Faculty of Engineering

Measurement Based Optimal Source Shaping in Integrated Services Packet Networks

A Thesis

Submitted for the Degree of

Master of Science (Engineering)

in the **Faculty of Engineering**

By

PARIJAT DUBE

Department of Electrical Communication Engineering

INDIAN INSTITUTE OF SCIENCE

BANGALORE – 560 012, (INDIA)

October 1999

DECLARATION

I hereby declare that the work reported in this thesis is entirely original. It was carried out by me in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, under the supervision of Professor Anurag Kumar. I further declare that it has not formed the basis for the award of any degree, diploma, membership, associateship or similar title of any University or Institution.

Parijat Dube
December, 1999
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore 560012, India.

Professor Anurag Kumar
December, 1999
Professor
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore 560012, India.

Contents

Abstract **xii**

List of Publications from this Thesis **xv**

Acknowledgements **xvi**

1 Introduction **1**

1.1 Information Flow Control in an ISPN 2

1.1.1 Admission Control, Shaping and Problem Motivation 4

1.2 Description of the Research Problem and Approach 6

1.3 Organisation of this Thesis 11

2 Literature Survey and Preliminary Background **13**

2.1 Introduction 13

2.2 Literature Survey 14

2.2.1 PIBCAC approaches 14

2.2.2 MBCAC approaches 15

3 Measurement Based Estimation of the Optimal Token Rate **19**

3.1 Introduction 19

3.2 The Leaky Bucket Shaper: A Review 20

3.3 Shaping for Minimum Cost Lossless Multiplexing: Characterisation of an Optimal Value of ρ 22

3.4 Issues in Measurement Based Estimation of ρ^* 29

3.5 Fast Simulation Based Stochastic Approximation Algorithm (FSSAA) 30

3.5.1 Review of Large Buffer Asymptotic Theory 31

3.5.2 Development of the Algorithm 32

3.5.3 Advantages of FSSAA 36

3.6 An On-Line Algorithm for Measurement Based estimation of ρ^* using FSSAA 38

3.6.1 Our algorithm and the conditions for convergence of RM algorithm . 39

3.6.2 The Algorithm 39

3.6.3 Detection of Change in Source Characteristics [41] 40

3.7 Algorithm Implementation and Simulation Results 42

3.7.1 Simulation Set-up 42

3.7.2 Simulation Results and Discussions 42

3.8 Conclusion 47

4 A Formulation for the Determination of the Optimal Bucket Size 49

4.1 Introduction 49

4.2 Shaping for Minimum Cost Lossy Multiplexing: Characterisation of an Optimal value of σ 50

4.3 Analytical Study 56

4.3.1 Preliminary Theory: 56

4.3.2 Proofs 58

4.4 An Off-Line Algorithm for determining σ^* 67

4.5 Algorithm Implementation and Simulation Results 69

4.5.1 Simulation Setup 69

4.5.2 Simulation Results and Discussions 71

4.6 An Approximate Analytical Calculation of σ^* 78

4.6.1 MMFP Characterisation of LB output 78

4.6.2 Determination of $c(\sigma)$ 80

4.6.3 Comparison of Analysis with Simulation 82

4.7 Conclusion 83

5 Measurement Based Estimation of Optimal Bucket Size 87

5.1 Introduction 87

5.2 The Optimisation Problem and the Overall Solution Approach 89

5.3 Ordinal Optimization 91

5.3.1 Terminology related to Ordinal Optimization [[57]] 92

5.3.2 Ordinal optimization as applied to our problem 93

5.4 Discrete Stochastic Optimization 95

5.4.1 Motivation for using discrete stochastic optimization 95

5.4.2 A Discrete Stochastic Optimization Algorithm 96

5.4.3 Our problem and ADSOA 99

5.5 The On-Line Optimization Algorithm 101

5.5.1 *Phase I:* Search Space Reduction 101
5.5.2 *Phase II:* Fine Tuning 106
5.6 Algorithm Implementation and Simulation Results 109
5.6.1 Simulation Set-up 109
5.6.2 Simulation Results 114
5.7 Conclusion 117

6 Related Issues **125**

6.1 Introduction 125
6.2 Renegotiation Issues 126
6.2.1 ρ^* estimation 127
6.2.2 σ^* estimation 128
6.3 A Measurement Based Call Admission Control Algorithm (MBCAC) based on ρ^* and σ^* 129
6.4 An Overall Block Diagram for Implementing our Proposed ρ^* and σ^* Algorithms 131
6.5 Non-Homogeneous Sources 132
6.5.1 Per Class Queueing: 133
6.6 Non-Linear Cost functions 134

7 Conclusion and Future Research Directions **135**

7.1 Future Research Directions 136

Bibliography **137**

List of Tables

3.1 Table showing the effect of the length of the measurement interval on the convergence of the ρ determination algorithm. The ρ^* obtained by EBW analysis is 139.1 43

4.1 Optimal shaper parameters and and per source multiplexer capacity and buffer for various approaches. $\Delta = 6$ and 9. 77

5.1 The rank dynamics table for our example. For each σ the corresponding ranks obtained at sampling points are shown. The term in () brackets gives the correct relative rank for the corresponding σ 116

List of Figures

1.1 A typical information flow pipe of ISPN 2

1.2 A typical network model 7

1.3 The network scenario under consideration. 8

1.4 The network model, showing the LB shapers and the network node. 9

3.1 The Leaky Bucket shaper. The buffer is infinite, but we want the probability of exceeding the buffer level B_s to be very small. 20

3.2 A single server queuing system, with service rate ρ and infinite buffer, that is equivalent to the (σ, ρ) leaky bucket shaper from the QVP (see text) point of view. 21

3.3 A typical $(B_s + \sigma)$ vs ρ trade-off curve; i.e., the function $h_{p_s}(B)$. Two state Markovian on-off source with mean on-time 5/3, mean off time 5/2, and peak rate $R = 170$; $p_s = 10^{-3}$. 22

3.4 Characterisation of ρ^*; $\overline{r}$ is the mean rate of the source. 24

3.5 Capacity requirement at the network node (c) as a function of total buffer (sum of cell buffer and token buffer) at the shaper (B); $\bar{r}$ is the mean rate of the source $S(t)$. 25

3.6 A typical $\log P(Q > B)$ Vs B curve. Note that the two main features of the curve are an intercept on the ordinate at $B = 0$, and a nearly constant asymptotic slope. 33

3.7 Obtaining the asymptotic slope by employing two virtual buffers B_1 and B_2. 34

3.8 By the use of the scaling property, the probability of occurrence of the required event (*cell-loss*) increases and hence the variance in loss measurements decreases. 36

3.9 By the use of two virtual buffers instead of one the estimate of asymptotic slope is much better. 37

3.10 By incorporating Intercept correction in the affine approximation to the curve of $\log P_{W,c}(Q > B)$, the calculated loss at a given buffer is much closer to the actual loss than that without Intercept correction. 38

3.11 Measurement windows for change detection. When the $-\ln(.)$ of means of *measured QVPs* in the two sliding windows differ by more than a predefined threshold, a change in source characteristics is assumed to have taken place. 41

3.12 A virtual buffer system for estimating ρ^*. Three virtual buffers at each source $S(t)$, are used to obtain an estimate of $-\ln p(\rho_k, T\rho_k)$, which is used for finding the next iterate of ρ^* using the RM algorithm. The actual LB parameters are updated only periodically after renegotiation. 42

3.13 An on-off Markon modulated source 43

3.14 The estimates of the measured QVP (ϵ) and the ρ^* as a function of the number of iterates for measurement intervals 10000, 1000 and 100 time units. 44

3.15 The estimates of the measured QVP (ϵ) and the ρ^* as a function of the number of iterates for two variable measurement interval techniques. Technique 1 works with 100 as the interval length upto 5 iterates and then shifts to 1000 for successive iterates. Technique 2 works with 100 as the first interval length and at each successive iterates it increase by 100 until the interval becomes 1000. For all successive iterates the interval length remains fixed at 1000. . . 46

3.16 The change detection capability of the windowing technique in the stochastic approximation algorithm based on loss measurements by the virtual buffer approach. The update time is 1000 time units. The value of ρ^* obtained by analysis for the four stationary regimes are 139.1, 166.3, 135.3, 165.8 and $\delta = 4$. 48

4.1 Network buffer (B_m) vs. per source capacity (c) trade-off curves at the multiplexer for *given* σ, $0 \leq \sigma \leq T\rho^*$. Shown are the lossless trade-off curve, the network delay constraint curve, and the lossy trade-off curve. 51

4.2 The feasible region for the network buffer vs per source capacity trade-off. The shaded region shows the (B_m, c) pairs that meet the CLR $\leq p_m$ as well as the delay requirement i.e., $\mathcal{R}$. 52

4.3 The regions $\mathcal{R}_{\sigma_1}, \mathcal{R}_{\sigma_2}, \mathcal{R}_{\sigma_3}$ with $\sigma_1 > \sigma_2 > \sigma_3$. 53

4.4 The shaded region shows the complete set of feasible (B_m, c) points for σ, i.e., the set $\mathcal{S}_\sigma$. 54

4.5 The extremal point for the approach followed in the proofs. 59

4.6 The cumulative output from (σ, ρ^*) and $(\sigma + \Delta, \rho^*)$ is bounded by $B_\sigma(0, t)$ and $B_{\sigma+\Delta}(0, t)$ respectively. 60

4.7 The arrangement for proving $Q_\sigma \leq Q_{\sigma+\Delta}$. 61

4.8 The Simulation Setup 70

4.9 Network buffer (B_m) vs per source capacity (c) tradeoff curves at the multiplexer for a *given* σ and n_1 and n_2, with $n_2 > n_1$. Shown are the lossy tradeoff curve (which is independent of n) and, the delay constraint and lossless tradeoff bound for n_1 and n_2. 71

4.10 The graph of per source extremal capacity and per source extremal buffer required at the mux vs σ for lossy multiplexing for $n = 50$ and $n = 150$. . . . 73

4.11 The per source extremal buffer-capacity tradeoff for lossy multiplexing for $n = 50$ and $n = 150$. 74

4.12 *Cost Curves* for $\Delta = 6$ and $\Delta = 9$ for $n = 50, 150$ showing the sensitivity of σ^* to Δ and n. 75

4.13 The two-state characterisation of the rate process at the input of the shaper 78

4.14 The approximate Markovian characterisation of the rate process at the output of the shaper 78

4.15 The Figure showing the characterisation of the maximum real root of Equation 4.24. 82

4.16 The comparison of approximate analysis with simulation of the per source extremal capacity Vs σ and the per source extremal buffer Vs σ curves for $n = 50$ and $n = 150$. 84

4.17 The comparison of approximate analysis with simulation of the per source extremal buffer-capacity tradeoff for $n = 50$ and $n = 150$. 85

4.18 The comparison of approximate analysis with simulation for $\Delta = 6$ of the *Cost Curves* for $n = 50$ and $n = 150$. 86

5.1 The curve showing the subset sizes required (s) for different alignment probabilities (P_A) by BP rule with $k = 1, g = 10, d = 1, \mid \Theta \mid= 691$ 95

5.2 The curve showing the subset sizes required (s) for different alignment probabilities(P_A) by BP rule with $k = 1, g = 10, d = 10, \mid \Theta \mid= 70$ 96

5.3 The three possible cases with σ^* being the true optima. 103

5.4 The two possible cases 105

5.5 A top-down representation of our optimisation problem depicting the motivation for using different techniques during the course of finding a solution . . 110

5.6 Equivalent Leaky Bucket Arrangements 111

5.7 Block diagram of the simulation set-up for *Phase I* of our algorithm 113

5.8 Block diagram of the Simulation Set-up for the *Phase II* of our algorithm at the mth iterate. Here σ_m^+ can be either $\sigma_m + 1$ or $\sigma_m - 1$ 114

5.9 The values of σ_m^* at the mth iterate obtained in the *Phase II* of our Algorithm. *off-line technique* gives a value of $\sigma^* = 110$. 117

5.10 The values of σ_m at the mth iterate obtained in *Phase II* of our Algorithm. An important feature to be observed is that the sequence $\{i_m\}(i_m = \sigma_m)$ also converges to σ^* as opposed to ADSOA where only $\{i_m^*\}$ converges and $\{i_m\}$ is a positive recurrent random walk. 118

5.11 Single leaky bucket shaper 119

5.12 Tandem arrangement of two leaky bucket shapers 120

5.13 The equivalent arrangement for L and $L_1 + L_2$ for $t \geq t_1$ 121

5.14 The equivalent arrangement for L and $L_1 + L_2$ for time t when $v_1 \leq t < w$. 122

6.1 The complete block diagram for implementing the ρ^* and σ^* determination Algorithm 131

6.2 Per homogeneous class based queueing at the multiplexer with P homogeneous classes and M_k being the number of sessions in the kth homogeneous class. . 133

Abstract

Integrated Services Packet Networks support a guaranteed transport service that is used by applications that have an intrinsic (unmodifiable) temporal behavior, and hence have tight end-to-end delay requirements. The traffic coming from a source is apriori unknown and usually does not have any deterministic bounds on the arrival statistics. Hence, invariably a source is shaped before feeding it into the network so that an enforceable traffic contract can be set up with the network.

A common procedure for shaping is the leaky bucket (LB) algorithm which provides deterministic bounds on the average rate (ρ) and the burst size (σ) to the shaper output. A LB shaped or (σ, ρ) conformant source can be provided a deterministic multiplexer delay guarantee of $\frac{\sigma}{\rho}$ by providing it a bandwidth equal to ρ and buffer equal to σ at the network node. Thus given the LB parameters the delay of the shaped source is bounded but the converse problem of determining the leaky bucket parameters for guaranteeing a given delay is more relevant for *stream traffic* due to their time varying and stringent delay requirements. Associated with each set of LB shaper parameters and delay requirements, there exists a value of network buffer and bandwidth such that a source's delay requirements are met. Also as the network resources (link bandwidth and buffers at the switches) are limited and scarce a cost is associated with their usage by the source (a source is charged for their use). If we will keep the shaper parameters to such a value so that for a given delay, the cost is minimised then the bandwidth and buffer corresponding to these parameters are *optimal* in the sense that:

- the "cost" is minimized.

- the delay is guaranteed.

The shaper parameters corresponding to these optimal value of network resources will be called **optimal shaper parameters**.

Apriori determination of optimal shaper parameters is clearly infeasible for on-line stream traffic (i.e., not stored). Hence, the best approach is measurement based *on-line* determination of parameters. In this thesis we are concerned with designing algorithms that can be implemented at the source, and/or at the switches for measurement based on-line estimation of optimal LB parameters ρ^* and σ^* for arbitrary sources. The problem we deal with has the following features:

- The network model consists of homogeneous sources shaped by a LB shaper and the shaped process is fed to the network. In this thesis we consider only a single hop network.

- Since the source will not conform to the LB, occasionally the source bits will need to be buffered in the LB and hence will get delayed. We are concerned with statistically bounding the delay in the shaper plus the delay in the network node (i.e., *shaping+multiplexing delay*).

- We work only with fluid model since the work was originally motivated by ATM. The arrival process to the shaper is inherently peak rate controlled and is assumed to be short range-dependent.

- ρ^* and σ^* are characterised as solutions of certain constrained optimization problems. Also, assumptions on the properties of various functions involved in the characterisation are justified through the EBW approach.

- Continuous and discrete stochastic optimization algorithm are employed for the on-line estimation ρ^* and σ^*. The simulations are done for fixed length packets.

- The cost function employed is linear in buffer and bandwidth.

We begin by formulating the problem of *lossless multiplexing* which results in the characterisation of ρ^* as the solution of an implicit equation. For lossless multiplexing the optimal value of σ^* is 0. We then developed a Measurement Based On-Line Algorithm for estimation of ρ^*. The algorithm tracks ρ^* for stationary as well as non-stationary sources. The convergence of the algorithm is studied through simulations.

We then consider the problem of *lossy multiplexing* for characterising $\sigma^* > 0$ to exploit the multiplexing gain. The characterisation involves various functions, the properties of which are analytically established. An *off-line* algorithm is then developed to verify the analytical conclusions through simulations. We then develop an approach for approximate analytical calculation of σ^* for the three-state Markovian characterisation of the output of the shaper the input to which is a two-state Markov modulated fluid source. A close match of the analytical results with those obtained by simulations is observed.

We next proceed towards the development of a *Measurement Based On-Line Algorithm* for estimation of σ^*. The algorithm incorporates the features of discrete and continuous stochastic approximation techniques. The efficiency of the algorithm for an example source is shown.

Finally, we discuss a dynamic LB parameter renegotiation approach based on our proposed algorithms and present a *Measurement Based Call Admission Control Algorithm.* We also discuss an approach for working with non-homogeneous sources and expand the domain of the cost functions to non-linear functions (with some general properties).

List of Publications from this Thesis

Natwar Modani, Parijat Dube and Anurag Kumar, "Measurement Based Optimal Source Shaping with a Shaping+Multiplexing Delay Constraint" in *Proc. IEEE INFOCOM 2000*, March 2000, Tel Aviv, Israel.

Acknowledgements

It is a pleasure to place on record my deep gratitude and admiration for my thesis supervisor Prof. Anurag Kumar. I have never met a person with such an impressive horizon of knowledge of networking concepts. I am grateful for his valuable suggestions, ideas, constructive criticism and his patience and appreciation for my often half-baked ideas. Under his guidance I have grown immensely in knowledge and confidence. I shall always be proud to be known as his student and shall carry a part of him always in me.

The research reported in this thesis is a part of an ongoing project. I am particularly thankful to Natwar and Nilesh for several useful discussions concerning my research. I am thankful to Vinay and Dinesh for helping me at several times during the course of my research.

The Students' room of the ERNET was my working place. I am thankful to Santosh for introducing me to the working environment and for always extending his helping hand in my earlier days at ERNET. I also thank Aditya, Arzad, Avijit, Gourishankar, Sharma, Shobha, Supratim, Sunil and more recently Ajoy, Munish, Ramkumar, Sayee, Sandeep and Sethu for making my stay in the Students' room pleasant. I also thank the ERNET project-staff and particularly Bindu and Jaysree for their friendship. I shall always be thankful to Ms. Chandrika and Manjunath for their kind cooperation.

I shall fondly remember my days in IISc with my friends Ajay, Priyank, Arzad, Devendra, Vijay, Vineet, Maneesh and Nandan.

The research of this Thesis was supported by a grant from Nortel Networks, USA. I thank Nortel Networks for their generous financial support.

