cost pricing⁵, then prices may even be set below average cost. In this case, the total long-run cost of providing the network would exceed the total long-run revenue, resulting in a loss-making situation for the network provider. Note that marginal cost pricing is a particular form of usage-based pricing. The higher the demand for bandwidth, the lower the price the user is willing to pay for an additional unit of bandwidth. The demand function for bandwidth is thus downwards sloping.

We point out that the above argument relies heavily on the forecast of abundant bandwidth availability. While this assumption may be valid in the long-run, we believe that it will not prevent network congestion over all time frames. Since flat pricing fails to exchange the appropriate signals between the users and the network providers, it may even lead to congestion and insufficient investment into network expansions [46, 112]. Demand for bandwidth is expected to grow at exponential rates, and network providers may find it hard to provide sufficient resources over all time periods, in particular in light of long-range dependent traffic and newly discovered killer applications by users (e.g. Napster or MP3 files currently).

Flat pricing also does not explicitly support individual QoS guarantees to the user. According to Parris, providers are even discouraged from provisioning QoS, since they cannot recover the associated costs [127]. Proponents here again counter with the “abundant-resource” argument. Odlyzco, in particular, argues that QoS could be adequately provided by combining a flat pricing scheme with resource over-provisioning. This could be financed by re-directing investment from private networks to the Internet and from research in QoS technologies to resource capacities [52, 122]. Analyzing 1997 statistics on Internet expenditure, he finds that only 2.6% of total IT investment was spent on expanding the bandwidth capacity of the Internet backbone. In contrast, the cost of providing private networks was 6 times higher. Considering the increasing importance of the Internet backbone to both

⁵i.e. setting the marginal cost to marginal revenue or price.

private and corporate users a larger proportion should be spent on the backbone.

While a flat pricing scheme does not preclude the network from running at high utilization levels, the latter are still typically low. To provide decent networking services, links have to remain lightly utilized to cope with the bursty nature of data traffic. Low utilization rates in this sense do not indicate inefficiency in general. For example, even though household phone lines are only utilized on average 4% of the time, many households decide that it is better to pay for additional lines for fax / Internet services rather than miss important calls. Similarly, comparing modem pool utilization rates at Columbia University and the University of Toronto, Odlyzco finds that the former experienced peak rates of 100% whereas the latter had peak rates well below 100% (56% on average). This implies that users in Toronto received a *better* service in terms of network accessibility⁶.

Flat pricing does not improve the economic efficiency of the network. Individual users' utility levels are not taken into account in the pricing scheme. This is explained by Edell et al. in [46] using economic theory. Figure 5.1 shows a demand function for bandwidth. Since the demand is not a function of price under flat pricing, the user demands D_F units of bandwidth and so a negative surplus for parts of the bandwidth demand is incurred. Under a usage-based pricing scheme, the user would only demand D_U units of bandwidth. Triangle *A* represents the user surplus. For bandwidth units D_U the user is charged less than he / she is willing to pay. Triangle *B* represents the waste to the user, since the user is charged more for the units $D_F - D_U$ than indicated by the demand function. For high prices p , the waste may even exceed the surplus, in which case a user would choose not to demand any bandwidth at all. Flat pricing thus allows network operators to obtain higher revenues. Edell et al. extend their analysis to show that below-average users subsidize above-average users. Their Internet Demand Experiment (INDEX) trial provides supporting numerical evidence about the inefficiencies of flat pricing. For details, see <http://www.index.berkeley.edu/public/index.phtml>.

⁶With regards to the Internet, Odlyzco remarks that the poor service which is sometimes perceived by users can be entirely attributed to server problems (e.g. mail-, web- or domain-name servers).

Figure 5.1: A user's demand function for bandwidth

5.3 Paris-Metro Pricing

In a slight variation of flat pricing, Odlyzco [123] proposes a scheme based on the pricing of the Paris Metro labeled 'Paris-Metro Pricing' (PMP). As indicated in figure 5.2, a PMP-based network is a set of logical networks. The total bandwidth capacity is divided into several subnetworks. Each logical network operates on a best-effort basis and is priced differently. Users choose one of these logical networks for the transmission of their traffic and this implicitly defines the service level. Network operators set the prices for each logical subnetwork, for example through customer surveys or feedback forms. Users make a selection based on the expected network congestion and their budget. Assuming that prices are kept stable over significant periods of time and that users are price sensitive, higher priced networks will experience lower utilizations and hence be able to provide a higher service level.

Our evaluation of PMP is similar to flat pricing. The main advantage of the scheme is its simplicity. Since each logical network implements flat pricing, the scheme is compliant with existing technologies. However, the network now has to keep track of each user's choice of network for billing and accounting.

Pricing indeed becomes a control for traffic management under PMP. Instead of defining and providing different QoS categories, the network operator provides a set of technically identical networks which are distinguished by their access price. PMP traffic management

Figure 5.2: Conceptual representation of PMP

encourages users to separate themselves into different classes.

While the perceived service level in each subnetwork varies with price, PMP still does not support individual QoS guarantees. Each network still operates on a best-effort basis. The proposal suggests a small number of logical networks, sufficient to satisfy the varying QoS demands by users. Prices, as well as the capacity division, have to remain stable over long time periods to allow users to predict the service level. Because of the price differential, utilization levels of higher priced networks are again low. A relatively better service level is expected for higher priced networks (these are in effect over-provisioned). However, lower priced networks are now seen to operate at higher efficiency levels.

PMP networks improve economic efficiency. The utility level of users is bound to increase by the choice of the different subnetworks. However, each of these networks still provides a flat pricing scheme and so PMP still does not use pricing as a means to optimize the overall utility of the user community.

A technical disadvantage of PMP networks is their potential for instability (e.g. caused by fractal traffic). During periods of congestion, price-insensitive users may choose a higher-priced network in expectation of receiving better service. This may lead to congestion in the higher priced networks and thus cause instability. Odlyzco claims that such behavior can be suppressed by appropriately setting prices.

PMP-pricing is currently seen as an initial proposal, far from deployment. A number

of technical issues have to be resolved, for example, the inter-operability between different ISPs and their splitting of revenues.

5.4 Priority Pricing

Providing priority services and charging accordingly is advocated in [18, 19, 71]. Any priority-based network has to be accompanied by a pricing scheme to deter each user from using the highest priority class as a default. Users are forced to indicate the value of their traffic by selecting a priority level. During periods of congestion the network can then carry the traffic by the indicated level.

Priority pricing requires a priority field in every packet header. Such a field is already provided in IP, ensuring compatibility with such networks. Like PMP, measurements are now required for billing and accounting to keep track of the priority level of each transmitted packet for each user.

Priority pricing allows the users to send appropriate signals to the network to facilitate traffic management, even at short time frames. During periods of congestion, traffic is transmitted by priority level. Low priority traffic is delayed or even dropped. Individual QoS guarantees still cannot be given. However, like PMP users are able to attain a higher *relative* positioning (with respect to traffic from other users) by paying an increased amount. The impact of priority pricing on QoS (for a typical user) and congestion is studied by Gupta et al. [71, 70] for the Internet. They show that if prices are calculated such that congestion is *prevented*, then the service level improves. The network is assumed to provide a large number of services (such as database, games, news, movies, stock prices, software) at server locations. 50 servers are modeled to provide 100 services at random locations. Users request services at random from the network. The size of these requests ranges from 10 Kb to 15 Mb. The link capacities vary between 128 Kbps to 10 Mbps. The network implements priority queues for these service requests at the server. Each request is given a value and a linear decay rate should it not be satisfied immediately. This allows the users to select the priority, the time and the type of service. Assuming that prices are calculated such that the user's cost is minimized, then the overall service level improves. Note that Gupta et

al. assume a long time frame and that prices are calculate such that network and economic efficiency is optimized.

In contrast to the previous pricing schemes, priority pricing assumes that resources will be scarce or running at high levels of utilization. Transmission capacities of tens of megabits or more will provide sufficient bandwidth to enable bandwidth-intensive applications to be run by an increasing community of users. Future applications will have a wide variety of service requirements accommodated by multiple service classes. Bandwidth will be re-distributed from performance-insensitive applications to applications running under strict performance requirements.

Since prices are an indicator for the user's utility, priority pricing raises the economic efficiency of the network. Any dropped packets will have a lower priority level and thus a lower value to the user. The impact on the economic efficiency of priority pricing as compared to flat pricing is studied by Cocci et al. [19]. They find that a network can always set prices such that the overall user satisfaction is higher under priority pricing than under flat pricing. The study assumes that users are sensitive to the network's performance and to pricing signals. The user's benefit decreases as the network performance degrades or as the price for transmission increases. These two assumptions allow Cocci et al. to derive a utility function. Under flat pricing users are not faced with monetary incentives and thus will demand the highest priority. Under priority pricing users will choose one of four service classes: high priority loss and delay, low-priority loss and delay, high-priority loss and low-priority delay and low-priority loss and high-priority delay. The users are charged the prices $3p_{priority}$, $2p_{priority}$, $2p_{priority}$, and $p_{priority}$ respectively. Four types of traffic are modeled in their example: voice (delay sensitive, loss insensitive), TELNET and FTP (both delay sensitive, but at the packet level and the message level respectively) and EMAIL (loosely delay sensitive). For each of these applications, the utility function is derived using the formula $U = -V - TC$, where U represents the utility received, V is a function describing the degradation of the performance of the application as a function of the network performance, and TC is the total cost for the application, derived from the price per packet as indicated above. Both schemes are required to achieve a fixed revenue. The comparison is based on the overall user satisfaction as derived from the utility functions.

Under flat pricing, each user selects high priorities for both service and loss. Under priority pricing, the user selects one of the four classes to maximize the overall level of satisfaction. The network processes all incoming packets by priority, maintaining different queues for each class. Packets are queued on a FCFS basis and transmitted by priority level. Different topologies (ranging from two stations connected through a network of two switches to six stations connected by a backbone of seven switches) and traffic mixes are considered. In contrast to Gupta's study, the time frame here is assumed to be shorter. However, we believe the assumption of known utility functions over such time frames is unrealistic. Note also that prices have to be carefully set to achieve the desired results (as in PMP).

While such a study illustrates how priority pricing can improve the economic efficiency under certain assumption, it does not show the impact on social fairness. A user's ability to pay may in this case not guarantee transmission by the network. With a large number of priority levels and / or high prices for high priority levels, traffic from poor users (in a relative sense) may be starved.

5.5 Smart-Market Pricing

A smart-market pricing scheme [110] focuses on the issues of capacity expansion and the social cost imposed on other users. In addition to a fixed charge to cover for the connection costs and a (possibly small) charge per packet to cover the incremental cost of sending a packet, MacKie-Mason and Varian introduce a usage charge when the network is congested (see also [109]). This charge is determined through an auction. The user associates a price with each packet, carried in the packet's header, communicating the user's willingness to pay for transmission. Typically, such a price would be derived from default values, with the option to the user of overriding the default value for special transmissions. The network collects and sorts all the bids. It then determines a threshold value and transmits all the packets whose bid exceeds the threshold value. The threshold value is determined by the network's capacity and represents the marginal cost of congestion. Each transmitted packet is then charged this marginal congestion cost, not the value of the bid. Figure 5.3 illustrates this concept.

Figure 5.3: Conceptual representation of Smart-market pricing

The introduction of an auctioning mechanism leads to non-compliance with existing technologies. Smart-market pricing requires significant technical changes to protocols and networking hardware. This is a considerable cost considering the speed of standardization in the field. For each transmitted packet the user's billing records have to be updated. This also implies considerable overhead, even if the auctioning mechanism is only applied during congested periods. The scheme also does not provide the users with service guarantees or even a guarantee of transmission. Like priority pricing, it only ensures that the packets are transmitted according to their *relative* priority, determined by the bid. MacKie Mason and Varian also point out a potential problem of social distribution. Prices may deter poor users from using the service, thus creating a case for government regulation.

On the other hand, smart-market pricing encourages both network and economic efficiency. Resources are again assumed to be scarce, at least over short time intervals, due to the introduction of bandwidth intense applications. "As long as users have access to unlimited Internet usage, they will tend to 'overgraze', creating congestion that results in delays and dropped packets for other users" [110, page 8]. The underlying network utilization is assumed to be high. Using an auctioning mechanism, the network also encourages economic efficiency. All transmitted packets have a positive benefit to the user. The bid indicates the benefit to the user of having the packet transmitted, thus representing its social value. The equilibrium price is the bid of the marginal user, the value of the bid of

the first packet not granted transmission. Therefore the charge is always less than or equal to the bid. Bandwidth is allocated according to these bids, and so it is given to those users whose traffic is socially most valuable. Furthermore, the revenues collected from such a pricing scheme also equal the optimal investment to expand the networks capacities. The bids carried in each packet could also be used to facilitate routing decisions. Packets with high bids could be routed over shorter paths, whereas packets with low bids may be routed through longer paths.

5.6 Edge Pricing

Shenker et al. in [138] advocate a re-direction of the research agenda away from optimal pricing schemes towards pricing schemes concentrating on structural and architectural issues. Edge pricing provides a conceptual contribution to shift the focus on locally computed charges based on simple expected values of congestion and route. An example of such an expected congestion cost would be time-of-day charges. Basing the charge on the expected distance between source and destination allows the charge to be applied at the edge of the network, either at the source or the destination. Prices may be communicated from any place in the network. Such a pricing scheme is much simpler and facilitates receiver payments. The value of multicasting and receiver-charging is particularly stressed. Multicasting increases network efficiency and should therefore be encouraged⁷. Receiver-charging is important for future multimedia applications, such as video-on-demand. Any pricing scheme should be flexible enough to enable charges to be billed to either sender or receiver. This implies that prices need to be determined locally.

Even though edge pricing is predominantly a conceptual contribution, we view it as compatible with ATM (asynchronous transfer mode) or RSVP (resource reservation protocol). Both protocols rely on CAC functions at the edge of the network, to which the proposed edge pricing scheme could be coupled. Since edge pricing is supposed to be applied over

⁷Multicasts usually have a common path for all the receivers, and this allows the network to only transmit a single instance of the data packets up to the point inside the network where the routes to the different receiver split. Multicast packets are thus very valuable to all the recipients and greatly reduce the network load.

short to medium time frames, it also influences the users transmission decisions. In this way and by association with ATM / RSVP, edge pricing can support traffic management. Traffic measurements for billing and accounting may still be required, but these would be at most locally applied at the edge of the network. Individual users could even be given QoS guarantees. Edge pricing promotes network efficiency by encouraging users to multicast, and again by compatibility with the above mentioned networking technologies. The underlying pricing time frames are assumed to be medium, varying possibly several times a day.

Economic efficiency is particularly de-emphasized in edge pricing. Shenker criticizes that those usage-based pricing schemes, which optimize overall user benefit, ignore architectural and technical issues. His criticism of such schemes centers around the assumption that prices are fixed over long time frames. The determination of the optimal price hinges on a well-defined demand / utility functions which are assumed to be stable over time. However, demand functions are also typically based on delay or loss. A user's utility varies at short notice and cannot be expressed through a long-term demand function. Shenker et al. point out that "The relationship between what happens to a packet traversing a network and the resulting change in a user's utility is extremely complicated... It is extremely difficult, if not impossible, for the network to infer the effect on the transfer time arising from delaying any of the individual packets, especially since the transfer time is also a function of the user's congestion control algorithm. To make matters even worse, often an entire suite of applications is used simultaneously, and then the user's utility depends on the relationship between the delays of the various traffic streams" [138, page 6]. A user's utility thus varies considerably with the state of the network, over short to medium time frames and determining prices according to long-term demand functions thus does not achieve economic efficiency. Note that such short-term demand functions are assumed to be known and fixed over time in the studies mentioned in section 5.4. Shenker's criticism implies that utility maximizing prices in priority pricing have to be determined over short time periods. Without a precise knowledge of user's utility functions over such intervals, this task becomes impossible. This is particularly relevant since both studies supporting priority pricing emphasize that prices have to be carefully set to induce economic efficiency.

5.7 Expected Capacity Pricing

Clark [16] also focuses on users' expectations and introduces the concept of expected capacity pricing. His scheme centers around the user specifying the required *expected* capacity. The user is then charged according to the expected capacity that the network provisions, not on actual usage, based on a long-term contract with the network. This indicates the capacity he or she expects to use when the network is congested. Such expectations can be specified in a number of ways, not just specifying the minimum capacity required. For example, the user could provide the maximum transfer time of data objects or use an effective bandwidth based characterization. The network is assumed to be able to provision for a users expected capacity using an implied CAC scheme, and to police the traffic.

Expected capacity pricing again seems to be compatible with ATM or RSVP. This implies support for congestion control or traffic management by encouraging users to determine the service level and charging accordingly. Individual QoS guarantees can therefore be given. One of the main advantages of expected capacity pricing is that charging is not related to actual traffic volume but expected traffic volume. Measurements are not required, saving significant overhead for the network. Combined with the concept of edge pricing, a user's bill can easily be computed at the edge. However, even though measurements are not required for billing, traffic nevertheless has to be policed to ensure conformance with the traffic contract. This may jeopardize the scalability of the pricing scheme. Policing individual flows at the UNI introduces additional computational requirements which have to be provisioned⁸.

If excess traffic is charged for under this scheme, the advantage of not relating billing to measurements is partially lost again. We still consider policing more efficient than billing on measurements, since the former only has to make an accept / reject decision. Note that expected capacity pricing also encourages receiver payments. In this case the receiver specifies an expected capacity contract with the network [15]. The traffic stream is then

⁸Recall that we view the user as the complement to the network. A service provider would thus police individual users. However, it would then aggregate traffic and establish a contract with the network core. From the core's point of view, this aggregate traffic is then policed by a single policer. In this way, traffic contracts and policing is hierarchical.

policed at the receiver, according to the receiver's contract. If a packet violates the policer, the receiver is informed about the violation and asked to act accordingly⁹. The scheme also handles inter-provider traffic. Service providers are assumed to enter into contractual agreements among themselves, thus establishing a user-network relationship which could be handled as outlined above. Each provider specifies the expected capacity required from the other provider and is metered accordingly. Alternatively, providers could agree to carry all traffic irrespective of the flags, and simply charge each other on a per packet basis.

The time frame for expected capacity pricing is assumed to be long term determined by the time frame of the underlying contract between user and provider. A dynamic version of this pricing scheme is proposed in [84], where the assumption of the long-term contract between the user and the network is relaxed by employing bandwidth brokers.

Like Shenker, Clark also points to the problem of defining utility functions for users. He identifies the difficulty of assigning a monetary value to a data packet. Network operators have an insufficient understanding of the value that users place on different services. Users typically react differently to service degradations. For example, some users may tolerate a slight increase in delay, whereas others may be very sensitive to even slight increases in delay. Our evaluation of this scheme in terms of economic and network efficiency therefore does not differ from edge pricing.

In our view, this scheme is socially fair in the sense that even relatively poor users can negotiate contracts with the network. The responsibility for resource allocation is pushed to the network. Assuming that the network fulfills this function, all users can gain access to the network irrespective of their ability to pay. Rich users may still be able to purchase "better" service.

Note that expected capacity pricing should be distinguished from resource-based pricing, where the user is charged according to the measured resource-usage. Such schemes are typically computationally more demanding. A resource-based pricing policy is discussed in [127, 24]. Edell et al. [45] devise a billing system for a resource-based pricing scheme, thus

⁹Alternatively, the sender could flag packets according to the receiver's contract. However, in this case the contract has to be communicated through the network, potentially traversing many ISP-networks. Furthermore, the handling of multicasts would be difficult. Such a scheme is thus less desirable.

demonstrating its feasibility from a computational perspective.

Kesidis [94] points out a potential opportunity for users when charged according to the declared or expected capacity. He considers the case where the user wishes to transmit both a VBR stream with mean rate m_{VBR} and a best-effort stream with mean rate m_{BE} . Each connection is assumed to be simultaneously (σ, ρ) - and $(0, \pi)$ -constrained. If the network charges a connection according to $UC(\rho, m) = p_1(\rho) + p_2 m$ (where m is the measured mean rate of the connection, p_1 is the price based on the declared value of ρ and p_2 is the price for the measured mean rate) then the user may have an incentive to declare only a single connection for both streams and interleave best-effort cells into the VBR stream. In particular, if the network allocates bandwidth according to the worst-case results introduced in section 2.5, then delay guarantees can be given to the connection. However, since the declared $(\sigma, \rho) \wedge (0, \pi)$ -constraints may not be tight, best effort cells could be inserted into the VBR stream without violating these declared values. This is referred to as in-band transmission of best-effort cells. If the condition

$$p_2 \leq \frac{UC(\rho_{VBR}, m_{VBR} + m_{BE}) - UC(\rho_{VBR}, m_{VBR})}{m_{BE}} \quad (5.1)$$

holds, then the user has an incentive for in-band transmission. If not, it is cheaper for the user to declare a separate connection for the best-effort stream. Of course, the incentive for in-band transmission increases with the introduction of connection charges. Kesidis also provides a schedule for such in-band transmission. Best-effort cells could be inserted during the idle periods of a virtual queue with service rate ρ , buffer size σ , fed with the VBR stream. This study highlight another issue which has to be considered when introducing more sophisticated pricing schemes.

5.8 Responsive Pricing

The concept of responsive pricing [117, 108, 115, 116, 114, 113] describes a dynamic price setting strategy by the network. It illustrates how the network can exploit the adaptive nature of users to increase economic and network efficiency. Price is emphasized as an alternative (and even improved as compared to traditional feedback mechanisms) means

for congestion control to ensure proper network operation, and in particular to guarantee different service levels. Similar to smart-market pricing, the charging mechanism only comes into operation during periods of congestion.

Responsive pricing is based on the realization that users are adaptive and respond to price signals. In case of high network utilization, resources are stressed and the network increases the prices for the resources. Adaptive users then, by definition, reduce the traffic offered to the network. Similarly, in case of low network utilization, the network decreases the price and the community of adaptive users increases their offered traffic. In this way, adaptive users do not just increase the network efficiency, but also economic efficiency.

Adaptive users fall into two classes: elastic and inelastic users. Elastic users cannot tolerate losses but are able to delay transmission (a similar model is studied in [75]). Inelastic users require strict delay guarantees, but can tolerate some degree of losses. Both classes of adaptive users value the service level they receive from the network using a benefit function. Elastic users have a benefit function of the form $ben_t(x, v) = v - h(d(x))$, where v is the benefit to the user if all the traffic were sent immediately, $d(x)$ represents the delay incurred for x number of packets. The function $h(\cdot)$ is concave, measuring the decrease in benefit as delay increases. Elastic users evaluate the benefit of transmitting x packets now versus later, given the price from the network. Inelastic users also have a benefit function, $ben_t(x)$, depending only on the number of packets transmitted in period t . This function is concave with respect to x , indicating that the benefit of sending x packets now increases more slowly as the number of packets increases. The benefit function is assumed to be fixed over a time interval (assumed to be short, of order 0.05 of the transmission time of a frame [117]), but it is allowed to vary between time intervals. Inelastic users determine the number of packets to transmit in interval t by a marginal cost argument. They equate the marginal benefit of sending another packet to the marginal cost for such a packet, i.e. $\frac{\partial ben_t(x)}{\partial x} = p_t$ and solve for x for a given price p_t from the network. Both types of users thus respond to the price signals given from the network by altering the amount of traffic they transmit.

The network sets the prices using either a closed-loop or a smart-market approach. In the former scheme, the network measures its resource utilization, for example the buffer occupancy at the user-network interface and then determines the price per packet. This

ensures a desired level of network utilization (under steady-state assumptions).

Simulation results of Murphy et al. show that network utilization can be smoothed out using such a scheme. Adaptive users are induced to transmit traffic in the off-peak hours, and deterred from transmissions during the peak hours. Furthermore, the economic efficiency increases in their scheme, since users have control over their transmission decisions. For the scenario reported in [117], the perceived benefits to the user from responsive pricing increase by 15% and the percentage of packets lost drops by 91% as compared to unpriced network operations. In [108] issues regarding feasibility, implementation and cultural effects of responsive pricing are addressed.

Responsive pricing was designed for ATM ABR services. We thus classify it as compatible with existing technologies and as being socially fair. Note however that ABR guarantees only a minimum cell rate. Furthermore, the underlying time frame is assumed to be very short. We perceive this as being a problem in the above description, since users' utility functions as well as prices are assumed to be able to change at such short notices. Such a system may also introduce instability, as experienced by traditional congestion control functions. Upon declining network utilization, the network would reduce prices. This encourages adaptive users to transmit, thereby causing congestion. Furthermore, under these assumptions the network operator has to keep detailed billing and accounting records, therefore incurring significant overhead.

5.9 Effective Bandwidth Pricing

Kelly's effective bandwidth pricing scheme [86, 89, 87] is designed to induce the user to declare the true values for the mean and the peak cell rates of general traffic sources during CAC. The functional form of the effective bandwidth of the user's traffic source is assumed to be known. The user is charged according to a linear function placed tangent to the effective bandwidth curve of the source. For example, for an on-off traffic source with unknown mean and peak rates, the user is charged according to a linear function $f(z, m, H) = a(z) + b(z)Z(m, H)$, where m and H represent the *measured* mean and peak rate of the source respectively, z is a value declared by the user and $Z(\dots)$ represents the moment

generating function of the traffic source as a function of the mean rate and the peak rate. In this case $Z(\dots)$ evaluates to

$$Z(\mu, \pi) = 1 + \frac{\mu}{\pi}(e^{\theta\pi} - 1)$$

and $a(z), b(z)$ are chosen such that $f(z, m, H)$ is tangent to the effective bandwidth curve $\alpha(\mu, \pi) = \frac{1}{\theta} \log(Z(\mu, \pi))$. The user's minimal cost declaration of z is $z = E_G Z(\mu, \pi)$, the expected value of the moment generating function with respect to the joint distribution of the mean and peak rates μ and π respectively.

If the user agrees to peak-rate policing and the value of π is known for CAC, effective bandwidth pricing simplifies to placing a tangent $f(\mu, m) = a(\mu) + b(\mu)m$ to the effective bandwidth curve

$$\alpha(m) = \frac{1}{\theta} \log \left[1 + \frac{m}{\pi}(e^{\theta\pi} - 1) \right]$$

at the point $m = \mu$, as shown in figure 5.4. Here, $\alpha(m)$ represents the effective bandwidth for an on-off source with mean rate m and peak rate π . θ is a QoS parameter related to the blocking probability. As can be seen from figure 5.4, the concavity of the function $\alpha(m)$ with respect to m then ensures that $f(\mu, m) \geq \alpha(m)$, with equality holding only at the point $m = \mu$. The user is charged a premium for not declaring the true mean rate of the source.

Figure 5.4: Effective Bandwidth Pricing

Kelly's pricing scheme ensures that the network can *deduce* the anticipated load generated by a user. The scheme allows the network to infer the actual (parameterized) function from the user's declaration. The user is assumed to act rationally in the sense of wanting to minimize the economic cost of the connection. The network then knows that the declared values truly reveal the anticipated load, and hence allows effective-bandwidth based CAC to work properly. Note that under effective bandwidth based CAC, QoS guarantees can be given to individual users and statistical multiplexing effects can be exploited to obtain high levels of utilization. The underlying time frame in this case is assumed to be in the order of magnitude of a call duration, i.e. minutes. As in the case of edge, expected capacity and responsive pricing, CAC ensures that access to the network is based on available resources, not on the ability to pay. We therefore consider effective bandwidth pricing socially fair.

Note that the user is only required to provide the expected value of the traffic stream. Any additional information about the joint distribution of the mean rate and the peak rate is not used in the scheme, and it would thus be needless for the user to describe the distribution's higher moments, or for the network to require such higher moments. In the case of on-off sources with known peak rates, the user only has to declare the expected mean rate, a quantity that can easily be measured or estimated. Furthermore, Kelly shows that if the user is indeed able to describe the effective bandwidth function more accurately, then the work spent on this description is proportional to the cost savings obtained. A disadvantage of this scheme is that the functional form of the effective bandwidth is assumed to be known in advance.

This pricing scheme can also be extended to allow for time-varying prices and to deter the user from splitting the source traffic. A user could be discouraged from transmitting a single stream over several connections by imposing a connection charge. Allowing users to re-negotiate the tariff periodically is important in cases where user traffic is highly variable over time. A periodic re-negotiation of the tariff would reveal to the network the anticipated load over a short-term interval, and thus improve the network's resource use. However, such re-negotiations would only make sense if the short-term interval exceeds the round-trip delay. To account for the additional complexity arising from the re-negotiation, Kelly proposed to bill the user for a set-up charge c for every re-negotiation. This would encourage the

user to re-negotiate the tariff only if the expected savings exceed the set-up cost c . The idea of basing a usage-charge on both measured and declared values is further explored in [22]. Courcoubetis et al. describe how to compute usage charges from simple measurements which may be used to bound the effective bandwidth.

Effective bandwidth pricing is further studied in [146, 23]. Courcoubetis and Siris evaluate usage based pricing by time / volume charges. They approximate the effective bandwidth by two alternative bounds. The first one is based on the effective bandwidth of a leaky-bucket. The second one is based on an inverted T pattern for on-off traffic, which is assumed to be the worst-case output of a leaky bucket. A comparison is made between measuring the time and volume over the entire call duration and measuring these quantities at time intervals less than the call duration. They conclude that the latter is more robust and accurate (as compared to the true effective bandwidth), although it involves a higher overhead. In [146] an interesting comparison is presented between effective bandwidth pricing and the pricing scheme of British Telecommunications' CellStream service. Details on these studies can be found under <http://www.ics.forth.gr/netgroup/charging> and under <http://www.statslab.cam.ac.uk/~frank/>.

Jiang and Jordan [81, 82] use a pure effective bandwidth pricing scheme for connection admission control, where the user is charged per unit of effective bandwidth used. In this unified CAC framework, user and network agents interact in two stages. In the first stage, the user agents characterize the information streams and define the desired performance parameters in terms of effective bandwidths. At the same time, network agents determine the available amount of network resources which are assumed to be allocated to a set of VPs. Network agents determine the acceptance region for new calls. In the second stage, user and network agents come together in a market environment to determine which traffic streams are admitted at what prices. Effective bandwidths provide the common unit for negotiation. The outcome of the negotiation is a price per unit of effective bandwidth, where demand equals supply. Note that such an optimization only makes sense if demand and supply are stable over the optimization time frame. Jiang therefore proposes a de-centralized scheme (see [81] for details) to obtain the solution. This significantly reduces the computational burden that would otherwise be placed on the network if market clearing prices had to be

calculated globally. The proportional fairness pricing scheme in section 5.10 shows how a similar global optimization can be decomposed into sub-problems such that the resulting solution of the sub-problems also constitute a solution to the initial problem. Note that the time frame here is assumed to be in the order of a connection duration. Shenker's criticism of varying short term utility functions thus does not apply here, since it assumes time frames which are in the order of packet delays. Jiang's scheme does not allow for demand functions varying over such time intervals. It assumes that the user's demand is known at the time of the connection establishment and fixed thereafter.

5.10 Proportional Fairness Pricing

The concept of Proportional Fairness Pricing (PFP) [59, 90, 92, 91, 147] was motivated by the desire to incorporate the notion of fairness into the allocation of network resources. In this scheme, a resource allocation is *fair* if it is in proportion to the users willingness to pay¹⁰.

Proportional fairness is derived from a number of perspectives. Consider a global perspective of the entire network. The network can be modeled as a collection of resources J each with capacity $C_j, j \in J$. Assume that each user in the set of users R has a unique route between source and destination¹¹ and denote by $U_r(x_r)$ be the utility of this route to the user at rate x_r . The utility function is assumed to be increasing, strictly concave and continuously differentiable with respect to x_r . Let $\{A\}$ be an indicator matrix where $A_{j,r} = 1$ if resource j is required to carry user r 's flow. Denote by $C = (C_j, j \in J)$ and $x = (x_r, r \in R)$. The systems's optimal rates are then determined by the following optimization problem:

$$\begin{aligned} \text{Max } & \sum_r U_r(x_r) && (OBJ_1) \\ \text{s.t. } & Ax \leq C && (C_1) \\ & x \geq 0. \end{aligned}$$

¹⁰Fairness in this section is not to be confused with our evaluation criterion of social fairness. Recall that we consider a pricing scheme socially fair if access to the network during congestion periods is not determined by a user's wealth.

¹¹Users and routes thus have a one-to-one correspondence.