Chapter 1

Introduction

At the present time, there are two pervasive communication networks, mainly telephone network and the Internet. The telephone network is a connection-oriented, circuit switched network, and provides guaranteed delay and zero in-cell loss to voice calls. It is generally viewed as a single rate (64 kbps PCM) network [1]. On the other hand, the Internet is a connectionless, packet switched, multirate network and does not provide any delay guarantees. Till recently, this was not a problem as the Internet was primarily concerned with carrying data traffic such as file transfers and e-mail that do not require bandwidth guarantees and can suffer long, unpredictable delays. In the recent times, the networking community conceived the idea of developing a single digital unified network structure, called an Integrated Services Packet Network (ISPN), to carry all the different information types. ISPN will carry audio, video and data traffic including the traffic hitherto carried by the Internet (data), and by the telephone network (voice calls). As pointed out in [35], *The Holy Grail of computer networking is to design a network that has the flexibility and low cost of the Internet, yet offers the end-to-end quality-of-service guarantees of the telephone network.*
Two possible strategies for structuring an ISPN are being considered [35]

1. Upgradation of the pervasive Internet by adding quality-of-service mechanisms, thus

[1] Though using Integrated Services Digital Network (ISDN) source a user can on-demand obtain multiples of such 64 kbps channels that can then be viewed for the duration of the call as a single channel.

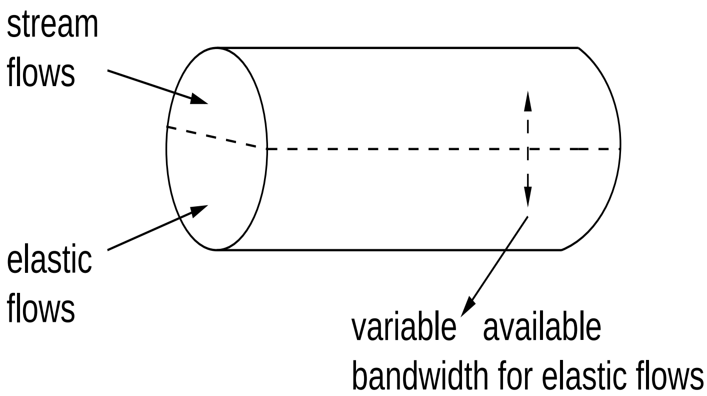

Figure 1.1: A typical information flow pipe of ISPN

making it suitable to carry delay sensitive traffic such as real-time audio and video.

2. Development and implementation of a new network from scratch that achieves the desired goal of providing quality-of service to individual streams, this being the stated goal of *Asynchronous Transfer Mode (ATM)* networks.

In the direction of the first solution, the Internet community is working on two approaches, viz., *Integrated Services* and *Differentiated services.* Considerable research in the area of providing guaranteed QoS support to individual traffic streams in ATM has been done and is going on.

1.1 Information Flow Control in an ISPN

It is now clear that integrated services network will be packet switched, multirate and multiquality, supporting audio, video and data information. All information flowing through a packet network suffers two types of distortions, temporal distortions due to delays, and content distortion which could be due to errors introduced by noise and/or losses caused by a mismatch of input and output rates at packet switches (or routers). The traffic in an ISPN can be broadly divided into two classes (see Figure 1.1) based on their tolerance to these two kinds of distortion:

1. *Stream Traffic:* Sources of this type have an intrinsic (unmodifiable) temporal behavior and hence have a tight end-to-end delay and delay variation requirement. They can tolerate some loss of the transmitted data without perceptible distortion at the receiving end. The sources can emit data into the network at a constant rate or a variable rate (called as CBR and VBR sources respectively in ATM terminology). Examples of constant rate stream traffic are circuit emulation, whereas of variable rate traffic are packet telephony and motion compensated variable rate coded video.

2. *Elastic Traffic:* Sources of this type do not have intrinsic temporal characteristics. They, in general, have loose delay but tight loss requirements. Examples are file transfer, e-mail, "web" browsing.

Due to the limited information carrying capacity of the physical layer (consisting of the bit transport infrastructure called the *channel* which may be wired like the copper cables or the optical fibres and/or wireless) there is a need to control the amount of information (of *stream and elastic traffic*) pumped into it from the higher control layers. The information flow control approaches can be broadly divided into two categories depending on the nature of traffic

1. To reject a source if its resource requirements cannot be met without degradation of information quality of other sources, called **admission control**. This is suitable for *stream traffic.*

2. To blindly accept any connection request and then dynamically (by some feedback control) adjust the rate at which the information is pumped from these sources to the dumb layer if congestion occurs. This is suitable for *elastic traffic.*

To be more specific, in the broad area of traffic control we shall be concentrating on the **admission control of *stream traffic* in this thesis**.

1.1.1 Admission Control, Shaping and Problem Motivation

An admission control problem can be subdivided into three subproblems,

- characterisation of the source's traffic parameters and the source's QoS requirements
- determination of the resource requirements for satisfying the source's requirements
- determination of the amount of free resources in the network, given the available resources and the already admitted sessions

The traffic parameters and the QoS requirements quantify the resource requirements of the source. Also, the admission control mechanism should have the features of fast decision making and high efficiency (in the sense that it should maximize the network resource utilization while simultaneously guaranteeing QoS to an admitted call).

The traffic coming from a source is apriori unknown. It does not have any *deterministic* bounds on its peak rate, average rate and burst size; in a sense the bounds are *stochastic*. Invariably a source is shaped before feeding it to the network. Shaping results in providing deterministic bounds on the traffic statistics. A common procedure for shaping is the Leaky Bucket (LB) Algorithm. The leaky bucket provides deterministic bounds on the burst size (called σ) and the long term average rate (called ρ) of the source. The source traffic is shaped and this traffic is then fed to the network node.

The two broad classes of admission control strategies from the QoS point of view are: deterministic QoS and statistical QoS allocation. Deterministic QoS allocation provide deterministic bounds on the received QoS and the bounds are never violated whereas in statistical QoS allocation there is a small probability of the QoS being violated. For CBR sources deterministic QoS allocations implies peak bandwidth allocation. Before discussing QoS allocation for VBR sources we shall first discuss the characteristics of VBR traffic which shall motivate two possibilities for reducing its resource requirement.

VBR sources have high rate active periods, low rate active periods and inactive periods. Thus if a VBR source is given a link bandwidth equal to its peak rate then there will be wastage of the bandwidth during the low rate active periods and the inactive periods. Since VBR sources do not always generate information at peak rate, if we have an appropriately sized buffer, a bandwidth less than the peak rate can be used. During the high rate periods of the source, the extra cells generated get buffered; these are then transmitted during the low input rate periods (when the allocated bandwidth is more than the input rate). This saving in bandwidth is called *buffering gain*. The (assumed) "statistical" independence between the sources sharing a link also reduces the total bandwidth requirement of the aggregate of sources. Owing to statistical independence, there is a small probability that many will be active at the same time. Thus the capacity requirement does not increase linearly with the number of sources. This reduction in bandwidth is called *multiplexing gain*. However buffering results in delay. Keeping the delay within tolerable bounds we can exploit this **buffer-bandwidth tradeoff** to work with less bandwidth along with a sufficient buffer.

For unshaped VBR sources peak bandwidth allocation is necessary. However if source parameters are known beforehand then one can employ stochastic analysis to determine the bandwidth-buffer tradeoff for achieving a statistical QoS. For shaped VBR sources deterministic QoS can be provided by resource allocation based solely on shaper parameters [2]. Although such an allocation exploits *buffering gain*, it is conservative owing to the fact that shaper parameters are only bounds. For shaped VBR sources statistical QoS allocation policies invariably admit the source with given shaper parameters like in peak rate allocation or deterministic QoS allocation etc. Then actual resource requirements of the source are learned on-line and bandwidth and/or buffer is compactified.

For a given set of shaper parameters and QoS requirements there exists a value of buffer and bandwidth such that the source's QoS requirements are met. There will also be an associated "cost" [3] by different sources which can be a function of bandwidth and/or buffer. Keeping the QoS requirements same, for different values of shaper parameters we can associate different cost.

[2]For LB shaped source, if the source is given a capacity of ρ and a buffer of σ at the multiplexer then the multiplexer delay is bounded by $\frac{\sigma}{\rho}$.

[3]Note that the network resources are provided by an operator, who will charge for their usage.

If we change the shaper parameters to a value, so that for the given QoS, the "cost" is minimized then the buffer and bandwidth corresponding to these parameters is optimal, in the sense that

- the delay requirements are met.
- the cost of providing service to this connection is minimised.

The problem thus concentrate on the **determination of optimal shaper parameters**. For stored VBR (audio/video on CD) sources these parameters can be known beforehand by doing off-line measurements. For on-line (non stored) sources like live telecasts, there is a need for quick, on-line estimation of these parameters. This parameter estimation time should be negligible as compared to the total life of the connection. Thus there is a need of algorithms for quick, measurement based on-line estimation of optimal shaper parameters for on-line VBR sources.

1.2 Description of the Research Problem and Approach

In the previous section we motivated the need for **quick and efficient measurement based on-line algorithms for the estimation of *optimal shaper parameters*.** The main objective of the research reported in this thesis is to design algorithms that can be implemented at the source and/or at the switches to compute the *optimal shaper parameters*. A typical network model is depicted in Figure 1.2. The path between the VBR source and the receiver consists of a LB shaper at the source end, followed by one or more intermediate switches. The goal is to estimate the optimal shaper parameters σ^* and ρ^* such that the shaped arrival process (i.e., the output of the LB shaper) is such that its network resource requirements (link bandwidth+buffer at the intermediate switches) minimizes the cost (a function of network resources), and simultaneously there is a statistical guarantee on the end-to-end delay (shaper delay + intermediate switch delay). The statistical guarantee on

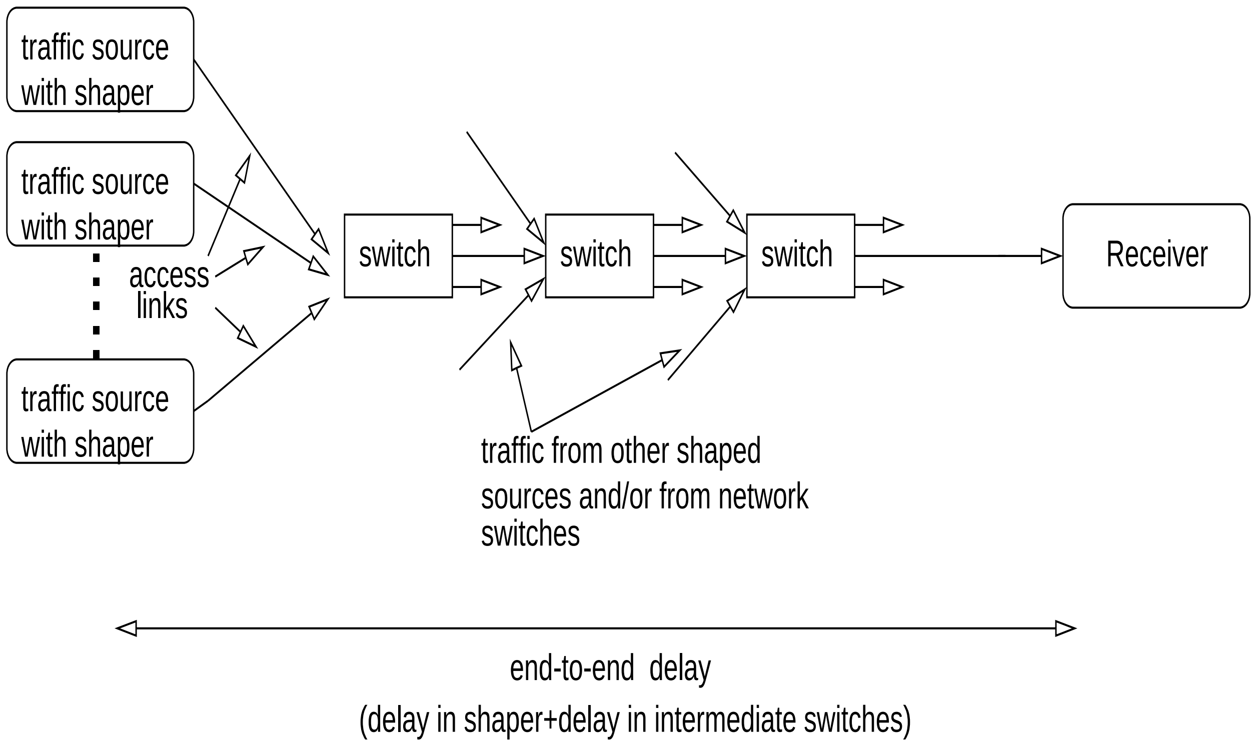

Figure 1.2: A typical network model

the end-to-end delay means that the delay can exceed a desired value only with a small probability.

In our thesis we deal with a subset of this problem by considering only a single hop network. The network scenario we considered is shown in Figure 1.3. There are n sources, assumed to be statistically identical (e.g., voice sources using the same coding and silence suppression scheme). Each source is shaped by a LB, and then the sources are multiplexed at a network node. Such a situation would arise, for example, between the packet voice "PBXs" of two enterprise locations connected by a serial link. The basic model is also relevant to end-to-end delay QoS approaches in which the QoS is managed by breaking it into a per-hop delay bound, and traffic is reshaped at each hop (in this case, of course, statistical homogeneity cannot be assumed after the first hop).

The detailed model that we work with is shown in Figure 1.4. Each source is shaped

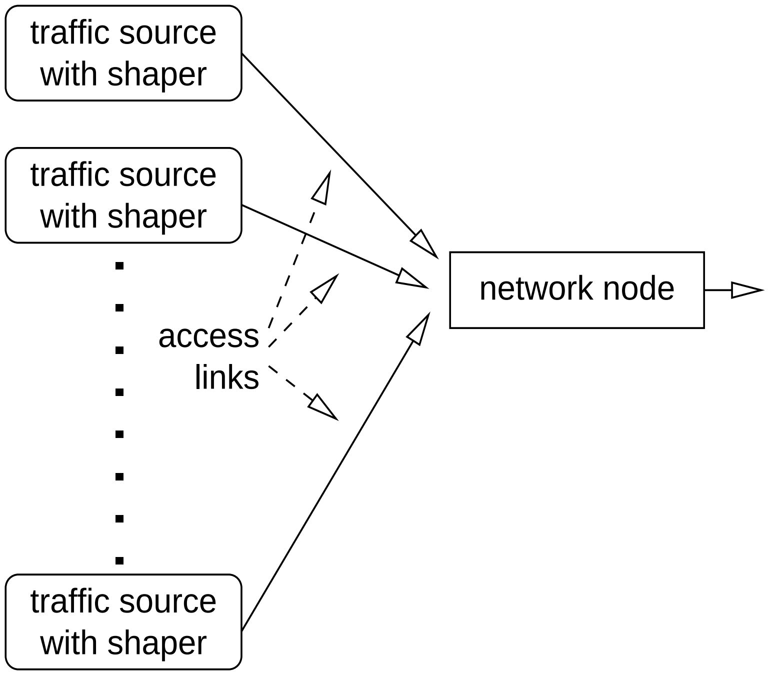

Figure 1.3: The network scenario under consideration.

by a LB. We are interested in choosing the shaper parameters ρ, σ and B_s so as to minimise the network resources required for providing a certain QoS. The network resources comprise the network link capacity and the buffers at the node. It is these network resources that are scarce and expensive (the shaping resources are in the client computers where the memory is very large and cheap), and hence we consider a linear network buffer-capacity cost. The QoS constraint is that the *shaping delay (in the shaper buffer)* ***plus*** *the multiplexing delay* can exceed T only with a small probability q.

In pursuing this goal we had to tackle the following issues

1. **Notion of Optimality and the Characterisation of ρ^* and σ^*:** As pointed out earlier our optimal LB parameters ρ^* and σ^* should shape the source such that its network resource requirements minimize the cost and simultaneously there is a statistical guarantee on the end-to-end (shaping+multiplexing for single node case). We first consider the case of *lossless multiplexing*. This yields ρ^* as the solution of an implicit equation with the capacity and the buffer requirement at the multiplexer being ρ^* and

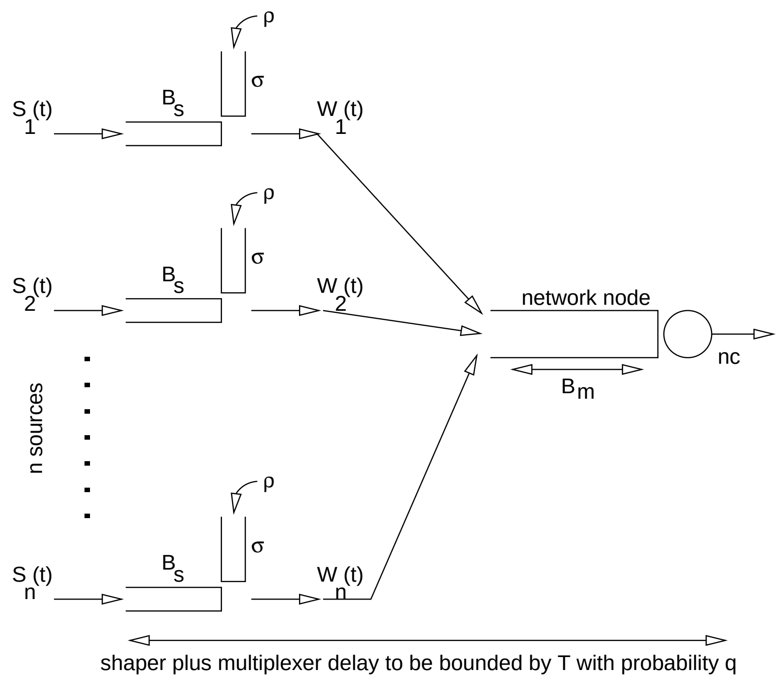

Figure 1.4: The network model, showing the LB shapers and the network node.

$\sigma^*(=0)$ respectively. Then for exploiting multiplexing gain we allow losses in the multiplexer, thus working with *lossy multiplexing.* This characterises σ^* as the solution of a constrained optimization problem and yields a value of capacity and buffer such that the network buffer-capacity cost for lossy multiplexing is less than the cost for lossless multiplexing.

2. **Development of Measurement Based On-Line Algorithms for Estimation of Optimal Shaper Parameters:** Before mentioning our approach for formulating the algorithms we shall first discuss the desired features of any measurement based on-line optimal shaper parameters estimation algorithm.

   Desired Features:

   (a) *Robustness and Fast Convergence:* The algorithm should be robust, i.e. it should be *insensitive to changes in underlying statistical assumptions* and should depend minimally on source modeling. Also the updates should converge to the true parameter values as fast as possible in the presence of *measurement noise.* In the case of *nonstationary* sources the algorithms should track the ρ^* and σ^* effectively.

(b) *Simple and Independent Updation with Minimum Source-Network Signalling:* The algorithms should have simple updation steps for easier implementation. Also the updation should involve minimum signalling between the source and the network.

(c) *Conformity with the Existing Protocols:* The proposed algorithms should be such that they can be implemented within the constraints imposed by existing standard protocols.

Thus to be of any "worth" our proposed algorithms should satisfy one or more of the above requirements. We shall now discuss our approaches for developing the on-line algorithms which will also reflect the degree of compliance of our algorithms with the three *desired features* mentioned above.

(a) **Measurement Based On-Line Estimation of ρ^*:** Since, we obtain ρ^* as the root of an implicit equation, we apply stochastic approximation techniques to determine ρ^* on-line. The stochastic approximation requires the measurement of a **rare event probability**. To tackle this issue we use a fast simulation technique for quick estimation of ρ^*. The algorithm tracks the ρ^* for stationary as well as piecewise-stationary (nonstationary sources with stationary regimes) sources. Also, the algorithm is applicable to any stationary and ergodic short-range dependent traffic which is a true characterisation for most stream traffic [4]. Another important feature of our algorithm is that it can be completely implemented at the source end, working independently from the network node.

(b) **Measurement Based On-Line Estimation of σ^*:** We characterise σ^* as the solution of an optimization problem. The optimization problem needs to be solved on-line. Again like the ρ^* determination algorithm, our σ^* determination algorithm is also devoid of any source modeling and is valid for any stationary and ergodic short-range dependent traffic. The algorithm searches over a discrete set

[4]Studies have indicated that some video traffic is long-range dependent

of σ. Each update involves the determination of a function which is itself obtained as an update of a continuous stochastic approximation algorithm. For a given ρ^*, the σ^* determination algorithm can be completely implemented at the network switch, working independently of the source except during the change in ρ^* estimate at the source end.

Thus, our proposed algorithms satisfy the first two *desired features* to a fairly good extent. The conformity with existing protocols has to be dealt with in depth. However, the incorporation within Integrated Services (IntServ) concept of Internet using RSVP [54] seems feasible at first glance.

An important remark to be made here is that *in the designing of these algorithms we have worked with fixed length packets (or cells as the work started with ATM in mind). The analysis is done with fluid* [5] *models whereas in the simulations we have worked with fixed length packets. However we have used no other feature specific to ATM technology making the algorithms equally relevant for any QoS supporting data networks and technologies.* Also, analysis with packetized source is undertaken in [44], which is an extension of the research reported in this thesis.

1.3 Organisation of this Thesis

In Chapter 2 we present a brief overview of some of the literature available in the area of measurement based admission control techniques. We conclude the chapter with a brief comment on how the research reported in this thesis fits into the existing literature.

In Chapter 3 we develop a formulation for characterisation of the optimal token arrival rate (called ρ^*) of the LB shaper. We then proceed to develop a measurement based online algorithm for the determination of ρ^*. In this process we develop a Fast Simulation

[5] As mentioned in [18] *"Several recent papers on packetized communications and ATM networks are based on stochastic fluid models ...; these studies have found the approach in [6], which is based on stochastic fluid models, to be effective in its accuracy and capacity to solve large systems".*

Based Stochastic Approximation Algorithm (FSSAA) for on-line estimation of the minimum capacity required by an arrival process for a given buffer and a given loss probability at the multiplexer. Adequate simulations validate the claims of the algorithm.

In Chapter 4 we develop a formulation for the characterisation of the optimal token bucket depth (called σ^*) of the LB shaper. Analytical proofs of the properties of various functions strengthen the formulation. We then develop an off-line algorithm for the estimation of σ^* which is supported by simulations. Next we proceed to develop an approximate analytical calculation of σ^* for Markov modulated fluid sources by using the Markovian approximation for the output of shaper provided in [18] and the *Perron-Frobenius* eigen value theory. The analysis is then compared with simulation results [16].

In Chapter 5 we develop a measurement based on-line algorithm for estimation of σ^*. The algorithm is a combination of discrete and continuous stochastic approximation techniques. Sufficient simulation results validate the efficiency of our proposed algorithm [17].

In Chapter 6 we discuss renegotiation and outlin a call admission control algorithm based on our proposed measurement based algorithms. We also discuss non-homogeneity of sources and non-linearity of cost functions. Finally we conclude in Chapter 7 with an insight into future research issues.

Chapter 2

Literature Survey and Preliminary Background

2.1 Introduction

The admission control approaches can be broadly classified into two categories

1. Category 1: These approaches do not have any apriori information about the source characteristics and depends on measurements for learning the traffic arrival statistics to estimate the resource requirements. We will call them *measurement based* approaches.

2. Category 2: These approaches by pass the subproblem of traffic parameter identification. They take it for granted that these parameter values are known and they directly provide techniques to calculate the resource requirements of the source. We will call such approaches as *prior information based* approaches.

Prior information based approaches are suitable for the situations where the source has a validated parameterised model with known parameters. For example in case of PCM voice calls, the peak rate is 64 kbps and the usual values of the mean on-time and the mean off-time is 400 ms and 600 ms respectively. Thus modeling a voice call as a two state Markov modulated fluid process (MMFP) with peak rate 64 kbps and the on and off period exponentially distributed with mean on-time 400 ms and mean off-time 600 ms is the standard procedure. Measurement based approaches are employed in situation where the traffic characteristics are highly source dependent. For example different video sources have different arrival statistics due to different type of pictures, different coding rates etc. Measurement based approaches can be further classified as *Off-Line* and *On-Line*. *Off-line* measurement based approaches are used for stored traffic (movie/songs on CD). Here whole of the traffic statistics is known apriori (i.e., before the connection set-up procedure). The parameters can then be determined using off-line measurement based techniques and resource requirements can be calculated. For *On-Line* VBR, apriori knowledge of traffic statistics is impossible thus signifying the need for on-line measurement based admission control. *On-Line* measurement based approaches may even be needed for stored media like video-on-demand as the user is expected to be able to exercise interactive control (such as pause, fast-forward etc.) [27].

In this chapter we shall provide an overview of the different call admission control (CAC) techniques concentrating on measurement based call admission control (MBCAC) techniques in particular.

2.2 Literature Survey

CAC strategies are abundant in literature (See [45] for a good review). Though there is a wealth of papers dealing with prior information based call admission control approaches (PIBCAC), the complementary approach involving MBCAC is the need of the hour because of the inherent unpredictability of multimedia traffic.

2.2.1 PIBCAC approaches

Notable among such approaches are the approaches based on Effective Bandwidth (EBW) ([19], [28], [24], [34], [22]) and Heavy Traffic Approximation ([50], [51]).

"The appeal of EBW schemes lies in their inherent simplicity: Given a quality of service requirement and system parameters such as the buffer space, these schemes represent a traffic stream via a single parameter - its equivalent bandwidth - which is independent of all other sources that it contends with" [47]. With this, the admission region in the multiclass case become a simplex; i.e., if the sum of the EBW's of a newly arriving call and those of the existing calls is less than the link capacity than the new call is accepted. Studies have indicated that the admission control techniques based on EBW approaches are tight in the asymptotic regime (infinite multiplexer buffer) and overly conservative for small/moderate buffer sizes ([10]). In [47] the author compares various EBW approaches and proposes enhancements to widen the admission region.

In the diffusion based approach [22], the dynamics of the queue level in the multiplexer are modeled by a diffusion equation. This results in representing the superposition of several sources as an equivalent three parameter source, for which the EBW can be calculated. In the heavy traffic approximation approach ([50], [51]), an approximation for bandwidth allocation based on the asymptotics of the tail of the queue-length distribution is done. Another interesting approach is based on a Bayesian formulation of the problem [26] where an attempt is made to minimize call blocking probability while bounding the cell loss ratio (CLR) in a call. The problem becomes a constrained optimization problem with statistical interpretations.

2.2.2 MBCAC approaches

MBCAC approaches ([25], [55], [37], [52]) are devoid of any source modeling, requiring some loose assumptions on the source traffic. The techniques are to a large extent general and invariably require on-line measurement of traffic statistics (peak rate, mean rate) and QoS requirements (cell loss ratio, delay, delay jitter).

The two major MBCAC approaches are,

- the call is admitted based on worst-case parameters (for example the peak rate). Then measurements are done and bandwidth is compactified ([37], [55]).

- the call is admitted virtually and its effect on the CLR offered by the network is simulated. If the CLR remains within the guaranteed bounds then the call is accepted ([11]); the time taken to do this simulation is a part of the call set up time, and thus these approaches require efficient estimates of CLR (a *rare event*). Some notable works in estimating such *rare event* probabilities are [11], [49], [53].

Another interesting class of CAC strategies are based on Neural Networks. Neural Networks based on back-propagation ([58]) can be trained to learn complex nonlinear relationships between ordered traffic and the resulting QoS. The resulting Neural Network Based Call Admission Controller (NNCAC) has the ability to interpolate or extrapolate from the past and is thus flexible, adaptive and simple to implement. They are either measurement based (with on-line training of neural networks like [30], [31], [42], [15], [43], [8]) or prior information based involving traffic modeling and subsequent off-line training with analytical results like the one developed in [33].

For LB controlled sources, the NMBCAC approach is to assume the output process of the LB to be worst case [20]. It is assumed that the worst case output of a LB is a periodic on-off source, however this assumption is not true for a buffered multiplexer.

All the approaches discussed and mentioned till now have one common feature *they are more or less* **dumb**, *in the sense they are only concerned with determining whether a given source can be accepted or not. They are not concerned with tailoring the arrival process to the network to meet the QoS requirements and thus, the source parameters and traffic arrival statistics remains untouched.* However as we motivated in Section 1.1.1, tailoring the traffic arrival statistics to such a direction that the resulting traffic coming to the network has resource requirements which minimizes the cost and simultaneously guarantee QoS (like end-to-end delay etc.) is an **intelligent** approach requiring, **Optimal Source-Shaping + Admission Control**.

Our work in this thesis is concentrated on incorporating this additional intelligence, of determining optimal source [*effective source (source + shaper) for shaped sources*] parameters to MBCAC. There are three notable references ([29], [13] and [40]) dealing with a problem akin to ours.

In [29] the authors study the problem of finding an optimal sustainable rate parameter based on network buffer-bandwidth cost considerations. They use a cost function of the form $b^{\beta}c^{\gamma}$, where b is the network buffer and c is the network bandwidth allocated to the call. For an LB based controller, and for a given cell rejection probability, they minimize the cost function over all feasible pairs (b, c) for lossless service in the network. They do not, however, consider any delay constraint, as we do in our thesis. Another related work is [13]. The objectives of the research reported in this paper are similar to ours, i.e., to choose optimal leaky bucket parameters subject to a QoS constraint. The approach and results are different, however. Whereas in [13] the author only considers delay in the LB buffer, we consider the problem of choosing LB parameters under a shaping plus multiplexing delay constraint.

Another recent published work is reported in [40]. The authors have dealt with the problem of real-time estimation of optimal leaky bucket parameters by developing an approximate statistical characterization for an arbitrary cell stream and mapping it to a set of optimal leaky bucket parameters. The optimality is in the sense of minimizing a network cost function, subject to meeting user-specified constraints on shaping delay. They have considered only shaping delay whereas our problem considers shaping+multiplexing delay. There network cost is simply the capacity required for a given network buffer whereas our cost is a linear function of capacity and buffer required at the network node thus providing a capacity-buffer tradeoff. The method they proposed for estimating the optimal token rate is heuristic and might gives suboptimal results for increased optimality requires increased implementation complexity. In contrast we have employed stochastic approximation based technique whose convergence is analytically established. Also, there proposed method works well for shaping probability in the neighborhood of 10^{-2} or 10^{-1} but for decreased values of shaping probability (even for 10^{-3}) the estimated parameters are overly conservative. We have wprked with shaping probabilities of the order of 10^{-5} in our simulations.

The work reported in this thesis was initiated in [41]. There the author developed a characterisation of optimal leaky bucket parameters ρ^* and σ^* and a measurement based approach to estimate ρ^* on-line. However the characterisation of ρ^* was incomplete and assumptions were not justified. We have furthered the work of [41] in this thesis. The main contributions of this thesis are,

1. A thorough analysis of the formulation for ρ^* and validation of the assumptions.

2. Development of an alternate and more efficient *Measurement Based On-Line Algorithm* for estimation of ρ^*.

3. Analytical study of the properties of various functions involved in the characterisation of σ^*. The use of an *off-line algorithm* to support the analytical results through simulations.

4. Development of an approximate analytical calculation of σ^*.

5. Development of a *Measurement Based On-Line Algorithm* for estimation of σ^*.

Chapter 3

Measurement Based Estimation of the Optimal Token Rate

3.1 Introduction

In this chapter first we provide the leaky bucket (LB) shaper model that we shall be using. We then provide a formulation of the Minimum Cost Lossless Multiplexing problem for the characterization of optimum value of ρ, i.e., ρ^*. The formulation was first partially developed in [41]. Here we thoroughly analyse the formulation and validate the assumptions made. We then proceed towards developing a measurement based technique for on-line estimation of ρ^*. In this direction we develop a *general and efficient* algorithm for on-line estimation of minimum required capacity for a given loss probability at a multiplexer with a given buffer size. The algorithm is called the *Fast Simulation Based Stochastic Approximation Algorithm (FSSAA)* and it incorporates the merits of the scaling property of EBW approach and the virtual buffers technique with a stochastic approximation algorithm. This algorithm is then applied for on-line estimation of ρ^*. The algorithm tracks the ρ^* for stationary as well as *piecewise-stationary* (non-stationary sources with stationary regimes) sources. Ample simulation results validate the claims of the algorithm.

An important point to be mentioned here is that *we work only with fluid models in* *the analysis since the work is originally motivated by ATM where the cells are small and fixed size.* The extension of the analysis in this thesis to bounded packet sizes and multiple hops s being pursued in [44].

3.2 The Leaky Bucket Shaper: A Review

Leaky Bucket Shaper is designed on the principle of Leaky Bucket (LB) algorithm, which is the standard algorithm for providing deterministic bounds on the input traffic to network. The algorithm is characterized by three parameters token arrival rate called ρ, token bucket depth called σ and a cell [1] buffer called B_s. The LB shaper can be viewed as consisting of two buffers the token bucket (of size σ) and the cell buffer. The token buffer is incremented periodically at rate ρ. If the token buffer is full when a token arrives, then the token is discarded. When a cell arrives at the LB, if the token buffer is not empty, the cell departs immediately and the token buffer content is reduced by one. Thus when the token buffer is not empty, the output rate is equal to the source rate. If the token buffer is empty then an arriving cell has to wait in the cell buffer and the output rate is equal to ρ. If the cell buffer is full the arriving cells are rejected.

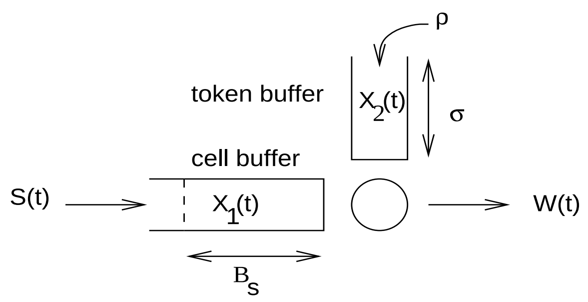

Figure 3.1: The Leaky Bucket shaper. The buffer is infinite, but we want the probability of exceeding the buffer level B_s to be very small.

Figure 3.1 shows the LB shaper, and the associated notation that we shall use. The shaper is used to put *deterministic* bounds on the arrival process from the variable rate

[1] We use the term “cells” since the work originally began with ATM in mind.

source. The processes $S(t)$ and $W(t)$ shown in Figure 3.1 are to be viewed as rate processes. In the analysis we will assume fluid processes, whereas the simulations will be done with discrete fixed length packets, or cells. We shall not concern ourselves with peak rate control, *assuming* that the input is already peak rate controlled to the rate R

We note that when there are cells in the cell buffer, since tokens are arriving at the rate ρ, the cell buffer is depleted at the rate ρ. If the cell buffer level exceeds B_s, and since the source would not drop its own cells, we view this as a QoS violation; i.e., B_s *does not represent a memory limitation, but a delay bound of* $\frac{B_s}{\rho}$. Thus our view is that the cell buffer "behind" the LB is infinite but the buffer level exceeds B_s with a small probability; we call this the *QoS Violation Probability (QVP).*

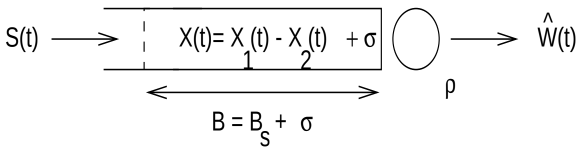

Figure 3.2: A single server queuing system, with service rate ρ and infinite buffer, that is equivalent to the (σ, ρ) leaky bucket shaper from the QVP (see text) point of view.

With reference to Figure 3.1, define $X(t) = X_1(t) - X_2(t) + \sigma$. The QVP is then just $P(X > B_s + \sigma)$, where X is the stationary marginal buffer occupancy (see Figure 3.2). For a fluid model, the system shown in Figure 3.2 is equivalent to the leaky bucket shown in Figure 3.1 from the QVP point of view (see also [18, 9]). Thus for a given ρ, the QVP depends on σ and B_s only through their sum. We will use the notation p_s to denote $P(X > B_s + \sigma)$. Writing $B := B_s + \sigma$, for fixed p_s we denote the B vs ρ tradeoff function by $g_{p_s}(\rho) = B$. Let $h_{p_s}(\cdot)$ denote the inverse of $g_{p_s}(\cdot)$ (an example of $h_{p_s}(\cdot)$ is in Figure 3.3).