Since the objective function is strictly concave, the maximizing vector of rates x exists and can be found using the Lagrangian method. Kelly interprets the Lagrange multipliers v_j of this problem as the implied cost of a unit flow through resource j , or as the shadow price of additional capacity at resource j .

This system optimization can be decomposed into a user and a network optimization. Considering the former, the user is assumed to be charged according to the shadow price of the resources on the route, i.e. the price $p_r = \sum_{j \in J} v_j A_{j,r}$. A user would then demand rates x_r which maximize his / her utility, i.e.

$$\begin{aligned} \text{Max } & U_r(x_r) - p_r x_r & (OBJ_2) \\ \text{s.t. } & x_r \geq 0 \end{aligned}$$

The network on the other hand attempts to maximize revenue from the flows. Assuming that the network is allowed to vary the flows x_r , then revenue is maximized by the following optimization problem:

$$\begin{aligned} \text{Max } & \sum_r p_r x_r & (OBJ_3) \\ \text{s.t. } & Ax \leq C & (C_3) \\ & x \geq 0. \end{aligned}$$

Thus, the Lagrange multipliers v_j provide the link between the user's and the network's optimization problems. Kelly shows that there exists a price vector $p = (p_r, r \in R)$ which solves the network's optimization problem, given the solutions for the rates x_r of the individual users. The vector of rates for all users then also solves the overall system problem. In other words, a price vector and a rate vector can be found to optimize all three problems above. If the users fix the rates, the network is able to set a price p_r to charge them proportionally to their rate.

The system optimization problem can alternatively be decomposed such that a user *chooses* his / her price and the network allocates the rates according to the (extended) proportional fairness criterion. The resulting allocation of the rates, given the willingness to pay for each user, again solves the system's problem.

Note that such an allocation of resources guarantees economic efficiency, since users' utilities are maximized. According to our definition of social fairness, this scheme is also fair. Every user who is willing to pay is allocated some bandwidth. This guarantees access even for "poor" users. Similarly, network efficiency is high since the network is always willing / able to allocate all its resources. Congestion in this scheme is avoided by allocating resources to the users. During periods of high demand, each user would get a proportionally smaller amount of bandwidth, and thus users would have to throttle their transmission at the edge of the network. Note that billing and measurements are not required, since each user is charged according to the indicated willingness to pay. The network only has to keep track of the willingness to pay and allocate resources accordingly. The underlying time frame is related to the stability of the user's willingness to pay. We believe that users prices are typically stable over short (possibly even medium) time frames. For further details on this scheme, see [59, 90, 92].

Courcoubetis et al. [27] extend this by replacing the users in the above formulation with intelligent agents. Such an agent would decide the willingness to pay in lieu of the user such that the user's utility is still maximized. It could learn the actual user's preferences and act in time frames in the order of magnitude of less than minutes. However, this would impose additional computational overhead on the network to change the optimal rate allocation.

5.11 Concluding Discussion

In this chapter we reviewed and evaluated a number of pricing schemes for broadband multiservice networks, summarized in table 5.11. Most of the schemes attempt to use pricing as a means to control congestion by exploiting the price sensitivity of users. In some cases, this is done by ordering user traffic relative to some form of priority (PMP, priority and smart-market pricing). Other schemes implicitly support congestion control and traffic management by associating pricing with CAC (edge, expected capacity or effective bandwidth pricing). Note that most schemes are designed for pricing best-effort services, and therefore do not support individual QoS guarantees. This does not mean that the service level is not improved in general. Priority schemes allow users to purchase a relative advan-

Table 5.1: Summary of pricing scheme features.

	Flat	PMP	Priority	Smart Market	Edge / Exp.Cap.	Resp.	Eff. Bandw.	PFP
Compliance	IP	IP. VN	IP		ATM. RSVP	ATM. VN	ATM	ATM. IP
Billing Meas.	No	No	Yes	Yes	Yes (local)	Yes	Yes (local)	No
Cong. Control Traf. Mgmt	No	Yes (rel.)	Yes (rel.)	Yes (rel.)	Yes (CAC)	Yes	Yes (CAC)	Yes
Indiv. QoS	No	No	No	No	Yes	Part	Yes	No
Network Eff.	Low	Var.	High	High	High	High	High	High
Economic Eff.	Low	Var.	High	High	Var.	High	High	High
Social Fairness	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Time frame	Long	Long	Short	Short	Medium / Long	Short	Short	Short

VN: virtual networks; RSVP: Resource Reservation Protocol; Var.: variable; rel.: relative

tage over other users, and so users paying for high priority enjoy a better service. However, in these cases no guarantee for transmission is made. In case of badly designed prices, all users might be able to purchase high priority service. This underlines the importance of well designed prices to make priority schemes effective. On the other hand, QoS guarantees for individual users can typically be given in pricing schemes which are related to CAC.

Major differences between the reviewed scheme can be related to differing views about the underlying time frames and resource availability. Flat pricing, PMP or even expected capacity pricing, for example, take long-term views. Flat and expected capacity pricing are based on long-term contracts between the user and the network. Under PMP the users have to learn about the service level provided by the network. Flat pricing and PMP argue that due to technological advances congestion can always be avoided by over-provisioning. The same argument is seen to guarantee high service levels. However, these pricing schemes do

not address the issue of short term congestion. If the underlying time frame turns out to be long-term, then the pricing scheme loses its ability to exploit users' price sensitivity to influence short term congestion.

Other pricing schemes take a different point of view, assuming that resources will be scarce, at least over short time scales where congestion typically occurs. This justifies the need for using prices as an alternative to congestion control. Price changes are used to induce users to alter their transmission behavior. However, schemes which attempt to *set* prices over short intervals such that economic efficiency is maximized may then be unrealistic. For example, in responsive pricing short term demand or utility functions are assumed to be available, which in our view they are not. The optimizations proposed by Jiang [81] or Kelly [90] overcome this problem by assuming that the utility / demand for a connection is known only at the time when the connection is established. However, they implicitly assume that the utility functions then remain stable over the connection duration. No allowances are made for demand functions which vary at time intervals shorter than the connection duration. If utility functions indeed vary at shorter time intervals, as suggested by Shenker, then the allocation may no longer be economically efficient. We note that this problem may be solved by agents configured to act in short time frames such that a user's long term utility increases. Allowing for re-negotiation, such agents also have to be carefully parameterized such that the computational burden introduced by a re-negotiation does not overwhelm the system. True economic efficiency for the entire network allowing for highly variable utility functions over short intervals seems to be an unrealistic goal. Nevertheless, pricing schemes designed to handle short-term congestion at least approach economic efficiency and also allow for higher network efficiency.

Viewed from another angle, most schemes assume that prices are fixed, either over long periods of time or by the user. The former is the underlying premise of PMP, flat, expected capacity, edge and effective bandwidth pricing. In the case of flat pricing and PMP this is required to allow users to form reasonable estimates of the network's state. Prices are fixed by the user in the case in PFP or smart-market pricing. Again, responsive pricing suffers from the assumption that prices can vary over short time intervals. Note that priority pricing, in particular, requires that prices are calculated such that network / economic

efficiency increases, even if long-term assumptions are made. This is one of the underlying assumptions made in the studies by Gupta [71] and Cocci [19]. These studies imply that if prices are not set in such a way, priority pricing may provide no gain over flat pricing, for example.

The time frame should also be considered when evaluating the pricing schemes requirements for billing and traffic measurements. Keeping track of each user's packet transmissions, possibly at each node, for billing adds considerable overhead to the network (even during hopefully limited periods of congestion as proposed in smart-market pricing). PMP, priority, smart-market and responsive pricing incur such an overhead. Itemized billing in these cases is also problematic, since users may be flooded with information about charges incurred by different packets at each (short) interval. Users may be unable in this case to verify their bills. In contrast, keeping track of a single price or recording charges locally (edge, effective bandwidth pricing) over short intervals requires less overhead.

Our criterion of social fairness is satisfied by all pricing schemes except priority and smart-market pricing. In these cases, rich users may gain a relative advantage over poor users by paying more. In their present form poor users may not gain access to the network in the extreme case. This drawback could be remedied by carefully designing the prices or by enhancing the pricing scheme with CAC. Note that we have classified those schemes which use CAC as socially fair. CAC regulates the access to the network not based on a user's willingness to pay, but by resource availability and FCFS level. Rich users cannot therefore bump poor users off the network.

Since pricing schemes attempt to exploit users' price sensitivity, a clear understanding on exactly how sensitive users are is required. Edell's Internet demand experiment [46] provides valuable data here. Users are given a variety of choices each month on a variety of service plans. These service plans provide different pricing service alternatives. Studies along those lines are valuable and still needed in the future to facilitate some of the above pricing schemes. For example, the assumptions in responsive pricing about users being adaptive could be quantified. Which type of users are adaptive and how adaptive are they? Which services do users value? Information is not just required to determine the precise form of long-run utility functions, but also in the short-run.

We conclude this chapter by pointing out that many additional aspects may be relevant when introducing pricing. Shenker's and Clark's focus on receiver payments or multicasting are typical examples. Another issue may be the disappearing distinction between users and the network. Future multi-service networks based on virtual networks may blur the distinction between carrier and user, between private and public networks. A carrier's users may themselves be carriers. Users will increasingly control the network, with access to the control functions through software. The resulting difficulty in defining specific services, tariffs and regulations will be hard to enforce by telecommunication regulators. Furthermore, pricing schemes charging users by usage almost invariably fail to reflect the inherent value of the information, i.e. the content of data packets. The particular content of a data packet may be extremely important to one user, but less to another user. The value of a data packet to the same user may also vary with time¹². Such issues are still open for future research.

¹²cf. real-time video: if the delay constraints on video traffic are not met, the data packets are meaningless to the user and may just as well be discarded.

Chapter 6

Minimum Cost Traffic Shaping

In this chapter we introduce the notion of *minimum cost traffic shaping* (MCTS). MCTS offers a procedure to establish the traffic contract between the user and the network dynamically. We arrive at this framework by taking a user's perspective on connection admission control. We examine how (σ, ρ) -constrained and effective bandwidth based CAC can be viewed by a user who wishes to gain access to the network, with the additional requirement that his / her stream requires a given level of QoS.

In the following section we provide the motivation for such a framework. In particular, we argue that both (σ, ρ) -constrained and effective bandwidth based CAC could benefit from tighter integration with the user requirements. We identify the traffic shaper as the main interface between the user and the network. We point out that the user may have many ways to parameterize the shaper due to the trade-off between buffer and bandwidth. In section 6.2 we then present the minimum cost traffic shaping framework. We identify an optimization problem faced by the user, the solution to which will provide the minimum cost parameters for the traffic shaper. We discuss in detail the QoS constraints of this minimization problem and optional CAC constraints. In section 6.3 we then apply this framework assuming (σ, ρ) -constrained CAC. We demonstrate minimum cost traffic shaping in two sets of examples, assuming Poissonian / Gaussian / on-off arrival streams and self-similar arrival streams respectively. In section 6.4 we outline the case where the network's CAC function is based on effective bandwidths. We motivate the cost function in this case

and again provide a concrete example using Gaussian sources. In the last section of this chapter we propose a pricing scheme which is tightly associated with the MCTS framework. Under this pricing scheme, the user is charged according the resources required to satisfy the desired QoS.

We use the terms ‘shaping’ and ‘regulating’ synonymously in this chapter.

6.1 Motivation

The principal motivation for the results presented in this chapter is to formulate a procedure which allows traffic contracts to be established dynamically. Currently, traffic contracts are assumed to be negotiated by the network operator and a human user. The contract is typically long term and the parameter settings are the result of the negotiation. In light of newly emerging applications with varying QoS requirements, we believe that a dynamic scheme to specify the traffic contract is desirable. MCTS offers such a scheme. The user determines the resources required using the MCTS optimization problem, given prices for resources by the network. The network then decides whether these resources can be provisioned. If this is the case, then the user’s stream is admitted. The resource requirement is subsequently used to parameterize the traffic shaper. All values used in MCTS are fixed in the traffic contract. If insufficient resources are available, the user’s stream is rejected and transmission has to be re-attempted at a later time. We arrive at the MCTS framework by examining two popular connection admission control schemes ((σ, ρ) -constraints and effective bandwidths) *from a user’s perspective*. As reviewed in chapters 2 and 3 both schemes are able to provide its users with QoS guarantees. In both cases, this is done by restricting access to the network in case of congestion so that the quality of service enjoyed by existing calls is not jeopardized.

In the case of (σ, ρ) -constrained CAC, such guarantees can be given under the assumption that all arrival streams are already (σ, ρ) -constrained. Guarantees can even be given to individual streams since resources are reserved on a per-stream basis. While it is well known that such (σ, ρ) -constraints can be achieved through traffic shapers or policers, an integrated view involving the user has only been taken by Low et al. [104]. In Low’s

scheme, the user is assumed to know the set of all burstiness curves which satisfy a QoS requirement and a decentralized algorithm is presented which maximizes total welfare of the network. Our framework is similar in that we also formulate a minimization problem. However, we specify the QoS constraints in terms of the effective bandwidth approximation, instead of burstiness curves (even if the CAC function is based on (σ, ρ) -constraints). We are motivated here by the many results on this approximation. The advantages of this characterization for the QoS constraints are highlighted in section 6.2.1 below.

In the case of effective bandwidth based CAC, guarantees can be given for aggregate arrival streams if their effective bandwidths are known. Individual guarantees can be given only if statistically similar sources are multiplexed. Traffic shaping or policing is not assumed in this case, even though industry standards such as ATM or the upcoming DiffServ standard recommend it. To the best of our knowledge, the integration of traffic shaping and effective bandwidth based CAC has never been examined from a user's perspective. In [35] both (σ, ρ) -constraints and effective bandwidth based CAC are integrated in a resource provisioning framework. Traffic shaping is only assumed for (σ, ρ) -constrained streams for which resources are reserved. However, the implications of such a provisioning framework on the user are not discussed in detail.

In addition, we propose a pricing scheme where the user is charged according to the resource requirements which satisfy the desired QoS. If network operators are treated increasingly as providers of a particular *service*, they will charge for the *service* and this in turn requires them to know what costs are incurred by providing the service. In a competitive environment providers are driven to apply marginal cost pricing (or else risk going out of business). The resources required¹ to provide a particular QoS level must then be precisely specified if users are to be charged by the QoS level. Examining the buffer / bandwidth tradeoff is therefore important. Since the result of our MCTS problem also determines the parameters for the traffic shaper, we take an integrated view of effective bandwidths, traffic shaping and pricing from a user's perspective.

Note that we use the term *user* in a generic sense. We view the user as being the complement of the network. This could be the actual human user (which is what we

¹We restrict ourselves to buffer and bandwidth resources in this thesis.

assume in our pricing scheme). It could be an application or a set of applications which request network resources. Alternatively, it could be an agent acting on behalf of the human user. We associate each user with a flow or packet stream. Such a flow could represent the traffic generated by a single application. Alternatively, it could represent the aggregate of individual streams, as is the case for service providers.

The buffer / bandwidth trade-off can be illustrated as follows. Consider again the case of effective bandwidth based CAC. It focuses on the available bandwidth capacity in relation to the effective bandwidth required by a newly arriving stream. The CAC condition (3.32) is formulated from a point of view of a stream requiring bandwidth. The effective bandwidth here also takes the stream's burstiness into account. However, the underlying condition could also be formulated in terms of a buffer requirement. With B, C and θ^{NW} fixed such that a pre-specified QoS is provided, and with $\alpha_{AG}(\cdot) = \sum_{j=1}^J \alpha_j(\cdot)$ and $\alpha_{J+1}(\cdot)$ denoting the aggregate effective bandwidth of the J existing sources and the newly arriving stream respectively, then the CAC condition could be expressed as

$$\theta^{NW} > \alpha_{AG}^{-1}(C) + \alpha_{J+1}^{-1}(C) \quad (6.1)$$

This situation is illustrated in Figure 6.1. The horizontal and vertical lines represent the available buffer and bandwidth capacities B and C respectively. The contours indicate the effective bandwidth approximation (3.16) with $\theta = \alpha^{-1}(\cdot)$ for the *aggregate* arrival stream (i.e. all $J + 1$ sources). The effective bandwidth CAC based on the large buffer asymptotic examines the intersection of these contours with the buffer restriction. The points of intersection represent the bandwidth required by the current aggregate. However, we could also compare the intersection of the aggregate contour with the vertical constraint for C . These points of intersection then provide us with the current QoS level represented by θ that the aggregate stream enjoys. The upper limit is given by θ^{NW} , which is achieved at the maximum buffer level B . In fact, any point on the current contour line for the aggregate arrival stream represents the buffer / bandwidth required to maintain the specified QoS level. In the framework described below, we use a cost function to discriminate between all points on the same contour. This trade-off has already been examined in [105, 103, 49, 163, 104] albeit without examining the effects on pricing for the user.

Figure 6.1: Admissible region for effective bandwidth based CAC ($\alpha(\cdot)$ here represents the aggregate effective bandwidth of all arrival streams).

A precise specification of the buffer / bandwidth requirements may also extend the concept of service classes currently proposed in the standards. Users are presently required to select a service class. However, these classes are not only limited in their number. They are sometimes even overlapping. For example, in ATM a VBR connection with the SCR set to the PCR effectively provides a CBR service to the user. Such inconsistencies in the design of the service classes are undesirable from a user's point of view. Rather than simplifying the specification of the QoS level, they complicate the selection for the user. In our view buffer and bandwidth are both tangible resources and it may be easy for the user to comprehend the association of resource requirements with the desired QoS level. Our minimum cost traffic shaping framework would thus provide the user with a continuum of choices, rather than a limited set of pre-specified (and possibly overlapping) service classes. We view this as an extension, rather than a replacement, since the underlying network operation may still use a limited set of classes for technical reasons. However, the allocation of the continuum of buffer / bandwidth requirements to a particular class would then be performed by the network operator, shielded from the user. Moreover, since users would be charged for the resources required to carry their stream under the specified QoS demands, a direct correlation is introduced between charging and QoS.

Finally, the exact knowledge of buffer / bandwidth requirements may also assist network planning and operations. Recent research efforts drive for self-healing and self-sizing networks [73]. The total network capacity, assumed to be large, is separated into virtual networks in a dynamic manner. A precise knowledge of the resource requirements of each virtual network can help to dynamically allocate the total bandwidth / buffer capacity between competing virtual networks. Allowing users to freely select the buffer / bandwidth capacities provides the network with an indication of the demand function. For a given price, the network can record the demand for resources. A shift in the demand function resulting from new applications or stricter QoS requirements can then easily be detected and handled in a transparent manner for the user.

6.2 Minimum Cost Traffic Shaping Framework

In this section we formally introduce the notion of *minimum cost traffic shaping* (MCTS). As outlined in section 4.2, providing QoS guarantees typically involves a contract between the user and the network. The user announces the characteristics of the traffic stream as well as the desired QoS. The network determines whether the available resources are sufficient to satisfy this QoS demand. If this is not the case the connection is refused. Otherwise, the connection is accepted and the traffic characteristics and the QoS are fixed in the contract. The user's stream is subsequently regulated, either by a traffic shaper or policer. This ensures that the contract is honored, which is particularly important since a violation of one user's traffic contract impacts on the QoS received by other users.

The process of establishing a contract introduces two difficulties. First, how should the user characterize the arrival stream? A simplistic characterization of the traffic stream is given by the mean. However, the mean is insufficient to negotiate QoS guarantees, since it fails to capture the higher moments of the arrival stream. On the other extreme, a complete characterization is given by the effective bandwidth of the source. However, the computation of the effective bandwidth is itself not trivial. The second difficulty is how to establish the parameters of the traffic regulator. The buffer / bandwidth trade-off discussed in the previous section illustrates that a number of combinations (b, c) of buffer b

and bandwidth c could be used to satisfy the desired QoS. The contract negotiation between the user and the network cannot possibly be performed for each of these combinations in turn. As a guideline, the rate of the traffic regulator is set to the mean arrival rate of the stream. The regulator's buffer size is determined by the maximum burst size. However, such a parameterization fails to make use of the buffer / bandwidth trade-off. For example, such a parameterization may lead to unacceptable delays for the user due to the large buffer size of the regulator. Such a delay constraint could be satisfied by increasing the regulator's rate. This in turn would lead to a smaller buffer size required for the regulator.

The traffic regulator can be identified as the main interface between the network and the user. Its parameterization is also seen by the International Telecommunications Union (ITU), the Internet Engineering Task Force (IETF) and the ATM Forum as a indicator for the amount of resources required to guarantee a specified QoS to a stream.

The framework of minimum cost traffic shaping centers around the determination of the regulator's parameters. In this scheme, the *user* determines a buffer / bandwidth requirement for his / her stream which satisfy the desired QoS requirements. The parameters are found as the solution to a minimization problem. We introduce a cost function to assign a cost to each pair (b, c) . This cost function determines how to trade-off buffer and bandwidth. In MCTS, the pair (b^*, c^*) which minimizes the cost function is selected. MCTS simplifies the contract negotiation. The network provides resource prices. The user determines the MCTS resources (b^*, c^*) . Using these values, the network performs its CAC function. If successful, the minimum cost values are then used in the contract negotiation. The traffic contract then consists of the desired QoS parameters and (b^*, c^*) . Instead of policing the contract, the network / user agree to parameterize a traffic regulator using (b^*, c^*) . The proposed MCTS framework thus allows traffic contracts to be dynamically established.

In this section we model the traffic regulator as a single server queue with fixed service rate c and infinite buffer capacity. As indicated in chapter 4 such a queuing system models the performance of both a FCFS queue and a buffered / unbuffered leaky bucket regulator. We assume that the arrival streams are short-range dependent and that their effective bandwidth is independent of the time parameter. In the following section we discuss the QoS constraints in detail. In section 6.2.2 we then provide a general formulation of the

minimization problem. In section 6.2.3 we then introduce the additional requirements that the output of the regulator also has to satisfy the connection admission control constraints imposed by the network.

6.2.1 QoS Constraints

The MCTS framework assumes that the user generates a traffic stream with particular QoS requirements. We assume that the user is able to specify probabilistic QoS requirements on the loss probability, the worst-case delay and the delay variation (jitter). Typically, such constraints are formulated as

$$P(Q > b) \triangleq f(b, c, \dots) \leq \epsilon_{\mathcal{L}} \quad (6.2)$$

$$P(\mathcal{D} > \delta) \triangleq g(b, c, \dots) \leq \epsilon_{\mathcal{D}} \quad (6.3)$$

$$P(\mathcal{J} > \gamma) \triangleq h(b, c, \dots) \leq \epsilon_{\mathcal{J}} \quad (6.4)$$

where Q represents the occupancy of the queue, $\mathcal{D}$ represents the delay and $\mathcal{J}$ represents the jitter. In particular, we assume that the user is able to specify the upper bounds on these probabilities $\epsilon_{\mathcal{L}}, \epsilon_{\mathcal{D}}, \epsilon_{\mathcal{J}}$ as well as the thresholds for the delay and the jitter δ and γ respectively. Note that we do not envisage that a human user would be able to specify these quantities numerically. Instead, a set of numerical values for the thresholds and probabilities could be engineered for particular applications or for different user types. Agents acting on behalf of the user or the MCTS function at the network layer could be parameterized accordingly.

In general, the user can use any such expression to indicate the constraints he / she requires on the resources to satisfy the QoS. For many traffic types, in particular under Markovian assumptions, analytical expressions exist [99, 133, 5]. We will demonstrate the feasibility of our framework using such analytical solutions in one of the examples in section 6.3.3. However, in order to formulate the problem independently from the assumed traffic type, we will specify the QoS constraints using the effective bandwidth approximations (3.16.3.17.3.18) introduced in section 3.3. Note that we are modeling the traffic regulator as a single server queue. Since we assume that this regulator is dedicated to the arrival

Figure 6.2: Feasible region with loss, delay and jitter constraints.

stream, we know that the effective bandwidth approximation is good with constant² $\kappa \approx 1$.

We can thus formulate the QoS constraints as

$$e^{-\theta b} \leq \epsilon_L \quad (6.5)$$

$$e^{-\theta \delta c} \leq \epsilon_D \quad (6.6)$$

$$e^{-\theta \gamma c} \leq \epsilon_J \quad (6.7)$$

The parameter θ indicates the tightness of the effective bandwidth approximation in each case. Recall that the tightest bound is achieved when $\alpha(\theta) = c$, and so we can replace θ in the above constraints by $\theta = \alpha^{-1}(c)$. This allows us to express the QoS constraints as functions of the buffer size b and the service rate c of the queue. Note that both the delay constraint (6.6) and the jitter constraint (6.7) are only functions of the service rate, not the buffer size. Intuitively, these constraints indicate that the buffer *threshold* b does not influence the delay / jitter, only the parameter c . This is of course a result from our assumption of an infinite buffer capacity. Figure 6.2 represents these constraints graphically. The shaded region depicts all pairs (b, c) such that these QoS constraints are satisfied (the *feasible region*).

Recall that if the arrival process has a finite peak rate π and mean μ , then the effective

²Recall that the constant $\kappa \ll 1$ when many streams are multiplexed into the queue.

bandwidth is bound by $\mu < \alpha(\theta) < \pi$ (see section 3.1). In this case, all three constraints will never exceed $c = \pi$. Of course, if the user's stream is shaped at peak rate, then there will never be any losses, and the delay and jitter will be zero (since $\alpha(\pi)^{-1} = \infty$). The feasible region reflects this.

The advantages of using the effective bandwidth approximation now becomes apparent. The shape of the feasible region is *always* as illustrated in figure 6.2, regardless of the traffic type. The loss constraint is an exponentially decreasing function of c . It approaches infinity at $c = \mu$ and tends to the supremum of the arrival stream (which may be infinite). The convexity of the loss constraint is determined by the loss probability requirement of the user. The smaller ϵ_L , the more convex the loss constraint (and vice versa). Both delay and jitter constraints are only functions of c . They are always below the peak rate π . Their location with respect to c is also determined by the user's requirements ϵ_D and ϵ_J . The fact that the feasible region is in such a standard form simplifies the solution of the subsequent optimization problem for the user!

6.2.2 Framework Formulation

Since the feasible region represents all resource pairs (b, c) of buffer and bandwidth which satisfy the user's QoS, we can discriminate between them if we assign a cost to each allocation. The user's minimum cost parameterization of the traffic regulator (b^*, c^*) is then the solution to the following minimization problem:

$$\text{Min } \mathcal{C}(b, c, \alpha(.)) \quad (\text{OBJ})$$

$$\text{s.t. } P(Q > b) \triangleq f(b, c, \alpha(.)) \leq \epsilon_L \quad (\text{C1})$$

$$P(\mathcal{D} > \delta) \triangleq g(b, c, \alpha(.), \delta) \leq \epsilon_D \quad (\text{C2})$$

$$P(\mathcal{J} > \gamma) \triangleq h(b, c, \alpha(.), \gamma) \leq \epsilon_J \quad (\text{C3})$$

$$\mathcal{C}(b, c, \alpha(.)) \leq \mathcal{B}(b, c, \alpha(.)) \quad (\text{C4})$$

where we have indicated that the effective bandwidth $\alpha(.)$ of the traffic stream may be a parameter of the QoS constraints and hence the cost function. The cost function $\mathcal{C}(.)$ here is a general measure to distinguish between the resource allocations. It is assumed to be non-decreasing and differentiable in both parameter spaces b and c . Such a cost function could be

imposed by the network to induce the user to trade-off buffer for bandwidth (or vice versa). It may be related to the networks charging mechanism to recover the cost of providing the resources at the network side. Or it may be a control mechanism by the network which is unrelated to the charging mechanism. The cost function may also be defined by the user to reflect the user's monetary cost of providing the buffer / bandwidth resources. In our case we always interpret this function as a *monetary* cost function reflecting the price the user has to pay for the resources required to provide the desired QoS. In this section we assume that the user himself / herself determines the cost function. We will relax this assumption in subsequent sections.

The slope of the cost function is of particular importance to the solution. It represents the marginal cost of substitution, indicating how the user could substitute buffer for bandwidth and vice versa to preserve the current total cost. For example a unit-reduction in bandwidth could be re-invested in $-\partial b / \partial c$ units of buffer. The characteristics of the feasible region described in the last section indicate that even a linear cost function suffices to find a unique solution to the minimization problem. Without loss of generality we will therefore restrict ourselves to linear cost functions in most cases from now onwards.

Since we interpret the cost function in a monetary way, we have also included a budget constraint in the general formulation of MCTS. Constraint (C4) reflects the fact that the user typically only has a limited budget $\mathcal{B}(b, c, \alpha(.))$. It represents the upper limit which the user accepts to pay. The inclusion of such a constraint has the consequence that the feasible region may collapse. For example, if the budget constraint is zero and the cost function for buffer and bandwidth increases positively in both parameters, then no allocation of (b, c) will satisfy the budget constraint. We typically assume that the budget constraint is not binding in the discussion which follows.

Economic theory often describes a user's choice problem in form of utility functions. Utility is an abstract concept describing the happiness or satisfaction the user derives from the consumption of a particular choice of goods. It may include many factors, such as price, friendliness of service, customer support etc. Since utility is such an abstract concept, it is hard to measure and define. Different users may have different utility functions, each depending on different factors. Users typically aim to maximize their utility. Note that

minimizing cost may indeed maximize the utility of a user. Cost may be one factor from which users derive negative utility. If utility is only derived from cost (ignoring other factors) then cost minimization also maximizes utility. While we acknowledge that utility maximization is more general, we restrict ourselves to cost minimization. In this way we are able to associate the minimum cost solution with a charging scheme for the user later on. It also simplifies the presentation of our framework. The general formulation of MCTS assumes that all variables are ergodic and stationary. The resource allocation (b^*, c^*) and the associated cost $\mathcal{C}(b^*, c^*, \alpha(.))$ are to be interpreted per unit of time for the duration of stream.

Assume now that the feasible region exists, for example by assuming an unlimited budget constraint. Due to the standard form of the feasible region, the nature of the solution under our assumption of a linear cost function is one of the following:

- Delay / Jitter allocation
- Loss allocation
- Peak-rate allocation

Which of these three types is optimal depends on the slope of the cost function. For example, a linear cost function with a slope $-\partial b / \partial c$ close to zero will result in a peak rate allocation. In this case, bandwidth is cheap relative to buffer, and so the minimum cost resource requirement is characterized by $(0, \pi)$. The three solution types are illustrated in figure 6.3.

The MCTS problem can in general be solved by the Lagrangian method under mild conditions for the constraints and the objective function. Arrow and Enthoven [3] have shown that if $\mathcal{C}(b, c, \alpha(.))$ is differentiable and quasi-concave³ and all of the constraints are differentiable and quasi-convex⁴ and assuming that a solution for (b^*, c^*) exists, then the Lagrangian method will find it. Furthermore, if the conditions satisfy the Kuhn-Tucker

³ $Y = f[x]$ is said to be quasi-concave iff. whenever $q[x_1] \geq z$ and $q[x_2] \geq z$ then also $q[kx_1 + (1-k)x_2] \geq z$, for all $0 < k < 1$.

⁴ $Y = q[x]$ is said to be quasi-convex iff. whenever $q[x_1] \leq z$ and $q[x_2] \leq z$ then also $q[kx_1 + (1-k)x_2] \leq z$, for all $0 < k < 1$.

Figure 6.3: Illustration of possible types of the optimal solution (D = delay allocation, L = loss allocation, P = peak rate allocation).

conditions and the quasi-concave programming conditions, then the solution is unique. Our first set of examples provides the details of such a calculation. We already note here that this calculation is general for short-range dependent traffic types.

At this point we would also like to point out the effect of a cost function based on bandwidth only. This could be the case if no charge is imposed for the buffer. In this case, the slope of the cost function $\partial b / \partial c = \infty$. The optimal solution would then be a mean rate allocation (∞, μ) . The engineering guideline of setting the traffic regulator's rate to the mean rate and the buffer size to the maximum burst size thus corresponds to a cost function only based on bandwidth.

6.2.3 CAC Constraints

The general MCTS introduced in the last section did not take into account that the output stream of the traffic regulator has to satisfy the networks CAC constraints. Without such constraints the user determines the minimum cost parameters for the traffic regulator without taking the resource availability of the network into account. Even the cost function was assumed to be determined by the user alone, reflecting the cost to the user of providing the resources for traffic regulation. However, the resulting output stream may be rejected by the network if insufficient resources are available. Also, the network may charge for the

resources required to carry the user's regulated stream.

A more complete formulation of the MCTS framework would include such constraints. The user's optimization problem to determine the resource requirements for a given QoS then becomes

$$\text{Min } \mathcal{C}(b, c, \alpha(.)) \quad (\text{OBJ})$$

$$\text{s.t. } P(Q > b) \stackrel{\Delta}{=} f(b, c, \alpha(.)) \leq \epsilon_{\mathcal{L}} \quad (\text{C1})$$

$$P(\mathcal{D} > \delta) \stackrel{\Delta}{=} g(b, c, \alpha(.), \delta) \leq \epsilon_{\mathcal{D}} \quad (\text{C2})$$

$$P(\mathcal{J} > \gamma) \stackrel{\Delta}{=} h(b, c, \alpha(.), \gamma) \leq \epsilon_{\mathcal{J}} \quad (\text{C3})$$

$$0 < b \leq B^* \quad (\text{C4})$$

$$0 < c \leq C^* \quad (\text{C5})$$

$$\mathcal{C}(b, c, \alpha(.)) \leq \mathcal{B}(b, c, \alpha(.)) \quad (\text{C6})$$

The addition of constraints (C4) and (C5) has several implications for the user. The network is now assumed to communicate the current buffer / bandwidth availability (B^*, C^*) to the user at the time the optimization is performed. We envisage this information exchange to be part of the protocol to establish the traffic contract. Furthermore, the user has to be aware whether the network's CAC is based on (σ, ρ) -constraints or effective bandwidths. Recall that such constraints were derived in sections 2.4 and 3.4 respectively.

Constraints (C4) and (C5) may also affect the feasible region. For example, the available bandwidth may be below the peak rate of the user's traffic stream. Additional upper bounds on both buffer and bandwidth are thus introduced. The modified optimization problem is depicted in figure 6.4.

If the cost function is set by the network it may change over longer time intervals. This would allow the network to respond to changes in the network status. For example, if the network resources become congested, the cost function may be increased to deter some user from transmitting (through the budget constraint). Changes in the cost function over short time intervals may indeed be technically feasible. However, such changes must not affect current transmissions, only newly arriving streams. Otherwise each user is required to perform another optimization at the time when the cost function changes. Some users may not be able to find a minimum cost solution anymore, in which case they would abort

Figure 6.4: Graphical representation of the optimization problem

their transmission. We view such a scenario as highly undesirable from a user's perspective. Moreover, *human* users may not respond well to cost functions which are hard to predict over time.

Finally, we point out that the nature of the cost function may also be affected by a tighter integration with the network. The cost function used in this section only reflects the charges for the resources required by the traffic shaper. The network may also impose charges for the resources required to carry the resulting output stream. A more general form of the linear cost function would be

$$\mathcal{C}(b, c, b^D, c^D, \alpha(.)) = p_1 b + p_2 c + p_3 b^D + p_4 c^D \quad (6.8)$$

where the superscripts D indicate the buffer / bandwidth resources of the departure process from the regulator. The coefficients $p_i; i = 1 \dots 4$ are again assumed to be non-negative. In the following two sections we will elaborate MCTS under the two CAC regimes in detail.

6.3 Minimum Cost Traffic Shaping under (σ, ρ) -constraints

In this section we discuss the details of MCTS in case the network's CAC function is based on (σ, ρ) -constraints.

Figure 6.5: Modified model for a VP including a traffic regulator.

6.3.1 Framework Formulation

We assume that a user wishes to establish a virtual circuit (VC) by gaining access to a virtual path (VP). In our model the VP consists of H nodes, modeled as single server queues. To simplify our discussion, we assume that each node has a bandwidth capacity of C and a buffer size of B . A total of J VCs have already been established by other users, each of which is (σ_j, ρ_j) -constrained, $j = 1 \dots J$. The user's QoS parameters for loss, delay and jitter are known to be ϵ_L, ϵ_D and ϵ_J , as are the threshold values δ, γ for the delay and jitter respectively. Without loss of generality, we assume that the user has an unlimited budget constraint. Figure 6.5 illustrates this setup.

Since we are assuming a particular VP, the QoS parameters have to be 'distributed' over all nodes in the VP and the traffic regulator as indicated in section 3.3.3. However, a consequence of (σ, ρ) -constrained CAC is that once a stream is admitted to the VP, no losses occur inside the VP. In our model losses can only occur at the traffic regulator and so ϵ_L applies to the regulator only. Concerning the delay constraint, since we assume all buffers inside the VP to have size B , an upper bound on the end-to-end delay is

$$d_{VP} \leq \frac{BH}{C} + \frac{H}{C} + \Pi \quad (6.9)$$

where the first term on the right represents the queuing delay, the second term represents the processing delay and Π represents the end-to-end propagation delay. Hence

$$\begin{aligned} P(\mathcal{D} > \delta) &= P(\mathcal{D}_{TS} + \mathcal{D}_{VP} > \delta) \\ &\leq P(\mathcal{D}_{TS} + d_{VP} > \delta) \\ &= P(\mathcal{D}_{TS} > \delta - d_{VP}) \end{aligned}$$

where $\mathcal{D}_{TS}$ and $\mathcal{D}_{VP}$ denote the delay inside the traffic shaper and the VP respectively. The delay constraint for the traffic regulator is then

$$e^{-\alpha^{-1}(c)(\delta-d_{VP})c} \leq \epsilon_{\mathcal{D}} \quad (6.10)$$

Finally, as argued in section 3.3.3 the jitter is in the range of the maximum transfer time of a single node, so for the traffic regulator we also have respectively

$$e^{-\alpha^{-1}(c)\gamma c} \leq \epsilon_{\mathcal{J}} \quad (6.11)$$

Recall that the output of a single server queue with fixed capacity c is $(1, c)$ -constrained (see section 2.2). In addition to the QoS constraints described by their effective bandwidth approximations, the following CAC constraints have to be satisfied:

$$c \leq C - \sum_{j=1}^J \rho_j \quad (6.12)$$

$$1 \leq B - \sum_{j=1}^J \sigma_j - H \quad (6.13)$$

Note that the output stream is not a function of the regulator's buffer size b . We assume that the network imposes a cost function of the following form

$$\mathcal{C}(b, c) = p_1 b + p_2 c \quad (6.14)$$

Under these assumptions the MCTS problem can be formulated as follows:

$$\text{Min } p_1 b + p_2 c \quad (\text{OBJ})$$

$$\text{s.t. } e^{-\alpha^{-1}(c)b} \leq \epsilon_{\mathcal{L}} \quad (\text{C1})$$

$$e^{-\alpha^{-1}(c)(\delta-d_{VP})c} \leq \epsilon_{\mathcal{D}} \quad (\text{C2})$$

$$e^{-\alpha^{-1}(c)\gamma c} \leq \epsilon_{\mathcal{J}} \quad (\text{C3})$$

$$0 < b \leq B^* \quad (\text{C4})$$

$$0 < c \leq C^* \quad (\text{C5})$$

where $B^* = B - \sum_{j=1}^J \sigma_j - H$ and $C^* = C - \sum_{j=1}^J \rho_j$.

With our interpretation of the cost function in monetary terms, the network in the above formulation charges the user for the resources required to regulate the stream such that the QoS and CAC constraints are satisfied. We note that the network in the case of (σ, ρ) -constrained CAC does not need to *allocate* the buffer requirement b *inside* the network, since the output stream of the regulator is $(1, c)$ -constrained. However, such a cost function may still be justified for the following reasons. First, the MCTS parameters for the traffic shaper indicate the resources required to carry the stream while providing the desired QoS. They can thus be interpreted as a translation of a QoS requirement into a resource requirement. The cost function then charges the user for QoS through the parameters (b, c) . Second, due to clumping effects described in section 2.2, a users stream may also occupy more than one space in a network buffer inside the VP. So even though the output stream of the regulator is bounded by one cell / packet per time unit, the buffer space occupied by cells / packets from the user's stream inside any of the network nodes may exceed one. The cost function accounts for this phenomenon. Third, in many cases the network may have to re-shape the user's traffic inside the network. The optimal parameters also indicate how the network has to parameterize the regulators inside the network. Finally, the physical resources for the shaper may actually be provided by the network UNI. In this case, even without re-shaping inside the network, the cost function would be justified since the network provides the resources for the initial shaping.

Of course the cost function could be based purely on the resources required to carry the regulator's output stream. In the case of a $G/D/1/b$ -queue regulator the output stream is $(1, b)$ -constrained, and so the respective cost function would be

$$\mathcal{C}(c) = p_3 + p_4 c^D$$

and so the minimum cost solution would be (∞, μ) . If all the arrival streams are guaranteed to be $(1, \rho)$ -constrained, the network could provision its buffers according to the maximum number of streams allowed on the VP per unit time. Each user would provide the buffer required to smooth his / her stream without incurring costs. The network is then unable to substitute bandwidth with buffer.

Note that in the above formulation for MCTS the effective bandwidth and the (σ, ρ) -paradigms are mixed. We use the effective bandwidth approximation to formulate the QoS constraints. However, the CAC constraints are based on (σ, ρ) -constrained CAC by assumption.

The next section illustrates the MCTS scheme under (σ, ρ) -constrained CAC using Gaussian, Poisson or on-off sources as an example. It shows the details for the solution to the MCTS problem for the general case.

6.3.2 Example: Gaussian, Poisson and On-Off Sources

We start by illustrating the scheme in this example using on-off sources, assuming that the user is able to completely specify the on-off traffic model operating in discrete time and modeled by a two state Markov modulated arrival process with transition probability matrix R . A single arrival occurs when the model is in its on-state. No arrivals occur during the off-state. The model jumps from the on-state to the off-state in any slot according to the matrix R . The effective bandwidth of such a source is given by [10, 68, 88, 97] as

$$\alpha(\theta) = \frac{1}{\theta} \log \left[\frac{r_{11} + r_{22}e^\theta + \sqrt{(r_{11} + r_{22}e^\theta)^2 + 4r_{12}r_{21}e^\theta}}{2} \right] \quad (6.15)$$

where⁵ $\theta = \sup_{\psi} [\alpha(\psi) \leq c]$. We also assume that the network has communicated the upper bound for the delay across this VP, $d_{V,P}$, to the user. Thus the user does not need to know exactly the number of hops on the VP to apply the QoS-requirements. The optimization problem stated in the last section can then be solved as follows (note that we have $\alpha(\theta) = c$ and therefore $\theta = \alpha^{-1}(c)$ is a function of c).

The form of the Lagrangian $\mathcal{F}$ is:

$$\begin{aligned} \mathcal{F} = & p_1 b + p_2 c \\ & - v_1(\theta b + \log(\epsilon_{\mathcal{L}})) \\ & - v_2(\theta c(\delta - d_{V,P}) + \log(\epsilon_{\mathcal{D}})) \end{aligned}$$

⁵We henceforth only consider the extremizing value θ as the solution to $c = \alpha(\theta)$.

$$\begin{aligned}
& -v_3(\theta c\gamma + \log(\epsilon_{\mathcal{J}})) \\
& -v_4(B^* - b) \\
& -v_5(C^* - c),
\end{aligned}$$

and the solution to this equation is the set of values for $\theta, b, c, v_1, v_2, v_3, v_4$ and v_5 for which the following Kuhn-Tucker conditions hold:

$$\begin{aligned}
b > 0 \quad \partial\mathcal{F}/\partial b &= 0 \\
c > 0 \quad \partial\mathcal{F}/\partial c &= 0 \\
v_i \geq 0 \quad \partial\mathcal{F}/\partial v_i \leq 0 \quad v_i(\partial\mathcal{F}/\partial v_i) &= 0; i = 1 \dots 5
\end{aligned}$$

The Kuhn-Tucker conditions provide the necessary and sufficient conditions that the optimal solution (b^*, c^*) must satisfy. Appendix B provides a brief derivation and justification of the necessity and sufficiency of the Kuhn-Tucker conditions.

If constraints (C4) and (C5) are binding, then the solution is trivially found by the intersection of the cost function with the corner solutions defined by these constraints. Assume therefore that constraints (C4) and (C5) are not binding, i.e. that sufficient resources are available to find a solution for (b, c) . Assume furthermore that only one of (C2) and (C3) is binding. Both these constraints have the same form and are only functions of c . We concentrate on constraint (C2) for the remainder of the calculation. In the calculation of $\partial\mathcal{F}/\partial c$ we need to find $\partial\theta/\partial c$. Using $\alpha(\theta) = c$ this is just

$$\frac{\partial\theta}{\partial c} = \frac{\partial\theta}{\partial\alpha(\theta)} = \frac{1}{(\partial\alpha(\theta)/\partial\theta)}$$

After some algebra the optimizing value of θ , θ^* , can be found as the solution to either

$$\theta^2 \frac{\partial\alpha(\theta)}{\partial\theta} = -\frac{p_1}{p_2} \log(\epsilon_{\mathcal{L}}) \quad (\text{C1 binding}) \tag{6.16}$$

$$\theta\alpha(\theta) = -\frac{\log(\epsilon_{\mathcal{P}})}{(\delta - d_{\mathcal{I} \cdot \mathcal{P}})} \quad (\text{C2 binding}) \tag{6.17}$$

Given θ^* the remaining values can be found using

Figure 6.6: Plots for $\Lambda(\theta)$ and $\theta^2 \frac{\partial \alpha(\theta)}{\partial \theta}$ for an on-off source.

$$c = \alpha(\theta^*) \quad (6.18)$$

$$b = -\frac{\log(\epsilon_{\mathcal{L}})}{\theta^*} \quad (6.19)$$

$$v_1 = \frac{p_1}{\theta^*} \quad (6.20)$$

$$v_2 = \begin{cases} 0 & \text{(C1) binding} \\ p_2 + \frac{\frac{p_1}{\theta^*} \log(\epsilon_{\mathcal{L}}) \frac{\partial \theta}{\partial c}}{(\delta - d_{\mathcal{L}} \cdot p)(\theta^* + \alpha(\theta^*)) \frac{\partial \theta}{\partial c}} & \text{(C2) binding} \end{cases} \quad (6.21)$$

$$v_{3,4,5} = 0 \quad (6.22)$$

Note that equations 6.16 and 6.17 can easily be solved numerically. Both $\theta^2 \partial \alpha(\theta) / \partial \theta$ and $\theta \alpha(\theta)$ are increasing in θ . Figure 6.6 depicts the graphs for the functions $\Lambda(\theta) = \theta \alpha(\theta)$ and $u(\theta) = \theta^2 \frac{\partial \alpha(\theta)}{\partial \theta}$ respectively for the on-off sources considered here. Note that $\theta^2 \partial \alpha(\theta) / \partial \theta$ reaches a limit in the case of on-off sources. For certain values $-\frac{p_1}{p_2} \log(\epsilon_{\mathcal{L}})$ may be larger than this limit. This corresponds to the case of a peak rate allocation, where the slope of the cost function is shallower than the slope of the loss constraint at the peak rate.

The minimum cost solution (b^*, c^*) has the desirable property of guaranteeing access to the network, provided the traffic stream is shaped by a $G/D/1/b^*$ queue with service rate c^* . The user's QoS will be met at a minimum cost per time unit of the connection.

We now turn our attention to Gaussian sources with mean μ and variance σ^2 . In this case, the effective bandwidth is given by

$$\alpha(\theta) = \mu + \frac{\sigma^2}{2} \theta \quad (6.23)$$

As the calculation above shows, we can determine θ^* as the solution to either equation (6.16) or (6.17), depending on whether constraint (C1) is binding or not. In this case,

$$\begin{aligned}\theta^2 \frac{\partial \alpha(\theta)}{\partial \theta} &= \theta^2 \frac{\sigma^2}{2} \\ \theta \alpha(\theta) &= \theta \mu + \frac{\theta^2 \sigma^2}{2}\end{aligned}$$

and so θ^* is

$$\theta^* = \begin{cases} \sqrt{-\frac{\sigma^2}{2} \left(\frac{p_1}{p_2} \log \epsilon_{\mathcal{L}} \right)} & \text{(C1) binding} \\ -\frac{\mu}{\sigma^2} + \sqrt{\frac{\mu^2}{\sigma^4} - \frac{2 \log \epsilon_{\mathcal{P}}}{\delta - d_{\mathcal{V}P}}} & \text{(C2) binding} \end{cases} \quad (6.24)$$

For the Poisson case with parameter λ , we have an effective bandwidth of

$$\alpha(\theta) = \frac{\lambda}{\theta} (e^\theta - 1)$$

and so

$$\begin{aligned}\theta^2 \frac{\partial \alpha(\theta)}{\partial \theta} &= \lambda [(e^\theta (\theta - 1) + 1)] \\ \theta \alpha(\theta) &= \lambda (e^\theta - 1)\end{aligned}$$

In this case if (C1) is binding θ^* is the solution to

$$-\frac{p_1}{p_2} \log \epsilon_{\mathcal{L}} - 1 = e^\theta (\theta - 1) \quad (6.25)$$

otherwise

$$\theta^* = \log \left[1 - \frac{\log \epsilon_{\mathcal{P}}}{\lambda (\delta - d_{\mathcal{V}P})} \right] \quad (6.26)$$

The solution method presented in this section has a number of desirable properties. First, we note that the solution is *general* for *any* short-range dependent traffic stream with known effective bandwidth. If either the loss constraint or the delay constraint is binding, equations (6.16) and (6.17) determine the final solution. Since the effective bandwidth

function is always increasing in θ , these equations can always be solved numerically, possibly even in real time⁶. This would allow the network to change prices to reflect the current state of the resource availability. A non-human user (e.g. a software agent acting for the human user) could respond to price fluctuations in this time-scale. On the other hand, even at time scales acceptable to the user (minutes or more), a fast solution to the MCTS problem is desirable. If either constraint (C4) or (C5) are binding, θ^* can be determined immediately from $\theta^* = -\frac{\log(\epsilon \mathcal{C})}{B^*}$ or $\theta^* = \alpha^{-1}(C^*)$.

Second, examining the region defined by the constraints, as depicted in figure 6.4, immediately reveals the feasibility of the solution. If this region defines an empty set, the user can re-examine the demanded QoS and repeat the process with alternative parameters. If the region defines a singleton, the solution follows immediately. If the region defines a set of values for b and c , the solution follows from the mathematical procedure outlined above. For QoS constraints characterized by the effective bandwidth approximation, a linear cost function suffices. In other cases, the network may have to adapt the cost function to guarantee a solution. If the constraints are quasi-convex, a quasi-concave cost function guarantees a solution. Similarly, if the constraints are quasi-concave, a quasi-convex cost function guarantees a solution.

Third, the values taken by the Lagrangian multipliers v_i also reveal important information on the sensitivity of the result. If $v_i > 0$, then constraint i is binding and furthermore indicates how the value of the objective function (the cost function) would change in response to a small relaxation of constraint i . Furthermore, if $v_i = 0$ then constraint i is not binding and we can compute the slackness of the constraint by introducing an artificial variable y . This variable takes on the value which would make constraint i binding. It also allows us to compute how much the input would have to change before the optimal solution changes (see [160], for example).

Finally, we note that the user and the network only have to exchange minimal information required to determine the CAC and QoS constraints. The network is assumed to contribute the cost function $\mathcal{C}(\cdot)$ and optionally the available resource capacities B^*, C^* to

⁶We solved equation 6.16 in less than 100 ms and equation 6.17 in less than 1 ms using Mathematica 3.0 on a 155MHZ Pentium PC running Windows 95 for the on-off case.

the optimization. The user is assumed to know his / her QoS parameters $\epsilon_{\mathcal{L}}, \epsilon_{\mathcal{D}}$ and $\epsilon_{\mathcal{J}}$, as well as the threshold values δ, γ for the delay and jitter respectively. Not all of these QoS constraints may be required by the user every time. The traffic contract can be immediately established using these values and the optimal solution. Note that if the network provides the available resource capacities (B^*, C^*) then these have to be 'locked' to prevent other users from concurrently accessing these resources. If the available resource capacities are not provided, then the user simply computes (b^*, c^*) and subsequently goes through a CAC procedure with the network. This procedure is greatly simplified for the network, since it only has to hunt for a VP to the destination with the specified capacities (b^*, c^*) . If insufficient resources are available, the stream is rejected. In this case the user may re-attempt transmission at a later time.

6.3.3 Example: Self-Similar Traffic Streams

This section extends the minimum cost traffic shaping scheme for self similar traffic, as shown in [39]. The traffic shaping problem is again interpreted as trying to find the minimum cost parameters of the traffic shaper which enable the user to gain access to the network while satisfying the QoS requirement of the connection. In this case, the QoS constraints are formulated *analytically*. As shown by Gallardo et al. [56], the effective bandwidth results in chapter 3 are not valid for processes where the moment generating function is ill-defined (i.e. having infinite higher moments)⁷. Analytical results do exist however for fractional Gaussian processes. In this section we illustrate the validity of our scheme for such processes. The user is required to provide basic characteristics of the traffic stream, such as the mean rate, the variance, and the degree of self-similarity. As above, the user specifies a maximum tolerated loss probability $\epsilon_{\mathcal{L}}$ and the network supplies the user with upper bounds on the available resources (B^*, C^*) as well as the form of the cost function by which the user is charged.

Before we formulate the framework under these assumptions, we provide a brief review of self-similarity.

⁷Gallardo et al. propose to use α -stable processes to model broadband traffic.

6.3.3.1 Self-similarity in broadband networks

Studies of traffic measurements for LAN traffic, WAN traffic and variable bit rate (VBR) traffic [161, 130, 20, 4] have shown that the actual network traffic is long range dependent (LRD) and would be better modeled by self-similar ("fractal") rather than traditional Poisson-based processes. Contrary to common beliefs that multiplexing traffic streams tends to produce smoothed-out aggregate traffic with reduced burstiness, aggregating self-similar traffic streams can intensify burstiness rather than diminish it. This has significant implications on the queuing behavior of the traffic streams, and hence on the performance measures such as the loss probability [153].

The need for studying self-similar traffic has motivated creation of appropriated models. There are many stochastic models which exhibit the self-similar property, such as the exactly self-similar fractional Gaussian noise (FGN) or the asymptotically self-similar fractional autoregressive integrated moving-average (F-ARIMA) processes. Associated with FGN is fractional Brownian motion (FBM), which is the integrated version of FGN. These models require only several parameters to be specified, typically the mean, variance and Hurst parameter $\mathcal{H}$ [121, 76].

An FGN process $A = A_t : t = (1, 2, \dots)$ is a stationary Gaussian process with mean $\mu = E[A_t]$, variance $\sigma^2 = E[(A_t - \mu)^2]$ and autocorrelation function

$$r(t) = \begin{cases} 1, & t = 0 \\ \frac{1}{2}(|t+1|^{2\mathcal{H}} + 2|t|^{2\mathcal{H}} + |t-1|^{2\mathcal{H}}), & t = 1, 2, \dots \end{cases} \quad (6.27)$$

Thus for $\frac{1}{2} < \mathcal{H} < 1$, the FGN process A is exactly second-order self similar with Hurst parameter $\mathcal{H}$.

In [150] an efficient and simple technique based on the Spatial Renewal Process (SRP) is proposed to generate a self-similar traffic stream. The SRP process can exactly match any given marginal distribution even sub-exponential autocorrelation functions corresponding to LRD-traffic. The SRP model is fit to the steady-state distribution and autocorrelation requirements that characterize a fractional Gaussian Noise (FGN) process. It has a linear computational requirement $O(n)$ (where n is the sample size), limited memory requirements and it does not require the sample size to be known at the start of the generation process.

For these reasons we chose the SRP-FGN generator to produce self-similar traffic streams.

The SRP renewal function $F_T(t)$ is as follows:

$$F_T(t) = \begin{cases} \omega(2^{2\mathcal{H}-1} - 2), & 0 \leq t < 1 \\ \omega\mathcal{H}((t+1)^{2\mathcal{H}-1} - 2t^{2\mathcal{H}-1} + (t-1)^{2\mathcal{H}-1}), & t \geq 1 \end{cases} \quad (6.28)$$

with $\omega = 1 - (2^{2\mathcal{H}-1} - 2)^{-1}$. The SRP renewal function $F_T(t)$ was calculated with Hurst parameters $\mathcal{H} = \{0.65, 0.75, 0.85, 0.95\}$ below. We use a $N(0, 1)$ random variable to characterize the background sequence A_t . To take advantage of the averaging effect of aggregation, we combined 25 independent SRP streams and computed the output by summing the traces and re-normalizing the sample variance to 1.

6.3.3.2 Framework Formulation

In the subsequent formulation, we assume that the user wishes to transmit a self-similar traffic stream with known mean cell rate μ , variance σ^2 and Hurst parameter $\mathcal{H}$. The user demands a guarantee on the loss probability $\epsilon_{\mathcal{L}}$. The network is able to provide this guarantee if the user agrees to voluntarily shape the traffic using a single server queue with finite buffer size b and deterministic service rate c and if b and c are within a region $B_{min} \leq b \leq B_{max}$ and $C_{min} \leq c \leq C_{max}$ prescribed by the network. This guarantee is obtained by provisioning a VC with buffer size b and bandwidth c along the source-destination path.

Furthermore, the network charges the user according to the demanded resources using a cost function $\mathcal{C}(b, c)$ increasing in b and c . We assume again that the user has an unlimited budget constraint.

In the case the formulation of the MCTS problem is:

$$\text{Min } \mathcal{C}(b, c) \quad (\text{OBJ})$$

$$\text{s.t. } P(Q > b) \leq \epsilon_{\mathcal{L}} \quad (\text{C1})$$

$$1 \leq B_{min} \leq b \leq B_{max} \quad (\text{C2})$$

$$0 < C_{min} \leq c \leq C_{max} \quad (\text{C3})$$

In order to incorporate constraint (C1) we require an expression for the loss probability as a function of the buffer size and the service rate of the traffic shaper. Fu [55] and Giordano

et al. [63] determined the following analytical form for the probability that the queue size Q exceeds level b when a single self-similar traffic stream with Hurst parameter $\mathcal{H}$ is fed into a single server queue with deterministic service rate c

$$P(Q > b) \leq e^{-\frac{(c-\mu)^2 \mathcal{H}}{2\sigma^2 \mu [(1-\mathcal{H})^2 + \mathcal{H} \mathcal{H}^2]} b^{2-2\mathcal{H}}} \quad (6.29)$$

Constraints (C2) and (C3) are linear and hence convex by definition [44]. Using equation (6.29) in constraint (C1) it can be shown that this constraint is convex. We assume that the objective function (*OBJ*) is concave in the decision parameters b and c . These properties again ensure that the problem has indeed a solution and that furthermore it is unique.

The above minimization problem can be solved by gradient based methods. In this case we have used the optimization routine DONLP2 [148], which is suitable to solve complex non-linear programming problems with highly non-linear equality and inequality constraints of the type of (C1). The results of the optimization problem are presented in the next sections.

6.3.3.3 Simulation Verification

We tested the proposed traffic shaping scheme for a number of input traffic streams with different Hurst parameters. First we performed several simulation runs to verify the validity of the analytical formula (6.29). For this purpose, we generated a single FGN traffic stream with Hurst parameter $\mathcal{H} = \{0.65, 0.75, 0.85, 0.95\}$, mean cell rate $\mu = 500$, variance $\sigma^2 = 2$ using the SRP model described in section 6.3.3. We fed this stream into a single server queue with varying service rates and buffer sizes. Figure 6.7a illustrates the results for the loss probability with varying buffer size b and fixed service rate $c = 540 \frac{\text{cells}}{\text{sec}}$. Figure 6.7b shows a similar comparison for a fixed buffer size of $b = 15 \text{ cells}$ and varying service rate c .

In both cases the simulated loss probability is between 1.5 and 3 orders of magnitude lower than the calculated loss probability from (6.29). Furthermore, the analytical and simulated loss probability functions are sufficiently parallel, indicating that the decay rate of the loss probability is correctly captured by the formula.

To verify the minimization algorithm we studied the sensitivity of the minimum cost

Figure 6.7: Simulated and Analytical loss probability

solution for the buffer size b and the service rate c of the traffic shaper to the constraints (C2) and (C3). Throughout our experiments, we use a quadratic cost function of the form $\mathcal{C}(b, c) = p_1(c - p_2)^2 + p_3(b - p_4)^2$ with the following coefficients: $p_1 = p_3 = 1; p_2 = p_4 = 0$. We thus model a quadratic cost function with global minimum at the origin. Figure 6.8 illustrates the contours of such an objective function and the feasible region. We set $\epsilon_{\mathcal{L}} = 1 \times 10^{-6}$.

In our first experiment, we removed the boundary constraints (C2) and (C3) on b and c . This provides us with the minimum cost traffic shaping parameters to satisfy the required loss probability for the user.

We applied the non-linear programming algorithm DONLP2 to our minimization problem and obtained the following solution:

$$c^* = 609.03 \frac{\text{cells}}{\text{sec}}; b^* = 108.30 \text{ cells}$$

corresponding to a minimum cost of $\mathcal{C}_{opt}(b^*, c^*) = 382650$. We verified this result analytically using Mathematica by obtaining the set of pairs (b, c) for which the loss probability constraint (C1) is satisfied and then restricting the objective function (OBJ) to this set. Figure 6.9 shows the value of the cost function for all points which satisfy the loss probability constraint where $108 \leq b \leq 110$ and $608.5 \leq c \leq 610$. Note that the cost increases strictly with b and c within the restricted region. The graph verifies the optimal solution obtained from the non-linear programming algorithm DONLP2.

Figure 6.8: Contour Plot of the Feasible Region

Figure 6.9: Value of the Cost Function restricted to the Feasible Region

Table 6.1: Minimization results without constraints on service and buffer

C_{min}	C_{max}	B_{min}	B_{max}	c	b	$\mathcal{C}(\dots)$
620	720	105	205	620	105	395000
720	820	205	305	720	205	560000
920	1000	200	300	920	200	886000

In our second experiment, we impose constraints (C2) and (C3) for different values of C_{min} , C_{max} , B_{min} and B_{max} . We chose the values for the constraints to create feasible regions excluding the optimal solution (b^*, c^*) . Furthermore, we move the feasible region further and further away from the optimal solution. In this way, we expect the minimization algorithm to converge to a boundary solution, satisfying the constraints (C2) or (C3).

The results in Table 6.1 indicate that the minimization algorithm indeed converges to a boundary solution, satisfying the lower bound conditions of the constraints (C2) and (C3). As expected, the value of the cost function $\mathcal{C}(\dots)$ rises the further we depart from the optimal solution.

In our third experiment to validate the minimization algorithm, we fix the range of the buffer size b to be $10 \leq b \leq 5000$, thus allowing the optimal solution b to be reached. However, we now splice the c -dimension to create feasible regions which mostly exclude the optimal value for c .

Table 6.2 again confirms the validity of our minimization algorithm. For the range $540 \leq c \leq 600$, the solution for c is at the upper bound C_{max} . For all ranges of $C_{min} > c^*$, the solution for c is at the lower bound C_{min} . Moreover, the results indicate how buffer is traded in for service to minimize the cost. As the solution for the service rate c increases due to the constraints on C_{min} and C_{max} , the value for the buffer size b decreases. The network forces the user to demand a service rate which is higher than the one the user would select without constraint (C2). In return, the user demands less buffer size while still satisfying the loss probability constraint (C1). Note that the value of the cost function is higher for all ranges where the optimal solution c^* is not included. Only in the range $600 \leq c \leq 620$

Table 6.2: Minimization results for different feasible regions spliced in the service-dimension

C_{min}	C_{max}	c	b	$\mathcal{C}(\dots)$
540	560	560.00	3195.93	10527597
560	580	580.00	626.05	728343
580	600	600.00	176.79	391254
600	620	609.03	108.31	382650
620	640	620.00	62.92	388358
640	660	640.00	26.26	410290
660	680	660.00	12.32	435752
680	700	680.00	10.00	442500

is the optimal solution c^* .

We repeated the last experiment allowing the service rate c to lie in the range $540 \leq c \leq 1000$. We now spliced the b -dimension to regions of width 20, thus excluding the optimal value of b^* for most of the resulting feasible regions. These regions are constructed both above and below the optimal value, thus again forcing the algorithm to converge to the bounds B_{min} and B_{max} respectively.

Table 6.3 illustrates that the expected boundary solutions are indeed reached by our minimization algorithm. For ranges of b which lie below the optimal value of $b^* = 108.031$, the solution converges to the upper bound B_{max} of the feasible region for b . For ranges of b which lie above the optimal value b^* the solution converges to the lower bound B_{min} . The buffer / bandwidth tradeoff is again apparent. As the network forces the user to demand more buffer than b^* , the service rate required to satisfy the loss probability constraint decreases. The cost function is again higher for all regions which exclude the optimal solution (b^*, c^*) .