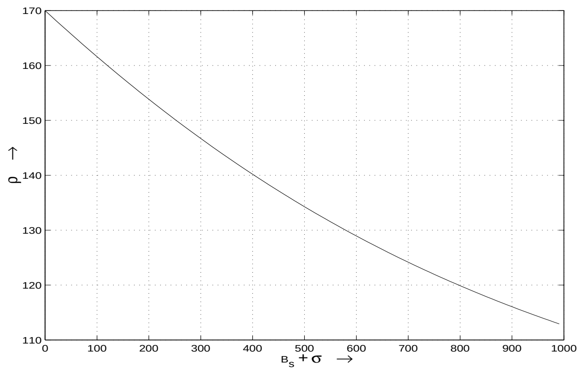

Figure 3.3: A typical $(B_s + \sigma)$ vs ρ trade-off curve; i.e., the function $h_{p_s}(B)$. Two state Markovian on-off source with mean on-time 5/3, mean off time 5/2, and peak rate $R = 170$; $p_s = 10^{-3}$.

3.3 Shaping for Minimum Cost Lossless Multiplexing: Characterisation of an Optimal Value of ρ

In this section we consider several statistically identical sources $(S_1(t), S_2(t), \ldots, S_n(t)$, see Figure 1.4), each shaped by the same LB parameters, feeding a buffered multiplexer. Each source requires a shaping plus muxing delay bound of T, which can be violated with the QVP of q. For this scenario, we develop the notion of an optimal LB token rate parameter ρ^*. *The optimality will be in the sense of minimizing the multiplexer service rate (link bandwidth) required for lossless multiplexing of the superposition of the LB controlled sources, with the constraint that the shaping plus multiplexing delay bound T is violated with a probability $\leq q$.*

For an isolated (σ, ρ, R) source fed into a buffered server with service rate c, the maximum (over all conforming sources) buffer occupancy b is related to c by

$$b = \frac{\sigma}{R - \rho}(R - c) \tag{3.1}$$

This occupancy is achieved by an on–off extremal periodic source with on-time $T_{on} = \frac{\sigma}{R-\rho}$ (at peak rate) and off-time $T_{off} = \frac{\sigma}{\rho}$ (see [20]). It is also easily seen that (see [46] for a more general result) the lossless multiplexing of n such sources will require a buffer of nb and capacity nc; i.e., for lossless multiplexing the segregated and aggregated systems are identical for resource requirements. Thus, for lossless service at the multiplexer, we need to consider each source with its own network queue of service rate c and buffer size b. With these ingredients, we will develop an optimization problem.

For given (σ, ρ, R), for lossless service it is necessary that the capacity at the multiplexer $c \geq \rho$. Also, we need to have

$$b \geq \frac{\sigma}{R-\rho}(R-c) \quad \Rightarrow \quad c \geq R - \frac{b}{\sigma}(R-\rho)$$

It follows that, for lossless multiplexing, given $b \geq 0$, we need

$$c = \max\left(\rho, R - \frac{b}{\sigma}(R-\rho)\right) \tag{3.2}$$

Now consider the shaping and multiplexing delay constraint T. Suppose we allow a target shaping delay of $\frac{B_s}{\rho}(\leq T)$. Hence the delay bound T is met provided the bound on the network node delay $\frac{nb}{nc} = (T - \frac{B_s}{\rho})$, i.e., $b = c(T - \frac{B_s}{\rho})$. Since our QVP is q, we can permit the buffer level in the shaper to exceed B_s with probability $p_s = q$. We seek the values of ρ, σ, b and B to minimise the multiplexer capacity c. Recalling some notation from Section 3.2 (in particular that $B = B_s + \sigma$), we have the following optimisation problem:

$$\min_{(b,B,\sigma,\rho)} \quad c = \max\left(\rho, R - \frac{b}{\sigma}(R-\rho)\right)$$

Subject to:

multiplexer delay constraint:

$$b \;\; = \;\; c\left(T - \frac{B-\sigma}{\rho}\right)$$

shaper delay constraint and delay QVP:

$$\frac{B-\sigma}{\rho} \leq T \qquad\qquad \text{and} \qquad\qquad B = g_q(\rho)$$

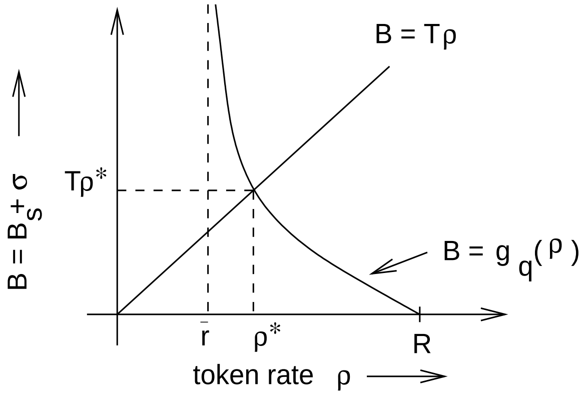

Figure 3.4: Characterisation of ρ^*; $\overline{r}$ is the mean rate of the source.

and, by definition

$$0 \leq \quad \sigma \quad \leq B$$

Equivalently, eliminating b between the multiplexer delay constraint and the objective function, we obtain

$$\min_{(B,\sigma,\rho)} \quad \max\left(\rho, \frac{R}{1+\frac{1}{\sigma}(T-\frac{B-\sigma}{\rho})(R-\rho)}\right)$$

Subject to

$$B = g_q(\rho), \quad 0 \leq \sigma \leq B \quad \text{and} \quad B \leq T\rho + \sigma$$

The solution to the above problem is provided by the following theorem.

Theorem 3.3.1 *For $g_q(\rho)$ a convex and decreasing function, the optimal value of the problem described above is given by the unique ρ^* that solves the equation*

$$T\rho = g_q(\rho) \tag{3.3}$$

Further, the optimal value of $B_s + \sigma = T\rho^*$, and for $0 \leq \sigma \leq T\rho^*$, the value of $b = \sigma$.

Proof : Looking at the objective function of the optimization problem, let us first examine

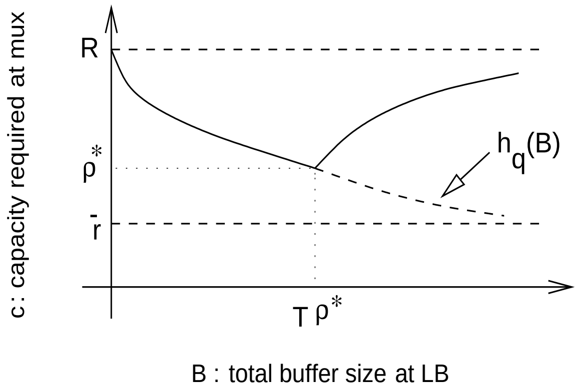

Figure 3.5: Capacity requirement at the network node (c) as a function of total buffer (sum of cell buffer and token buffer) at the shaper (B); $\bar{r}$ is the mean rate of the source $S(t)$.

the condition for the following inequality to hold

$$\rho \leq \frac{R}{1 + \frac{1}{\sigma}(T - \frac{B-\sigma}{\rho})(R - \rho)}$$

i.e.,

$$\rho + \frac{\rho}{\sigma}(T - \frac{B - \sigma}{\rho})(R - \rho) \leq R$$

Since $R \geq \rho$, we have

$$\frac{\rho}{\sigma}(T - \frac{B - \sigma}{\rho}) \leq 1$$

from which we get

$$T\rho \leq B$$

Thus, independent of σ, $\rho > \frac{R}{1+\frac{1}{\sigma}(T-\frac{B-\sigma}{\rho})(R-\rho)}$ for $B < T\rho$. We recall that $h_q(\cdot)$ is the inverse function of $g_q(\cdot)$; since $g_q(\cdot)$ is decreasing and convex, the same properties hold for $h_q(\cdot)$. Now ρ and B are related by $\rho = h_q(B)$, and ρ^* is defined by the solution of $\rho = h_q(T\rho)$. Since $h_q(B)$ is decreasing in B, $Th_q(B) > B$ for $B < T\rho^*$. Thus for $\rho = h_q(B)$, we can write (see Figure 3.5),

$$c(B) = \begin{cases} h_q(B) & if \quad B < T\rho^* \\ \frac{R}{1+\frac{1}{\sigma}(T-\frac{B-\sigma}{\rho})(R-\rho)} & if \quad B \geq T\rho^* \end{cases} \tag{3.4}$$

We see that the c vs B curve is just the $h_q(B)$ curve upto $B = T\rho^*$, and is hence decreasing upto $B = T\rho^*$. We will now show that for $B > T\rho^*$, $c(B) = \frac{R}{1+\frac{1}{\sigma}(T-\frac{B-\sigma}{\rho})(R-\rho)} \geq \rho^*$. This will establish the result. Notice that for $B > T\rho^*$, and $B = g_q(\rho)$, we have $B > T\rho$ (see Figure 3.4). Also we have, from the constraints, that $B - T\rho \leq \sigma \leq B$. It is then easily seen that, for $B > T\rho^*$

$$\frac{R}{1 + \frac{1}{\sigma}(T - \frac{B-\sigma}{\rho})(R - \rho)} \geq \frac{R}{1 + \frac{T}{B}(R - \rho)}$$

Hence it suffices to show that, for $B > T\rho^*$,

$$\frac{R}{1 + \frac{T}{B}(R - \rho)} \geq \rho^*$$

But this follows since, by the convexity of $h_q(\cdot)$, we have, for $\rho < \rho^* < R$, and $\rho = h_q(B)$ (see Figure 3.4)

$$\frac{R - \rho^*}{R - \rho} \geq \rho^* \frac{T}{B}$$

In Figure 3.5 the increasing solid curve to the right of $B = T\rho^*$ is a sketch of the lower bound to c, i.e., $\frac{R}{1+\frac{T}{B}(R-\rho)}$. The geometry of the solution is depicted in Figure 3.4.

Discussion of Theorem 3.3.1: The only assumption that the above result makes about the source $S(t)$ is that the function $g_q(\rho)$ is convex and decreasing. *Also the optimal sustainable rate parameter ρ^* obtained by this approach depends only on the source process, the maximum delay (T) desired by the source, and the QVP q. Thus there is a possibility that ρ^* can be determined autonomously by the source in real-time by making measurements.* We explore an algorithm for doing this in the Section 3.6. Another observation is that since we have the optimal value of $c = \rho^*$, if the source is shaped by (σ, ρ^*), $0 \leq \sigma \leq T\rho^*$ we must use a buffer of size σ, to ensure a lossless service to the shaped source at the multiplexer.

The function $g_q(\cdot)$: With reference to Figure 3.2, an important approximate approach to determining the service rate ρ so that the overflow probability $P(X > B) < q$ is to use the asymptotic approximation developed in [19]. Such an approach would be particularly applicable, for example, to VBR voice sources, for which a two state Markov model with exponential on and off times is a standard model. If we write the negative of the slope of the tail of $\ln P(X > B)$ as $\eta(\rho)$, then one approach is to design the shaper by taking

$$\eta(\rho) = \frac{-\ln q}{B}$$

Hence, with this approach, we have

$$g_q(\rho) \quad = \quad \frac{-\ln q}{\eta(\rho)} \tag{3.5}$$

Lemma 3.3.1 *If $S(t)$ is a Markov modulated fluid process, then the function $g_q(\rho)$, as defined in Equation 3.5, is convex and decreasing.*

Proof: For a Markov modulated source, $\eta(\rho)$ is the negative of the *dominant eigen value* for the eigen value problem,

$$z\phi(\Lambda - \rho I) = \phi M$$

where, Λ is the rate matrix M is the generator matrix, ϕ is the eigen vector and I is the identity matrix [2]. Let the dominant eigen value be $-z_1$, then

$$\eta(\rho) = z_1$$

Given that,

$$g_{p_s}(\rho) = \frac{-\log p_s}{\eta(\rho)}$$

$$\Rightarrow \dot{g_{p_s}}(\rho) = \frac{\log p_s}{(\eta(\rho))^2}\dot{\eta}(\rho) \tag{3.6}$$

Since, $\dot{\eta}(\rho) > 0$ (See [19]), thus $\dot{g}_{p_s}(\rho) < 0$. Thus $g_{p_s}(\rho)$ is a *decreasing function.* Now, again differentiating Equation 3.6 with respect to ρ,

$$\ddot{g}_{p_s}(\rho) = \log p_s \left(\frac{\eta^2(\rho)\ddot{\eta}(\rho) - 2\eta(\rho)\dot{\eta}^2(\rho)}{\eta^4(\rho)} \right)$$

For $g_{p_s}(\rho)$ to be convex we require,

$$\eta^2(\rho)\ddot{\eta}(\rho) - 2\eta(\rho)\dot{\eta}^2(\rho) < 0$$

$$\Rightarrow \eta(\rho)\ddot{\eta}(\rho) - 2\dot{\eta}^2(\rho) < 0 \tag{3.7}$$

[2]For details please refer to [19].

By the inverse function theorem,

$$\frac{d}{dz_1}\eta^{-1}(z_1) = \left(\dot{\eta}(\eta^{-1}(z_1))\right)^{-1}$$

$$\Rightarrow \frac{d^2}{dz_1^2}\eta^{-1}(z_1) = -\left(\dot{\eta}(\eta^{-1}(z_1))\right)^{-2}.\ddot{\eta}(\eta^{-1}(z_1)).\left(\dot{\eta}(\eta^{-1}(z_1))\right)^{-1}$$

Let $g_1(z)$ be the perron root of the inverse problem i.e,

$$g(z)\phi = \phi A(z)$$

where,

$$A(z) = \Lambda - \frac{1}{z}M$$

Then it is shown in [19] that,

$$2\dot{g}_1(z_1) + z_1\ddot{g}_1(z_1) < 0 \tag{3.8}$$

Also, $g_1(z_1) = \rho$ so that $-g_1 = -\eta^{-1}$. Expressing Equation 3.8 in terms of η

$$\frac{-2}{\dot{\eta}(\rho)} + (-\eta(\rho))\left(\frac{-\ddot{\eta}(\rho)}{(\dot{\eta}(\rho))^3}\right) < 0$$

$$\Rightarrow 2(\dot{\eta}(\rho))^2 - \eta(\rho)\ddot{\eta}(\rho) > 0$$

$$\Rightarrow \eta(\rho)\ddot{\eta}(\rho) - 2(\dot{\eta}(\rho))^2 < 0 \tag{3.9}$$

Observe that, Equation 3.9 is same as Equation 3.7, which is the condition for convexity of $g_{p_s}(\rho)$.

Minimum Cost Lossless Multiplexing: $S_1(t), S_2(t), \ldots, S_n(t)$ are statistically identical sources, each shaped according to (σ, ρ, R) (for given R), and being served in a lossless manner at a FCFS multiplexer with per source buffer b and per source capacity c. Consider the problem of *minimising the linear buffer-bandwidth cost function* $\gamma c + \beta b$ *(where* γ *and* β *are the per unit cost of capacity and buffers, respectively;* $\gamma > 0$ *and* $\beta > 0$*) over all choices of the LB parameters, and under the QoS constraint: shaping plus multiplexing delay* $\leq T$ *with*

$QVP = q$. Since the pair $c = \rho^*, b = 0$ is feasible for this problem, and ρ^* is the least feasible value of c, it follows that the linear cost function is minimized for leaky bucket parameters $\sigma = 0$ and $\rho = \rho^*$. Note that for a fluid source we interpret $\sigma = 0$ to mean that all fluid arrival from a source queues up at its LB buffer, and is served at the rate ρ (i.e., as the fluid "tokens" arrive). In practice, with discrete arrivals, σ will need to be at least the maximum packet size.

3.4 Issues in Measurement Based Estimation of ρ^*

In the model of Figure 3.2, with X the stationary queue length, define $p(\rho, B) = P(X > B)$. Then we define $d(\rho, B)$ as

$$d(\rho, B) = -\ln p(\rho, B) + \ln q \tag{3.10}$$

where q is the desired QVP. Recalling Theorem 3.3.1, our problem is now to find the root ρ^* of the function $f(\rho) = d(\rho, T\rho)$. Observe that to know the value of $f(\rho)$ at a given ρ, we need to know the value of $p(\rho, T\rho)$ i.e., the probability of the queue length exceeding $T\rho$ in a multiplexer with infinite buffer, arrival process $S(t)$ and a server of capacity ρ.

Since we are interested in on-line estimation of ρ^*, we shall be obtaining the value of $p(\rho, T\rho)$ through real-time measurements over a finite time interval and then shall employ an iterative algorithm which hopefully will converge to ρ^*. Since, typically QVP will be very small (10^{-5} to 10^{-9}), determination of $p(\rho, T\rho)$ involves the measurement of the probability of a *rare event*. Unless special techniques are used, a low variance measurement would require a very long time interval. Since this measurement is used in a control loop, a quick and accurate measurement becomes all the more important. We, therefore, utilise some speed-up techniques for the measurement of this small probability.

Also since the estimates of $p(\rho, T\rho)$ are noisy as they are obtained by measurements over a finite time interval. Thus, we get noisy estimates of $f(\rho)$ for each value of $\rho \in (\bar{r}, R)$. This put us in the framework of *Stochastic Approximation Techniques*.

From the above discussion we conclude that to determine ρ^* we need a Algorithm which imbibes

- A speed-up technique to obtain quick and low-variance measurement of $p(\rho, T\rho)$.
- A Stochastic Approximation Algorithm to get the root of $f(\rho) = 0$.

With these objectives in mind we now proceed to develop a general algorithm to determe the minimum capacity required for a given loss probability[3] in a multiplexer with a given buffer. We shall call it **Fast Simulation Based Stochastic Approximation Algorithm (FSSAA)**. This FSSAA will then be employed to determine $p(\rho, T\rho)$.

3.5 Fast Simulation Based Stochastic Approximation Algorithm (FSSAA)

In this section we develop a general and efficient algorithm for measurement based on-line estimation of the minimum required capacity for a given loss probability at a multiplexer with a given buffer. Our Algorithm is,

- general as it is devoid of any source modeling and very loose assumptions on the traffic characteristics like short range dependence, stationarity and ergodicity.
- efficient as it has features of fast convergence and thus can be employed for on-line estimation of capacity in a closed loop feedback control. Also the algorithm is shown to work for stationary as well as *piecewise-stationary* sources.

Also note that the buffer can be fixed or can be a function of capacity. Our algorithm is based on large buffer asymptotics and so gives correct results when the buffer sizes are large enough.