Finally, we tested our algorithm for different degrees of self-similarity. In this case, we generated traffic streams with Hurst parameters $\mathcal{H} = \{0.65, 0.75, 0.85, 0.95\}$, mean arrival rate $\mu = 500 \frac{\text{cells}}{\text{sec}}$ and variance $\sigma^2 = 2$. We fixed the maximum tolerated loss probability at

Table 6.3: Minimization results for different feasible regions spliced in the buffer-dimension

B_{min}	B_{max}	c	b	$\mathcal{C}(\dots)$
40	60	621.01	60.00	389253
60	80	615.02	80.00	384649
80	100	610.58	100.00	382805
100	120	609.03	108.31	382650
120	140	607.08	120.00	382942
140	160	604.20	140.00	384662
160	180	601.78	160.00	387735

Table 6.4: Minimization results for input streams with different Hurst parameters for a target loss probability of 1.00×10^{-2} . The simulated loss probability is shown for comparison.

$Hurst$	c^*	b^*	$\mathcal{C}(\dots)$	$P(Q > b)_{sim}$
0.65	534.67	99.98	295870	8.90×10^{-5}
0.75	546.04	91.68	306565	7.20×10^{-5}
0.85	560.60	77.60	320290	5.20×10^{-5}
0.95	580.61	49.78	339586	3.00×10^{-5}

$\epsilon_{\mathcal{L}} = 1.00 \times 10^{-2}$ and the allowable ranges for b and c to $520 \leq c \leq 620$ and $10 \leq b \leq 500$ respectively. We used the same cost function as above and again applied the minimization algorithm DONLP2. The results for this experiment are shown in Table 6.4.

In this case, the optimal c^* and b^* are not at one of the boundary conditions B_{min} , B_{max} , C_{min} or C_{max} . This indicates that the solution represents the global minimum cost which satisfies the loss probability constraint (C1). Furthermore, the minimum cost solution increases with the Hurst parameter, indicating that traffic with a higher degree of self-similarity requires more resources from the network to guarantee a specified loss probability.

6.4 Minimum Cost Traffic Shaping under Effective Bandwidths

In this section we discuss the details of MCTS under the assumption that the user's output stream has to satisfy the constraints of effective bandwidth based CAC.

6.4.1 Framework Formulation

Recall that a newly arriving stream with effective bandwidth $\alpha_{j+1}^A(\theta)$ can be admitted to a queue with available capacity $C^* = C - \sum_{j=1}^J \alpha_j^A(\theta)$ (with J streams already admitted to this node) if

$$\alpha_{j+1}^A(\theta) \leq C^*$$

(see equation (3.32)). The queue then guarantees a QoS according to the effective bandwidth approximations for the *aggregate* stream. If QoS guarantees are desired for individual streams, then only statistically similar streams with similar QoS requirements may be multiplexed together. This could be provided by a PSN architecture with N queues per node, where N is the number of combinations of statistically different streams and QoS requirements.

The traffic shaping problem under effective bandwidth based CAC is almost the same as stated in the last section. In particular, the QoS constraints imposed by the user remain unchanged. The two CAC constraints introduced in section 6.2.3 are now replaced by a single constraint of the type (3.32), with $\alpha_{j+1}^A(\theta)$ replaced by $\alpha_{j+1}^D(\theta, c)$ (see equation (3.10)), derived as

$$\alpha^D(\theta, c) = \begin{cases} \alpha^A(\theta) & \text{if } \beta^A(\theta) < c \\ c - \frac{1}{\theta} \Lambda_A^*(c) & \text{otherwise} \end{cases} \quad (6.30)$$

In this way we impose the restriction that the output of the traffic regulator has to satisfy the CAC constraints of the network. Nevertheless, as compared to (σ, ρ) -constrained CAC, an important difference can be identified:

- Cell losses can now occur inside the network along the end-to-end path. Even though the stream may still be part of a VP, resources are not reserved such that *no* cell losses occur inside the network.

This difference is easily accounted for using the arguments of section 3.3.3. The end-to-end loss requirement can be ‘distributed’ over the $H + 1$ nodes (including the traffic regulator) along the end-to-end path, giving us a loss requirement per node of $\epsilon_L/(H + 1)$. Similarly, the delay requirement can be adjusted to allow each node to delay the packet / cell by a portion of the end-to-end threshold δ . For example, δ may be uniformly allocated to the $H + 1$ nodes, giving each node a delay threshold of $\delta/(H + 1)$. Finally, the jitter requirement was seen to be of the order of magnitude of a single node delay. If we require a threshold of γ at each node then the jitter constraint along the entire end-to-end path would be satisfied. If we assume that the user is charged linearly according to the buffer / bandwidth resources of the traffic regulator, the MCTS problem under effective bandwidth CAC is

$$\text{Min } p_1 b + p_2 c \quad (\text{OBJ})$$

$$\text{s.t. } e^{-\alpha^{-1,A}(c)b} \leq \frac{\epsilon_L}{H+1} \quad (\text{C1})$$

$$e^{-\alpha^{-1,A}(c)\frac{\delta}{H+1}} \leq \epsilon_D \quad (\text{C2})$$

$$e^{-\alpha^{-1,A}(c)\gamma} \leq \epsilon_J \quad (\text{C3})$$

$$\alpha^D(\theta^{NW}, c) \leq C^* \quad (\text{C4})$$

where $C^* = C - \sum_{j=1}^J \alpha_j(\theta^{NW})$, θ^{NW} indicates the QoS provided by the network, and where $\alpha^D(\cdot)$ represents the output effective bandwidth of the traffic regulator of the user’s stream.

This formulation of MCTS is solvable as outlined in section 6.3.2. Constraint (C4) with θ^{NW} given by the network is a function of c only, and so no new additional free variables are introduced. Specifically, let us ignore constraints (C2) and (C3) and consider first the case where $C^* > \alpha^A(\theta^{NW})$ in a bounded arrival process with peak rate π . In this case constraint (C4) is never binding, as shown in the following lemma.

Lemma 6.1 *Consider a bounded arrival process with effective bandwidth $\alpha^A(\theta)$, mean rate μ and peak rate π . If $C^* > \alpha^A(\theta^{NW})$ in the above MCTS formulation then constraint (C4) is never binding.*

Proof: In the first part of the proof we first show that for any $\mu < c < \pi$ we have

$$\alpha(\theta) \geq c - \frac{1}{\theta} \Lambda^*(c) \quad (6.31)$$

Note that this implies that the output effective bandwidth is greater if $\beta^A(\theta) < c$ for any parameter c of the traffic regulator. This can easily be shown by the following argument

$$\begin{aligned} c - \frac{1}{\theta} \Lambda^*(c) &\leq \alpha(\theta) \\ \Leftrightarrow \theta c - \Lambda^*(c) &\leq \Lambda(\theta) \\ \Leftrightarrow \theta c - \Lambda(\theta) &\leq \Lambda^*(c) \\ \Leftrightarrow \theta c - \Lambda(\theta) &\leq \sup_{\theta} [\theta c - \Lambda(\theta)] \end{aligned}$$

Therefore, since $C^* > \alpha^A(\theta^{NW})$ by assumption, we also have $C^* > c - \Lambda^*(\theta^{NW})/\theta^{NW}$.

In other words, irrespective of the rate of the shaper c , the output effective bandwidth $\alpha^D(\theta^{NW}, c) \leq C^*$. $\square$

Otherwise, (C4) may be binding, in which case the rate of the shaper has to be set such that the output stream is admitted. The minimum cost value c^* is then the solution to $C^* = c - \Lambda^*(c)/\theta^{NW}$. If constraints (C2) or (C3) are binding, then (C4) is not binding and so it does not contribute to the solution.

A drawback in the above formulation is that the user is not charged for the resources provided *inside* the network to deliver the QoS guarantees. The cost function is again only dependent on the shaper's parameterization. While we have provided reasons for such a cost function in the last section, this may not be desirable under effective bandwidth based CAC. In particular, the network may enjoy a higher statistical multiplexing gain ($\kappa \ll 1$) and be able to admit more streams than under (σ, ρ) -constrained CAC. In the current formulation, each user is nevertheless charged *as if* the resources were reserved. The advantage of statistical multiplexing are thus not fully passed on to the user in the cost function. The following section proposes a solution via the notion of a *virtual dedicated queue*.

6.4.2 Charging for Resources under Effective Bandwidths

The problem addressed in this section is how to relate the effective bandwidth to a particular pair (b^D, c^D) of buffer / bandwidth reflecting its resource requirement, under the assumption that the network enjoys high statistical multiplexing gains. This would allow the user to be charged by the amount of resources attributed to the stream, not as if the resources were reserved. Such a charge should be less than the equivalent reservation systems because of multiplexing. In particular, if (b, c) reflects the resource usage with reservation, then we expect $\mathcal{C}(b^D, c^D) < \mathcal{C}(b, c)$. We do this by incorporating the parameter κ from the large buffer effective bandwidth approximations. An alternative approach would be for the network to set the prices for the resources as a function of κ (e.g. lower the resource prices if $\kappa \ll 1$). Such an approach is not discussed in this section.

This problem relates to the fact that the effective bandwidth of a stream is not directly related to a buffer / bandwidth requirement inside the network. The effective bandwidth *does* capture the burstiness of a stream since it is typically above the mean rate of the stream⁸. However, this increase in the bandwidth requirement is not expressed in terms of buffering.

We propose a solution based on a *virtual dedicated queue* to translate the effective bandwidth requirement into a buffer / bandwidth requirement. In this model the user's stream is cloned into a virtual single server queue which serves *only* the user's stream. Yet, it incorporates the statistical multiplexing gain of the actual queue by using the full effective bandwidth approximation (including the constant κ). The buffer size and bandwidth capacity of this virtual dedicated queue are then determined by the solution to a MCTS optimization. Figure 6.10 illustrates this concept. We assume that the buffer and bandwidth capacity of the network queue are fixed at B and C respectively. We also assume that the network knows or sets the parameter κ . We denote the QoS parameter for this queue by θ^{NW} . The bandwidth and buffer capacities for the virtual dedicated queue on the other hand are variable and denoted by b^D and c^D . This queue's QoS parameter is denoted by $\tilde{\theta}$. Note that the user's arrival stream is directly fed into either queue, without being

⁸The more bursty the arrival stream, the higher its effective bandwidth to reflect the additional bandwidth capacity required to accommodate the burstiness such that a QoS guarantee can still be made.

Figure 6.10: Virtual dedicated queue model.

regulated. We restrict ourselves to a loss probability constraint only for now to facilitate the presentation of our argument.

Recall from section 3.4 that the statistical multiplexing parameter κ can be incorporated into the CAC decision (see the argument in that section, in particular equation 3.34).

$$\alpha_{j+1}^A(\tilde{\theta} + (\log \kappa)/B) \leq C - \sum_{j=1}^J \alpha_j^A(\tilde{\theta} + (\log \kappa)/B) \quad (6.32)$$

We can thus view $\alpha_{j+1}^A(\tilde{\theta} + (\log \kappa)/B)$ as the effective bandwidth for the user's stream, i.e.

$$\alpha_{j+1}^A(\tilde{\theta} + (\log \kappa)/B) = c^D \quad (6.33)$$

Since $\kappa \leq 1$ we have $\alpha_{j+1}^A(\tilde{\theta} + (\log \kappa)/B) \leq \alpha_{j+1}^A(\tilde{\theta})$ and so the additional κ -term implies that the effective bandwidth attributed to the stream is *smaller* with $\kappa \ll 1$ than in the case where $\kappa = 1$.

Consider now the virtual dedicated queue. With a single stream feeding this queue, the loss probability is approximated by $e^{-\tilde{\theta}b^D}$. From 6.33 we have

$$\tilde{\theta} = \alpha_{j+1}^{-1,A}(c^D) - \frac{\log \kappa}{B}$$

and so

$$P(Q > b^D) \approx e^{-(\alpha_{j+1}^{-1,A}(c^D) - \frac{\log \kappa}{B})b^D} \quad (6.34)$$

where Q here indicates the occupancy of the virtual dedicated queue. In this equation the buffer / bandwidth trade-off is again apparent. We will use it as a basis for charging

Figure 6.11: Effect of κ on the shape of the loss constraint.

the user. To summarize, the higher statistical multiplexing gain enjoyed by the network through the parameter κ can be passed on to the user by shifting the QoS parameter $\tilde{\theta}$ (for the user) downwards by $\log \kappa / B$. This implies that the resource requirement is smaller than in the case where $\kappa = 1$. Figure 6.11 illustrates the buffer / bandwidth trade-off for the user under the virtual dedicated queue model for different values for κ .

Based on the above derivation we propose a modified charging scheme. The following MCTS formulation determines the minimum cost parameters for the virtual dedicated queue, taking the higher statistical multiplexing gain κ into account. We now include loss, delay and jitter constraints.

$$\text{Min } p_3 b^D + p_4 c^D \quad (\text{OBJ})$$

$$\text{s.t. } e^{-(\alpha^{-1} A(c^D) - \frac{\log \kappa}{B}) b^D} \leq \epsilon_{\mathcal{L}} \quad (\text{C1})$$

$$e^{-(\alpha^{-1} A(c^D) - \frac{\log \kappa}{B}) c^D \delta} \leq \epsilon_{\mathcal{D}} \quad (\text{C2})$$

$$e^{-(\alpha^{-1} A(c^D) - \frac{\log \kappa}{B}) c^D \gamma} \leq \epsilon_{\mathcal{J}} \quad (\text{C3})$$

The objective function represents the cost function for the resources determined via the virtual dedicated queue model. Constraints (C1), (C2) and (C3) represent the QoS constraints, which discount the resource requirements according to the statistical multiplexing factor κ . Note that we have omitted the CAC constraint. The above optimization is purely used for charging purposes. The stream would be admitted if equation 6.32 holds. Note that we have restricted ourselves to a single network queue. The above formulation does

not assume a virtual path with H hops.

Recall that the MCTS formulation in the last section focused only on the traffic regulator. In contrast, this formulation does not assume that the user's stream is regulated. We can obtain a complete MCTS formulation by combining the MCTS formulation in section 6.4.1 with the above formulation:

$$\text{Min } p_1 b + p_2 c + p_3 b^D + p_4 c^D \quad (\text{OBJ})$$

$$\text{s.t. } e^{-\alpha^{-1}(c)b} \leq \frac{\epsilon c}{H+1} \quad (\text{C1})$$

$$e^{-\alpha^{-1}(c)c \frac{\gamma}{H+1}} \leq \epsilon_D \quad (\text{C2})$$

$$e^{-\alpha^{-1}(c)c^\gamma} \leq \epsilon_J \quad (\text{C3})$$

$$e^{-(\alpha^{-1,D}(c^D, c) - \frac{\log \kappa}{B})b^D} \leq \frac{\epsilon c}{H+1} \quad (\text{C4})$$

$$e^{-(\alpha^{-1,D}(c^D, c) - \frac{\log \kappa}{B})c^D \frac{\gamma}{H+1}} \leq \epsilon_D \quad (\text{C5})$$

$$e^{-(\alpha^{-1,D}(c^D, c) - \frac{\log \kappa}{B})c^D \gamma} \leq \epsilon_J \quad (\text{C6})$$

$$\alpha^D(\theta^{NW}, c) \leq C^* \quad (\text{C7})$$

In this case, the user is charged for the resources required for shaping and for the resources attributed to the output stream of the regulator. By including prices for the regulator, the above formulation results in MCTS parameters. Note that constraints (C4), (C5) and (C6) now involve the output stream of the traffic regulator $\alpha^{-1,D}(c^D, c)$, instead of $\alpha^{-1,1}(c^D)$. We have also included a CAC constraint to ensure that the parameterization for the traffic shaper results in an output stream which can be admitted to the network with QoS parameter θ^{NW} . The following section illustrates how this formulation can be solved in the case of Gaussian traffic streams.

6.4.3 Example: Gaussian Sources

To demonstrate MCTS under effective bandwidth based CAC in an example, we assume that the user's arrival stream has a Gaussian distribution with mean μ and variance $\sigma^2 < \infty$. The user's stream is again shaped by a G/D/1/b queue with service rate c , where (b, c) are to be determined. The output stream of the shaper is multiplexed inside the network, requiring H hops to reach its destination. We represent the network using a single server queue and assume that the statistical multiplexing gain is known to be κ . We assume that

the network charges the user for both the parameterization of the traffic shaper as well as a share of the resource capacity of the network. Finally, we assume that the user demands QoS in terms of losses only, represented by parameter $\epsilon_{\mathcal{L}}$.

The effective bandwidth, decoupling bandwidth and the output effective bandwidth of the arrival stream are given by

$$\alpha^A(\theta) = \mu + \frac{\theta\sigma^2}{2} \quad (6.35)$$

$$\beta^A(\theta) = \mu + \theta\sigma^2 \quad (6.36)$$

$$\alpha^D(\theta, c) = \begin{cases} \alpha^A(\theta) & \text{if } \beta^A(\theta) < c \\ c - \frac{(c-\mu)^2}{2\sigma^2\theta} & \text{otherwise} \end{cases} \quad (6.37)$$

where c again represents the service rate of the shaper. The inverses of the effective bandwidth and the output effective bandwidth are computed as

$$\alpha^{-1,A}(c) = \frac{2(c-\mu)}{\sigma^2} \quad (6.38)$$

$$\alpha^{-1,D}(c^D, c) = \begin{cases} \frac{2(c-\mu)}{\sigma^2} & \text{if } \frac{c-\mu}{2} \geq c^D \\ \frac{(c-\mu)^2}{2\sigma^2(c-c^D)} & \text{otherwise} \end{cases} \quad (6.39)$$

The complete MCTS formulation in this case is

$$\text{Min } p_1b + p_2c + p_3b^D + p_4c^D \quad (\text{OBJ})$$

$$\text{s.t. } e^{-\frac{2(c-\mu)}{\sigma^2}b} \leq \frac{\epsilon_{\mathcal{L}}}{H+1} \quad (\text{C1})$$

$$e^{-(\alpha^{-1,D}(c^D, c) - \frac{\log \kappa}{B})b^D} \leq \frac{\epsilon_{\mathcal{L}}}{H+1} \quad (\text{C2})$$

$$\alpha^D(\theta^{NW}, c) \leq C^* \quad (\text{C3})$$

and its Lagrangian $\mathcal{F}$ is

$$\begin{aligned} \mathcal{F} &= p_1b + p_2c + p_3b^D + p_4c^D \\ &- v_1\left(\frac{2(c-\mu)}{\sigma^2}b + \log \epsilon_{\mathcal{L}}\right) \\ &- v_2\left((\alpha^{-1,D}(c^D, c) - \frac{\log \kappa}{B})b^D + \log(\epsilon_{\mathcal{L}}/(H+1))\right) \\ &- v_3(C^* - \alpha^D(\theta^{NW}, c)) \end{aligned}$$

Table 6.5: MCTS parameters under Gaussian traffic for EB based CAC

Number	p_1, p_2, p_3, p_4	b	c	b^D	c^D	$\mathcal{C}$	Run Time (sec.)
1	1.1.1.1	2.92	5.46	0.00	5.46	13.84	0.05
2	4.4.1.1	2.30	5.84	0.00	5.84	38.48	0.03
3	1.1.4.4	4.61	4.92	0.00	4.92	29.27	0.04
4	1.1.1.4	4.61	4.92	0.00	4.92	29.27	0.04
5	1.1.1.20	9.46	4.45	0.00	4.45	102.94	0.04
6	4.1.1.4	1.03	8.13	4.13	5.03	36.54	0.04

The solution of this problem using the Lagrange multiplier approach requires the computation of $\partial\alpha^D(\cdot)/\partial c$, $\partial\alpha^{-1,D}(\cdot)/\partial c$ and $\partial\alpha^{-1,D}(\cdot)/\partial c^D$. The calculation of these derivatives is conditional on the domains of the respective variables (depending on the *if*-clause), as is the case in (6.39) or (6.37) for example. This complicates the Lagrange approach such that the solution may have to be found by dividing the original problem into several sub-problems, one for each *if*-clause. Alternatively, a numerical solution can be found using DONLP2.

Table 6.5 shows the results from numerous numerical evaluations of this problem. We assumed that the users input stream is distributed according to a Gaussian distribution with mean $\mu = 4$ and variance $\sigma^2 = 0.7$. This stream is first fed into a traffic regulator, and then fed into a FCFS queue representing the network. The network queue is assumed to have buffer $B = 1000$, QoS parameter $\theta^{NW} = 0.001$, statistical multiplexing parameter $\kappa = 0.01$, and an available bandwidth capacity that ensures admission. We restrict ourselves to loss constraints only. A representative MCTS problem with these numerical values in AMPL format is shown in appendix C.

Row 1 represents the case where all resources are priced equally. Notice that no charge is imposed for the network buffer. In all cases where the rate of the traffic regulator is ‘small’ enough, the cost function does not induce the network to charge for buffer instead of bandwidth. Rows 2 and 3 represent the case where the regulator’s resources and the network

resources are charged more respectively. Higher prices for the regulator lead to a smaller regulator buffer. The bandwidth of the regulator however increases as compared to row 1. Furthermore, this bandwidth is directly charged for inside the network too. Similarly, cheaper prices for the regulator lead to a larger regulator buffer allocation and corresponding smaller bandwidth allocation. The bandwidth of the regulator is again matched inside the network. Rows 4 and 5 set prices to induce the user to provision more resources by charging more for the network bandwidth. In the case of row 4, the results are identical to row 3 because no network buffer is charged for. However, increasing the price for network bandwidth to 20 shows how bandwidth is substituted by *buffer at the regulator*. Instead of charging more for the network buffer as a result of the higher price for network bandwidth, the stream is buffered at the regulator. Row 6 shows the effect of charging more for both the regulator buffer and the network bandwidth. In this case, the rate of the regulator is increased and the network now starts to charge for buffer too.

The above table also lists the execution run times of each MCTS problem on the NEOS server⁹ (see <http://www-neos.mcs.anl.gov/neos>). In all cases the solution was obtained in less than 50 ms.

We repeated the above experiment assuming the user's traffic follows an on-off source with the same mean rate. However, in this case we parameterized the source to be very bursty (on-phase is exponentially distributed with mean 9, off-phase is exponentially distributed with mean 1, and the rate during the on-phase was set to 40). Table 6.6 presents the results in this case.

As in the case for Gaussian sources, an increase in the price for network bandwidth results in a higher buffer allocation for the traffic shaper. Notice also that as the price for bandwidth rises, the allocated rate approaches the mean rate of the source. Like before, charging for both regulator buffer and network bandwidth increases the bandwidth allocation for the regulator. Since the regulator output is now higher, the network also starts to charge for buffer.

The above experiments demonstrate the full characterization of the MCTS problem

⁹The NEOS server is a DEC Alpha 500/500 (500MHz 21164 EV36 chip, 512 MB, DEC-Unix-4.0E). The reported execution times should thus be interpreted as lower bounds to be conservative.

Table 6.6: MCTS parameters for on-off traffic under EB based CAC

Number	p_1, p_2, p_3, p_4	b	c	b^D	c^D	$\mathcal{C}$	Run Time (sec.)
1	1.1.1.1	32.43	19.01	0.00	10.81	110.62	0.04
2	1.1.1.4	56.50	10.82	0.00	10.81	110.62	0.02
3	1.1.1.20	94.38	6.98	0.00	6.98	240.99	0.02
3	1.1.1.200	225.36	4.93	0.00	4.93	1217.89	0.02
4	4.1.1.4	8.43	39.40	34.79	16.64	144.16	0.03

under the assumption that the network charges for resources inside the network according to the proposed virtual dedicated queue model. The next section introduces a pricing scheme related to MCTS. We also examine the cost functions in more detail.

6.5 QoS Pricing Scheme

In the above discussion we have presented our MCTS-framework to determine the resources required to provision a user's stream such that the demanded QoS is satisfied. The notion of resource costs is used in a *generic* sense to manage the buffer bandwidth tradeoff. However, we have already interpreted this cost in a monetary sense. The user is charged the value of the cost function, determined by the resource prices set by the network and the resources required determined as the solution to the MCTS problem. In this section we discuss such a pricing scheme in detail. As a consequence of our monetary interpretation, we perceive the user as being *human* in this section¹⁰, and we assume that human users are not responsive to price changes in time frames of the order of magnitude of minutes or less. Technically, the prices could vary over short time intervals to reflect the current resource availability. We envisage that prices may be changed in time intervals in the order of minutes or hours. The end user may receive information about the expected charge at different times. At the time

¹⁰As opposed to software agents who may act for the human user.

of connection establishment, the actual charges could be communicated by the network. Price fluctuations over shorter intervals are only undesirable because of the human user's desire for price stability.

The pricing scheme proposed in this thesis is to charge the user according to the resources required to satisfy the demanded QoS. In general, the total charge to the user can be expressed by

$$\mathcal{C}(b^*, c^*, \alpha(.))t \quad (6.40)$$

where (b^*, c^*) represent the buffer / bandwidth parameters for the traffic regulator, t represents the duration of the connection and where $\alpha(.)$ represents the effective bandwidth of the user's stream. Such a cost function has a number of desirable properties: the total charge

- is directly related to resources associated with the user's stream
- is minimal to provide the demanded QoS when combined with MCTS
- can be calculated at the edge of the network
- can be foreseen by the user
- can be adapted for receiver-based charging

We will discuss each of these properties in turn.

Regarding the first property, the cost function (6.40) indicates that the user is charged according to the resource allocation of the traffic shaper per time unit. The traffic regulator's parameters are indicative of the resources required inside the network to satisfy the user's QoS requirements. Expressed in terms of (σ, ρ) -constraints, if the traffic regulator is parameterized by (b, c) then the output stream is (b, c) -constrained. Note that the *actual* resource requirement inside the network may be smaller than (b, c) . For example, under (σ, ρ) -constrained CAC, if the user's stream is shaped by a $G/D/1/b$ queue with service rate c , the output stream is $(1, c)$ -constrained. Similarly for effective bandwidth based CAC,

the actual resources required may be less due to statistical multiplexing gains inside the network. However, we still consider such a charge as adequate, because the network may

1. provide the initial shaping resources for the user
2. experience clumping effects of the user's stream increasing its burstiness
3. have to re-shape the user's stream inside the network periodically

As far as the second property is concerned, we envisage that users will want to minimize the cost of their connection such that the QoS required for their streams are still satisfied. This is precisely the problem addressed in MCTS. The cost function then emphasizes the relation between the charges and the QoS demanded by the user. However, we only recommend that users employ this framework, without making it a hard requirement. Users may parameterize the traffic regulator in a different way, for example such that their *utility* is maximized. In this case, the resources would be allocated such that the user's utility is maximized subject to the constraints that the stream has to be admitted to the network and that the QoS constraints are met. In case of (σ, ρ) -constrained CAC where the resources corresponding to the output stream are reserved, users may even demand capacities in excess of their actual requirements. In this case, they are also charged excessively. This seems fair since reserved resources are unavailable to other users by the CAC function.

Thirdly, the above charging function has the desirable property that it can be computed at the edge of the network. We view it as an example of Shenker's edge pricing scheme discussed in section 5.6. The charges are entirely calculated at the user-network interface (UNI) or the network-network interface (NNI). Nodes in the interior of the network are not involved in determining the total charge. They do not need to keep measurements of each user's stream. This greatly facilitates accounting and billing. Once the traffic regulator's parameters are determined, the accounting and billing functions only need to measure the duration of the connection. Our QoS pricing scheme thus introduces minimal computational overhead for accounting and billing.

Regarding the fourth point, our pricing scheme also has the desirable property that the user can form a reasonable expectation of the charges incurred by the connection at

the time the traffic contract is established. In this sense our scheme is compatible with Clark's expected capacity pricing, without requiring long-term traffic contracts. Given the expected duration of the connection t , the total cost to the user can immediately be determined. Unlike some of the pricing schemes discussed in the last chapter (e.g. smart-market pricing), the cost to the user is not dependent on the state of congestion inside the network. We view this as an advantage, since users are not surprised by unexpected bills. Itemized billing can easily be verified by the user and possibly compared to the user's log file.

Finally, the proposed pricing scheme is also amenable to receiver-based charging. In this case the benefit of the stream is associated with the receiver, who should therefore take responsibility for the charges incurred. The receiver would determine the sender's parameters for the traffic regulator and communicate the results back to the sender. The sender would then parameterize the device accordingly. In this case, in addition to the network contributing the resource prices to the receiver, the sender would have to communicate the characteristics of the traffic stream. Alternatively, the receiver could transmit a request for a stream to the sender along with the QoS requirements and budget constraint. The sender could then perform the parameterization of the traffic regulator using MCTS.

6.5.1 Alternative Cost Functions for MCTS

In the discussion of MCTS in sections 6.2 and 6.4 we have also proposed to charge the user for the actual resources attributed to the stream *inside* the network. In the case of effective bandwidth based CAC, we proposed to determine these resources according to a virtual dedicated queue model. In general, the cost function can be related to two issues:

1. recovering the cost of the resources provided by the network and
2. influencing the user's transmission behavior

Consider the first issue: using $\mathcal{C}(b^D, c^D) = p_3 b^D + p_4 c^D$ as a cost function recovers the cost of the resources attributed to the stream inside the network. However, the prices p_3 and p_4 are unable to influence the user's parameterization of the traffic shaper, as indicated in

the example in section 6.4.2. If the cost function is based solely on the resources attributed to the stream inside the network, the user is induced to parameterize the shaper according to the stream's mean rate and the shaper's buffer size according to the maximum burst size of the stream (assuming that the delay and jitter constraints are met). In this way, the output stream of the regulator is smoothed to the largest extent. The resources attributed to the stream inside the network are substituted by the traffic regulator's resources as much as possible. Consequently the network is unable to influence the user's output stream of the traffic regulator using pricing. In contrast, the QoS pricing scheme above allows the network to influence the user's parameterization. While the resulting charges do not capture the resources attributed to the stream inside the network accurately, they nevertheless are a good proxy for the reasons outlined in the last section. Thus our QoS pricing scheme is both a good proxy for the resource requirement of the stream to provide the demanded QoS to the user *and* a control mechanism for the network to induce the user to trade-off buffer and bandwidth resources.

Furthermore, to accurately recover the cost for resources inside the network, prices (b^D, c^D) would have to be attributed to every node along the end-to-end path. If we consider effective bandwidth based CAC for example, each node can multiplex a number of streams, each with possibly different destinations. The multiplexing gain and charging scheme outlined in the section 6.4.2 differ for each node. A complete cost function would then be

$$\mathcal{C}(b_h^D, c_h^D) = \sum_{h=1}^H p_3^h b_h^D + p_4^h c_h^D : h = 1 \dots H$$

where we have indicated that the output stream between nodes may also differ between each hop by b_h^D and c_h^D , $h = 1 \dots H$. Such a cost-recovering function would eliminate most of the properties stated in the last section. It would not be computable at the edge, require measurements of statistical multiplexing gains and individual streams inside the network and may even prevent the user from calculating an expected bill.

With this in mind, even a cost function of the form (6.8) is inappropriate. In this case the user is charged for both the traffic shaping parameters and the resources attributed to the stream inside the network using the virtual dedicated queue model for example. Such a

cost function clearly mixes the issues of cost recovery and controlling the user's transmission behavior. Prices p_1 and p_2 are used for control, whereas prices p_3 and p_4 are used for cost recovery. We believe that such a cost function is hard to justify for the user. Furthermore, as outlined in section 6.4.3, the resulting MCTS problem is hard to compute even for the case of known Gaussian traffic sources. Solutions based on numerical methods would prevent a solution in real-time.

Cost functions of the higher order polynomials or exponential form on the other hand may be used in our MCTS scheme. In case of high resource utilization, the cost function could incorporate an element of a prohibitive cost [158] by letting the cost function for the resources increase as utilization rises. A justification for the network of such cost functions would be that as resources inside the network become scarce, their implicit value rises. As resource utilization rises, more and more users would not be able to satisfy their budget constraints and the resources would be allocated to those users who are willing and able to pay the higher price. This would increase the economic efficiency of the network. Users with 'urgent' connections can still access the network. For the network, such a cost function implies higher revenues. The MCTS scheme is then still valid, provided the new cost function is still differentiable and quasi-concave. For example, the network may impose an exponential charging function per time unit of the form

$$\mathcal{C}(b, c) = (p_1 e^b + p_2 e^c) t \quad (6.41)$$

or a hyperbolic charging function

$$\mathcal{C}(b, c) = \left(-\frac{p_1}{b - B^*} - \frac{p_1}{c - C^*} \right) t \quad (6.42)$$

where (B^*, C^*) represent the total available buffer / bandwidth capacities inside the network. Figure 6.12 illustrates these functions. As a disadvantage, the total cost to the user may again be dependent on the state of the network (as in equation (6.42)) and a human user would not be able to form reasonable expectations on the resulting charges.

The above arguments lead us to propose a linear cost function

Figure 6.12: Exponential or hyperbolic cost functions.

$$\mathcal{C}(b, c) = (p_1 b + p_2 c)t \quad (6.43)$$

as the default for our MCTS problem. As outlined in section 6.2, the form of the feasible region based on the effective bandwidth approximation is such that linear cost functions suffice to enable a unique solution. This also reduces the time required to solve the MCTS problem, possibly to real-time dimensions. Furthermore, it is simpler than cost functions with higher polynomials, yet allows the network to influence the user's buffer / bandwidth tradeoff. Justification of such a cost function to the human user would also be simplified. A disadvantage of such a cost function would be that the network retains all the benefit from statistical multiplexing. However, in a competitive environment between service providers, the statistical multiplexing gain could be passed to the user by adjusting the resource prices p_1 and p_2 . A provider with higher multiplexing gain could charge less than a provider with little statistical multiplexing gain. If all users and service providers would employ MCTS, competition would then be based on resource prices.

6.5.2 Comparison with other Pricing Schemes

As indicated in the above, the proposed QoS pricing scheme based on MCTS is compatible with both expected capacity pricing and edge pricing. However, the proposed QoS pricing scheme also differs in that we base it on MCTS. Edge pricing as described in section 5.6 is predominantly a conceptual contribution. No details are specified about the implementation. Furthermore, edge pricing does not stress the relationship between charges and QoS.

In contrast, our pricing scheme provides details on how such a scheme could be realized. Expected capacity pricing assumes long-term contracts between the user and the service provider. The resources provisioned for a user during congestion periods are based on such long-term expectations. The QoS pricing scheme proposed here does not make such a requirement. Technically, we allow charges to be applied at much smaller time scales. The traffic characterizations in our case are assumed to be known or measured by the user. Rather than forming expectations about the required capacity, users in our case are only assumed to form expectations about the traffic type.

Our QoS pricing scheme differs significantly from priority pricing or smart-market pricing. Under the latter schemes, each packet is priced according to its priority level or given a bid. However, there is no guarantee that a packet will be transmitted during congestion periods. Priority and smart-market pricing only gives the user a mechanism to indicate and pay for a higher priority mechanism. Rich users can choose to pay more in order to increase the *relative* priority of their traffic. The entire network has to support priorities or an auctioning mechanism, and each node has to maintain records for each user for accounting and billing. In contrast, in our QoS pricing scheme the emphasis is put on charging for QoS, based on either (σ, ρ) -constrained or effective bandwidth based CAC. In both cases, QoS guarantees are given by restricting access to the network during times of congestion. With a linear cost function based on MCTS users can not gain a relative advantage over each other during times of congestion. Network access is on a FCFS basis and all admitted users are guaranteed their respective QoS. However, with a hyperbolic cost function, for example, the network could increase the prices as resources become scarce. This would favor ‘rich’ users who are willing to pay for their transmission during congestion periods.

Our proposed pricing scheme differs less from effective bandwidth pricing, where the user is charged according to a linear function placed tangent to the effective bandwidth of a worst-case on-off source at the measured mean rate of the user’s stream. This pricing scheme has the desirable effect that users are induced to declare their true mean rate (the peak rate of the traffic is assumed to be known). However, this pricing scheme implies that the user is also charged for the QoS provided by the network, as indicated by θ^{NW} in our discussion. This is not apparent in the discussion of effective bandwidth pricing in section

5.9, in particular not in figure 5.4 which shows $\alpha(m)$, where m indicates the measured mean rate. Since $\alpha(\theta^{NW})$ is typically above the mean arrival rate of the traffic stream it takes the stream's burstiness into account. Our MCTS scheme translates this burstiness into a buffer / bandwidth requirement (b, c) to determine the regulators parameters. The proposed QoS pricing scheme then charges the user for these parameters. The effective bandwidth pricing scheme only charges for bandwidth required in the network to provide the QoS θ^{NW} . The charges thus differ in that our scheme charges for the traffic regulator's parameters, whereas Kelly's scheme charges for θ^{NW} assuming a worst-case on-off source. However, in both cases QoS guarantees are given to the user and effective bandwidth based CAC is assumed.

Chapter 7

Measuring Effective Bandwidths

The MCTS framework introduced in the last chapter made extensive use of the effective bandwidth of the user's arrival stream. In some cases such a characterization may be available. For example, we envisage researchers and in particular application developers establishing traffic models for traffic generated by certain applications. Analytical forms for effective bandwidths for many classes of traffic models have been calculated [97]. However, applying definition (3.1) in a practical environment is not trivial. It requires a full characterization of the underlying process, since $\alpha(\theta, t)$ is a function of the moment generating function. The effective bandwidth may be modeled parametrically, but this approach introduces the following problems.

Parametric modeling requires first a determination of the analytical form of the source's effective bandwidth. The traffic stream may then be used to estimate the parameters. Only then is the stream fully characterized by the analytical form of the EB. Both procedural steps introduce approximations leading to inefficient resource allocation. Since equation (3.15) is an upper bound for the probability of the workload of an infinite queue crossing a threshold B , allocation schemes based on analytical EBs are already conservative. Estimating the parameters of the analytical form for the EB introduces approximations. Using parametric models of the source's EB therefore only compounds the over-allocation of available resources.

Furthermore, a given trace is only a finite time realization of a stochastic process. In

such a trace we may not necessarily observe all the possible values of the underlying process, particularly the theoretical peak value. Using analytical EBs leads to a resource allocation where such values are taken into account, despite the fact that they may not be realized in the given trace. The amount of resources allocated may consequently exceed the actual resource requirements for the given trace, compounding the inefficiencies arising from parametric modeling itself.

The above discussion motivates the importance of *measuring* the effective bandwidth of a source (either by application developers or by users directly). In this chapter we analyze four different methods of measuring the effective bandwidth for given, finite time traffic traces: a *direct estimator*, a *block estimator*, an estimator based on the *Kullback-Leibler* distance and an estimator based on *linear regression*. We are particularly interested in studying their accuracy as compared to analytical results. For this reason, we restrict our analysis to three traces generated from traffic models where the analytical effective bandwidth is known (Poisson, on-off, Fractional Gaussian Noise). We point out possible assumptions which may restrict their use. Furthermore, we seek to understand the required parameter space. Some of the estimators require both a temporal and a spacial parameter. Others only require the spatial parameter. We intend to determine the range for these parameters where the measurements are sufficiently accurate. In this chapter we also place a particular emphasis on the *finiteness* of the traces. In any practical context, traffic can only be observed over a finite temporal window. The accuracy of asymptotically efficient estimators thus has to be investigated in this context.

We note here that the majority of the results presented in this chapter were developed jointly with Sandra Tartarelli, University of Pisa, during her research term at Carleton University (with the exception of section 7.8).

7.1 Effective Bandwidth Estimators

The four estimators can be classified by the underlying method. The direct- and block estimator are derived from equation (3.1). The estimators based on the Kullback-Leibler distance and linear regression on the other hand are based on the large buffer effective

bandwidth approximation (3.15). They thus measure the decay rate of the queuing process arising from feeding the arrival stream A into a single-server queue with deterministic service rate c . The next section provides a brief overview of these estimation methods. We also point out any assumptions which may limit their applicability. In the subsequent section we then describe the traces used in our analysis.

To differentiate effective bandwidth measurements from their analytical counterparts, we use the notation $\hat{\alpha}(\dots)$ for the former and $\alpha(\dots)$ for the latter. We also use the terms “measure” and “estimate” synonymously.

7.1.1 The direct estimator

Obviously, the effective bandwidth can be estimated directly using equation (3.1) as shown by Gibbens [62]. However, instead of considering the probabilistic expectation of the underlying process in equation (3.15), the temporal average is applied. The estimator takes the form

$$\hat{\alpha}(\theta, t) = \frac{1}{\theta t} \log \frac{1}{t_N - t} \int_0^{t_N - t} e^{\theta A(\tau, \tau + t)} d\tau \quad (7.1)$$

where

$$A(\tau, \tau + t) = \sum_{i=1}^N A_i 1(\tau \leq t_i \leq \tau + t) \quad (7.2)$$

for $0 \leq \tau \leq t_N - t$ and t_N indicating the trace length expressed in time units. We thus replace the expectation in equation (3.1) by the sample mean of the arrivals measured over a sliding window of length t . Since the direct estimator is based on the underlying definition of the analytical effective bandwidth, it is not restricted by any additional assumptions. This estimator is applicable wherever the analytical effective bandwidth is defined.

Note that the integral in equation (7.1) may be hard to obtain numerically due to finite precision arithmetic. A large number of arrivals over the interval $(\tau, \tau + t)$ may lead to numerical overflows. The estimator is limited in both parameter spaces by such potential overflows.

7.1.2 The block estimator

A block estimator for the effective bandwidth has been proposed by [42, 83]. It takes the form

$$\hat{\alpha}(\theta, t) = \frac{1}{\theta t} \log \frac{t}{N} \sum_{i=1}^{N/t} e^{\theta \tilde{A}_i} \quad (7.3)$$

where N is the trace length and

$$\tilde{A}_i := \sum_{k=(i-1)t+1}^{it} A_k \quad (7.4)$$

for a given time t and for $i \leq \frac{N}{t}$. Note that this estimator is also based on equation (3.1). However, in contrast to the direct estimator, this estimator considers blocks of arrivals over an interval of length t . Thus, instead of using a sliding window of length t , the windows are non-overlapping. The estimator is based on the assumption that the block arrivals are realizations of independent and identically distributed (i.i.d.) random variables. Duffield et al. [42] show that the estimator is asymptotically unbiased in this case as the number of samples in the trace N tends to infinity. However, this assumption limits the applicability of the estimator. In particular, it prevents the estimator from being applied to a self-similar traffic stream.

Györfi et al. [72] derived confidence intervals for the effective bandwidth which are related to the direct estimator. Furthermore, their CIs are constructed without assumptions about the underlying distribution of the process. Under the assumption that $\{A\}$ is bounded (e.g. by passing the process through a traffic regulator) a $1 - e^{-q}$; $q > 0$: confidence interval for the effective bandwidth is given by

$$(\underline{\hat{\alpha}}(\theta, t), \overline{\hat{\alpha}}(\theta, t)) \quad (7.5)$$

where

$$\underline{\hat{\alpha}}(\theta, t) := \frac{1}{\theta t} \log \left(\left[\sum_{i=1}^N w(i, N) e^{\theta A(t, i)} - \epsilon(N) \right] \vee 1 \right)$$

$$\begin{aligned}\bar{\alpha}(\theta, t) &:= \frac{1}{\theta t} \log\left(\left[\sum_{i=1}^N w(i, N) e^{\theta A(t, i)} + \epsilon(N)\right] \wedge \pi\right) \\ \epsilon(N) &:= \left[\frac{q}{2}(e^{\theta t \pi} - 1)^2 \sum_{i=1}^N w(i, N)^2\right]^{\frac{1}{2}}\end{aligned}$$

Here $w(i, N)$ are weights such that $0 \leq w(i, N) \leq 1$ and $\sum_{i=1}^N w(i, N) = 1$ and $A(t, i)$ represents independent arrivals over an interval of length t , indexed by i . Note that this confidence interval requires independent measurements of the arrival process. $\epsilon(N)$ represents an error term which tends to zero as N becomes large. Its size also depends on the choice of weights $w(i, N)$, which may be chosen to be $w(i, N) = \frac{1}{N}$. Since obtaining independent samples for $A(t, i)$ may be difficult in practice, the arrival process may have to be measured for a significant period of time before the confidence interval becomes acceptable. This is one of the drawbacks of this approach.

7.1.3 The Kullback-Leibler Distance (KLD) estimator

The Kullback-Leibler distance (KLD) estimator [37] is the first of the two estimators based on a virtual buffer method. It is thus fundamentally different from the direct or the block estimators, in that it uses equation (3.16) instead of equation (3.1). If the arrival stream A is fed into a single server queue with service rate C , then the probability of the queue length exceeding a threshold B decays exponentially with rate θ . We can therefore estimate θ by observing the decay rate of the queue for a given service rate C . Thus, instead of estimating $\bar{\alpha}(\theta)$ for a given value of θ , the value of θ is estimated for a given value of $C = \alpha(\theta)$. In this case, the estimate for θ is obtained by minimizing the Kullback-Leibler distance¹ between the observed queue size distribution for a given buffer threshold B_1 and between $p(B) = e^{-\bar{\alpha}^{-1}(C)B}$. The Kullback-Leibler estimator for the value of θ corresponding to a given service rate C of the single server queue evaluates to

$$\hat{\theta} = \log\left(\frac{\sum_{b=B_1}^{\infty} \pi(b)}{\sum_{b=B_1}^{\infty} b\pi(b) - B_1 \sum_{b=B_1}^{\infty} \pi(b)}\right). \quad (7.6)$$

The effective bandwidth function is thus measured for various values of C between

¹The Kullback-Leibler distance is defined in general as $I(f, \hat{f}) := \int f(x) \log\left(\frac{f(x)}{\hat{f}(x)}\right) dx$.

the mean rate and the peak rate of the arrival stream². This implies that the mean and the peak values of the arrival stream are determined a priori. However, these values are typically easily obtained. As a result, the range of the values for the measured effective bandwidth $\hat{\alpha}(\theta)$ are known and need not be provided as parameters to the estimation function. Furthermore, since the values of θ are the result of the estimation, the domain of the effective bandwidth function is automatically determined. Note also that the KLD estimator for $\hat{\alpha}(\theta)$ involves no time parameter t . This results from using the effective bandwidth approximation, which is in turn based on the definition of the effective bandwidth given by (3.1). Details can be found in [97, 83]. The above KLD estimator is therefore only applicable for stationary, short-range dependent sources. For long-range dependent traffic, a curve-fitting technique may be applied using Duffield's modified exponential bound for an arrival sequence with Hurst parameter $\mathcal{H} > 0.5$ [40]

$$P(Q > B) \approx e^{-\theta B^2 - 2\mathcal{H}} \quad (7.7)$$

Using this result, the parameters B and $\mathcal{H}$ could be found using the KLD approach (as well as the coefficient to improve the approximation). Note also that the exponential bound is valid for large B , so the estimator (7.8) should be applied for a given threshold B_1 , i.e. the sums should be taken for $b > B_1$.

7.1.4 The linear regression (LR) estimator

The linear regression estimator is very similar to the KLD estimator. It is also based on the effective bandwidth approximation. As above, the decay rate of the queue size resulting from feeding the arrival stream into a single server queue with service rate C is obtained. Since the decay rate is exponential with rate θ , the log of the probability of the queue size exceeding a threshold B is linear. We can thus use linear regression to obtain an estimate for θ for a given value of $C = \alpha(\theta)$. Thus, for a given value of C , the decay rate θ is estimated using

²If C is less than the mean rate, the queue becomes unstable. If C exceeds the peak rate, the queue size is always empty.

$$\hat{\theta} = -\frac{\sum_{i=1}^M b_i \log(\pi(b_i)) - M \frac{1}{M} \sum_{i=1}^M x_i \frac{1}{M} \sum_{i=1}^M \log(\pi(b_i))}{\sum_{i=1}^M b_i^2 - M(\frac{1}{M} \sum_{i=1}^M b_i)^2} \quad (7.8)$$

where M is the number of bins in the histogram for the queue size. Since the LR estimator is also based on the effective bandwidth approximation the same restrictions as for the KLD estimator apply here. For long-range dependent traffic, a regression technique may be applied using Duffield's result stated in the last section. Note that in this case the decay rate is no longer exponential. To apply a linear regression technique, we transform the problem into a logarithmic domain.

Alternative estimation techniques for the effective bandwidth have been proposed by Gibbens [62] and by Vesilo and Solo [157]. The former estimator is based on a subsampling technique. The latter estimator has the desirable property of being adaptive. The effective bandwidth is estimated on-line, thus eliminating the requirement of collecting a large number of arrival observations. These estimators are not considered in this chapter.

7.2 Traffic streams

In order to evaluate the different effective bandwidth estimators, we generate finite time traces from known traffic models. This allows us to compare the measured effective bandwidth against the analytical results. The three traffic models chosen for the finite time traces are

- Poisson
- On-off Fluid
- Fractional Gaussian Noise (FGN)

Note that both the Poisson and the FGN models generate discrete time arrivals, whereas the on-off fluid model is in continuous time. We thus implement the four estimators in both discrete and continuous time.

The Poisson trace was generated with parameter $\lambda = 4$. With a time-slot definition of 0.271 ms, this amounts to an average arrival rate of 6.25 Mbps. The default length of the

trace is $N = 100,000$ slots, corresponding to 27 seconds of traffic. We consider this length as more than sufficient for a Poisson traffic stream. Note that the trace length is also varied in one of our experiments below. In this way, we are able to study the estimators as a function of the trace length.

The on-off fluid trace was generated based on the Bellcore specification for VBR I traffic. According to this specification, both on and off phases are geometrically distributed with a mean of 240 slots and 720 slots respectively, where a slot is defined to have a length of $2.83 \mu s$. A cell arrives with period of 6 slots during the on phase. The average number of cell arrivals per on-phase is thus 40 cells. The average rate of the stream evaluates to 6.24 Mbps. Since we use an on-off fluid model for comparison, we converted these parameters into continuous time measured in *ms*. Both on and off phases are now exponentially distributed with rates 1.47 and 0.49 respectively. Cells arrive at a rate of 58.9 cells/ms, giving again a total of 40 cells per on-phase or 6.24 Mbps for the stream. The default trace length was set to $N = 25,000$ cycles³, which amounts to 1.13 minutes of traffic. Again, the trace length is varied in some experiments to study its impact on the accuracy of the estimators.

The FGN trace has also been parameterized to match the Bellcore rates. We again model the number of cell arrivals in a discrete time slot of length $0.271 ms$. The mean number of arrivals and the corresponding variance were set to $\mu = 4$ cells/slot and $\sigma^2 = 0.36$ cells/slot respectively. We used a Hurst parameter of $\mathcal{H} = 0.7$. With these values, the average arrival rate of the stream is 6.25 Mbps. Where not differently specified, the number of slots in the trace was set to $N = 1,048,576$. This models a trace of length 284 seconds. As before, we vary the trace length in one of the experiments to assess the sensitivity of the estimation results on N . The FGN trace was generated using a fast Fourier transform method based on Paxson's generator [129] (available from [http : //ita.ee.lbl.gov/html/contrib/fft-fgn.c.html](http://ita.ee.lbl.gov/html/contrib/fft-fgn.c.html)). We also test the sensitivity of the measurement results by using a random-midpoint displacement (RMD) algorithm for the generation of the trace.

³We define a cycle as the duration of an on phase plus an off phase.

7.3 Accuracy of measured effective bandwidths

In this section we assess the accuracy of the four different estimation methods for effective bandwidths. For each of the three traffic traces we apply the estimators to a trace of length N , as specified in the last section. We also construct 95% confidence intervals by repeating the estimation 30 times, each time with a trace length N . We present our findings for the Poisson trace, the on-off trace and the FGN trace in succession.

7.3.1 Poisson traffic stream

We start our analysis in this case by applying the four estimators to a single trace of length $N = 100,000$ slots.

The estimated effective bandwidth surfaces for the direct and the block estimator are shown in figure 7.1. The θ -values in the plot are varied between 0.00001 and 5 with a multiplicative step of 1.5. The time values range between 1 and 30 slots, with a step of 1. The estimated effective bandwidth is thus shown in a log scale in figure 7.1 in the θ -dimension. Note that both surfaces are almost identical. This is to be expected in this case, because of the independence of arrivals in each slot. Applying a sliding window technique, as is done in the direct estimator, thus yields the same empirical distribution of the number of arrivals $A(\tau, \tau + t)$ in an interval of length t as the block method.

However, as time increases, the estimated effective bandwidth also decreases. This seems to be unexpected in this case. In fact, Kelly [88] shows that if the arrival process has independent increments, then the analytical effective bandwidth $\alpha(\theta, t)$ does not depend on time. We would thus expect the estimated effective bandwidth to be constant over time, which is clearly not the case in the measured effective bandwidth. This can be attributed to the finiteness of the trace length. For large aggregation intervals, the variance between the observed aggregated values decreases. Furthermore, the observed scaled maximum (the maximum of the aggregated arrivals divided by the time interval) also decreases. Since the measured effective bandwidth tends to this value, it decreases as the aggregation interval increases. We will discuss this phenomenon in detail in section 7.6.

To compare the estimated effective bandwidth for the block and the direct methods

Figure 7.1: Measured effective bandwidths for the Poisson trace using the direct and the block estimators. Note that the θ -axis is in log scale ranging from 0.00001 to 5 with a multiplicative step of 1.5.

Figure 7.2: Analytical and measured effective bandwidths for the Poisson trace for the four different methods.

with the results obtained from the other two estimators and with the analytical results, we have to collapse the time dimension. The above comparison between the measured effective bandwidth and the analytical effective bandwidth leads us to select $t = 1$. At this value, the measured effective bandwidth is closest to its analytical counterpart. Recall that this time value implies that the estimates from both methods are identical (with a window size 1, a sliding window is equivalent to a jumping window or block).

The resulting comparison is shown in figure 7.2 for a wide range of θ . All estimators seem to track the analytical effective bandwidth adequately for small values of θ . However,

none of the estimation methods is able to track the analytical effective bandwidth for large values of θ . The point at which the estimators start to deviate is almost identical for the different methods (at around $\theta \approx 1$). The direct / block estimators seem to perform best, degrading gradually. The KLD estimator follows the pattern of the direct / block estimators, albeit more irregular. The LR estimator stands out by retracting. This poor performance of the virtual buffer based estimators can be attributed to the difficulty in obtaining a satisfactory empirical distribution for the queue length. If the virtual queue is served at a high service rate (compared to the number of arrivals), the maximum queue size remains small. Consequently, only few observations are available to fit a line through the log of the empirical queue size distribution. The accuracy of the fitted line is then very poor. Similarly for the KLD estimator. The estimators may be improved by implementing a heuristic, requiring for example, a minimum number of observations in order to compute an estimate.

In general, the degradation of the estimators in the θ -space can be explained by the property that the effective bandwidth approaches the maximum value of $A(\tau, \tau + t)$ as $\theta \rightarrow \infty$ (see [88]). We prove this property in section 7.6.1 below. Whereas the analytical EB tends to infinity as θ grows, the measured effective bandwidth tends to the maximum of the observed values (for finite t). For the underlying Poisson process, $P(A(0, t) = x)$ is positive for all $x = 1, 2, \dots$, in particular for large values of x . By the argument of Laplace the analytical EB tends to infinity. In contrast, using a particular realization of this process we observe a maximum number of arrivals for $A(0, t)$. Hence, again by the argument of Laplace, this value determines the asymptotic behavior of the measured effective bandwidth. In our case, the measured maximum number of arrivals was observed to be 15.

This deviation has significant implications for CAC. For a QoS requirement of $\theta > 3$ in this case, a CAC method based on analytical effective bandwidth would require a bandwidth capacity in excess of 25.44 cells/slot. This is clearly inappropriate when the maximum observed value of the trace is only 15 cells/slot. If analytical effective bandwidths are used in the CAC procedure, the current traffic stream may well be rejected even though sufficient bandwidth may be available to carry it. Such values for θ are common if very small loss probabilities with small buffer sizes are required. Such is the case for real time traffic. We

Figure 7.3: Analytical and measured effective bandwidths for the Poisson trace with 95% CIs

can derive a realistic range for θ from equation (3.16) by assuming that $P(Q > B) \leq 10^{-10}$. With a mean arrival rate of 4 cells/slot the buffer size allocated to the stream is likely to exceed 4 cells. In this way we obtain $\theta < 3.45$. Similarly, assuming a cell loss probability of 10^{-2} and a large buffer size of 1000 cells we obtain $\theta = 0.0046$. A realistic range for θ therefore seems to be $0.001 < \theta < 3.5$.

We will investigate the discrepancy between the empirical and the analytical effective bandwidths in more detail in section 7.7.

The above results are confirmed by repeated estimations. Figure 7.3 shows the measured effective bandwidth with 95% CIs for the direct / block method and the KLD / LR method on the left and right respectively. Note that the axes for the virtual buffer based methods are reversed. In this case, we estimate θ for given values of $C = \alpha(\theta)$. The CIs are thus for the estimated θ -value. The axis representing the estimated effective bandwidth is again in log scale. All CIs were computed over 30 sample traces, each with a length of $N = 100,000$ slots. The tightness of the 95% CIs confirms the accuracy of the above analysis. The measurements appear to be sufficiently accurate for $\theta < 1$. For $\theta > 1$, the analytical effective bandwidth falls outside the CIs for all the four measurement methods. Note that this range for θ may indeed be significant, as indicated above.

Figure 7.3 also illustrates the similarities between the different estimation methods. With $t = 1$, the direct estimators and the block estimators are identical. The LR estimator

performs insignificantly better than the KLD estimator for small values of θ . Both virtual buffer based estimators degenerate significantly for $\theta > 1$. This result favors the direct /block method. Furthermore, their execution time is smaller than for the remaining two methods. The direct method took on average 640 ms to compute a single estimation point on a Sun Ultra 1. The estimation time for the KLD method on the same machine varied between 500 ms for large θ and 1.02 seconds for small¹ θ .

7.3.2 Comparison with truncated Poisson process

As a demonstration of the difference between the analytical and the measured effective bandwidths, we now compare the latter with the effective bandwidth of a truncated Poisson process. We thus show that the discrepancies between the analytical effective bandwidth and the measured effective bandwidth are resulting from the finiteness of the trace.

The mean rate and the truncation level of the truncated process are set to the observed mean and the observed maximum value respectively. In this way, we approximate the empirical histogram of the number of arrivals per slot with a truncated Poisson process. Denoting the truncated process by A_T , the corresponding distribution function by $P_T(A_T = x)$ and the finite time realization of the Poisson process by $\hat{A}$, we have

$$P_T(A_T = x) = \begin{cases} gP(A_T = x) & \text{if } x \leq \max(\hat{A}) \\ 0 & \text{otherwise} \end{cases}$$

where

$$g = \frac{e^\lambda}{\sum_{i=0}^{\max(\hat{A})} \frac{\lambda^i}{i!}}.$$

The analytical effective bandwidth for this process then evaluates to

$$\alpha_T(\theta) = \frac{1}{\theta} \log \frac{\sum_{i=0}^{\max(\hat{A})} (\lambda e^\theta)^i / i!}{\sum_{i=0}^{\max(\hat{A})} \lambda^i / i!} \quad (7.9)$$

The resulting comparison is shown in figure 7.4. We illustrate the measured effective bandwidth for the block estimator, as well as the analytical effective bandwidths of both

¹Recall that time is measured in slots here. A slot is defined to be 0.271 ms.

Figure 7.4: Comparison of the measured EB against the analytical EB for both the Poisson and the truncated Poisson processes.

the Poisson process and the truncated Poisson process. In this case, the maximum observed number of arrivals per slot was 15. The three effective bandwidth functions are very close for small values of θ . However, as θ increases, the effective bandwidth of the Poisson process tends to infinity, whereas its truncated counterpart tends to the maximum value. The graph shows that the truncated Poisson process is a more accurate model for a finite time trace than a pure Poisson process. This experiment leads us to conclude that the finiteness of arrival streams is an important characteristic which has to be considered for practical applications of effective bandwidths.

7.3.3 On-off fluid traffic stream

We now present the results for measured effective bandwidths for the on-off fluid. Again, we start by analyzing the direct and the block estimator. As before, we collapse the time dimension of these two methods to incorporate the results into a comparison of all four estimators against the analytical values. Finally, we present results of confidence intervals.

Figure 7.5 illustrates the estimated surfaces of the direct and block estimators for the on-off fluid trace. It also shows the analytical effective bandwidth surface for an on-off fluid process. The θ -values in the plot are varied between 0.00001 and 5 with a multiplicative step of 1.5. The time values range between 1 and 30, again with a step of 1. Note that the

Figure 7.5: Analytical and measured effective bandwidths for the on-off fluid trace using the direct and the block estimators. Note that the θ -axis is in log scale ranging from 0.00001 to 5 with a multiplicative step of 1.5.

θ -axis is thus shown in a log scale. As in the case for Poisson traffic, the estimated surfaces from the direct and the block estimators are remarkably similar, showing the same behavior in both the time and the θ space. For small values of θ and over the entire time-range, the measurements seem to track the analytical effective bandwidth reasonably well. Similarly, the accuracy seems to be satisfactory for large values of θ and small time values. However, as the time window increases, the estimators start to degenerate again for large values of θ . For the direct estimator, the measurements start to degenerate with $t \geq 5$. For the block estimator, the measured effective bandwidth degenerates with $t \geq 4$. The analytical effective bandwidth on the other hand does not exhibit such a behavior. These inaccuracies are further illustrated in figure 7.6 showing the squared error between the analytical effective bandwidth and the measured effective bandwidth. Our explanation offered in the last section also seems to apply for on-off traces. As the time parameter increases, the observed maximum value of $A(0, t)$, scaled by t , decreases. Since the measured effective bandwidth tends to this value as θ increases, the estimates degenerate in this parameter region.

Figure 7.6: Squared error of the effective bandwidth measurements for the on-off fluid trace using the direct and the block estimators. Note that the θ -axis is in log from 0.00001 to 5 with a multiplicative step of 1.5.

To compare the measured effective bandwidth with the KLD- and LR estimators, we again collapse the time dimension using equation (7.15). In this case, the choice of a representative time scale is less obvious. Without any prior assumption about the application of the effective bandwidth, we resort to the following heuristic: we restrict ourselves to the range of values for t where the measurements do not degenerate, giving us $1 \leq t \leq 5$. We then examine the squared error plots shown in figure 7.6 and select the largest value of t for which the squared error is small. We thus arrive at a value of $t = 5$ for our comparison. The results are shown in figure 7.7. The θ -dimension is again shown in a log scale. In this case, the measured effective bandwidth seems to be sufficiently close to the analytical effective bandwidth for all four methods. Note how our argument explaining the discrepancy between the analytical and measured effective bandwidths for the Poisson trace is less apparent here. For a fixed time interval t , the arrivals are limited by the peak rate of the source. For on-off fluid sources, this peak rate is clearly finite. Consequently, even the analytical effective bandwidth varies between the mean and the finite peak rate. The estimators capture this peak rate well (assuming the appropriate time value is chosen for the direct and block estimators).

The accuracy of the measured effective bandwidth is confirmed by figure 7.8, showing 95% CIs for all four methods. Again, we separate the plots for the direct / block estimators and the KLD / LR estimators, with the axes on the latter plot reversed. As before, we

Figure 7.7: Analytical and measured effective bandwidths for the on-off fluid trace

computed the 95% CIs over 30 estimations, each with a length of $N = 25,000$ cycles⁷ per estimation. The tight error bars in figure 7.8 indicate high accuracy for the measured effective bandwidth. For the direct / block estimators, the CIs include the analytical values only for small θ -values. For $\theta > 0.05$, the analytical effective bandwidth once again falls outside the 95% CI. The KLD / LR estimators on the other hand seem to achieve a better accuracy for this traffic type. But even in this case, a closer examination of the CIs reveals that for parts of the θ -range, the analytical effective bandwidth falls outside the 95% CI (in this case, for $\theta < 0.05$). Consistent with our previous findings, the measured effective bandwidths are below the analytical EB as θ increases. Both functions tend to the peak value of the source.

When applying effective bandwidths to CAC, this has again the consequence that a stream may be rejected by the network even though sufficient bandwidth may be available. For a given QoS value θ of the VP, the analytical effective bandwidth exceeds the measured effective bandwidth. Using condition (3.32) may then lead to a rejection.

In comparison with each other, we find again that the direct / block and the KLD / LR estimators perform similarly. For the direct / block estimators, the CIs are overlapping over the entire θ range. The same holds true for the KLD / LR estimators for smaller θ -values. The direct estimator runs between 0.16 seconds and 0.38 seconds for small and large

⁷Recall that we define a cycle as the duration of an on-phase plus and off-phase.