[3]More particularly the buffer overflow probability

3.5.1 Review of Large Buffer Asymptotic Theory

Consider a multiplexer with infinite buffer, with a stochastic cell arrival process and a constant service rate c. Let the arrival process A be stationary and ergodic, and let $A(t)$ be the total number of arrivals in the interval $(0, t)$. We assume that the log moment generating function of $A(t)$ exists, and define, for $\theta > 0$,

$$\Gamma_A(\theta) = \lim_{t\to\infty} \frac{1}{t} \log E \exp\left(\theta A(t)\right)$$

Assuming the arrival process to be stationary and ergodic, it is proved in [36] that for any $\theta > 0$ such that $c > \frac{\Gamma_A(\theta)}{\theta}$, (with B denoting the buffer threshold and Q the stationary queue length in the buffer)

$$\lim_{B\to\infty} \frac{1}{B} \log(P(Q > B)) \leq -\theta \tag{3.11}$$

$\frac{\Gamma_A(\theta)}{\theta}$ is called the Effective Bandwidth (EBW) required for the asymptotic behavior in Equation 3.11. For given $\theta > 0$, define

$$c_A(\theta) = \frac{\Gamma_A(\theta)}{\theta}. \tag{3.12}$$

Define, for given c,

$$\lim_{B\to\infty} \frac{1}{B} \log(P(Q > B)) = -\theta_A(c)$$

i.e., $\theta_A(c)$ is the magnitude of the asymptotic slope of the $\log(P(Q > B))$ vs B curve.

Given a positive integer ϕ, let $\phi A(t)$, denotes the number of arrivals in $(0, t)$ for an arrival process that brings ϕ units of data each time A brings one unit. Observe that,

$$\Gamma_{\phi A}\left(\frac{\theta}{\phi}\right) = \Gamma_A(\theta)$$

i.e.,

$$\frac{\Gamma_{\phi A}(\frac{\theta}{\phi})}{\frac{\theta}{\phi}} = \phi \frac{\Gamma_A(\theta)}{\theta} \tag{3.13}$$

Equation 3.13 states that *the EBW for the ϕA arrival process with asymptotic slope $\frac{\theta}{\phi}$ is ϕ times the EBW of the A arrival process with aymptotic slope θ* (see [3]) We call this the **Scaling Property of EBW**. Let $\bar{r}_W$ denote the average rate of A. The above observation and a related result constitute the following Lemma.

Lemma 3.5.1 *For ϕ positive, and a stationary and ergodic arrival process A the following are true:*

$$\text{(i)} \quad \text{for } \theta > 0,\ c_{\phi A}(\frac{\theta}{\phi}) = \phi c_A(\theta)$$

$$\text{(ii)} \quad \text{for } \mathrm{\bar{r}_A < c < R},\ \theta_{\phi A}(\phi c) = \frac{\theta_A(c)}{\phi}$$

Proof: The proof of (i) follows from Equation 3.13 and Equation 3.12. To prove (ii), observe that we know that, for $x > 0$,

$$\theta_A(x) = \theta \Leftrightarrow c_A(\theta) = x$$

By (i),

$$c_A(\theta) = x \Rightarrow c_{\phi A}(\frac{\theta}{\phi}) = \phi x$$

Hence, $\frac{\theta_A(x)}{\phi} = \theta_{\phi A}(\phi x)$.

3.5.2 Development of the Algorithm

Observe that for a given capacity, the graph of $\log P(Q > B)$ vs B is of the form shown in Figure 3.6; the Figure also shows the solution we seek. We use an affine approximation for this curve (see also [7] and [12]). Writing $P_{A,c}(Q > B)$ for a given A and c, we express this affine approximation as

$$\log P_{A,c}(Q > B) \simeq (\textit{Asymptotic Slope})\ B + (\textit{Intercept on ordinate at } B = 0)$$

$$\log P_{A,c}(Q > B) \simeq -\theta_A(c)\ B + \log P_{A,c}(Q = 0) \tag{3.14}$$

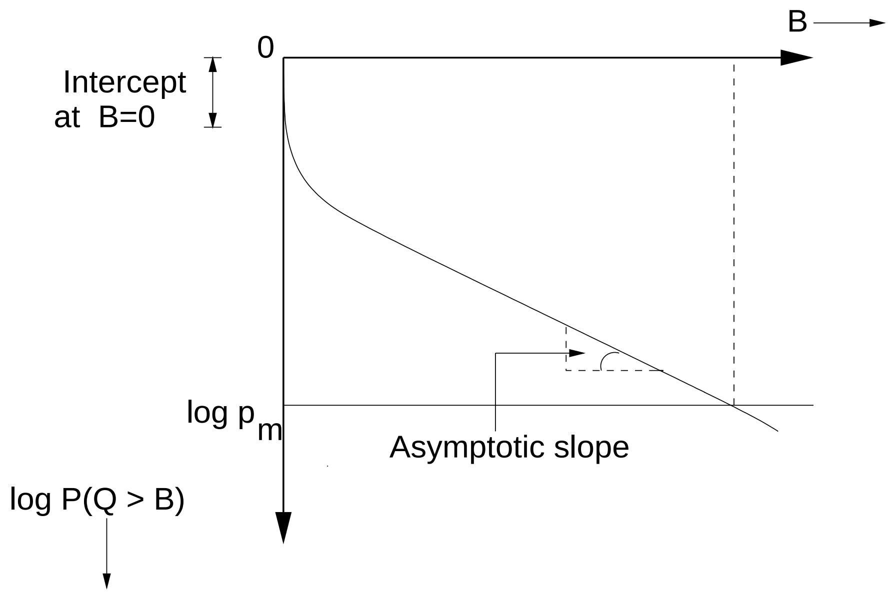

Figure 3.6: A typical $\log P(Q > B)$ Vs B curve. Note that the two main features of the curve are an intercept on the ordinate at $B = 0$, and a nearly constant asymptotic slope.

From our loss requirements, we seek the value of c, at which the right-hand-side of Equation 3.14 is equal to $\log p_m$. Thus our problem of determination of c reduces to that of finding the solution of the equation $f(c) = 0$, where,

$$f(c) = -\theta_A(c)B + \log P_{A,c}(Q = 0) - \log p_m \tag{3.15}$$

The solution of Equation 3.15 requires the determination of $\theta_A(c)$ and $\log P_{A,c}(Q = 0)$. Observe that $\theta_A(c)$, which is the asymptotic slope, as shown in Figure 3.7, can be approximated by taking two large buffer sizes B_1 and B_2 and determining $P_{A,c}(Q > B_1)$ and $P_{A,c}(Q > B_2)$. Then $\theta_A(c)$ can be expressed as (see Figure 3.7),

$$\theta_A(c) = \frac{-\log P_{A,c}(Q > B_2) + \log P_{A,c}(Q > B_1)}{B_2 - B_1} \tag{3.16}$$

Now, from the *Lemma 1*, $\theta_A(c) = \phi\ \theta_{\phi A}(\phi c)$; thus, for a fixed integer $\phi > 0$, we can

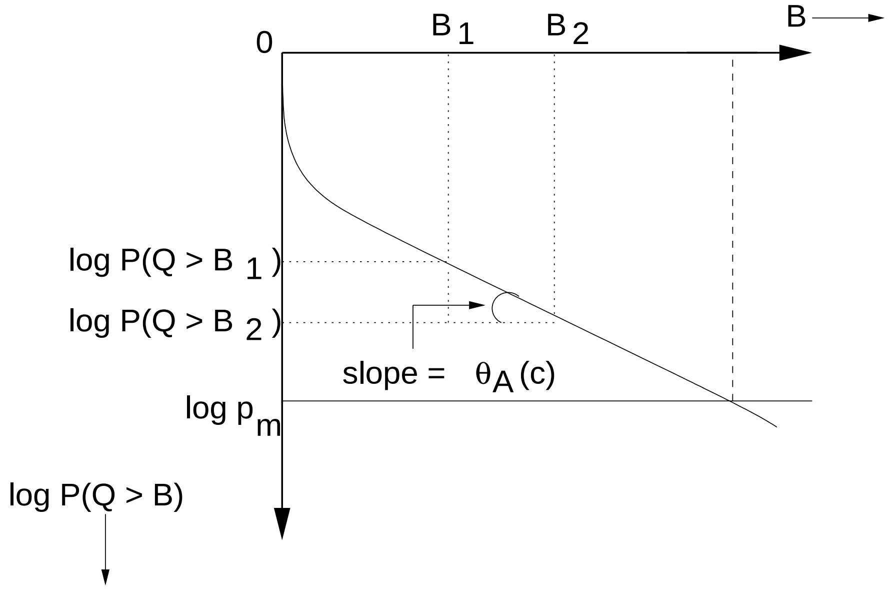

Figure 3.7: Obtaining the asymptotic slope by employing two virtual buffers B_1 and B_2.

express Equation 3.15 as,

$$f(c) = -\phi\theta_{\phi A}(\phi c)B + \log P_{A,c}(Q = 0) - \log p_m \tag{3.17}$$

where $\theta_{\phi A}(\phi c)$ is given by,

$$\theta_{\phi A}(\phi c) = \frac{-\log P_{\phi A,\phi c}(Q > B_2) + \log P_{\phi A,\phi c}(Q > B_1)}{B_2 - B_1} \tag{3.18}$$

Observe that we are determining $\theta_{\phi A}(\phi c)$ and $\log P_{A,c}(Q = 0)$ through simulations and thus these estimates are noisy; i.e., for each chosen c, we have a noisy observation of $f(c)$. Thus we enter the framework of *Stochastic Approximation Algorithms*. We shall be using the *Robbins-Monro(RM)* stochastic approximation approximation algorithm to get root of $f(c)$. The RM algorithm addresses the problem of finding the root of a function when we can only observe the function values corrupted by noise. It is an iterative algorithm that uses noisy measurements of the function for given values of the argument, and iteratively obtains an estimate of the root.

Recall that, given the argument c we can observe $f(\rho)+v$ where v is the measurement noise. In the RM algorithm, at the kth iteration, the current estimate c_k is updated as follows

$$c_{k+1} = c_k - a_k(f(c_k) + v_{k+1}) \tag{3.19}$$

where $\{a_k\}$ is a "gain" sequence. For a suitably nice function $f(\cdot)$, sufficient conditions for the convergence of the RM algorithm are [39]:

- Conditions on the gain coefficient sequence a_k
  1. $\sum_{k=0}^{\infty} a_k = \infty$
  2. $\sum_{k=0}^{\infty} a_k^2 < \infty$
- Conditions on noise:
  1. $\forall\ k \geq 0\ E(V_{k+1}|c_0, (v_i, c_i), 1 \leq i \leq k) = 0$
  2. $E(V_{k+1}^2|c_0, (v_i, c_i), 1 \leq i \leq k) < H$, for some H finite.

We digress here to mention the use of stochastic approximation approach in [32] to estimate the capacity required by a call at the network node. They considered the problem of finding the minimum capacity to support the call for a given network buffer size, arguing that for a given buffer and bandwidth, the delay is upper bounded. Here, we are finding the capacity and buffer required at the network node by a *Leaky Bucket* controlled source, having optimal leaky-bucket depth σ^*, subject to a constraint on the probability of violation of an end-to-end (shaper plus multiplexer) delay bound.

If our function $f(.)$ and the measurement noise satisfy the requirements for convergence of the RM algorithm [39], then our update at the $(k+1)th$ iteration is given by,

$$c_{k+1} = c_k - a_k(f(c_k) + v_k) \tag{3.20}$$

Here, ε_k is the noise in the k^{th} iteration. Substituting the value of $f(c_k)$ from Equation 3.17, Equation 3.20 becomes

$$c_{k+1} = c_k - a_k \left(\phi\ \theta_{\phi A}(\phi c_k) B + \log P_{A,c_k}(Q=0) - \log p_m + v_k\right) \tag{3.21}$$

3.5.3 Advantages of FSSAA

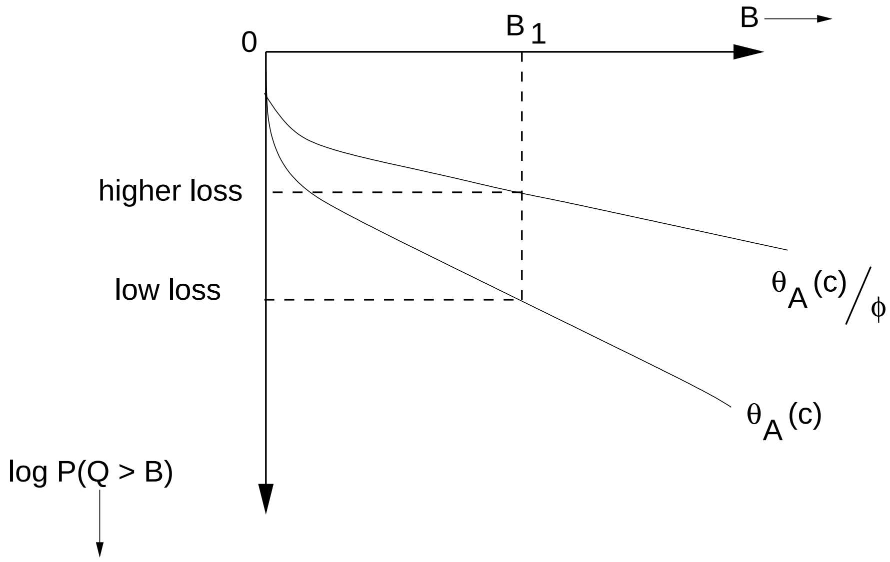

Figure 3.8: By the use of the scaling property, the probability of occurrence of the required event (*cell-loss*) increases and hence the variance in loss measurements decreases.

Observe that, in obtaining the estimate of $f(c)$ we are combining the merits of the *Scaling Property of EBW* (see [2]) with the *Virtual Buffer Technique* (see [11]). We need to determine $\theta_{\phi A}(\phi c)$ which is the same as $\frac{\theta_A(c)}{\phi}$ by *Lemma 1*. Since $\frac{\theta_A(c)}{\phi} < \theta_A(c)$, the **rare event (cell loss)** with very low probability (order of 10^{-5} or less) in the simulation, with arrival process A and capacity c (unscaled system), is made **not so rare** in the simulation with arrival process ϕA and capacity ϕc (scaled system), for the same buffer size in both the simulations; the reason for this is clear from Figure 3.8. For example, with $\phi = 4$, a CLR of 10^{-8} in the unscaled system is translated to a CLR of 10^{-2} in the scaled system.

The other issue is the estimation of $\theta_A(c)$ for given A and c. As shown in Figure 3.9, if we employ a single large virtual buffer B_2 to determine the asymptotic slope, $\theta_A(c)$ and

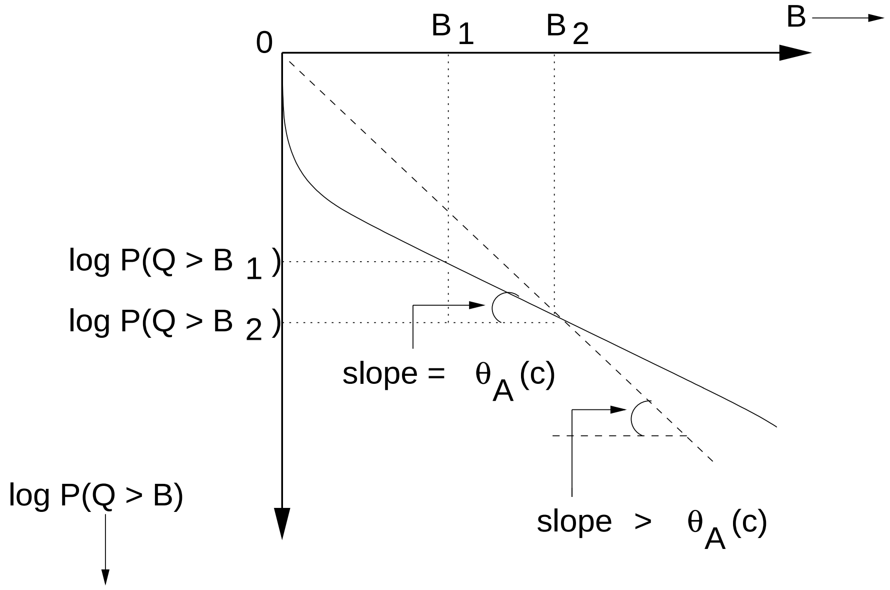

Figure 3.9: By the use of two virtual buffers instead of one the estimate of asymptotic slope is much better.

express it as $\frac{\log P(Q>B_2)}{B_2}$ then our approximation is simply the slope of the line joining a point $(B_2, \log P(Q > B_2))$ to the origin. As shown in Figure 3.9, this approximation is highly inaccurate. A more accurate estimate of the slope is proposed in [2] by considering two virtual buffers B_1 and B_2, and approximating $\theta_A(c)$ by Equation 3.16. Observe that, the slope as approximated by Equation 3.16 is a much better approximation of $\theta_A(c)$ as shown in Figure 3.9.

Since we have incorporated *Scaled Arrivals*, *Two Virtual buffers* and the *Intercept correction*, our technique for estimation of CLRs in the multiplexer should be comparatively faster than the approach proposed in [11], and more accurate than the approach proposed in [2]. In [11] the loss in a multiplexer was expressed in terms of losses in three virtual buffers of reduced sizes (obtained by appropriate scaling down of B). Due to the *Scaling Property* that we employ, the cell loss probability is increased further in the virtual buffers. Thus the variance in the CLR measurements on the virtual buffers is reduced. In [2] the loss was determined by using a linear approximation to the $P(Q > B)$ curve, without making the *Intercept correction*. Hence the loss measure was conservative. By incorporating the

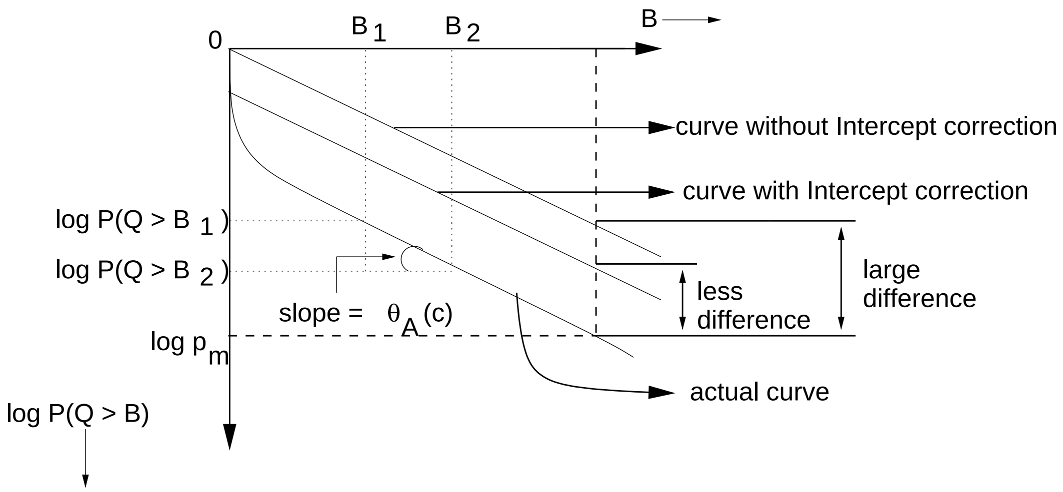

Figure 3.10: By incorporating Intercept correction in the affine approximation to the curve of $\log P_{W,c}(Q > B)$, the calculated loss at a given buffer is much closer to the actual loss than that without Intercept correction.

Intercept correction we have improved our loss calculations and they are now expected to be closer to the actual value as evident from Figure 3.10 (also see [7]).

3.6 An On-Line Algorithm for Measurement Based estimation of ρ^* using FSSAA

In this section we shall provide an on-line algorithm for estimating ρ^*. The algorithm is a direct application of FSSAA. Recall that our required ρ^* is the root of $f(\rho)$. If in FSSAA we replace c by ρ, put $B = T\rho$ and with S (the output of the VBR source) as the arrival process, then we get an iterative algorithm for estimating ρ^*, where at the kth iteration ρ is updated as,

$$\rho_{k+1} = \rho_k - a_k(f(\rho_k) + v_{k+1}) \tag{3.22}$$

where,

$$f(\rho_k) = -\phi\theta_{\phi S}(\phi\rho_k)T\rho_k + \log P_{S,\rho_k}(Q = 0) - \log p_m \tag{3.23}$$

3.6.1 Our algorithm and the conditions for convergence of RM algorithm

In our algorithm we have taken the gain sequence $\{a_k\}$ to be,

$$a_k = \frac{R}{(k+J)(-\ln q)D}$$

with J an integer, D a real number. J and D can be used to control the transient behavior and the convergence of the algorithm. Also, R and $-ln(q)$ are used to scale the gain properly. It is easy to verify that $\sum_{k=0}^{\infty} a_k = \infty$ and $\sum_{k=0}^{\infty} a_k^2 < \infty$. The conditions on noise hold approximately. The first condition requires the measurement to be conditionally unbiased. It can be argued that if we obtain the estimate of cell loss using the Virtual Buffer technique, the measurements are asymptotically unbiased as the measurement interval becomes large. Also, we are making a heuristic modification to take care of unbounded values. Whenever we encounter a zero loss (leading to an unbounded function value), the function value is artificially bounded (e.g., by taking the loss to be a small nonzero value a few orders of magnitude smaller than q). This modification ensures that the RM algorithm steps are executed only on bounded values of the function. Hence the conditional second moment of the observations *used* by the RM algorithm is bounded.

3.6.2 The Algorithm

1. Let at $k = 0$, $\rho_k = \rho_0$.

2. (a) With $B = T\rho_k$, find suitable values of B_1 and B_2 (by scaling down B). There is a trade-off in choosing the size of B_1 and B_2. If these virtual buffers are too large than our CLR measurements will be optimistic in the sense that the measured loss will be less than the true loss. On the other hand, if these virtual buffers are chosen to be small then the asymptoticity assumption may be violated and the value of $\theta_{\phi S}(\phi\rho_k)$ as obtained from Equation 3.18 may not be a good estimate of the asymptotic slope.

(b) Determine the losses in multiplexers with buffers B_1 and B_2 each with the arrival process ϕS and the server of capacity $\phi\rho_k$ over T. Calculate $P_{S,\rho_k}(Q > B_i)$ for $i = 1, 2$ which is same as CLR.

$$CLR = \frac{\text{total no. of cell lost in } T \text{ due to buffer overflow}}{\text{total no. of cells arriving in } T \text{ at the multiplexer}}$$

Then calculate $\theta_{\phi A}(\phi\rho_k)$ from Equation 3.18.

(c) Simultaneously determine $P_{S,\rho_k}(Q > 0)$ for a multiplexer with zero [4] buffer, server capacity ρ_k and arrival process S.

(d) Determine $f(\rho_k)$ through Equation 3.23.

3. Update ρ_k to ρ_{k+1} by using Equation 3.22. If $\mid \rho_k - \rho_{k+1} \mid < \epsilon$ (for some threhold ϵ) then go to Step 4 else update $k \longleftarrow k + 1$ and go to Step 2.

4. $\rho^* = \rho_k$. END

3.6.3 Detection of Change in Source Characteristics [41]

In practice the source characterisation will change with time, and hence the algorithm should be able to track the value of ρ^*. We view the source as displaying several stationary regimes, each characterized by different LB parameters. In the above algorithm, the gain goes to zero, and hence the algorithm loses its responsiveness to change as the number of iterations increases. There are two approaches. One is to use a small fixed gain in the RM algorithm; this approach is taken, in a different context, in [38]. A better approach is to *detect* a change in the source and then reset the stochastic approximation gain. We report results for the latter approach here.

We have used a simple change detection scheme based on a windowing technique. We define two windows (Figure 3.11). The first window is called the *Long Term Averaging (LTA)* window, and the second the *Change Detection (CD)* window. Let the length of the

[4] in practice a buffer of size 1 to hold the packet being served

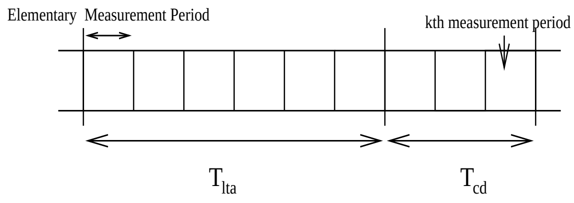

Figure 3.11: Measurement windows for change detection. When the $-\ln(.)$ of means of *measured QVPs* in the two sliding windows differ by more than a predefined threshold, a change in source characteristics is assumed to have taken place.

LTA window be T_{lta}, and that of the CD window be T_{cd}, both integer multiples of the basic measurement period. We take T_{lta} to be sufficiently large, and T_{cd} is small. Also, define

$$\epsilon_{lta}(k) \triangleq -\ln \frac{1}{T_{lta}} \sum_{i=k-T_{lta}-T_{cd}+1}^{k-T_{cd}} p_i$$

where p_i is the QVP at ith iteration. Similarly define

$$\epsilon_{cd}(k) \triangleq -\ln \frac{1}{T_{cd}} \sum_{i=k-T_{cd}+1}^{k} p_i$$

We declare that the process has changed if the two differ by more than a threshold value δ; i.e., if $|\epsilon_{lta}(k) - \epsilon_{cd}(k)| > \delta$. The choice of δ is based on a trade off between reacting to measurement errors, and thus causing false alarms, and a delayed response to actual changes in the source statistics (an analysis of such issues in a queueing context is presented in [48]). Also, after a change, the value of ϵ_{lta} will keep changing till the full window shifts into the area where the source has the new characteristics; if we keep running this detection algorithm during that period, we will get false alarms. Hence, once we detect a change, we turn off the detection algorithm for $(T_{lta} + T_{cd})$ iterations.

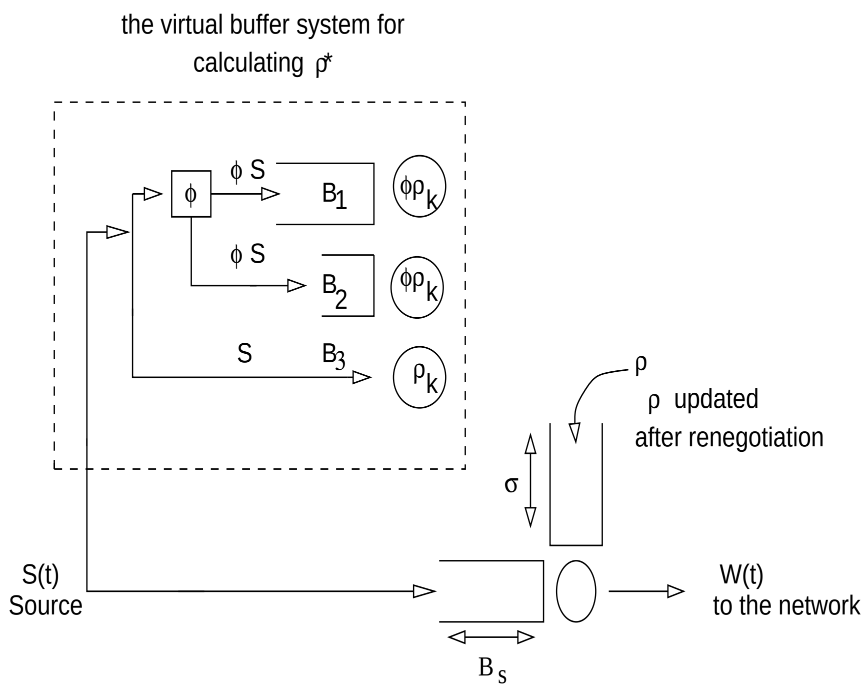

Figure 3.12: A virtual buffer system for estimating ρ^*. Three virtual buffers at each source $S(t)$, are used to obtain an estimate of $-\ln p(\rho_k, T\rho_k)$, which is used for finding the next iterate of ρ^* using the RM algorithm. The actual LB parameters are updated only periodically after renegotiation.

3.7 Algorithm Implementation and Simulation Results

3.7.1 Simulation Set-up

The simulation set-up is shown in Figure 3.12. There are three virtual buffers of sizes B_i. At the kth iterate of the algorithm, $B_i = \frac{T\rho_k}{k_i}$, where k_i is the scale down factor for virtual buffers. $B_3 = 1$ for all iterates. B_1 and B_2 are served with a capacity of $\phi\rho_k$ and B_3 with ρ_k. The arrival process to B_1 and B_2 is ϕS and to B_3 is S. B_1 and B_2 are used for calculating the asymptotic slope and B_3 for the zero buffer loss in our algorithm.

3.7.2 Simulation Results and Discussions

In the simulation, an on–off Markov modulated source (See Figure 3.13) with a mean on-time of $\frac{5}{3}$ and a mean off-time of $\frac{5}{2}$, $R = 170$ cells/unit time, delay bound $T = 5$ and QVP

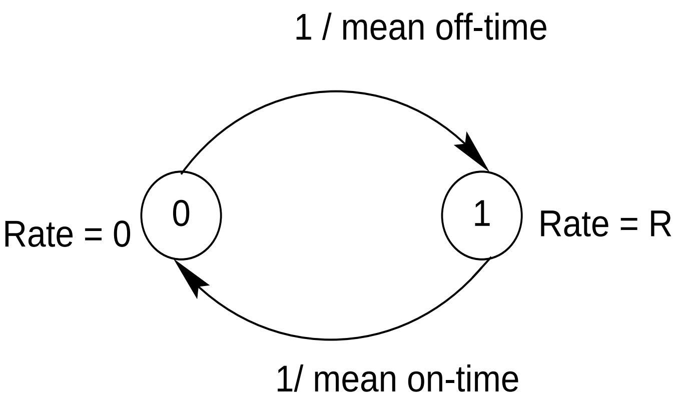

Figure 3.13: An on-off Markon modulated source

$= 10^{-5}$ was used. If we take the unit of time to be 10ms and 48 bytes of payload per cell, then these parameters will correspond to a mean on-time of 16.67ms, and a mean off-time of 25ms, peak rate of about 6.5 Mbps, mean rate of 2.6Mbps and delay constraint of 50ms. This could be a characterization of an interactive video source. The delay requirement has to be stringent for lip-synching, if the voice and video are sent by different connections. Also define $\epsilon = -\ln(q)$; it is ϵ that we plot in the simulation results; note that $-\ln 10^{-5} = 11.513$. We have worked with three measurement intervals of length 10000, 1000 and 1000 time units. The gain sequence parameter used are: $J = 3$ and $D = 4$; at the kth iteration, the virtual buffer values were $B_1 = \frac{T\rho_k}{6}$ and $B_2 = \frac{T\rho_k}{12}$; the scaling factor $\phi = 4$. Figure 3.14 shows the simulation results. There are two columns of plots in the figure; the first column shows the measured value of QVP (ϵ), and the second column shows the iterates of ρ.

Iterate No.	*Length Interval*		
	10000	1000	100
1st	141.98	145.75	129.59
5th	140.30	139.85	136.99
10th	139.94	139.39	138.13
30th	139.81	139.57	138.48

Table 3.1: Table showing the effect of the length of the measurement interval on the convergence of the ρ determination algorithm. The ρ^* obtained by EBW analysis is 139.1

- **Effect of Measurement Interval:**

  Table 3.1 gives the successive iterate of ρ for three iterate intervals. The effective bandwidth calculation gives $\rho^* = 139.1$. We notice the effect of choosing a shorter update

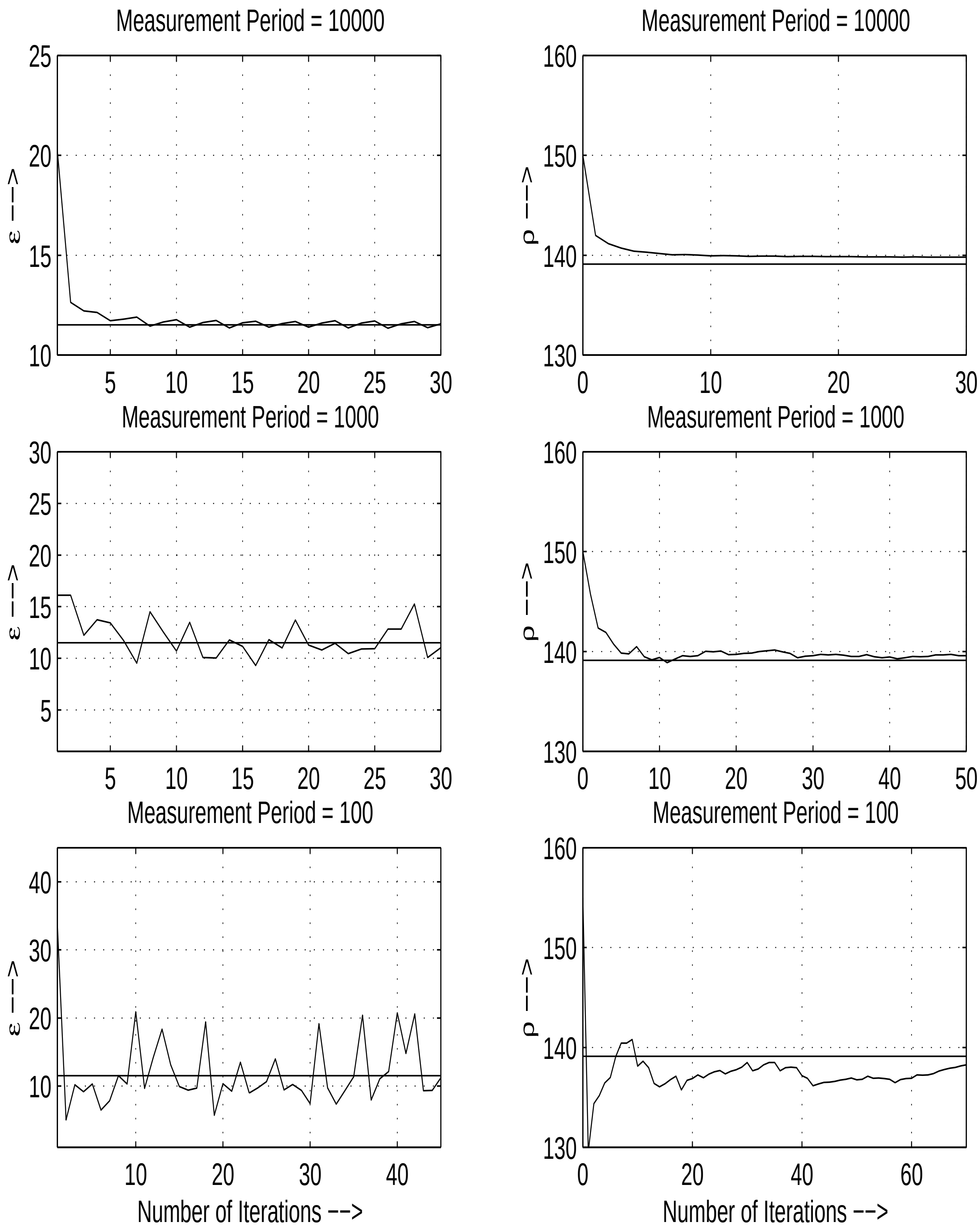

Figure 3.14: The estimates of the measured QVP (ϵ) and the ρ^* as a function of the number of iterates for measurement intervals 10000, 1000 and 100 time units.

period. As the update time period is increased the measurements are less noisy and are closer to being unbiased. Convergence itself is affected when the measurement time is very small (convergence is not evident in the third row of plots in Figure 3.14, corresponding to an update period of 100 time units, i.e., 1 second). However, convergence is reasonable for 1000 time units, which is equal to 10 seconds for the example above. The plots show that working with 10000 time units gave no significant reduction in the convergence time.

- **Variable Measurement Interval:**

  We also did the simulations with varying measurement intervals. The idea is to initially use small measurement intervals to get closer to the true ρ^* and then to increase the measurement intervals so as to get a finer estimate (Refer [21]). We worked with two types of variable interval techniques. In the first technique we initially worked with measurement interval of length 100 upto 5 iterates and then from 6th iterate onwards we changed the interval to 1000 time units. In the second technique we initially started with a measurement interval of 100 and at each successive iterate we increased the interval by 100. Once the interval becomes 1000 then we have stopped increasing the interval for successive iterates. Thus the intervals are $\{100, 200, \ldots, 1000, 1000, \ldots, 1000\}$. The simulation results are shown in Figure 3.15.