Figure 7.8: Analytical and measured effective bandwidths for the on-off fluid trace with 95% CIs

values of θ respectively. The corresponding values for the block estimator are 0.032 seconds and 0.079 seconds. The run times for the KLD estimators are between 45 seconds and 12 seconds for a single estimation point. The LR estimator takes between 43 and 13 seconds. Note that these time values are dependent on the implementation and the trace length. The virtual buffer estimators for fluid sources are more complicated to implement. Much of the time is spent on arranging the data structures to compute workload areas and on the subsequent integration. We thus do not place too much significance on the actual value. However, we do note that the virtual buffer estimators, although being more accurate, take significantly longer to obtain a single estimation point. This is a considerable disadvantage of these methods. For on-off fluid traces, the trade-off between run-times and accuracy leads us again to prefer the block estimator.

7.3.4 FGN traffic stream

Our analysis of measured effective bandwidths differs significantly for the FGN trace. Most of the estimation methods cannot be applied to such a trace because of their underlying assumptions. For this reason, in this section we restrict ourselves to the direct estimator. In addition to the comparison between the analytical and the measured results, we examine the sensitivity of the measurement results to the traffic generator and the behavior for large

Figure 7.9: Analytical and measured effective bandwidth for the FGN trace using the direct estimator

values of θ .

Figure 7.9 shows the analytical effective bandwidth and the measured effective bandwidth of the FGN trace described in section 7.2. The θ -values vary between .00001 and 10 1/cells with a multiplicative factor of 1.5. The range for the t -values is instead between .271 ms and 10 seconds, with multiplicative factor of 2. The figures clearly show that for large values of θ , the measured effective bandwidth and the analytical effective bandwidth are once again significantly different. In this case, they differ by many orders of magnitude. For large θ values, the analytical effective bandwidth increases in time. The measured effective bandwidth on the other hand decreases as t increases. This latter observation is consistent with our findings from the on-off and Poisson sources where the measured effective bandwidth also degenerates in time.

For small values of θ , the plots in figure 7.9 seem to indicate reasonable accuracy of the estimator. However, this is not at all the case, as illustrated in the squared error plot shown in figure 7.10. Even for small values of θ and t , the discrepancies between the analytical and the measured effective bandwidths are significant. Recall that the average traffic rate is 14760.15 cells/sec.

To demonstrate that this finding does not depend on the underlying traffic generator, we generated a second trace with the same parameter values using the RMD method [102]. Since the FGN generator is only asymptotically unbiased, we hope to demonstrate in this

Figure 7.10: Squared Error between the analytical and the measured effective bandwidths for the FGN trace.

way that the observed behavior is independent of the selected generation scheme. We compared the estimated effective bandwidth for both traces (FGN and RMD) with length $N = 131,072$ slots. To illustrate the differences, we restrict ourselves to two time values only: $t_1 = 1$ slot and $t_2 = 32$ slots. The results are illustrated in figures 7.11 and 7.12. For t_1 , the estimated effective bandwidths are very close. For t_2 , the measurements differ slightly as θ increases but are never the less still sufficiently close to one another. We therefore conclude that the differences between the measured and the analytical effective bandwidth are independent of the underlying traffic generator.

7.3.4.1 Asymptotic behavior in θ

In this section we provide evidence supporting our hypothesis that the estimated effective bandwidth tends to the measured maximum aggregated value A_{max} over the time interval $(0, t)$ as θ grows. A full proof is given in section 7.6. In this experiment, we fix $\theta = 9.707$, its largest value considered here. Tables 7.1, 7.2, 7.3 show the estimated effective bandwidth at this value, as well as the observed maximum A_{max} for different values of time and trace lengths of 512, 131072 and 1048576 slots respectively. In table 7.1 we also report the value of the analytical effective bandwidth at $\theta = 9.707$. As shown, the measured A_{max} depends both on the trace length and on t . The observed maximum aggregated value A_{max}

Figure 7.11: Comparison between the RMD and the FFT generators for $t = 1$ slot and $N = 131,072$.

Figure 7.12: Comparison between the RMD and the FFT generators for $t = 32$ slots and $N = 131,072$.

decreases as the time parameter increases. The asymptotic value of the estimated effective bandwidth gets smaller and smaller as time increases. This behavior is in sharp contrast with the theoretical effective bandwidth for FGN processes, which grows extremely fast with time. The longer the trace length, the bigger A_{max} . However, from tables 7.1, 7.2, 7.3 we can also infer that increasing the trace length even by several orders of magnitude does not affect A_{max} significantly.

time(ms)	$\hat{\alpha}(\theta_{max}, t)$	A_{max}	Theor. EB
0.27	19418.75	21554.14	905204.43
0.54	19041.21	20170.03	1189708.37
1.08	18391.32	18959.32	1396592.12
2.16	17573.16	17854.69	1565113.62
4.33	16901.72	17045.42	1709857.75

Table 7.1: Behavior of an FGN trace for large values of θ . $N = 512$.

time(ms)	$\hat{\alpha}(\theta_{max}, t)$	A_{max}
0.27	20514.31	24599.02
0.54	20717.09	22873.12
1.08	20299.11	21402.92
2.16	19333.94	19880.40
4.33	18424.75	18700.81
8.67	17726.63	17864.60
34.68	16616.70	16651.21
138.75	15824.72	15833.30
555	15358.79	15360.93

Table 7.2: Behavior of an FGN trace for large values of θ . $N = 131.072$.

time(ms)	$\hat{\alpha}(\theta_{mar}, t)$	A_{mar}
0.27	20691.64	25484.57
0.54	21054.80	23595.59
1.08	20537.77	21824.39
2.16	19839.07	20484.69
4.33	18944.90	19268.83
8.67	18153.15	18315.65
34.68	16849.34	16889.82
138.75	16020.53	16030.64
555	15519.72	15522.26
2220	15193.11	15193.74
8880	14993.34	14993.50

Table 7.3: Behavior of an FGN trace for large values of θ . $N = 1,048,576$.

We proceed to show the independence of these findings of the trace length N .

7.4 Effect of the trace length on measured effective bandwidths

Our experiment consists of re-running the above measurements over the same parameter space for different trace lengths. We subsequently restrict ourselves to the block estimators and the KLD estimators for the Poisson trace and the on-off fluid trace. For the FGN trace, we use the direct estimator. Recall that for Poisson traces with $t = 1$ slot, the block estimator and the direct estimator are identical. All results in this section are based on 30 estimations of the traces with length N .

7.4.1 Poisson traffic stream

Figure 7.13 shows the estimated effective bandwidths for the block estimator and the KLD estimator for the trace lengths $N = 1024$ slots and $N = 65,536$ slots respectively. In both cases, the two measured curves are remarkably similar. In particular, the measured effective bandwidth seems to be consistent for a significant portion of the range. This holds for both the block and the KLD estimators. Increasing the trace length only seems to affect the measurements for larger values of θ . The implications of this result are twofold. First, we conclude that a reasonable estimate of a trace's effective bandwidth can be obtained by a relatively small trace. A trace length of $N = 1024$ slots corresponds in our case to 277 ms. Obtaining reasonably accurate estimates for the effective bandwidth is important when applied to CAC. Second, this result supports our argument that the behavior of the effective bandwidth for large values of θ is dominated by the maximum observed value. Additional observations which are below the maximum observed with a small trace seem to add little information to the estimated effective bandwidth. However, those observations which exceed the current maximum are responsible for increasing the tail of the estimated effective bandwidth curve for large values of θ .

The measurements for intermediary trace lengths of $N = 4096$ slots and $N = 16,384$ slots confirm these conclusions. They are not shown here to simplify the presentation of our findings.

7.4.2 On-off fluid traffic stream

The main conclusion from the last section also carries over to the on-off trace. The results of comparing trace lengths $N = 500$ cycles and $N = 32,768$ cycles are shown in figure 7.14, with the block estimation shown on the left and the KLD estimations shown on the right. For the block estimation, an increase in the trace length has again the effect of pushing the tail of the estimated effective bandwidth curve up. However, even with a small trace length the estimator performs surprisingly well. In the case of the KLD estimator increasing the trace length has a different effect in this case. Rather than affecting the tail of the effective bandwidth function, its shape improves for smaller values of θ . Note that

Figure 7.13: Comparison of the measured effective bandwidth for different trace lengths for the Poisson trace using the block and the KLD estimator.

even for small traces, the bursty nature of on-off traffic allows us to observe the peak value of the traffic stream. We attribute the accuracy of the KLD estimators for high θ -values to this fact. Considering the short length of the trace $N = 500$ cycles, we conclude that the KLD estimator is acceptable.

Again we confirmed these results with traces of length $N = 2048$ and $N = 8192$ cycles respectively, but we suppress these here to simplify our presentation.

7.4.3 FGN traffic stream

The effect of the trace length on the measured effective bandwidth for the FGN trace is more significant than for the previous two traces. In this case, we evaluated traces of length $N = 512$ slots and $N = 1,048,576$ slots. Figure 7.15 illustrates the estimated EB for a fixed time $t_1 = 1$ slot. Figure 7.16 shows the corresponding graph for $t_2 = 16$ slots. For small time scales, the performance of the estimators is not dependent on the trace length. The estimated effective bandwidths are very close for significant sections of the range. However, as we increase the time scale the difference in the tail of the estimated effective bandwidths also increases. This suggests that additional observations are more valuable for larger time scales.

Figure 7.14: Comparison of the measured effective bandwidth for different trace lengths for the on-off trace using the block and the KLD estimator.

Figure 7.15: Estimates of the effective bandwidth for the FGN trace taken over traces 1048576 and 512 samples long, at $t = 1$ slot.

Figure 7.16: Estimates of the effective bandwidth for the FGN trace taken over traces 1048576 and 512 samples long, at $t = 16$ slots.

7.5 Summary of Findings

The above results can be summarized as follows:

- Accuracy:** the methods considered here are only able to match the analytical effective bandwidth to a high degree of accuracy for small values of the θ range. This is consistent for all traffic streams considered. For large values of θ , the estimators fail to track the analytical effective bandwidth at the 95% level of confidence. This can be explained by the property of the measured effective bandwidth approaching the scaled maximum observed value over a time interval. Demanding strict QoS requirements (very low loss probabilities or cell delays) may however require bandwidths in the range where θ is large. The analytical effective bandwidth cannot therefore be successfully measured using any of the methods described above over the entire parameter range.
- Degeneration in time:** for the direct and the block estimators, which are dependent on time, we observe a degeneration of the measured effective bandwidth in time for sufficiently large values of θ . We attribute this phenomenon to the finiteness of the arrival stream. As the time interval increases, the variance of the aggregate number of arrivals over the intervals decreases. Furthermore, the *scaled* maximum over a longer time interval decreases, causing the degeneration.

- **Trace Length:** effective bandwidth estimates can be obtained with a relatively small number of observations. Increasing the trace length improves the estimate most at large values of θ . The effective bandwidth can thus be estimated for a small trace. This initial estimate may be used during CAC and updated subsequently as more observations are collected.
- **Parameter space:** The estimators perform better for smaller values of θ . For larger θ -values, the estimated effective bandwidths fail to include the analytical effective bandwidth in all four estimation methods. For very small cell loss probability requirements or buffer sizes, it is precisely those large θ -values which are important. The four measurement methods thus only estimate the analytical effective bandwidth for a limited range of the θ -space.

For short-range dependent traces, the degradation in time can be omitted by selecting an appropriate time parameter. The analytical effective bandwidth is even time invariant if the trace exhibits independent increments. For long-range dependent traffic, the time scale is critical. The measured effective bandwidth differs substantially from its analytical counterpart in this dimension. Further research into selecting an appropriate time scale for different applications is required.

- **Estimation Method:** For short-range dependent traffic, the block estimator achieves the nicest results. It appears to be faster and less brittle than the estimators based on virtual buffers. On the other hand, the KLD and LR estimators achieve a higher accuracy for small θ values. Their estimates become unstable because of the difficulty of observing the complementary queue length probability at service rates approaching the peak rate of the source.

A further remark highlights the importance of these results: the above differences were obtained by experiments with finite-time traces of Poisson, on-off fluid and FGN traffic. We know that the traces match the underlying analytical model closely. It is unlikely that this is the case with real traffic traces. We thus anticipate the differences between analytical effective bandwidths and estimated effective bandwidths to worsen when applied to real

traces. When applied to CAC, for example, it may be preferable to measure the effective bandwidth instead of determining its analytical form.

7.6 Empirical Effective Bandwidths

The above limitations motivate us to formally differentiate measured and analytical effective bandwidths by introducing the notion of *empirical effective bandwidths (EEB)*. Let A again be a given, finite time realization of a weakly-dependent, stationary stochastic arrival process of length N . Inspired by the definition of the effective bandwidth given in [88], we *define* the empirical effective bandwidth of such a trace with parameters θ , t and N as

$$a(\theta, t, N) = \frac{1}{\theta t} \log \hat{E}_N[e^{\theta A(0,t)}] \quad 0 < \theta; 0 < t < N \quad (7.10)$$

where $A(0, t)$ indicates the aggregate number of cell arrivals in an interval of length t slots and $\hat{E}_N[e^{\theta A(0,t)}]$ is the measured EB over a trace consisting of N samples. To simplify the notation, we restrict ourselves to the discrete time case, where A_t denotes the number of cell arrivals in time slot t of the finite trace of length N slots. We note however that all results derived below carry over to the continuous time case. We assume the arrival process is fed into an infinite queue served at a deterministic rate of C cells per slot. We relate equation (7.10) to the probability $P(Q > B)$ of the queuing process Q exceeding a given threshold B .

Note that we retain the parameter space of Kelly's effective bandwidth in our definition. Applying the empirical effective bandwidth for CAC, the θ parameter determines the quality of service received by the source when fed into a single server queue (c.f. equation (3.16)). Alternatively, in the setting where many sources are multiplexed over a single link, θ provides an indication of the degree of statistical multiplexing. The time parameter t relates to the most likely duration of the busy period before the buffer reaches the threshold B . In this sense, it identifies the time scales of the arrival process which are relevant for buffer overflows. Note also that we do not prescribe a specific function to obtain $\hat{E}e^{\theta A(0,t)}$. For short-range dependent traffic any of the estimators presented above may be used.

7.6.1 Properties of Empirical Effective Bandwidths

In this section we establish some properties of the empirical effective bandwidth definition for a given traffic trace A_t , $t = 1 \dots N$, with sample mean $m = \frac{1}{N} \sum_{t=1}^N A_t$ and sample variance $s^2 = \frac{1}{N-1} \sum_{t=1}^N (A_t - m)^2$. We discuss these properties in three parts, according to the behavior in the θ -space, the time space and with respect to the trace length. In each section, we contrast the results obtained for EEBs with the results for the analytical effective bandwidth. In particular, we show that the behavior of the EEB as $\theta \rightarrow \infty$ differs significantly from analytical effective bandwidths.

Throughout this section, we assume that equation (7.3) has been used as estimator for $\hat{E}_N e^{\theta A(0,t)}$ and hence that the block sums are independent. Following the notation in [42] we denote by $\tilde{A}_t := \sum_{j=(i-1)t+1}^{it} A_j$, $i = 1 \dots \lfloor \frac{N}{t} \rfloor$ the number of cell arrivals over an interval of length t . We also let $M_t = \lfloor \frac{N}{t} \rfloor$.

7.6.1.1 Behavior in the θ space

Property 1: For a fixed value of $t < \infty$,

$$\lim_{\theta \rightarrow 0} a(\theta, t, N) = \frac{1}{tM_t} \sum_{i=1}^{M_t} \tilde{A}_i = m \quad (7.11)$$

Proof: The proof is analogous to the one given for the analytical effective bandwidth in theorem 3.2. $\square$

Note that the sample mean is an unbiased estimator for the population mean. The analytical effective bandwidth and the empirical effective bandwidth thus exhibit a similar behavior as $\theta \rightarrow 0$.

Property 2: For a fixed value of $t < \infty$,

$$\lim_{\theta \rightarrow \infty} a(\theta, t, N) = \frac{1}{t} \max[\tilde{A}_i] \quad (7.12)$$

Proof: Again, the proof is analogous to the one for theorem 3.3, with the essential supremum replaced by the maximum observed block sum $\max[\tilde{A}_i]$. $\square$

Note that the analytical effective bandwidth tends to the *essential supremum* of the process, which may be infinite. However, for a particular finite time realization of the process

the probability of observing the theoretical supremum becomes less likely as t increases. For example, for the underlying process there is a positive probability of observing the peak rate over the *entire* interval $(0, t)$, even as t grows large. For a finite time realization of the process such observations are less likely, in particular as t grows large. As a result the EEB is determined by the *measured* maximum value of $\tilde{A}_i, i = 1 \dots M_t$ as $\theta \rightarrow \infty$. Note how this property explains the discrepancy between the analytical effective bandwidths and their measured counterparts observed in the last section.

In summary, the analytical effective bandwidth varies between the population mean μ and the supremum of $A(0, t)/t$ for any fixed t . In contrast, the empirical effective bandwidth varies between the sample mean m and the measured maximum $\max[\tilde{A}_i]$. The importance of this difference is our principal justification for the introduction of EEBs.

7.6.2 Behavior in the t space

The last observation also allows us to describe the difference between the EEB and the analytical EB in the t -space. To further illustrate our point using an example, let us assume that A is a Markov-modulated arrival process with peak rate π and mean rate $m < \pi$. Then, $P(A(0, t) = \pi t)$ is positive. However, for any finite time realization this probability tends to zero as t increases, since $\lim_{t \rightarrow \infty} A(0, t) = mt < \pi t$. The analytical EB is limited by the supremum⁶ of $A(0, t)/t$. However, the EEB is limited by the maximum of $A(0, t)/t$, which tends to mt as t increases.

In the special case where the central limit theorem (CLT) holds for A_i , we can characterize the behavior of the EEB by the following additional property:

Property 3: Assume that the trace A_n is a finite realization of a weakly-dependent stochastic process with finite population mean μ and population variance σ^2 . Assume also that t is large enough such that the block sums $\tilde{A}_i, i = 1 \dots M_t$ are independent and that the CLT holds. For large N, t with $t < N$ and for a given θ the EEB degenerates towards

$$\lim_{t \rightarrow N} a(\theta, t, N) = m + \frac{s^2 \theta}{2} \quad (7.13)$$

⁶Which may indeed be increasing in t .

Proof: By the central limit theorem, the block sums $\tilde{A}_i \sim N(t\mu, t\sigma^2), i = 1 \dots M_t$ [125] for t sufficiently large. Furthermore, in this case the sample mean m and the sample variance s^2 are unbiased estimators for μ and σ^2 of a function of a random variable. Equation (7.10) can then be written as

$$\begin{aligned} \lim_{t \rightarrow \infty} a(\theta, t, N) &= \lim_{t \rightarrow \infty} \frac{1}{\theta t} \log E[e^{\theta \tilde{A}_i}] \\ &= \lim_{t \rightarrow \infty} \frac{1}{\theta t} \log e^{\theta \mu t + \frac{\sigma^2 \theta^2 t}{2}} \\ &= \mu + \frac{\sigma^2 \theta}{2}. \end{aligned}$$

The result follows from the unbiasedness of m and s^2 . $\square$

This property explains why we observed a degeneration of the measured effective bandwidth in the last section for large time values. In many cases, the EEB thus decreases as t increases, provided $m < \pi$. Note that this condition excluded the trivial case of a constant source. In this case the analytical EB and the empirical effective bandwidths are actually equal and we do not observe a degeneration in time. The behavior of analytical EBs in the t -space is not necessarily decreasing. For a stochastic process, the probability of observing the peak rate even for large values of time is positive. This probability tends to zero for a finite time process for large t . A complete characterization of the underlying process can hardly be obtained from a single finite realization of the process with sufficient accuracy in an acceptable time frame. The empirical effective bandwidth consequently deviates significantly from the analytical EB.

7.6.3 Behavior with respect to the trace length

Property 4: As $N \rightarrow \infty$

$$\lim_{N \rightarrow \infty} a(\theta, t, N) = \alpha(\theta, t). \quad (7.14)$$

Proof: The proof is given by Duffield in [42]. $\square$

Note that the last property holds in both the t and θ spaces. As the trace length increases, the EEB approaches its analytical counterpart.

7.7 Experimental Results for CAC

To show the relevance of the EEB concept in a practical environment we now report simulation results. We apply EEBs for the finite time realizations of on-off fluids, Poisson and FGN arrival processes to the CAC problem. We are thus able to compare the allocated resources using EEBs with the results obtained from using the analytical counterpart. The latter is assumed to be known and approximated using a parametric model. Note that we also assume that the entire finite time realization is known, as is the case for pre-recorded video, for example. We present our results by first describing our experimental methodology, followed by a separate presentation of the results for the on-off, the Poisson and the FGN traces.

7.7.1 Simulation Methodology

To show the applicability of EEBs we proceed as follows: we generate a test trace of length N . We then select one of the estimators for $\hat{E}[e^{\theta \cdot A(0,t)}]$ in equation (7.10) to obtain numerical results for the EEB. We considered the block estimator for on-off and Poisson traffic and the LR estimator for the FGN trace. We subsequently apply the EEB to our CAC problem. We assume that the given arrival stream wishes to gain access to a single server queue with various buffer sizes B and a specified quality of service requirement δ , expressed as a cell loss ratio (CLR). We use the effective bandwidth approximation to evaluate the required θ -value, called θ^* . Since we only estimate the effective bandwidth for a limited number of θ -values, we apply a linear interpolation to derive the numerical value at θ^* . We subsequently determine the corresponding service rate C using $\alpha(\theta^*, t^*) = C$. Here, t^* indicates the time scale at θ^* where the empirical effective bandwidth takes the maximum value, i.e.

$$t^* = \arg \sup \{t \in \mathcal{R}^- : a(\theta^*, t)\}. \quad (7.15)$$

Note that the KLD and LR estimators do not require t^* , since they are inherently related to the relevant time scale for the queuing performance. Finally, we feed the given trace into the single server queue with the given buffer size B and the determined service rate C . We observe the simulated CLR and compare it against the specified requirement δ .

In the case of Poisson and on-off fluid traces we repeat this experiment using the analytical EB to establish a comparison. As indicated in the introduction, effective bandwidth based CAC is based on equations (3.15) and (3.1). For this reason, we use $t \rightarrow \infty$ for the analytical results. In order to obtain confidence intervals for our procedure, we then repeat this experiment 30 times.

7.7.2 Results for the Poisson trace

Table 7.4 illustrates the advantages that can be gained when applying EEBs for CAC using the Poisson trace. For this experiment, we again used the 30 traces to obtain 95% CIs. We assumed a quality of service criterion of $\delta = 0.001$ for various buffer sizes. This criterion is met using both the analytical and the empirical effective bandwidths. In this case the bandwidth allocated using the EEB is slightly higher in general as compared to the analytical effective bandwidth. As the required CLR and the buffer size decrease this behavior tends to invert. For example, in the case of $B = 5$, the cell loss probability requirement leads to a value of $\theta^* = 1.38$. As shown in section 7.3.1 it is those high values of θ where the empirical and the analytical effective bandwidths differ significantly. For larger buffer sizes under this QoS requirement, the estimated effective bandwidth is very close to its analytical counterpart, and hence we observe little difference in the results. For very small values of these two parameters a simple computation as shown in section 7.3.1 reveals that the analytical EB would even exceed the observed peak value.

7.7.3 Results for the on-off fluid trace

The advantages of EEBs over analytical EBs become apparent when using the on-off trace. Table 7.5 lists the simulation results for our experiment for various buffer sizes. The quality of service criterion for this experiment was set to $\delta = 0.0001$. Again, this requirement is clearly met using both the EEB and the analytical effective bandwidths. Both methods exceed the requirement by at least one order of magnitude. This can be explained by the inequality in equation (3.16) and the resulting upper bound on which the allocation is based.

Recall that our analysis in section 7.3 showed that the estimated and the analytical effective bandwidths are very close. This can be attributed to the nature of the traffic

Table 7.4: Simulation results for the Poisson trace (top: empirical effective bandwidth: bottom: analytical effective bandwidth)

Buffer	$a(\theta^*, t^*, N)$	$\hat{P}(Q > b)$	CI
5	8.26	2.75e-05	(2.70e-05, 2.81e-05)
10	5.79	6.46e-05	(6.35e-05, 6.58e-05)
20	4.78	8.32e-05	(8.03e-05, 8.61e-05)
30	4.50	6.16e-05	(5.89e-05, 6.43e-05)
100	4.14	4.49e-05	(4.16e-05, 4.83e-05)
500	4.03	1.22e-05	(9.37e-06, 1.51e-05)
Buffer	$\alpha(\theta^*)$	$\hat{P}(Q > b)$	CI
5	8.63	1.65e-05	(1.61e-05, 1.69e-05)
10	5.76	7.19e-05	(7.07e-05, 7.31e-05)
20	4.77	8.84e-05	(8.54e-05, 9.14e-05)
30	4.49	6.68e-05	(6.39e-05, 6.96e-05)
100	4.14	5.27e-05	(4.89e-05, 5.64e-05)
500	4.02	1.86e-05	(1.42e-05, 2.29e-05)

stream, and so it is not surprising that our EEB again performs well.

However, notice that for this type of source we allocate less bandwidth to the stream for all buffer sizes using EEB. While this is not necessarily always the case, it does highlight the advantage of EEBs: the allocated bandwidth is not based on the population characteristics, but on the measured characteristics of the stream. This results in a bandwidth allocation which is tailored to the actual traffic stream.

7.7.4 Results for the FGN trace

The validity of using EEBs for CAC is also demonstrated when we use the FGN trace. Table 7.6 illustrates the simulation results for our experiment with FGN realizations. The CLR for this experiment was set to $\delta = 0.0001$. In this case we use equation (7.7) instead of the effective bandwidth approximation to derive θ^* . This is necessary since the latter is not applicable to long-range dependent traffic. For the same reason we are not able to determine an appropriate time scale analytically. In order to circumvent this difficulty we used the results from the LR estimator. This estimator in fact should inherently provide

Table 7.5: Simulation results for the on-off Fluid trace (top: empirical effective bandwidth; bottom: analytical effective bandwidth)

Buffer	$a(\theta^*, t^*, N)$	$\hat{P}(Q > b)$	CI
5	57.90	1.03e-05	(1.02e-05, 1.05e-05)
10	56.93	1.89e-05	(1.86e-05, 1.92e-05)
20	54.99	3.94e-05	(3.89e-05, 3.98e-05)
30	53.16	5.44e-05	(5.37e-05, 5.51e-05)
100	42.23	0.0001	(9.80e-05, 0.0001)
500	22.30	3.56e-05	(3.11e-05, 4.00e-05)
1000	18.16	2.58e-05	(2.24e-05, 2.91e-05)
1500	16.91	1.34e-05	(1.04e-05, 1.65e-05)
Buffer	$\alpha(\theta^*)$	$\hat{P}(Q > b)$	CI
5	58.10	9.24e-07	(8.77e-07, 9.72e-07)
10	57.31	1.78e-06	(1.68e-06, 1.87e-06)
20	55.76	3.23e-06	(3.05e-06, 3.41e-06)
30	54.24	5.06e-06	(4.80e-06, 5.32e-06)
100	44.63	1.99e-05	(1.83e-05, 2.15e-05)
500	22.41	3.16e-05	(2.75e-05, 3.57e-05)
1000	18.20	2.17e-05	(1.89e-05, 2.46e-05)
1500	16.95	1.16e-05	(8.93e-06, 1.44e-05)

Table 7.6: Simulation results for the FGN trace (empirical effective bandwidth only)

Buffer	θ^*	$a(\theta^*, t^*, N)$	$\hat{P}(Q > b)$	CI
5	3.50	4.71	7.05e-06	(6.95e-06, 7.15e-06)
10	2.31	4.53	5.83e-06	(5.67e-06, 5.99e-06)
20	1.52	4.39	5.38e-06	(5.19e-06, 5.58e-06)
30	1.19	4.32	4.81e-06	(4.56e-06, 5.05e-06)
100	0.58	4.19	2.43e-06	(2.16e-06, 2.70e-06)
500	0.22	4.09	3.98e-06	(3.27e-06, 4.69e-06)
1000	0.14	4.06	0	(0, 0)

results at the time scale of interest.

Table 7.6 shows that the QoS constraint is met for all the considered buffer dimensions.

To obtain insights on the relevant time scale, in figure 7.17 we compared the curve resulting from the LR estimator with the direct estimation, taken at different time scales, namely $t = .271$ msec, $t = .542$ ms, $t = 1.084$ ms and $t = 2.168$ ms. Figure 7.17 seems to indicate that the bigger the value of θ the smaller the relevant time scale in terms of queuing performance. In particular figure 7.17 shows that for $\theta < 5.5$ the relevant time scale is smaller than $t = .271$ ms. For $2 < \theta < 5.5$ the relevant time scale is in the range $.271 \text{ ms} < t < .542 \text{ ms}$ and so on. Further investigation is necessary to determine this dependence in a more rigorous way.

Notice that equation (7.7) implies that the θ values of interest for given QoS and buffer sizes are higher in case of self-similar traffic. This is also apparent from the values reported in table 7.6 and explains why a virtual buffer estimator provides accurate results in a different range respect to what observed for on-off and Poisson traffic (see section 7.3).

7.8 Minimum Cost Traffic Shaping and Measured Effective Bandwidths

In this section we briefly outline how empirical effective bandwidths can be used in the proposed MCTS framework. To simplify the discussion below, we restrict ourselves to the

Figure 7.17: Relevant time scale for the FGN trace for queuing performance

block estimator unless stated otherwise. Also, we assume that the time parameter is given, or that the traffic streams have independent increments such that the time parameter in the effective bandwidth formulation is irrelevant.

Estimating effective bandwidths for MCTS is more complicated as compared to the estimation for CAC. In the latter, the network provides the QoS parameter θ , and the estimation algorithm therefore only has to estimate the effective bandwidth for the given parameter. For our MCTS framework however, we require the shape of the effective bandwidth *curve*. Each estimator thus has to cover a *range* of θ -values. This introduces the additional problem of determining the appropriate parameter range for θ . For bursty sources, the effective bandwidth is close to the peak rate for small values of θ . For less bursty sources, large θ values are required before the effective bandwidth is sufficiently close to the measured peak rate. Figure 7.18 illustrates the effective bandwidth of two on-off fluid sources with the same mean and peak rates, but different transition probabilities. Note how the critical θ -range for the bursty source is $(0, 2)$, whereas the critical θ -range for the less bursty source is $(0, 30)$.

Recall that under the assumption that the network has sufficient resources to admit the stream, the MCTS solution can be classified either into a delay / jitter allocation, a loss allocation, or a peak rate allocation (see figure 6.3). A delay allocation satisfies the delay constraint tightly, and is obtained as the solution to equation 6.17. A loss allocation satisfies

Figure 7.18: Illustration of the effective bandwidth of two on-off sources with different burstiness but the same mean and peak values

the delay constraint tightly, and is obtained as the solution to equation 6.16. The graphs for the relevant functions are depicted in figure 6.6. A peak rate allocation is characterized by the allocation $(0, \pi)$. It is effective when equation 6.16 does not yield a solution. Since this case is easily determined, we focus on the former two solution types. Recall also that empirical effective bandwidths have the desirable property of always tending to their observed peak rate. For this reason, the graph of the effective bandwidth for an on-off source is typical *for all finite-time EEBs*.