- **Change Detection Algorithm:**

  The values of the change detection algorithm parameters used were: $T_{lta} = 10, T_{cd} = 4, \delta = 4$. We take a two-state *piecewise-stationary* source having four stationary regimes. The values of (mean on-time, mean off-time) for the four regimes are $(\frac{5}{3}, \frac{5}{2})$, $(\frac{50}{3}, \frac{5}{2})$, $(\frac{5}{3}, \frac{50}{2})$ and $(\frac{50}{3}, \frac{50}{2})$. Analysis, based on the results of Section 3.3 and the effective bandwidth approach, gives a value of ρ^* as 139.1, 166.3, 135.3, 165.8 for the four regimes respectively. The update time used here is 1000 time units. In the first subplot, we plot with a solid line the ρ^* given by analysis, and with a dashed line the value given by stochastic approximation algorithm. We see that our algorithm detected all the changes without any false alarm. The stochastic approximation algorithm gives estimate of ρ^* that are near to that given by analysis. In the second subplot in

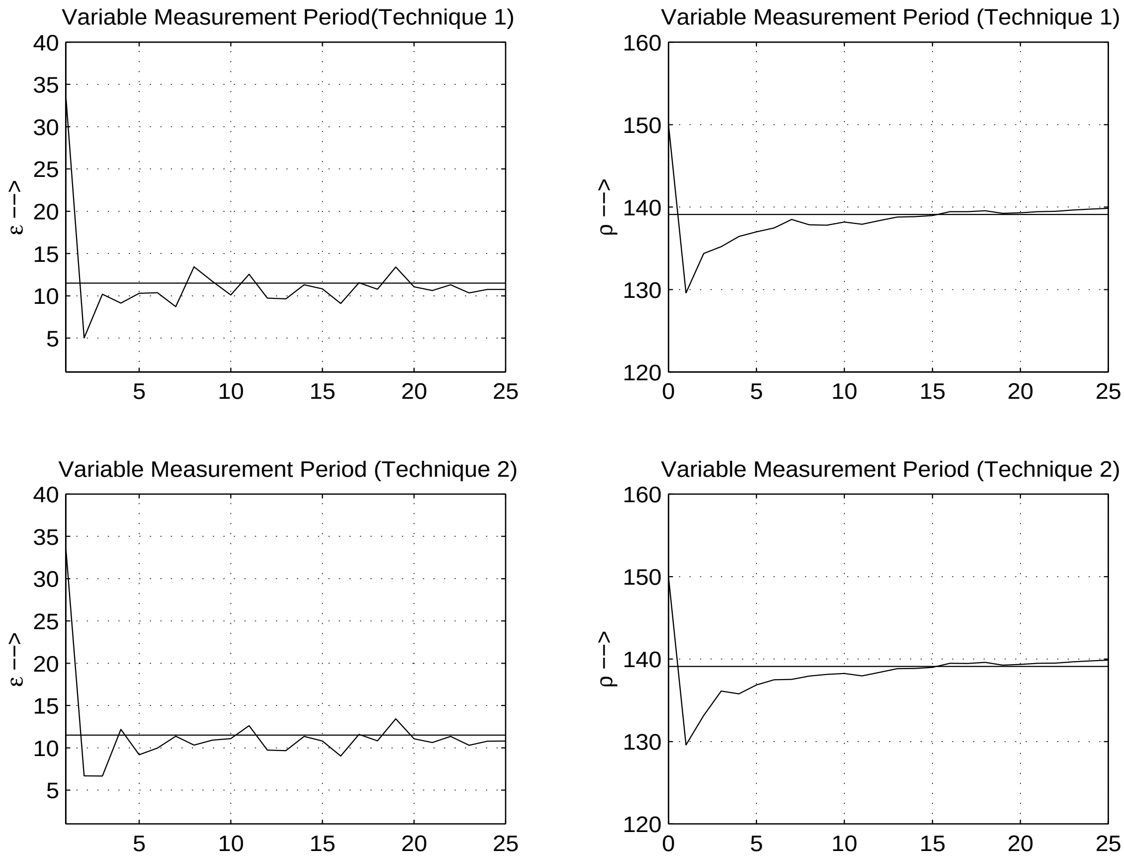

Figure 3.15: The estimates of the measured QVP (ϵ) and the ρ^* as a function of the number of iterates for two variable measurement interval techniques. Technique 1 works with 100 as the interval length upto 5 iterates and then shifts to 1000 for successive iterates. Technique 2 works with 100 as the first interval length and at each successive iterates it increase by 100 until the interval becomes 1000. For all successive iterates the interval length remains fixed at 1000.

Figure 3.15, we plot the observations of $-\ln(p_s)$.

3.8 Conclusion

In this chapter we presented the characterisation of ρ^* for Minimum Cost Lossless Multiplexing. The characterisation is taken from [41]. We here corrected the Proof for optimality of ρ^* given in [41], which was incomplete and lacking thoroughness. We have also shown the truth of the assumption on convexity of $g_q(\rho)$ for Markov modulated sources. We then proceeded to develop an on-line measurement based technique for estimation of ρ^*. In this process, we developed a *general and efficient* algorithm for on-line estimation of the minimum required capacity for a given loss probability at a multiplexer with a given buffer. The algorithm called the *Fast Simulation Based Stochastic Approximation Algorithm (FSSAA)*, incorporates the merits of the scaling property of EBW approach and the virtual buffers technique with RM algorithm.

Simulations done for an example source have validated the claims of the algorithm. We worked with a windowing technique for detecting changes (due to nonstationarities) in the source characteristics. Simulations have showed that the windowing technique tracks the changes in ρ^* due to changes in source characteristics quite efficiently.

We have noticed in simulations that the values of J and D plays a significant role in the convergence of the algorithm. For some combination of J and D the algorithm converges very fast. Thus there is a need for developing a theoretical basis for finding the values of J and D. From the simulations with variable measurement intervals we have seen that the convergence time can be greatly improved by initially working with small measurement intervals and later increasing the intervals. Thus a method for the determination of the optimum measurement interval has to be formulated.

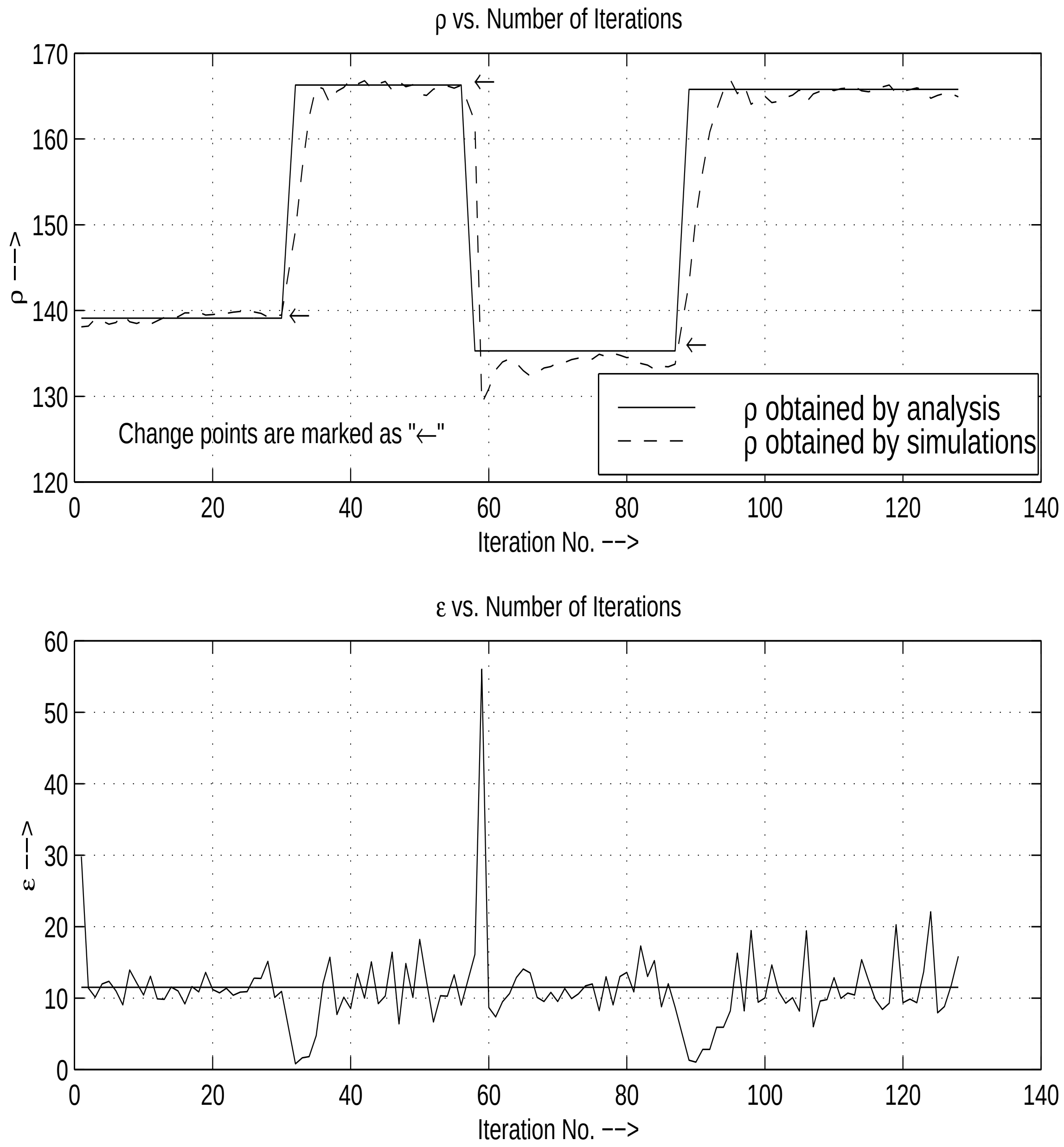

Figure 3.16: The change detection capability of the windowing technique in the stochastic approximation algorithm based on loss measurements by the virtual buffer approach. The update time is 1000 time units. The value of ρ^* obtained by analysis for the four stationary regimes are 139.1, 166.3, 135.3, 165.8 and $\delta = 4$.

Chapter 4

A Formulation for the Determination of the Optimal Bucket Size

4.1 Introduction

Observe that with $\rho = \rho^*$ our network model now consists of n homogeneous (σ, ρ^*, R) sources, $W_1, W_2, \ldots, W_n$, being fed into a queue with buffer size B_m and service rate $C = nc$. We now need to determine the optimal token bucket depth σ^*.

In this chapter we provide a characterisation of the optimal token bucket depth σ^* as the solution of a constrained optimization problem. The characterisation was initiated in [41]. The characterisation involves various functions of σ and a conjecture. In this chapter we study those functions analytically and obtain important insights into their properties. We verify our analytical conclusions by simulations. We next use an approximate Markov modulated fluid process (MMFP) model [18] for characterising the output process of the leaky bucket, the input to which is a MMFP. With this approximate model we develop the maximum eigenvalue problem for characterisation of σ^*. The comparison between the function curves obtained by the analytical results of this approximate MMFP model of the output of leaky bucket and those obtained with simulations for the two state MMFP as input source is then shown.

4.2 Shaping for Minimum Cost Lossy Multiplexing: Characterisation of an Optimal value of σ

We argued in the previous chapter that for lossless multiplexing and a shaping delay constraint of T with a QVP $= q$, the optimal leaky bucket parameters are $\sigma^* = 0$ and ρ^*. This corresponds to a per source buffer requirement of $b = \sigma^* = 0$ and a per source capacity requirement of $c = \rho^*$. A positive σ would, however, facilitate statistical multiplexing, and *if a cell loss ratio comparable to the QVP* [1] *is permitted then the network resource requirement could be reduced.* We denote the CLR by p_m. In this section we develop an approach that will finally yield the shaping parameters (σ^*, ρ^*, R) with $\sigma^* > 0$, and such that the buffer-bandwidth cost will be minimized over all operating points that yield a delay bound of T with QVP $\approx q(p_s = q$ and network CLR $p_m = q)$. For lossy multiplexing, the following bulleted list develops an approach for identifying an optimal $\sigma > 0$ for minimal cost.

- Our view of the system is that n homogeneous leaky bucket controlled sources are being fed into the multiplexer at the network, of capacity nc and buffer size B_m (see Figure 1.4). The multiplexer inputs are the LB controlled sources $W_1(t), W_2(t), \ldots, W_n(t)$, each being the output of the corresponding source shaped by (σ, ρ^*, B_s).

- Let the CLR at the multiplexer be denoted by $\lambda_{\sum_{i=1}^n W_i}(B_m, nc)$. We shall again use the effective bandwidth (EBW) approach, and assume the following additivity property of effective bandwidths

$$\lambda_{\sum_{i=1}^n W_i}(B_m, nc) = -\ln(p_m)$$

iff

$$\lambda_{(\sigma,\rho^*)}(B_m, c) = -\ln(p_m)$$

where $\lambda_{(\sigma,\rho^*)}(B_m, c)$ is ($-\ln$ of) the cell loss ratio for the output of a (σ, ρ^*, B_s) shaped source fed into a multiplexer with buffer B_m and capacity c. This says that if we scale

[1] *Note that if we think of a cell that is delayed by more than T as being equivalent to cell loss, then a cell loss ratio of q at the network node still yields a QVP of (assuming independence of the two phenomenon i.e., the delay in the shaper buffer and the loss at the network node) $1-(1-q)^2 \approx 2q \approx q$ for q small.*

the capacity with the number of sources n, with the same buffer, then the CLR does not change. *Hence, we only work with the total multiplexer buffer and the per source capacity.* Notice that this is a conservative assumption as the effective bandwidth approach ignores statistical multiplexing. *The conservatism of the EBW stems from the fact that it imposes a linear structure in a region where such linearity does not hold. This is the region where, because of statistical multiplexing, the bandwidth required to carry a given set of sources with a certain QoS is less than the sum of the bandwidths that would be needed to carry each source separately with the same QoS.* [2]

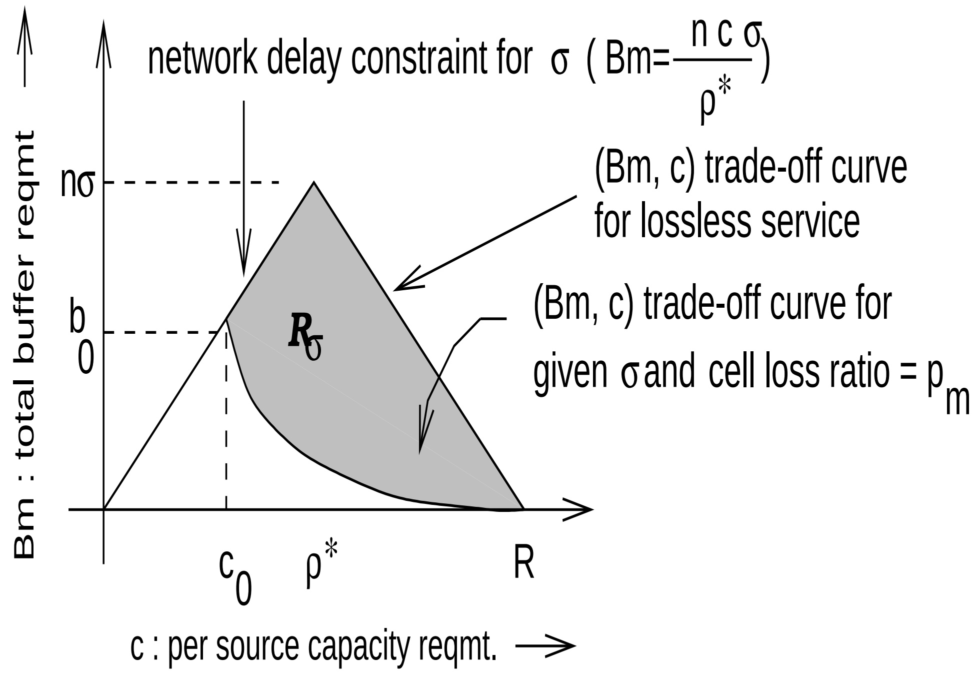

Figure 4.1: Network buffer (B_m) vs. per source capacity (c) trade-off curves at the multiplexer for *given* σ, $0 \le \sigma \le T\rho^*$. Shown are the lossless trade-off curve, the network delay constraint curve, and the lossy trade-off curve.

- Fix σ, $0 \le \sigma \le T\rho^*$, and consider n (σ, ρ^*, R) leaky bucket controlled sources feeding the network node in Figure 1.4. For lossless multiplexing we get the capacity-buffer tradeoff is given by $c = R(1 - \frac{B_m}{n\sigma}) + \rho^* \frac{B_m}{n\sigma}$ for $\rho^* \le c \le R$, shown as the straight line with negative slope in Figure 4.1. Also, the network delay constraint is given by $\frac{B_m}{nc} = \frac{\sigma}{\rho^*}$; the slanting line with positive slope in Figure 4.1. Observe that for fixed σ, the

[2] Refer to [47]

extremal points of the lossless capacity-buffer trade-off curve are $(\rho^*, n\sigma)$ and $(R, 0)$; this trade-off curve intersects the delay constraint (which is also linear) at $(\rho^*, n\sigma)$, the apex of the triangle. The triangles for increasing σ have the same base, and are nested, the apex increasing in height to $nT\rho^*$ as $\sigma \uparrow T\rho^*$. Also shown in Figure 4.1 (for the same value of σ as the one for which the triangle is drawn) is the lossy (B_m, c) trade-off curve, for CLR p_m; i.e., this is the graph $\{(B_m, c) : \lambda_{(\sigma,\rho^*)}(B_m, c) = -\ln p_m\}$. The lossy (B_m, c) trade-off curve intersects the delay constraint line at (c_0, B_0), where $c_0 < \rho^*$ and $B_0 < n\sigma$. Note that $B_0 = n\frac{\sigma}{\rho^*}c_0$, and hence c_0 solves the equation $\lambda_{(\sigma,\rho^*)}(\frac{n\sigma c_0}{\rho^*}, c_0) = -\ln(p_m)$; clearly c_0 depends on σ. We will call this *the extremal point for* σ, $0 \le \sigma \le T\rho^*$.