Due to these observations, we propose two alternative methods to solve the MCTS using measured effective bandwidths:

- Equivalent 2-state Markov fluid effective bandwidth
- Interpolation

Under the first method, we propose to fit the analytical form of a general 2-state Markov fluid to the estimated values for the effective bandwidth, given by Kesidis et al. [97] as

$$\alpha(\theta) = \frac{1}{2\theta}(-f_1(\theta) + \sqrt{f_1^2(\theta) - 4f_2(\theta)}) \quad (7.16)$$

where

$$\begin{aligned}
f_1(\theta) &= r_{1,2} + r_{2,1} - \theta(\phi_2 - \phi_1) \\
f_2(\theta) &= \theta^2 \phi_2 \phi_1 - \theta(r_{1,2} \phi_2 + r_{2,1} \phi_1)
\end{aligned}$$

Here $r_{i,j}$ are the transition rates between states i and j , and ϕ_i represents the arrival rate in state i , with $\phi_1 \leq \phi_2$. The motivation for this approach is that EEBs always resemble the analytical effective bandwidth for such a 2-state Markov fluid source. In particular, they are always at the mean rate for $\theta = 0$, and rise towards the observed peak rate as θ increases. Note that we do not claim that the arrivals process is equivalent to the arrival process from such a model. We only match that the empirical effective bandwidth with the analytical effective bandwidth of a 2-state Markov fluid. This allows us to obtain an expression for the effective bandwidth across the entire θ -range, not just the selected θ -values. Furthermore, we can use the analytical expressions for $\Lambda(\theta)$ and $u(\theta) = \theta^2 \frac{\partial \alpha(\theta)}{\partial \theta}$ of such a source to solve for the solution to the underlying MCTS problem. These are given by

$$\Lambda(\theta) = \frac{1}{2}(-f_1(\theta) + \sqrt{f_1^2(\theta) - 4f_2(\theta)}) \quad (7.17)$$

$$\theta^2 \frac{\partial \alpha(\theta)}{\partial \theta} = \theta \frac{\partial}{\partial \theta} \Lambda(\theta) - \Lambda(\theta) \quad (7.18)$$

Under the second method, we use the estimates of the empirical effective bandwidth to estimate the functions $\Lambda(\theta)$ and $u(\theta) = \theta^2 \frac{\partial \alpha(\theta)}{\partial \theta}$ directly. Given K estimation points $(\theta_k, a(\theta_k))$; $k = 1 \dots K$, we can estimate points on the CGF using

$$\hat{\Lambda}(\theta_k) = \theta_k a(\theta_k) \quad (7.19)$$

To obtain points on the second function $u(\theta)$ we have to estimate the slope of the empirical effective bandwidth function. The slope at point θ'_k can be estimated as

$$a'(\theta'_k) = \frac{a(\theta_{k+1}) - a(\theta_k)}{\theta_{k+1} - \theta_k}; k = 1 \dots K - 1 \quad (7.20)$$

where $\theta'_k = \frac{\theta_{k+1} + \theta_k}{2}$. The function $u(\theta)$ can then be approximated by

$$\hat{u}(\theta'_k) = (\theta'_k)^2 a'(\theta'_k) \quad (7.21)$$

For both solution types we then use a linear interpolation between successive points θ_k or θ'_k respectively; $k = 1 \dots K - 1$; to obtain the functional values for intermediary θ -values.

Both the curve fitting and the interpolation methods introduce additional delays to find the MCTS solution. The accuracy of each method depends on the number of estimation points θ_k ; $k = 1 \dots K$. Appropriate values for K thus have to be traded-off with the computation time required for each method to obtain a solution. We now compare each method for realizations of finite traces from Poisson, on-off, Gaussian and Pareto distributions. We assume that for each trace K θ -values have been used to estimate the EEB. The Poisson and on-off EEBs are taken from the estimation obtained in sections 7.3.1 and 7.3.3 respectively. For the Gaussian traffic stream we generated $N = 1000$ samples with mean $\mu = 40$ and $\sigma^2 = 8$. Similarly, in the Pareto-case, we generated $N = 1000$ samples with minimum value 4 and shape parameter 1.5. In these cases we assumed that each sample indicates the arrivals per slot and we used the block estimator with $t = 1$ to obtain estimates of the associated EEB.

For each source type we plot the fitted 2-state Markov fluid model against the EEB. This illustrates the accuracy of the curve fitting method. We also plot graphs for $\hat{\Lambda}(\theta)$ and $\hat{u}(\theta)$ to illustrate the smoothness than can be obtained with the interpolation method. In this case we make no further comparisons with respect to the accuracy, since we do not have an analytical expression for the EEB. Figures 7.19 to 7.26 illustrate these graphs for the Poisson, on-off, Gaussian and Pareto arrival streams respectively.

As shown in these graphs, the curve fitting method is able to achieve remarkable accuracy in most cases. Only in the case of Pareto arrivals is the fitted curve inaccurate. Experiments with different Pareto traces confirm this conclusion. Only in a few realizations of Pareto traces were we able to obtain a good fit. For some traces the fitted model did not even bound the EEB from above. The interpolated graphs for $\hat{\Lambda}(\theta)$ and $\hat{u}(\theta)$ also appear accurate and smooth enough to yield an acceptable MCTS solution. However, we point out that the accuracy of both methods is dependent on the suitable choice for the θ_k values. These should

Figure 7.19: Graph for the fitted EEB for the Poisson case.

Figure 7.20: Graphs for the interpolated EEB for the Poisson case.

Figure 7.21: Graph for the fitted EEB for the on-off case.

Figure 7.22: Graphs for the interpolated EEB for the on-off case.

Figure 7.23: Graph for the fitted EEB for the Gaussian case.

Figure 7.24: Graphs for the interpolated EEB for the Gaussian case.

Figure 7.25: Graph for the fitted EEB for the Pareto case.

Figure 7.26: Graphs for the interpolated EEB for the Pareto case.

be distributed in the ‘increasing’ part of the EEB curve, rather than its tail⁷. For the above evaluation we have used the following heuristic algorithm to determine $\theta_k; k = 1 \dots K$:

- Define K , a lower bound on the tail slope a'_{min} and set $\Delta\theta = 1$
- Set $k = 1, \theta_1 = 1, a(\theta_0) = -\infty$ and estimate $a(\theta_1)$.
- While $a(\theta_k) - a(\theta_{k-1}) > a'_{min}$, evaluate $k = k + 1, \theta_k = \theta_{k-1} + \Delta\theta$ and $a(\theta_k)$.
- Set $\Delta\theta = \frac{\theta_k}{N - \theta_k}$ and $\theta_{k-1} = \Delta\theta$.
- While $k \leq K$, evaluate $a(\theta_k), k = k + 1, \theta_k = \theta_{k-1} + \Delta\theta$, ignoring values of θ_k previously calculated.

This algorithm estimates the EEB at integer values of θ until the slope of the estimated curve is less than a'_{min} . The remaining points are then evenly distributed, starting at 0. In this way the desired K values for which the EEB is to be estimated covers the significant range. Of course, other algorithms may be used, possibly giving lower θ -values higher significance by using a logarithmic distribution of the θ s.

The computation times required under each method is shown in Table 7.7 for each of the traffic types. We also indicate the number of θ -values used to obtain the EEB. All evaluations were computed on a 155 MHZ Pentium PC using Mathematica 3.0 under Windows 95.

The computation times reported in table 7.7 clearly favor the interpolation method. Note that the computation times only apply to the curve fitting / interpolation methods described in this section. They do not include the time required to estimate $a(\theta_k)$. The order of magnitude of those computations was indicated in section 7.3. Neither are the times to obtain the MCTS solution included. This involves solving equations 6.16 or 6.17. Execution times for these equations were reported in section 6.3.2. Using the interpolation method, there is a strong indication that the total computation time required to obtain a solution to the MCTS problem is in the order of magnitude of a second or less. If a curve-fitting

⁷The KLD estimator achieves the desirable range for θ automatically, since it estimates the θ 's for the EEB between the mean and the peak rate.

Table 7.7: Computation times for the curve-fitting and interpolation methods.

Traffic Type	K	Computation Time (sec.)	
		Curve Fitting	Interpolation
Poisson	30	1.98	< 0.001
on-off	30	2.74	< 0.001
Gaussian	40	2.08	0.06
Pareto	40	1.26	< 0.001

method is used, this value changes to several seconds, possibly as much as 10 seconds. For this reason the curve-fitting methods seems inappropriate for an on-line solution approach.

Chapter 8

Conclusions and Suggestions for Future Research

8.1 Conclusions

In this thesis we have introduced the concept of minimum cost traffic shaping. This concept was derived by taking a user's perspective on connection admission control. In this framework, the *user* can formulate an optimization problem to determine the buffer and bandwidth resources required to a) satisfy the desired QoS and b) gain admission to the network. We show that the use of effective bandwidths to characterize the traffic stream and the QoS constraints leads to a standard form for this optimization problem.

The proposed view pushes much of the computational burden associated with CAC to the user. In particular, we require the user to know the effective bandwidth of the traffic stream. In our view, application developers could be required to provide such a characterization of the traffic stream generated by their application. Alternatively, the user could directly measure or estimate the effective bandwidth as indicated in the previous chapter. Furthermore, we assume that the user computes the MCTS solution. The advantages of this approach are severalfold. First, MCTS can be viewed as a dynamic scheme to establish the traffic contract. Given prices for resources and their availability, the user can determine the minimum cost resources required to satisfy the desired QoS and gain access to

the network. The traffic contract would then be reduced to such a buffer / bandwidth pair. Any traffic regulator associated with the user's stream would be parameterized accordingly. In this way, MCTS would minimize the negotiation process between the network and the user to establish the traffic contract. Second, the associated guaranteed service pricing scheme allows the user to foresee any charges resulting from the connection. The MCTS solution provides the user with valuable feedback, for example which of the constraints is binding or shadow prices. For the network, this pricing scheme recovers some of the costs for provisioning the resources by charging for QoS. Third, the traffic shaper is identified as a clear interface between the user and the network. In MCTS, the user is not restricted to selecting a member of a limited set of service categories. Instead, the user can freely determine the resource requirement for the stream. This provides valuable information to the network about the relative demands for the resources. Furthermore, it allows the network to engineer the service classes transparently from the user. We believe this is particularly important in light of new, QoS-intensive applications.

Effective bandwidths play an integral role in the proposed framework. We use it to characterize the traffic stream and show that the formulation of the QoS constraints in form of the effective bandwidth approximation is advantageous. Our analysis of estimated effective bandwidths particularly underlines the implications of finite time traces. The finiteness of traffic streams results in an effective bandwidth which is bounded above by the observed peak rate. Empirical effective bandwidth are thus *always* between the mean and the peak rate. This facilitates the characterization of a finite time stream, even in light of self-similar traffic. We find that the block estimator provides a reasonable estimate for the effective bandwidth.

In the derivation of the details of MCTS we paid special attention to the associated computational requirements. Our measurements of execution times for the various processes (measurements of effective bandwidths, optimization) indicates that the problem formulation can be computed in a reasonable on-line time frame (of the order of magnitude of a few seconds).

8.2 Suggestions for Future Research

The work presented in this thesis could be extended in a number of ways. We briefly outline a few suggestions for future research.

Integrated QoS environment : A complete formulation of MCTS optimization problem requires the estimation of the effective bandwidth, an optionally fitted effective bandwidth function and the numerical solution to the optimization problem. These three computations could be integrated into a single procedure. Moreover, such a procedure could be linked into the protocol hierarchy at the sender. If the MCTS solution characterizes the resource requirements at the packet level, adjustments have to be made to determine the resource requirements at the link level (i.e. taking link layer overhead, segmentation and queuing into account). Alternatively, if MCTS is integrated at the link layer, then the QoS requirements of higher layers have to be taken into account. A complete QoS-capable environment would also require the integration of QoS characteristics at the server, in case of a client-server application. Furthermore, a graphical user interface could be imposed to allow human users to vary the desired QoS, possibly in form of a slide bar on the user interface.

Demand experiments : Inspired by the INDEX experiment, we suggest an experiment to derive a demand function for buffer and bandwidth resources. Assuming a MCTS enabled environment, the network could alter the relative prices between buffer and bandwidth and observe the resulting demands from users. We would expect price changes not only to impact the relative use between the resources, but also to have an impact on the QoS demand. For example, higher prices may lead some users to demand a lower QoS to reduce their expenditure. Such a demand experiment would provide network operators with valuable information regarding the demand for network resources. It could be realized in a simulated environment, as currently developed at the broadband networks laboratory.

Critical Time Scales for Empirical Effective Bandwidths : Our main result for empirical effective bandwidth highlighted the importance of the finiteness of an arrival

stream. However, in our experiments we have selected the time parameter for the empirical effective bandwidth by inspecting the three-dimensional plot over both the θ and the time dimensions. We suggest further research to determine the critical time scale of traffic sources, particularly for finite-time self-similar traffic streams.

Appendix A

Description of ATM Service Categories

A.1 Constant Bit Rate Service (CBR)

The CBR service is intended for real-time applications which require a fixed bandwidth in order to guarantee a certain end-to-end transmission delay of the cells. Typical applications of this service would be video conferencing, voice calls, video / audio distribution or retrieval, as well as interactive video. These applications require a constant stream of cells to arrive with a tight delay constraint. Any cell which does not conform to these delay constraints is considered less valuable for the application and hence can be dropped.

This traffic type can be compared to a traditional circuit-switched traffic, where the complete end-to-end bandwidth from source to destination is exclusively reserved. This is not really the case in ATM networks, however, the perception to the user is the same, since the network also guarantees a fixed bandwidth.

The quality of service parameters required for this service class are PCR, CDVT, peak-to-peak CTD, maximum CTD and CLR.

This service class is handled by AAL1.

A.2 Real-time Variable Bit Rate Service (rt-VBR)

The rt-VBR service is similar to the CBR service in that it is intended for real-time traffic. Here however, the traffic is typically more bursty in nature, and hence lends itself to statistical multiplexing. The tight constraints on the end-to-end cell delay are still required.

The list of possible applications is therefore similar for this service class. It would cover those applications which transmit real-time traffic at intervals (bursts).

The QoS parameters required for this service class are PCR, SCR, CDVT (which may differ for PCR and SCR), MBS, peak-to-peak CDV, maximum CTD and CLR.

This service class is handled by AAL2.

A.3 Non real-time Variable Bit Rate Service (nrt-VBR)

This service category is the first of the three non real-time service categories. It is intended for bursty traffic types where cell loss is still not tolerated. The application expects and end-to-end transmission delay with a certain probability.

Typical applications are airline reservation applications, banking transactions or Frame Relay internetworking.

The QoS parameters required for this service category are PCR, CDVT, SCR, MBS, mean CTD and CLR.

This service class is handled by AAL3/4.

A.4 Unspecified Bit Rate Service (UBR)

The UBR service category is intended for traditional traffic types such as email or file transfer applications. These are typically characterized by varying transmission rates and may be statistically multiplexed. The end-to-end transmission delay of cells is not critical for applications subscribing to this service category.

This service category is somewhat an exception to those mentioned above, since the network here does not negotiate the bandwidth for each connection and also does not commit to any guarantees of cell delivery. For this reason the parameters negotiated between the

network and the user are limited to PCR and CDVT, and even those might not be considered for CAC and UPC.

This service class is handled by AAL5.

A.5 Available Bit Rate Service (ABR)

For traffic subscribing to the ABR service category, the network guarantees to use the available bandwidth for transmission. Since this bandwidth may vary over time, the transfer characteristics may change during the transmission phase of such traffic. The traffic is not assumed to be delay-sensitive. Furthermore, the application is expected to respond to the feedback provided by the network in form of a flow control function defined for this category. The source is expected to throttle its transmission in times of heavy network traffic. In return, the network guarantees reasonable cell loss ratios and a fair bandwidth allocations if the source 'behaves'.

The ATM Forum lists critical data transfers and super computer applications as typical candidates for this service [54].

The following parameters are specified for ABR traffic: PCR, CDVT, MCR, CLR and flow-control feedback. The ATM Forum does not specify a corresponding AAL for this class.

Appendix B

Brief Review of Kuhn-Tucker Conditions

This appendix provides a brief review of the derivation of the Kuhn-Tucker conditions (based on [124]). We start by deriving the Lagrangean first using the following equality constraint

Maximize $f(x, y)$ subject to $g(x, y) = c$

If both $f(\cdot)$ and $g(\cdot)$ are *differentiable*, then at the optimal solution to this problem (x^*, y^*) the tangents to both functions must have equal slope, i.e.

$$-\frac{df/dy}{df/dx}|_{(x^*, y^*)} = -\frac{dg/dy}{dg/dx}|_{(x^*, y^*)}$$

or

$$\frac{df/dy}{dg/dy}|_{y^*} = \frac{df/dx}{dg/dx}|_{x^*}$$

under the assumption that $\frac{dg}{dx}|_{x^*} \neq 0$ and $\frac{dg}{dy}|_{y^*} \neq 0$. If we let v be the common value of the slope of the tangents. The above condition can then be written as the following set of equations:

$$f'_y(x^*, y^*) - v g'_y(x^*, y^*) = 0$$

$$f'_x(x^*, y^*) - v g'_x(x^*, y^*) = 0$$

$$c = g(b^*, c^*)$$

where $f'_x(x^*, y^*) = \frac{df}{dx}|_{(x^*, y^*)}$ and similarly for $g(\cdot)$. We have included the last equation to ensure that the constraint is indeed satisfied. Note that the first two conditions equate the derivatives of the *Lagrangian* function $\mathcal{F} = f(x, y) - v(g(x, y) - c)$ with respect to x and y to zero.

Consider now the problem of an inequality constrained optimization

Maximize $f(x)$ subject to $g(x) \leq c$

Note that this problem is more general, since we can replace any equality constraint $g(x) = c$ by two inequality constraints $g(x) \leq c$ and $g(x) \geq c$. In the case of inequality constraints, the solution x^* can either satisfy the constraint or not. In the first case, we have just the problem discussed above, so the conditions are

$$\begin{aligned}\frac{d\mathcal{F}}{dx}|_{x^*} &= 0 \\ v(g(x^*) - c) &= 0 \\ g(x^*) &\leq c\end{aligned}$$

The second equation implies that either $v = 0$ or $g(x^*) - c = 0$ or both $v = g(x^*) - c = 0$. In the case where the solution does not satisfy the constraint with equality, we can set $v = 0$ to get $\frac{df}{dx} = \frac{d\mathcal{F}}{dx}$ for all x . This again satisfies the last set of conditions. Note that in this case the constraint at the optimal solution x^* is not binding, so we allow the function $f(x)$ to find its unconstrained maximum. At the point x^* we thus have $f'(x^*) = 0$ (the standard condition for an unconstrained maximum), which implies for the Lagrangian $\mathcal{F}'(x^*) = 0$. The above set of conditions is known as the Kuhn-Tucker conditions.

From the above argument we can formulate the sufficiency for the Kuhn-Tucker conditions. Suppose that $f(\cdot)$ is twice differentiable and quasi-concave and that $g(\cdot)$ is continuously differentiable and quasi-convex. If there exists a number v and a value x^* such that (x^*, v) satisfies the Kuhn-Tucker conditions, and if $f'(x^*) \neq 0$, then x^* solves the inequality-constrained maximisation problem.

Similarly, we can formulate the necessity for the Kuhn-Tucker conditions. Suppose that x^* solves the inequality-constrained maximization problem. Suppose further that $f(\cdot)$ and

$g(\cdot)$ are continuously differentiable, and that either $g(\cdot)$ is concave or $g(\cdot)$ is convex and there is some x such that $g(x) < c$ or $g(\cdot)$ is quasi-convex, $g'(x^*) \neq 0$ and there is some x such that $g(x) < c$. Then there exists a unique v such that (x^*, v) satisfies the Kuhn-Tucker conditions.

Thus, a set of conditions for which the Kuhn-Tucker conditions are both necessary and sufficient is that the objective function is concave and that the constraint is convex.

Note that the above results hold for a *set* of inequality constraints. In this case we have a Lagrange multiplier for each constraint. Note also that no conditions about the independence of constraints is imposed. Finally, we point out that any minimization problem can be converted into a maximization problem by multiplying the objective function by -1 .

Appendix C

MCTS Problems in AMPL Format

The following is a description of a MCTS problem in AMPL format. The user traffic is assumed to be Gaussian with mean $\mu = 4$ and variance $\sigma^2 = 0.7$. The network is assumed to have bandwidth capacity $C = 800$ packets / cells per time unit and a buffer capacity of $B = 1000$ packets / cells. The constant representing the statistical multiplexing gain inside the network is set to $\kappa = 0.001$. The optimization only considers losses. The required loss probability ϵ_L is set to 10^{-5} . We use a cost function where the user is charged for both the resources of the traffic shaper as well as the resources attributed to the stream inside the network according to the virtual dedicated queue model.

```
# MCTS Program
# EB based CAC with cost function TS resources
# can be submitted in this form to
#      http://www-neos.mcs.anl.gov/neos/solvers/NC0:DONLP2/solver-www.html
# Gaussian case
# alpha(th) = mu + th*sigma^2/2
# beta(th) = mu + th*sigma^2
# LAMBDA(th) = mu*th + th^2*sigma^2/2
# LAMBDA*(c) = (c - mu)^2/(2*sigma^2)

# PARAMETERS
```

```

param p{1..4};
param epl := 0.00001;
param kappa := 0.001;
param H := 1;
param B := 1000;
param thNW := 0.01;
param Cstar := 800;
param peakrate := 100;
param mu := 4;
param sigma := 0.8366; #corresponds to sigma^2=0.7

var th >= 1e-6, := 0.1;
var b >= 0.000001;
var c >= mu, <= peakrate;
var thd >= 1e-6, := 0.1;
var bd >= 0.00001;
var cd >= mu, <= peakrate;

minimize Cost:
    p[1]*b + p[2]*c + p[3]*bd + p[4]*cd;

s.t. LossTS: epl/(H+1) >= exp(-th * b);
s.t. LossNW: epl/(H+1) >= exp(-(thd + log(kappa)/B) * bd);
s.t. CAC: Cstar >= if (mu+thNW*sigma^2 <= c) then (mu + thNW*sigma^2/2)
                else (c - ((c-mu)^2/(2*sigma^2*thNW)));
s.t. EBNW: c == mu + th*sigma^2/2;
s.t. OEBNW: cd == if (mu+thd*sigma^2 <= c) then (mu + thd*sigma^2/2)
                else (c - ((c-mu)^2/(2*sigma^2*thd)));

data;

```

```
param p :=  
    1    1  
    2    1  
    3    1  
    4    20;
```

Appendix D

Table of Symbols

The following table provides a list of symbols used in each chapter along with a short description. Unless stated otherwise, the interpretation of symbols from previous chapters remains in effect.

Symbol Chapter Description

A_t	2	arrival process, either number of cells in slot t or aggregate number of arrival in interval $(0, t)$
A^*		worst case arrival process
$\mathcal{A}$		set of arrival processes
a		arrival time of a cell
$a(t)$		arrival curve, representing an upper bound on the total number of arrivals in the interval $(0, t)$
$b(.)$		burstiness curve
C		bandwidth capacity of a PSN
C		bandwidth allocated along the VP
d		departure time of a cell
f		virtual finishing time of a cell
H		total number of nodes along the end-to-end path
h		index for a particular node along the end-to-end path
i, j		index for cells, cell sequence number

Symbol Chapter Description

J	2	total number of arrival processes
j		index for arrival processes
N		number of FCFS queues in PSN
n		index of FCFS queue in PSN
Q		queue occupancy process
$s(t)$		service curve, representing a lower bound on the total number of departures from a queue in the interval $(0, t)$
t		principal index for time
t_1, t_2		limits for time intervals
W		workload process of a queue
γ		cell delay variation
δ		maximum end-to-end cell delay
ϵ		upper bounds on QoS probabilities
μ		minimum bandwidth property parameter
Π		total propagation and switching delay along the VP
$\Pi_{...}$		propagation and switching delay between two nodes along the VP
π		peak rate of an arrival stream
σ, σ'		burstiness of arrival process
ρ		rate of arrival process
ρ_n		proportional share of bandwidth capacity for queue n
$A(0, t)$	3	arrivals in the interval $(0, t)$
B		either fixed buffer size or buffer threshold
$C^{\mathcal{L}}_{...}$		bandwidth allocation to meet the loss requirement
$\mathcal{D}$		delay process
$\mathcal{J}$		jitter process
K		number of different classes of streams for CAC
k		index for different classes
T		finite time value, end of time interval

Symbol Chapter Description

t	3	time parameter
$\alpha^A(.)$		effective bandwidth of the arrival process
$\alpha^A(.)$		effective bandwidth of the aggregate arrival process
$\alpha^D(.)$		effective bandwidth of the departure process from a queue
$\alpha^W(.)$		effective bandwidth of the workload process of a queue
$\beta^A(.)$		decoupling bandwidth of arrival process A
γ		threshold for the tolerated jitter
δ		threshold for the tolerated delay
ϵ		positive bound on $\theta\alpha^W(\theta, \infty)$
ϵ_L		threshold on the tolerated loss probability
ϵ_D		threshold on the tolerated probability of the delay exceeding δ
ϵ_J		threshold on the tolerated probability of the jitter exceeding γ
ι		constant multiplicative factor for κ
κ		constant in front of the effective bandwidth approximation. degree of statistical multiplexing
$\Lambda_1^*(.)$		Legendre transform of the CGF
$\Lambda_1(.)$		CGF of the arrival process A
μ		mean arrival rate of a process
ω		exponential rate of decay for κ
θ		quality of service parameter
(θ^*, t^*)		parameter values for (θ, t) corresponding to the operating point of the link
ν		exponential decay rate in many-sources asymptotic
ϱ		quality of service requirement in many-sources asymptotic
σ^2		variance of the arrival process
φ		index of dispersion for intervals
B^P	4	capacity of the packet / cell buffer in a buffered leaky bucket
B^T		capacity of the token buffer in a buffered leaky bucket

Symbol Chapter Description

Q^P	4	occupancy of the packet / cell buffer in a buffered leaky bucket
Q^T		occupancy of the token buffer in a buffered leaky bucket
S_t		counter for the number of arrivals in an interval of length W slots
W		window size in policing algorithms
$\alpha^A(.)$		true effective bandwidth
$\beta^A(.)$		declared effective bandwidth
λ		arrival rates
θ		desired QoS parameter
$\hat{\theta}$		measured QoS parameter
ζ		weighting factor in exponentially weighted moving average policing algorithm
$ben(t)$	5	benefit function
C_J		capacity for each resource in J
c		cost for setting up a connection
D_F		demand under flat pricing
D_U		demand under usage-based pricing
$d(.)$		delay function
$f(.)$		function for effective bandwidth pricing
H		measured peak rate
$h(.)$		function measuring the benefit decay resulting from delay
J		collection of resources
m		measured mean rate
p		price
p_1, p_2		prices
TC		total cost
t		time
U		utility
$UC(\rho, m)$		charging function based on the declared ρ and the measured m

Symbol Chapter Description

V	5	performance degradation
v		benefit from immediate transmission
x		number of packets
$Z(\dots)$		moment generating function of arrival stream
z		user-declared value in effective bandwidth pricing
$\alpha(\dots)$		effective bandwidth for the arrival stream
μ		declared mean rate
π		declared peak rate
ν		Lagrange multipliers
A	6	indicator matrix to denote if resource j is required to carry user r 's flow
B		buffer size inside the network
$B(\cdot)$		user's budget
B^*		available buffer capacity in a network node
b		buffer size of the traffic regulator
b^D		buffer capacity of virtual dedicated queue. proxy for buffer inside the network
b^*		MCTS parameter for the regulator's buffer size
C		allocated service rate
C^*		available bandwidth capacity in a network node
$\mathcal{C}(\cdot)$		total cost to the user
c		rate of the traffic regulator
c^D		bandwidth capacity of virtual dedicated queue. proxy for bandwidth inside the network
c^*		MCTS parameter for the regulator's rate provided by the network
d_{VP}		end-to-end delay of a VP
$F_T(t)$		SRP renewal function
$\mathcal{F}$		Lagrangian function

Symbol Chapter Description

$f(b, c, \cdot)$	6	general functional representation for the loss constraint
$g(b, c, \cdot)$		general functional representation for the delay constraint
H		number of hops along the VP
$\mathcal{H}$		Hurst parameters
h		index for a particular node along the VP
$h(b, c, \cdot)$		general functional representation for the jitter constraint
J		number of arrival streams into a particular network node
j		index for arrival streams
k		arbitrary constant. used to explain quasi-convexity / quasi-concavity
m		measured mean rate of the arrival process
n		sample size
p_1, p_2, p_3, p_4		prices
Q		queue occupancy for the regulator
$q[\cdot]$		arbitrary function. used to explain quasi-convexity / quasi-concavity
R		set of users
R		transition probability matrix for an on-off source
$r_{i,j}$		entries of R . transition rates
$r(k)$		autocorrelation function for a self-similar FGN process
t		time index. also used in charging function
x_r		transmission rate for user r
y		slack variable for a constraint
z		arbitrary variable. used to explain quasi-convexity / quasi-concavity
$\alpha(\cdot)$		effective bandwidth of the arrival stream
$\alpha^A(\cdot)$		effective bandwidth of the user's arrival process
$\alpha^D(\cdot)$		effective bandwidth of the departure process from a traffic regulator
$\beta^A(\cdot)$		decoupling bandwidth of the arrival process
γ		tolerated jitter
δ		tolerated delay

Symbol Chapter Description

ϵ_D	6	upper bound for the tolerated probability of the delay exceeding δ
ϵ_J		upper bound for the tolerated probability of the jitter exceeding γ
ϵ_L		upper bound for the tolerated loss probability
κ		factor indicating a higher statistical multiplexing gain
$\Lambda(.)$		CGF of the arrival process
$\Lambda^*(.)$		Legendre transform of $\Lambda(.)$
λ		parameter for Poisson distribution
μ		mean rate of the user's arrival process
Π		end-to-end propagation and switching delay of a VP
ω		factor in the renewal function
π		peak rate of the user's arrival process
ρ		rate of the arrival stream
σ		burstiness of the arrival stream
σ^2		variance of the arrival process, used for Gaussian distribution
θ		parameter indicating the QoS for the user
θ^*		QoS parameter θ which corresponds to the MCTS solution
θ^{NW}		QoS parameter
ψ		variable QoS parameter: θ is the supremum of ψ
ν		Lagrange multipliers
$l(.)$	7	indicator function
A_i		number of arrivals in slot i
A_{max}		measured maximum number of arrivals per slot
$\tilde{A}_l$		block sum of arrivals
$A(i, j)$		total number of arrivals in the interval (i, j)
$A(t, i)$		independent samples of arrivals in an interval of length t
		used to explain distribution free CIs
$a(.)$		empirical effective bandwidth
$a'(\theta)$		slope of the EEB

Symbol Chapter Description

B_1	7	buffer threshold used in the KLD estimator
b		buffer variable used in a summation
$f_i(\theta)$		components of the analytical expression for a 2-state Markov fluid
g		constant factor for the truncated Poisson distribution
$\mathcal{H}$		Hurst parameter
i, k		time indices
K		total number of θ -values used in estimation of EEB
k		index for number of θ -values used in estimation of EEB
M		largest bin in the histogram for the queue size
m		sample mean of a given arrival trace
N		sample size, trace length
q		parameter determining the width of the distribution-free CI: $q = -\log(1 - CIwidth)$
$r_{i,j}$		transition rate of 2-state Markov fluid
s^2		sample variance of a given arrival trace
t		time variable
t^*		relevant time scale, time scale which is most likely to lead to a buffer overflow
t_N		arrival time of the last sample
$w(i, N)$		weights, between 0 and 1, used to explain distribution free CIs
$\hat{\alpha}(\cdot)$		measured effective bandwidth
$\alpha(\cdot)$		analytical effective bandwidth
$\underline{\hat{\alpha}}(\cdot)$		lower bound for CI
$\overline{\hat{\alpha}}(\cdot)$		upper bound for CI
$\epsilon(N)$		error term for distribution free CI
π		peak rate of arrival stream
$\pi(b)$		stationary probability of the queue length being b
τ		index for time

Symbol	Chapter	Description
θ^*	7	QoS parameter determined from the effective bandwidth approximation indicating the required QoS for the user
$\hat{\theta}$		estimated QoS parameter using the KLD or LR methods
ϕ		arrival rates of 2-state Markov fluid

θ^*	7	QoS parameter determined from the effective bandwidth approximation indicating the required QoS for the user
$\hat{\theta}$		estimated QoS parameter using the KLD or LR methods
ϕ		arrival rates of 2-state Markov fluid

Bibliography

- [1] L. Anania and R. J. Solomon. Flat - the minimalist price. In L.W. McKnight and J. P. Bailey, editors, *Internet Economics*, pages 91-118. Cambridge, Massachusetts, 1997. MIT Press.
- [2] V. Anantharam and T. Konstantopoulos. Burst Reducing Properties of the Leaky Buckle Flow Control Scheme in ATM Networks. *IEEE Transactions on Communications*, 42(12):3085-3089, December 1994.
- [3] K. J. Arrow and A. C. Enthoven. Quasi-concave programming. *Econometrica*, 29:779-800, 1961.
- [4] J. Beran, R. Sherman, M.S. Taqqu, and W. Willinger. Long-Range Dependence in Variable-Bit-Rate Video Traffic. *IEEE Transactions on Communications*, 43(2-4):1566-1579, 1995.
- [5] A. Berger. Performance Analysis of a Rate-Control Throttle where Tokens and Jobs Queue. *IEEE Journal on Selected Areas in Communications*, 9(2):195-170, February 1991.
- [6] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, New Jersey 07632, second edition, 1992.
- [7] S. Blake, D. Black, M. Carlson, E. Davies, and Z. Wang. An Architecture for Differentiated Services. *IETF RFC2475*, December 1998. available from URL www.ietf.org/html.charters/diffserv-charter.html.

- [8] J. A. Bucklew. *Large Deviation Techniques in Decision, Simulation and Estimation*. John Wiley & Sons, Inc., Toronto, 1990.
- [9] M. Butto, E. Cavallero, and A. Tonietti. Effectiveness of the Leaky-Bucket Policing Mechanism in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(3):335-342, April 1991.
- [10] C. S. Chang. Stability, Queue Length, and Delay of Deterministic and Stochastic Queueing Networks. *IEEE Transactions on Automatic Control*, 39(5):913-931, May 1994.
- [11] C. S. Chang. On Deterministic Traffic Regulation and Service Guarantees: A Systematic Approach by Filtering. *IEEE Transactions on Information Theory*, 44(3):1097-1110, May 1998.
- [12] C. S. Chang and J. A. Thomas. Effective Bandwidth in High-Speed Digital Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1091-1100, August 1995.
- [13] C. S. Chang and T. Zajic. Effective Bandwidths of Departure Processes from Queues with Time Varying Capacities. *IEEE Infocom'95, Boston, MA*, pages 1001-1009, 1995.
- [14] G. L. Choudhury, D. M. Lucantoni, and W. Whitt. Squeezing the Most Out of ATM. *IEEE Transactions on Communications*, 44(2):203-217, February 1996.
- [15] D. D. Clark. Combining Sender and Receiver Payments in the Internet. *Telecommunications Research Policy Conference*, October 1996.
- [16] D. D. Clark. Internet cost allocation and pricing. In L.W. McKnight and J. P. Bailey, editors, *Internet Economics*, pages 216-252. Cambridge, Massachusetts, 1997. MIT Press.
- [17] D. D. Clark, S.J. Shenker, and L. Zhang. Supporting real-time Applications in an Integrated Services Packet Network: Architecture and Mechanism. *Proceedings of the ACM Sigcomm '92*, pages 14-26, 1992.

- [18] R. Cocchi, S. Shenker, D. Estrin, and L. Zhang. Pricing in Computer Networks: Motivation, Formulation and Example. *ACM/IEEE Transactions on Networking*, 1:614–627, 1993. available from <ftp://ftp.parc.xerox.com/pub/net-research/pricing1.ps.Z>.
- [19] R. Cocchi, D. Estrin, S. Shenker, and L. Zhang. A Study of Priority Pricing in Multiple Service Class Networks. In *Proceedings of Sigcomm '91*. Oxford University Press, September 1991. available from <ftp://ftp.parc.xerox.com/pub/net-research/pricing2.ps.Z>.
- [20] M. Conti, E. Gregori, and A. Larsson. On the Relevance of Long Term Correlation in MPEG-1 Video Traffic. In *Proceedings of the International IFIP-IEEE Conference on Broadband Communications*, pages 52–63, 1996.
- [21] C. Coucoubetis and G. Fouskas. On the Performance of an Effective Bandwidth Formula. In *Proceedings of International Teletraffic Conference ITC 14*, pages 201–212, 1994.
- [22] C. Coucoubetis, F. Kelly, and R. Weber. Measurement-based Usage Charges in Communication Networks. *Statistical Laboratory Research Report 1997-19*, University of Cambridge, to appear in *Operations Research*, 1997. available from URL www.statslab.cam.ac.uk/Reports/1997/1997-19.html.
- [23] C. Coucoubetis and V. A. Siris. An Evaluation of Pricing Schemes that are based on Effective Usage. In *Proc. of IEEE International Conference on Communications (ICC'98)*, Atlanta, Georgia, USA, June 1998. available from URL www.ics.forth.gr/netgroup/publications/.
- [24] C. Coucoubetis, V. A. Siris, and G. D. Stamoulis. Integration of Pricing and Flow Control for Available Bit Rate Services in ATM Networks. In *Proc. of IEEE Globecom '96*, London, UK, November 1996. available from URL www.ics.forth.gr/netgroup/publications/.
- [25] C. Coucoubetis, V. A. Siris, and G. D. Stamoulis. Application and Evaluation of Large Deviation Techniques for Traffic Engineering in Broadband Networks. *Pro-*

- ceedings of ACM Sigmetrics '98/Performance '98 Joint International Conference on Measurement and Modeling of Computer Systems*, pages 1–10, June 1998. available from URL www.ics.forth.gr/netgroup/publications/.
- [26] C. Coucoubetis, V. A. Siris, and G. D. Stamoulis. Application of the many sources asymptotic and effective bandwidths to traffic engineering. *Telecommunication Systems*, 12:167–191, 1999. available from URL www.ics.forth.gr/netgroup/publications/.
- [27] C. Coucoubetis, G. D. Stamoulis, C. Manolakis, and F. Kelly. An Intelligent Agent for Optimizing QoS-for-Money in Priced ABR Connections. *To appear in Telecommunication Systems, Special Issue on Network Economics*, 1998. available from URL www.ics.forth.gr/netgroup/charging/#PUBLICATIONS.
- [28] C. Coucoubetis and R. Weber. Effective Bandwidths for Stationary Sources. *Probability in Engineering and Informational Sciences*, 1995. available from URL www.ics.forth.gr/proj/race/publications/index.html#journals.
- [29] C. Coucoubetis and R. Weber. Buffer Overflow Asymptotics for a Buffer handling many Traffic Sources. *Journal of Applied Probability*, 33:886–903, 1996.
- [30] C. Courcoubetis, G. Kesidis, A. Ridder, and J. Walrand. Admission Control and Routing in ATM Networks using Inferences from Measured Buffer Occupancy. *IEEE Transactions on Communications*, 43(2/3/4):1178–1784, February/March/April 1995.
- [31] R. C. Cruz. A calculus of network delay, Part 1: Network Elements in isolation. *IEEE Transactions on Information Theory*, 37:114–131, January 1991.
- [32] R. C. Cruz. Quality of Service Guarantees in Virtual Circuit Switched Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1048–1057, August 1995.
- [33] G. deVeciana. Leaky Buckets and Optimal Self-Tuning Rate Control. In *Proceedings of IEEE Globecom '94*, pages 1207–1211, 1994.

- [34] G. deVeciana, C. Courcoubetis, and J. Walrand. Decoupling Bandwidths for Networks: A Decomposition Approach to Resource Management. In *Proc. IEEE Infocom*, pages 466–473, 1994.
- [35] G. deVeciana and G. Kesidis. An Approach to Provisioning for Real-Time VBR Video Teleconferencing via End-to-End Virtual Path Connections. In *Proc. IEEE International Conference on Communications*, pages 632–636, 1996.
- [36] G. deVeciana and G. Kesidis. Bandwidth Allocation for Multiple Qualities of Service Using Generalized Processor Sharing. *IEEE Transactions on Information Theory*, 42(1):268–272, January 1996.
- [37] G. deVeciana, G. Kesidis, and J. Walrand. Resource Management in Wide-Area ATM Networks Using Effective Bandwidths. *IEEE Journal on Selected Areas in Communications*, 13(6):1081–1090, August 1995.
- [38] L. Dittmann, S. Jacobsen, and K. Moth. Flow Enforcement Algorithms for ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(3):343–350, April 1991.
- [39] G. Doubrovina, M. Falkner, M. Devetsikiotis, and I. Lambadaris. Optimal Cost Traffic Shaping with Self-Similar Input Sources. *Proceedings of IEEE Globecom '99, Rio de Janeiro, Brazil*, December 1999.
- [40] N. G. Duffield, J.T. Lewis, and N. O'Connell. Predicting Quality of Service for Traffic with Long-Range Fluctuations. *IEEE ICC'95*, pages 473–477, 1995.
- [41] N. G. Duffield and N. O'Connell. Large Deviations and overflow probabilities for the general single-server queue, with applications. *Math. Proc. Cambridge Phil. Soc.*, pages 363–375, 1995.
- [42] N.G. Duffield, J.T.Lewis, N.O'Connell, R. Russell, and F. Tomey. Entropy of ATM Traffic Streams: A Tool for Estimating QoS Parameters. *IEEE Journal on Selected Areas in Communications*, 13(6):981–990, August 1995.

- [43] Z. Dziong, M. Juda, and L. G. Mason. A Framework for Bandwidth Management in ATM Networks - Aggregate Equivalent Bandwidth Estimation Approach. *IEEE / ACM Transactions on Networking*, 5(1):134–147, February 1997.
- [44] J.G. Ecker and M. Kupferschmid. *Introduction to Operations Research*. John Wiley & Sons, 1988.
- [45] R. Edell, N. McKeown, and P. Varaiya. Billing Users and Pricing for TCP. *IEEE Journal on Selected Areas in Communications*, 13(7):1–14, September 1995.
- [46] R. Edell and P. Varaiya. Providing Internet Access: What we learn from INDEX. *Submitted to IEEE Network*, April 1999. available from URL www.INDEX.Berkeley.EDU/public/index.phtml.
- [47] A. Elwalid and D. Mitra. Stochastic Fluid Models in the Analysis of Access Regulation in High Speed Networks. In *Proceedings of IEEE Globecom '91*, pages 1626–1632, 1991.
- [48] A. Elwalid and D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, June 1993.
- [49] A. Elwalid, D. Mitra, and R. Wentworth. A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node. *IEEE Journal on Selected Areas in Communications*, 13(6):1115–1127, August 1995.
- [50] M. Falkner, M. Devetsikiotis, and I. Lambadaris. Issues on Fast Network Simulation by use of Effective and Decoupling Bandwidths. *ACM Transactions on Modeling and Simulation*, 9(1):45–58, January 1999.
- [51] P. Ferguson and G. Huston. *Quality of Service: delivering QoS on the Internet and in Corporate Networks*. J. Wiley & Sons, Inc., New York, USA, 1st edition, 1998.
- [52] P. C. Fishburn and A. M. Odlyzko. Dynamic Behavior of Differential Pricing and Quality of Service Options for the Internet. *Proc. First Intern. Conf. on Information and Computation Economics (ICE-98)*, pages 128–139, June 1998. available from URL www.research.att.com/~amo.

- [53] S. Floyd. submitted to Comments on Measurement-based Admission Control for Controlled Load Services. In *CCR*. available from URL www-nrg.ee.lbl.gov/nrg-papers.html. July 1996.
- [54] The ATM Forum. Traffic Management Specification 4.0. 1996. available from URL www.atmforum.com/atmforum/specs/approved.html.
- [55] W. Y. Fu and H. Latchman. A self-similarity based Call Admission Control Scheme for Output Buffered ATM Switches. In *7th IEEE Workshop on Computer-Aided Modeling, Analysis and Design of Communications Networks and Links (CAMAD'98)*, Rio de Janeiro. August 1998.
- [56] J. Gallardo, D. Makrakis, and L. Orozco-Barbosa. Use of α -stable Self-Similar Stochastic Processes for Modeling Traffic in Broadband Networks. *Performance Evaluation*, 40:71–98, 2000.
- [57] R. Gibbens and P. J. Hunt. Effective Bandwidths for the multi-type UAS channel. *Queueing Systems*, 9:17–27, 1991.
- [58] R. J. Gibbens and F. P. Kelly. Measurement-based connection admission control. *Proceedings of the 15th International Teletraffic Congress*, June 1997.
- [59] R. J. Gibbens and F. P. Kelly. Resource Pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999. available from URL www.statslab.cam.ac.uk/~frank/evol.html.
- [60] R. J. Gibbens, F. P. Kelly, and P. Key. A Decision-Theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1101–1113, August 1995.
- [61] R. J. Gibbens and Y.-C. Teh. Critical time and space scales for statistical multiplexing in multiservice networks. In *Proceedings of the 16th International Teletraffic Congress – ITC 16*, 1999. Accepted for publication.

- [62] R.J. Gibbens. Traffic characterisation and effective bandwidths for broadband network traces. *Statistical Laboratory Research Report 1996-9, University of Cambridge.*, 1996. available from URL www.statslab.cam.ac.uk/cgi-bin/resreps.pl?term=1996&field=year.
- [63] S. Giordano, M. Pagano, R. Pannocchia, and F. Russo. A New Call Admission Control Scheme Based on the Self Similar Nature of Multimedia Traffic. In *Proc. IEEE ICC*, pages 1612–1618, 1996.
- [64] S.J. Golestani. A self-clocked fair queueing scheme for broadband applications. In *Proceedings of IEEE Infocom 1994*, pages 636–646, 1994.
- [65] T. R. Griffiths. Analysis of a Connection Acceptance Strategy for Asynchronous Transfer Mode Networks. *Proceedings of the IEEE Global Conference on Communications, Globecom '90*, pages 862–867, 1990.
- [66] M. Grossglauser and D. Tse. Measurement-based Call Admission Control: A Heavy Traffic Framework. *Proceedings of the 36th Conference on Decision and Control, San Diego, California, USA*, pages 1792–1797, December 1997.
- [67] M. Grossglauser and D. Tse. A Framework for Robust Measurement-Based Admission Control. *IEEE/ACM Transactions on Networking*, 7(3), 1999.
- [68] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas in Communications*, 13(7):968–981, September 1991.
- [69] F. Guillemin and J.W. Roberts. Jitter and Bandwidth Enforcement. In *Proceedings of IEEE Globecom '91*, pages 261–265, 1991.
- [70] A. Gupta, D. O. Stahl, and A. B. Whinston. Pricing of services on the internet. In F. Phillips and W.W. Cooper, editors, *IMPACT: How ICC Research Affects Public Policy and Business Markets. A Volume in Honor of G. Kozmetsky*, Quorum Book, CT, MIT Press, forthcoming, available from URL cism.bus.utexas.edu/.

- [71] A. Gupta, D. O. Stahl, and A. B. Whinston. Priority pricing of integrated services networks. In L.W. McKnight and J. P. Bailey, editors, *Internet Economics*, pages 323–352. Cambridge, Massachustetts, 1997. MIT Press.
- [72] L. Györfy, A. Rácz, K. Duffy, J.T. Lewis, and F. Toomey. Distribution-free Confidence Intervals for Measurements of Effective Bandwidth. *Journal of Applied Probability*, pages 1–20, 1999. To appear. Also DIAS Technical Report DIAS-STP-99-06 available from URL www.stp.dias.ie/APG/1999.html.
- [73] Q. Hao, S. Tartarelli, and M. Devetsikiotis. Design and Optimization of High Speed Multiservice Networks: A Self-Sizing Methodology. *Presented at the 5th INFORMS Telecommunications Conference, Boca Raton, Florida.*, March 5–8 2000.
- [74] J. Heianen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. *IETF RFC2597*, June 1999. available from URL www.ietf.org/html.charters/diffserv-charter.html.
- [75] M. K. Honig and K. Steiglitz. Usage-Based Pricing of Packet Data generated by a heterogeneous user population. In *Proceedings of IEEE Infocom'95, Boston, MA*, April 1995.
- [76] C. Huang, M. Devetsikiotis, I. Lambadaris, and A. R. Kaye. Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach. In *Proceedings of ACM Sigcomm 95*, pages 114–125, 1995.
- [77] A. Hung and G. Kesidis. Bandwidth scheduling for Wide-Area ATM Networks Using Virtual Finishing Times. *IEEE/ACM Transactions on Networking*, 4(1):49–54, February 1996.
- [78] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. *IETF RFC2598*, June 1999. available from URL www.ietf.org/html.charters/diffserv-charter.html.

- [79] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang. A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks. *IEEE / ACM Transactions on Networking*, 5(1):56-69, February 1997.
- [80] S. Jamin, S. J. Shenker, L. Zhang, and P. B. Danzig. An Admission Control Algorithm for predictive real-time service (extended abstract). *Proceedings of the 3rd Int. Network and Operating Systems Support for Digital Audio and Video Workshop*, November 1992.
- [81] H. Jiang and S. Jordan. Connection Establishment in High-Speed Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):1150-1161, September 1995. available from URL www.eng.uci.edu/~sjordan/pubs/Pricing/index.htm.
- [82] H. Jiang and S. Jordan. The Role of Price in the Connection Establishment Process. *European Transactions on Telecommunications and Related Technologies*, 6(4), July-August 1995. available from URL www.eng.uci.edu/~sjordan/pubs/Pricing/index.htm.
- [83] J.T.Lewis and R. Russell. An Introduction to Large Deviations for Teletraffic Engineers. pages 1-37, October 1996. available from URL www.stp.dias.ie/~russell/.
- [84] S. Kalyanaraman and T. Ravichandran. Dynamic Capacity Contracting: A framework for Pricing the Differentiated Services Internet. *First International Conference on Information and Computation Economics*, May 1998. Submitted.
- [85] F. Kelly. Effective Bandwidths at Multi-class Queues. *Queueing Systems*, 9:5-16, 1991.
- [86] F. Kelly. On Tariffs, policing and admission control for multiservice networks. *Operations Research Letters* 15, pages 1-9, 1994. available from URL www.statslab.cam.ac.uk/~frank/PAPERS/tariff.html.
- [87] F. Kelly. Tariffs and effective bandwidths in multiservice networks. *International Teletraffic Conference ITC'14*, pages 401-410, 1994.

- [88] F. Kelly. Notes on Effective Bandwidths. In *Royal Statistical Society Lecture Notes Series*, volume 4, pages 141–168. Oxford, 1996. Oxford University Press. available from URL www.statslab.cam.ac.uk/~frank/eb.html.
- [89] F. Kelly. Charging and accounting for bursty connections. In L.W. McKnight and J. P. Bailey, editors, *Internet Economics*, pages 253–278. Cambridge, Massachusettts, 1997. MIT Press. available from URL www.statslab.cam.ac.uk/~frank/charge.html.
- [90] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997. available from URL www.statslab.cam.ac.uk/~frank/elastic.html.
- [91] F. Kelly. Rate control in communication networks. *Presentation to Nortel Networks, Kanata, Ontario*, November 6 1998.
- [92] F. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* 49, pages 237–252, 1998. available from URL www.statslab.cam.ac.uk/~frank/rate.html.
- [93] G. Kesidis. *ATM Network Performance*. Kluwer Academic Publishers, Boston, MA, USA, 1st edition, 1996.
- [94] G. Kesidis. Per-connection Management of Variable bit-rate traffic with an implication of Pricing. *EECE Technical Report, University of Waterloo, Canada*, June 1999. available from URL odysseus.uwaterloo.ca/~kesidis/papers.html.
- [95] G. Kesidis and T. Konstantopoulos. Extremal Traffic and Worst-Case Performance for Queues with Shaped Arrivals. *Proc. Workshop on Analysis and Simulation of Communication Networks, The Fields Institute*, November 1998. available from URL odysseus.uwaterloo.ca/~kesidis/papers.html.
- [96] G. Kesidis and T. Konstantopoulos. Performance Bounds for a Multiclass Queue with Deterministically Shaped Arrival Processes. *IEEE Communication Letters*, January 2000.

- [97] G. Kesidis, J. Walrand, and C. S. Chang. Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources. *IEEE ACM Transactions on Networking*, pages 424-428, August 1993.
- [98] K. Kilkki. *Differentiated Services for the Internet*. Macmillan Technical Publishing, 1st edition, 1999.
- [99] L. Kleinrock. *Queueing Systems Volume 1: Theory*. John Wiley Interscience, New York, USA, 1st edition, 1975.
- [100] T. Konstantopoulos and V. Anantharam. Optimal Flow Control Schemes that Regulate the Burstiness of Traffic. *IEEE/ACM Transactions on Networking*, 3(4):423-432, August 1995.
- [101] H. Kröner, M. Eberspächer, T.H. Theimer, P.J. Kühn, and U. Breim. Approximate analysis of the end-to-end delay in ATM networks. *Proceedings IEEE Infocom 1992*, 2:978-986, 1992.
- [102] W.C. Lau, A. Erramilli, J.L. Wang, and W. Willinger. Self-similar traffic generation: The Random Midpoint Displacement Algorithm and its Properties. In *Proceedings of ICC'95*, pages 446-472, June 1995.
- [103] S. Low and P. Varaiya. A simple theory of traffic and resource allocation in ATM. In *Proceedings of IEEE Globecom '91*, pages 1633-1637, 1991.
- [104] S. Low and P. Varaiya. A new approach to Service Provisioning in ATM Networks. *IEEE/ACM Transactions on Networking*, 1(5):547-553, 1993.
- [105] S. Low and P. Varaiya. Burstiness Bounds for Some Burst Reducing Servers. In *Proceedings Infocom '93*, pages 2-9, March 1993.
- [106] S. Low and P. Varaiya. Burst reducing servers in ATM networks. *Queueing Systems*, 20:61-84, 1995.
- [107] S. Low and P. Varaiya. Burst Reducing Servers in ATM Networks. *Queueing Systems*, 20:61-84, 1995.

- [108] J. MacKie-Mason, L. Murphy, and J. Murphy. Responsive pricing in the internet. In L.W. McKnight and J. P. Bailey, editors. *Internet Economics*, pages 279–303. Cambridge, Massachustetts, 1997. MIT Press.
- [109] J. MacKie-Mason and H. Varian. Pricing Congestibel Network Resources. *IEEE Journal on Selected Areas in Communications*, 13(7):1141–1149, September 1995.
- [110] J. K. Mackie-Mason and H. R. Varian. Pricing the Internet. In *International Conference on Telecommunication Systems Modelling*, pages 378–393, Nashville, TN, USA, March 1994. available from URL www.spp.umich.edu/papers/listing.html.
- [111] D. E. McDysan and D. L. Spohn. *ATM: Theory and Application*. McGraw-Hill, New York USA, 1st edition, 1995.
- [112] L.W. McKnight and B. Stiller. Charging and Accounting for Internet Services. *ACM Sigcomm '98*, September 1998. Tutorial T1.
- [113] J. Murphy and L. Murphy. Bandwidth Allocation by pricing in ATM networks. *Second International IFIP Conference on Broadband Communications, BB-94*, pages 333–351, march 1994. available from URL www.eeng.dcu.ie/~murphyj/publ/publ.html.
- [114] J. Murphy, L. Murphy, and E.C. Posner. Distributed Pricing for embedded ATM networks. In *Proceedings of the International Teletraffic Congress ITC-14*, pages 1053–1063, Antibes, France, June 1994. available from URL www.eeng.dcu.ie/~murphyj/publ/publ.html.
- [115] L. Murphy and J. Murphy. Feedback and Pricing in ATM networks. In *Proceedings of the IFIP TC6 Third Workshop on Performance Modelling and Evaluation of ATM Networks*, pages 68/1–68/12, Ilkley, England, July 1995. available from URL www.eeng.dcu.ie/~murphyj/publ/publ.html.
- [116] L. Murphy and J. Murphy. Pricing for ATM network efficiency. In *Proceedings of the 3rd International Conference on Telecommunication Systems Modelling and Analysis*, pages 349–356, Nashville, USA, March 1995. available from URL www.eeng.dcu.ie/~murphyj/publ/publ.html.

- [117] L. Murphy, J. Murphy, and J. MacKie-Mason. Feedback and Efficiency in ATM Networks. In *Proc. IEEE ICC*, pages 378–393. Ilkley, England, July 1995.
- [118] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. *IETF RFC2474*, December 1998. available from URL www.ietf.org/html.charters/diffserv-charter.html.
- [119] K. Nichols and B. Carpenter. Differentiated Services Operational Model and Definitions. *Internet Draft*, February 1998. available from URL www.ietf.org/internet-drafts.
- [120] K. Nichols, V. Jacobsen, and L. Zhang. A Two-Bit Differentiated Services Architecture for the Internet. *Internet Draft*, November 1998. available from URL diffserv.lcs.mit.edu/.
- [121] I. Norros. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE Journal on Selected Areas in Communications*, 13(6):953 - 962, 1995.
- [122] A. M. Odlyzko. The Economics of the Internet: Utility, utilization, pricing, and Quality of Service. July 1998. available from URL www.research.att.com/~amo.
- [123] A. M. Odlyzko. Paris Metro Pricing for the Internet. *Proc. ACM Conference on Electronic Commerce (EC'99)*, pages 140–147, 1999. available from URL www.research.att.com/~amo.
- [124] M. J. Osborne. Tutorial on optimization theory and difference and differential equations. 2000. available from URL www.chass.utoronto.ca/~osborne/MathTutorial.
- [125] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, New York, 1st edition, 1965.
- [126] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

- [127] C. Parris and D. Ferrari. A Resource-based Pricing Policy for Real-time Channels in a packet-switching Network. *International Computer Science Institute, Berkley, Technical Report tr-92-018*. 1992. available from URL www.icsi.berkeley.edu/techreports/index.html.
- [128] C. Parris, S. Keshav, and D. Ferrari. A Framework for the Study of Pricing in Integrated Networks. *International Computer Science Institute, Berkley, Technical Report tr-92-016*. 1992. available from URL www.icsi.berkeley.edu/techreports/index.html.
- [129] V. Paxson. Fast Approximation of Self-similar Network Traffic. *Technical Report LBL-36750/UC-405*. April 1995. available from URL ita.ee.lbl.gov/html/contrib/fft-fgn.c.html.
- [130] V. Paxson and S. Floyd. Wide Area Traffic: The Failure of Poisson Modeling. *IEEE ACM Transactions on Networking*, 3(3):226–244. June 1995.
- [131] H. G. Perros and K. M. Elsayed. Call Admission Control Schemes: A Review. *IEEE Communications Magazine*, pages 82–91. November 1996.
- [132] A. Ràcz. How to build robust Call Admission Control Based on On-Line Measurements. In *Proceedings of the IEEE Conference on Global Communications (GlobeCom '99), Rio de Janeiro, Brazil*, pages 1634–1640. December 1999.
- [133] E. Rathgeb. Policing Mechanisms for ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(3):325–334. April 1991.
- [134] J.W. Roberts, editor. *Performance Evaluation and Design of Multiservice Networks*. John Wiley & Sons, October 1991. Final Report, Cost 224 Project, Luxembourg: Commission of the European Communities.
- [135] H. Saito and K. Shiomoto. Dynamic Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):982–989. September 1991.
- [136] M. Schwartz. *Broadband Integrated Networks*. Prentice Hall, New Jersey, USA. 1st edition, 1996.

- [137] T. Shan and O. Yang. An efficient control scheme for the real-time VBR traffic in the ATM network: deterministic bandwidth allocation. In *Computer Communications*, to appear.
- [138] S. Shenker, D. Clark, D. Estrin, and S. Herzog. Pricing in Computer Networks: Reshaping the Research Agenda. *ACM Computer Communication Review*, pages 19–43, 1996.
- [139] S. Shenker, C. Partridge, and R. Guerin. Specification of the Controlled-Load Network Element Service. *IETF RFC2211*, September 1997. available from URL www.ietf.org/html.charters/intserv-charter.html.
- [140] K. Shiomoto and S. Chaki. Adaptive Connection Admission Control using Real-Time Traffic Measurements in ATM Networks. *IEICE Transactions on Communications*, E78-B(4):458–464, April 1995.
- [141] K. Shiomoto, S. Chaki, and N. Yamanaka. A Simple Bandwidth Management Strategy based on Measurements of instantaneous Virtual Path Utilization in ATM Networks. *IEEE/ACM Transactions on Networking*, 6(5), October 1998.
- [142] K. Shiomoto, N. Yamanaka, and T. Takahashi. Overview of Measurement-Based Connection Admission Control Methods in ATM Networks. *IEEE Surveys*, 2(1):2–13, January 1999. available from URL www.comsoc.org/pubs/surveys/.
- [143] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis*. Chapman & Hall Stochastic Modeling Series, 2–6 Boundary Row, London, SE1 8HN, UK, first edition, 1995.
- [144] M. Sidi, W.Z. Liu, I. Cidon, and I. Gopal. Congestion Control through Input Rate Regulation. In *Proceedings of IEEE Globecom '89*, pages 1764–1768, November 1989.
- [145] V. Siris. Large Deviations for Teletraffic Engineers, pages 1–37, October 1996. available from URL www.ics.forth.gr/netgroup/msa/.

- [146] V. A. Siris, D. J. Songhurst, G. D. Stamoulis, and M. Stoer. Usage-based charging using effective bandwidths: studies and reality. *16th International Teletraffic Congress (ITC-16)*, Edinburgh, UK, June 1999. available from URL www.ics.forth.gr/netgroup/charging/#PUBLICATIONS.
- [147] D.J. Songhurst, editor. *Charging Communication Networks: From Theory to Practice*. Elsevier, Amsterdam, The Netherlands, 1st edition, 1999.
- [148] P. Spelucci. DONLP2 User's Guide. *Technical University Darmstadt*, 1999. available from 'FTP://ftp.mathematik.tu-darmstadt.de/pub/departement/software/opti' or from URLs plato.la.asu.edu/donlp2.html, www-neos.mcs.anl.gov/neos/solvers/NCO:DONLP2/.
- [149] K. Sriram and W. Whitt. Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data. *IEEE Journal on Selected Areas in Communications*, 4(6):833-846, September 1986.
- [150] T. Taralp, M. Devetsikiotis, I. Lambadaris, and A. Bose. Efficient Fractional Gaussian Noise Generation Using the Spatial Renewal Process. In *Proceedings IEEE International Conference on Communications, Atlanta, USA*, June 1998.
- [151] T.E. Tedijanto and L. Gun. Effectiveness of Dynamic Bandwidth Management Mechanisms in ATM Networks. *Proceedings of the IEEE Infocom '93*, pages 358-367, 1993.
- [152] D.P. Tranchier, Boyer, Rouaud, and Mazenas. Fast Bandwidth Allocation in ATM Networks. *Technical Report, CNET-Lannion*, 1992.
- [153] B. Tsybakov and N. Georganas. On Self-Similar Traffic in ATM Queues: Definitions, Overflow Probability Bound and Cell Delay Distribution. *IEEE ACM Transactions on Networking*, 5(3):379-409, 1997.
- [154] J. S. Turner. Managing Bandwidth in ATM Networks with Bursty Traffic. *IEEE Network*, pages 50-58, September 1992.

- [155] R.A. Vesilo and J.W. Mark. ATM Call Admission using Burst Peakedness. *Proceedings of the 3rd Asia-Pacific Conference on Communications (APCC'97)*. Sydney, pages 1582–1586. December 1997. available from URL www.elec.mq.edu.au/~rein.
- [156] R.A. Vesilo and J.W. Mark. ATM Call Admission using Generalized Peakedness. *Proceedings of the IEEE Singapore International Conference on Networks (SICON'97)*. pages 1–15. April 1997.
- [157] R.A. Vesilo and V. Solo. Techniques for Adaptive Estimation of Effective Bandwidth in ATM Networks. in *Proceedings of Globecom '97, Phoenix*. pages 1344–1347. November 1997. available from URL www.elec.mq.edu.au/~rein.
- [158] J. Walrand and P. Varaiya. *High-performance Communication Networks*. Morgan Kaufmann. New Jersey, USA. 1st edition. 1996.
- [159] P. White. RSVP and Integrated Services in the Internet: A Tutorial. *IEEE Communications Magazine*. pages 100–106. May 1997.
- [160] H.P. Williams. *Model Building in Mathematical Programming*. J. Wiley and Sons. Chichester, UK. 3rd edition. 1990.
- [161] W. Willinger, R. Sherman M. Taqqu, and D. Wilson. Self-Similarity Through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. In *Proceedings of ACM Sigcomm 95*. pages 100–113. October 1995.
- [162] D. Wischik and A. Greenberg. Admission Control for Booking Ahead Shared Resources. In *Proceedings of IEEE Infocom '98*. pages 873–882. 1998.
- [163] M. Wong and P. Varaiya. A Deterministic Fluid Model For Cell Loss In ATM Networks. In *of ACM Sigcomm 95*. pages 395–400. October 1993.
- [164] J. Wroclawski. Specification of Guaranteed Quality of Service. *IETF RFC2212*. September 1997. available from URL www.ietf.org/html.charters/intserv-charter.html.

- [165] L. Zhang. VirtualClock: A new traffic control algorithm for packet-switched networks. *ACM Transactions on Computer Systems*. 9(2):101–124. May 1993.