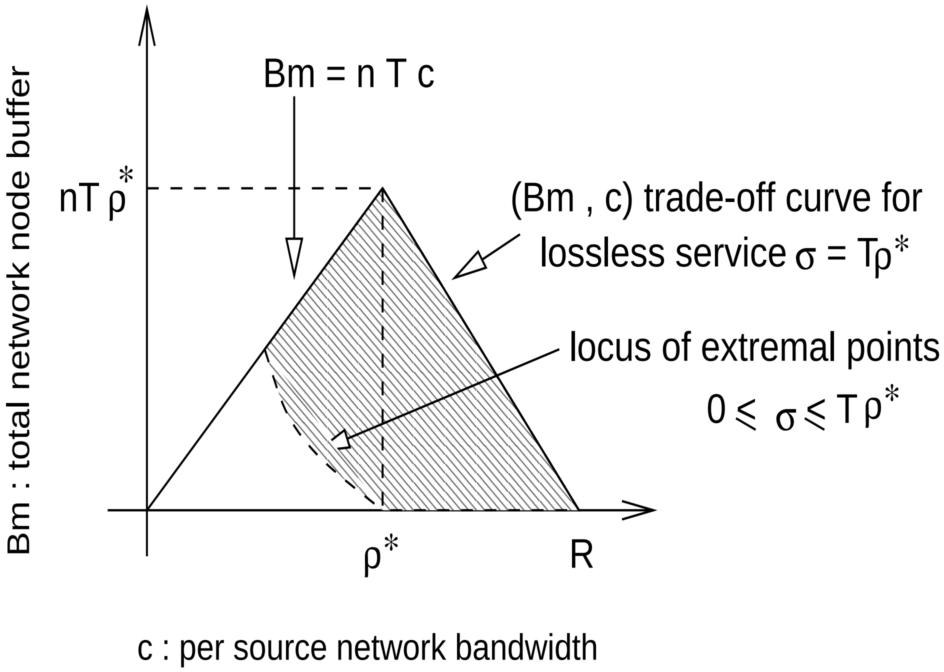

Figure 4.2: The feasible region for the network buffer vs per source capacity trade-off. The shaded region shows the (B_m, c) pairs that meet the CLR $\le p_m$ as well as the delay requirement i.e., $\mathcal{R}$.

- As σ varies over $0 \le \sigma \le T\rho^*$, the locus of the extremal points yields a curve as shown in Figure 4.2. The triangle shown in Figure 4.2 is the one shown in Figure 4.1 but for $\sigma = T\rho^*$. Note that for $\sigma = 0$, the extremal point is $(B_m = 0, c = \rho^*)$. Define, for $0 \le \sigma \le T\rho^*$, $c(\sigma)$ by $\lambda_{(\sigma,\rho^*)}(\frac{n\sigma c(\sigma)}{\rho^*}, c(\sigma)) = -\ln(p_m)$. Then the locus of the extremal

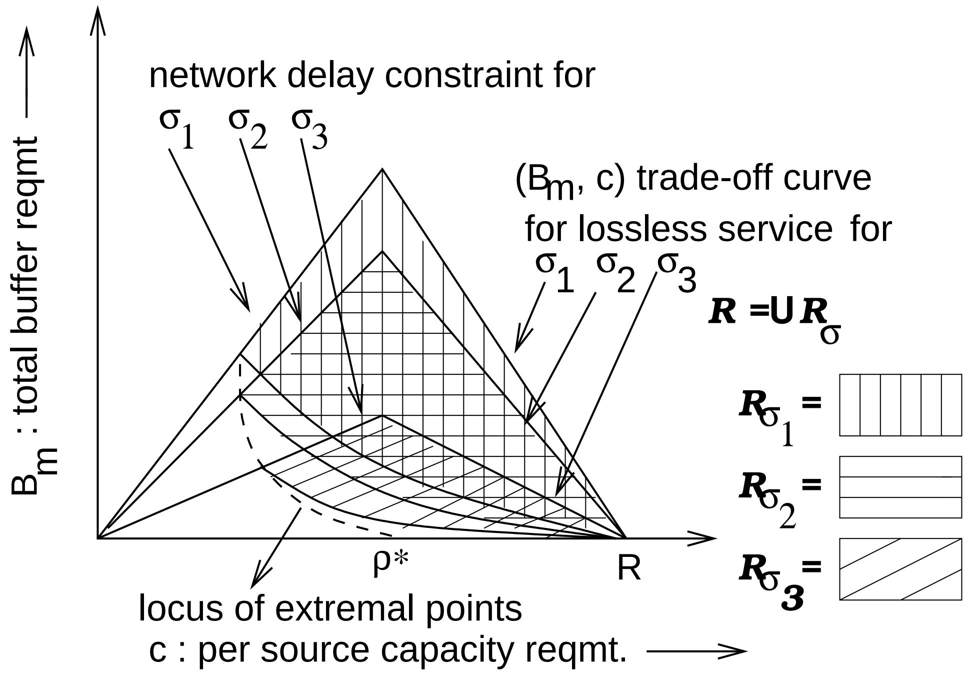

Figure 4.3: The regions $\mathcal{R}_{\sigma_1}, \mathcal{R}_{\sigma_2}, \mathcal{R}_{\sigma_3}$ with $\sigma_1 > \sigma_2 > \sigma_3$.

points shown in Figure 4.2 is defined by $\left(c(\sigma), \frac{n\sigma c(\sigma)}{\rho^*}\right)$, for $0 \leq \sigma \leq T\rho^*$. Along this curve, σ increases from the bottom to the top. It was *conjectured* in [41] that this is a convex curve as shown.

- The shaded area in Figure 4.2 is bounded by the curves $\{(B_m, c) : B_m = \frac{n\sigma c(\sigma)}{\rho^*}, c = c(\sigma), 0 \leq \sigma \leq T\rho^*\}$, $\{(B_m, c) : B_m = nTc, 0 \leq c \leq \rho^*\}$, and $\{(B_m, c) : B_m = nT\rho^* \frac{(R-c)}{(R-\rho^*)}\}$. *This shaded region is the set of* (B_m, c) *pairs for each of which there exists a* σ*,* $0 \leq \sigma \leq T\rho^*$*, such that the CLR and the delay requirements are met if the sources are controlled by* (σ, ρ^*). The truth of the preceeding claim can be established as follows. For a given σ, $\mathcal{R}_\sigma$ as shown in Figure 4.1, is the set of $\{B_m, c\}$ values such that for them the delay bound is satisfied and the loss requirements are met. Writing the mean rate of the shaper output as $\bar{w}$, $\mathcal{R}_\sigma = \{(B_m, c) : B_m > \frac{-\ln p_m}{\eta(c)}, \bar{w} \leq c \leq R\} \bigcap \{(B_m, c) : B_m < \frac{n\sigma(R-c)}{(R-\rho^*)}, \rho^* \leq c \leq R\} \bigcap \{(B_m, c) : B_m < \frac{nc\sigma}{\rho^*}, 0 \leq c \leq \rho^*\}$

Also, let $\mathcal{R}$ represent the shaded region in Figure 4.2. Then, $\mathcal{R}$ can be expressed as:

$$\mathcal{R} = \bigcup_{\sigma \in [0, T\rho^*]} \mathcal{R}_\sigma \tag{4.1}$$

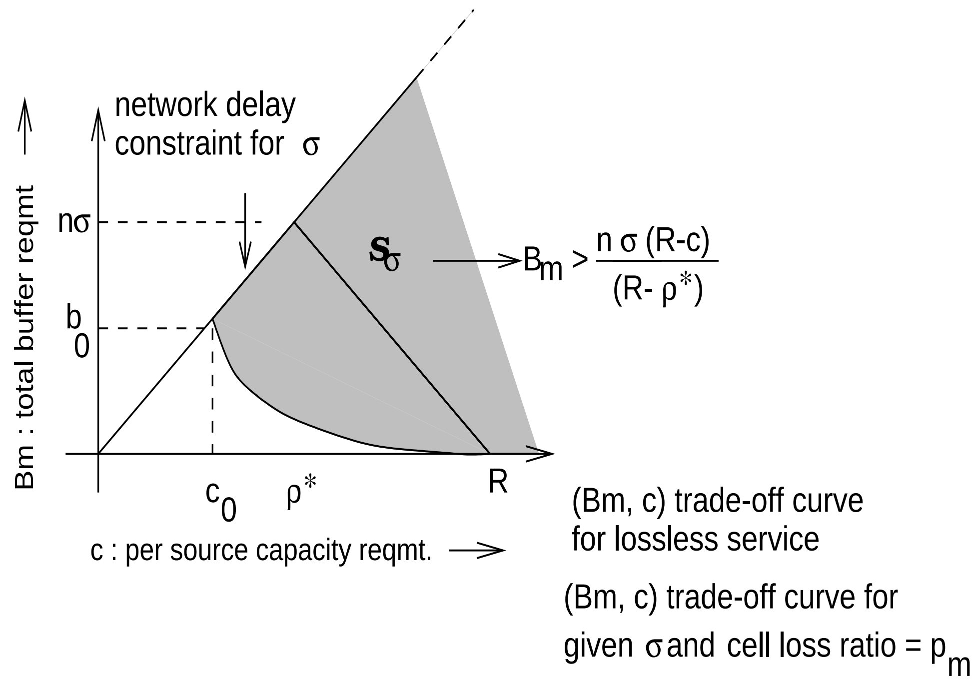

Figure 4.4: The shaded region shows the complete set of feasible (B_m, c) points for σ, i.e., the set $\mathcal{S}_\sigma$.

Figure 4.3 explains $\mathcal{R}$. Thus, since the (B_m, c) pair corresponding to each point in $\mathcal{R}_\sigma$ is such that the CLR and delay requirements are met if the source is controlled by (σ, ρ^*) and since $\mathcal{R}$ is the union of such $\mathcal{R}_\sigma$ $(0 \leq \sigma \leq T\rho^*)$ hence, for each pair (B_m, c) in the shaded region in Figure 4.2 the CLR and delay requirements are met.

An important point to be mentioned here is that *for a given σ, region $\mathcal{R}_\sigma$ as shown in Figure 4.1 is only a subset of the feasible values of (B_m, c) pair. The complete set of feasible values (let us call it $\mathcal{S}_\sigma$) is shown shaded in Figure 4.4 for a given σ. In our problem we have considered only $(B_m, c) \in \mathcal{R}_\sigma$ as feasible pair because any $(B_m, c) \in \mathcal{S}_\sigma \backslash \mathcal{R}_\sigma$ has $B_m > \frac{n\sigma(R-c)}{(R-\rho^*)}$ (i.e., the lossless constraint) and hence is surely conservative.* Define,

$$\mathcal{S} = \bigcap_{\sigma \in [0, T\rho^*]} \mathcal{S}_\sigma \setminus \mathcal{R}_\sigma$$

Let $\mathcal{U}$ be the set $\{(B_m, c) : 0 \leq B_m \leq \infty, 0 \leq c \leq \infty\}$. Let $\mathcal{U}' = \mathcal{U} \setminus \mathcal{S}$. We will consider $\mathcal{U}'$ as the universal set from now and hence, $\mathcal{R} \bigcup \bar{\mathcal{R}}$ [3] $= \mathcal{U}'$.

[3]For any set A we shall represent the complement set as $\bar{A}$

Consider now some $(B_m, c) \notin \mathcal{R}$. Now $(B_m, c) \notin \mathcal{R} \Rightarrow (B_m, c) \in \bar{\mathcal{R}} \Rightarrow (B_m, c) \in \bigcap_{\sigma \in [0, T\rho^*]} \bar{\mathcal{R}}_\sigma \Rightarrow \forall\, \sigma \in [0, T\rho^*] (B_m, c) \in \bar{\mathcal{R}}_\sigma$. Observe that $\bar{\mathcal{R}}_\sigma = \{(B_m, c) : B_m < \frac{-\ln p_m}{\eta(c)}, \bar{w} \leq c \leq R\} \bigcup \{(B_m, c) : B_m > \frac{nc\sigma}{\rho^*}, 0 \leq c \leq \rho^*\}$. Thus, $(B_m, c) \in \bar{\mathcal{R}}_\sigma \Rightarrow (B_m, c) \in \{(B_m, c) : B_m < \frac{-\ln p_m}{\eta(c)}, \bar{w} \leq c \leq R\}$ (thus the CLR requirement is violated for σ) or $(B_m, c) \in \{(B_m, c) : B_m > \frac{nc\sigma}{\rho^*}, 0 \leq c \leq \rho^*\}$ (thus the delay requirement is violated σ). Thus, we claim that *no $(B_m, c) \notin \mathcal{R}$ will have a (σ, ρ^*) that will meet the CLR and delay requirements.*

Let the per source cost function be $\gamma c + \frac{\beta B_m}{n}$, where γ and β are the per unit cost of capacity and buffers, respectively. The conjectured convexity of the curve $\{(B_m, c) : B_m = \frac{n\sigma c(\sigma)}{\rho^*}, c = c(\sigma), 0 \leq \sigma \leq T\rho^*\}$ guarantees that the minimum of the buffer-bandwidth cost function (linear) will be achieved on this boundary of the shaded region in Figure 4.2. Hence we have the optimisation problem:

$$\min_{0 \leq \sigma \leq T\rho^*} \gamma c(\sigma) + \frac{\beta \sigma c(\sigma)}{\rho^*} \tag{4.2}$$

where for each σ, $c(\sigma)$ solves (for c),

$$\lambda_{(\sigma, \rho^*)}(\frac{n\sigma c}{\rho^*}, c) = -\ln p_m \tag{4.3}$$

Let Δ denote the ratio of per unit cost of capacity to per unit cost of buffer, i.e. $\Delta = \frac{\gamma}{\beta}$. We will work with $\Delta > 1$, which means link bandwidth is more costly then node buffers. This is true as buffers are memory at the switches which is typically very large (as switches are basically computers and due to the ever-increasing technologies for memory enhancement, memory is not a limitation). However we will associate a cost parameter $\beta > 0$ to this buffer usage by the source, as the buffer space is limited and shared, a queued data of one source means that a buffer position is not available for storing other source packets; thus to discourage a connection from hogging the buffer, the cost parameter β is introduced.

Recall that our very purpose of going for *lossy* from *lossless* multiplexing is because we want a reduction in cost by working with $\sigma > 0$. Notice that, for lossless case the per source capacity and buffer requirement at the multiplexer was ρ^* and 0 (fluid) respectively. A reduction in cost can only occur if by working with a capacity less than ρ^* and buffer

> 0, the cost decreases. Since, in our optimisation problem for lossy multiplexing we are only minimizing over the extremal points (i.e., the lower boundary of $\mathcal{R}$, also the curve $\{(B_m, c) : B_m = \frac{n\sigma c(\sigma)}{\rho^*}, c = c(\sigma), 0 \leq \sigma \leq T\rho^*\})$ we shall first analytically study properties of this curve in the next section. We will study the the variation of extremal capacity and extremal buffer with σ.

4.3 Analytical Study

In this section we aim to establish analytical proofs of the variation of extremal capacity and extremal buffer with σ. The arrival process to the multiplexer is considered to be stationary and ergodic. To establish the proofs we follow the Effective Bandwidth (EBW) approach. We shall first review some preliminary theory and then shall proceed towards the proofs.

4.3.1 Preliminary Theory:

EBW of a stationary and ergodic arrival process

We consider an infinite queue with a stationary and ergodic stochastic arrival process and a constant service rate. Our aim is to find the constant service rate for the queue such that the overflow probability for a given threshold is bounded above by a given value.

We first set down some notations and assumptions. Let c denote the service rate, and $A(t_1, t_2)$ denote the number of arrivals in an interval (t_1, t_2). Since the arrival process is stationary, $A(0, t)$ has the same distribution as $A(\tau, t + \tau)$. We assume that a stationary queue length process exists for this arrival and departure process. Define $W(t)$ by,

$$W(t) = A(-t, 0) - ct$$

We assume that the log moment generating function of the process $W(t)$ exists, i.e.,

$$\Gamma_W(\eta) = \lim_{t\to\infty} \frac{1}{t} \log E(\exp \eta W(t))$$

The queue length Q at time 0 is given by [1]

$$Q = \sup_{t \geq 0} W(t)$$

The distribution of Q is the stationary queue length distribution. Let B denote the buffer threshold, and denote by q the desired upper bound on the probability $P(Q > B)$.

By a result in [2] we know that

$$\Gamma_W(\eta) < 0 \Leftrightarrow \lim_{n \to \infty} \frac{1}{B} \log\left(P(Q > B)\right) \leq -\eta$$

Now, consider the log moment generating function of $A(-t, 0)$. Denote it by $\Gamma_A(\eta)$.

$$\Gamma_A(\eta) = \lim_{t \to \infty} \frac{1}{t} \log E \exp\left(\eta A(0, t)\right) \tag{4.4}$$

Also,

$$\Gamma_A(\eta) = \Gamma_W(\eta) + \eta c$$

If we want $\Gamma_W(\eta) < 0$, then the requirement on the service rate c is obtained as follows,

$$\Gamma_W(\eta) = \Gamma_A(\eta) - \eta c < 0$$

$$\Rightarrow c > \frac{\Gamma_A(\eta)}{\eta}$$

For a given η, $c = \frac{\Gamma_A(\eta)}{\eta}$ is the required EBW. Thus,

$$c(\eta) = \frac{\Gamma_A(\eta)}{\eta} = \frac{1}{\eta} \lim_{t \to \infty} \frac{1}{t} \log E \exp\left(\eta A(0, t)\right) \tag{4.5}$$

Lemma 4.3.1 *For $\eta > 0$, $\Gamma(\eta)$ as defined in Equation 4.4 is*

1. *non-negative*

2. *non-decreasing*

3. *convex*

Proof:
Refer to [14].

With these ingredients, we proceed towards the proofs in our next subsection.

4.3.2 Proofs

The proofs in this section are valid for any stationary and ergodic output of the LB (which is the arrival process to the multiplexer) for which $\Gamma(\eta)$ is defined. For such an arrival process we have reviewed the EBW theory. In the succeeding proofs we have followed *an approach* where we will design B_m by using the equation (with c being the EBW),

$$B_m = \frac{-\ln p_m}{\eta(\sigma, c)} \tag{4.6}$$

Also, by the delay requirements we need that,

$$B_m = \frac{n\sigma c}{\rho^*} \tag{4.7}$$

From Equation 4.6 and Equation 4.7 we get,

$$\frac{n\sigma c}{\rho^*} = \frac{-\ln p_m}{\eta(\sigma, c)} \tag{4.8}$$

Observe that *for $0 \leq \sigma \leq R\rho^*$, Equation 4.8 is taken to define the lower boundary of $\mathcal{R}$ in our approach.* Thus Equation 4.8 is essentially stating the condition for the point P shown in Figure 4.5. At P, c is $c(\sigma)$, which is characterised as the solution of the Equation 4.8 and B_m corresponds to $\frac{n\sigma c(\sigma)}{\rho^*}$.

Observe that, the adopted approach (i.e., of designing B_m by taking $B_m = \frac{-\ln p_m}{\eta(\sigma,c)}$) implies that for a fixed σ the source is charged for a capacity $c(\sigma)$, obtained as a solution to Equation 4.8 and for a buffer $\frac{\sigma c(\sigma)}{\rho^*}$. In the following discussion we shall suppress the parameter ρ^* as the variable is now only σ.

Theorem 4.3.1 *For a fixed c, $\eta(\sigma, c)$ is non-increasing with increasing σ.*

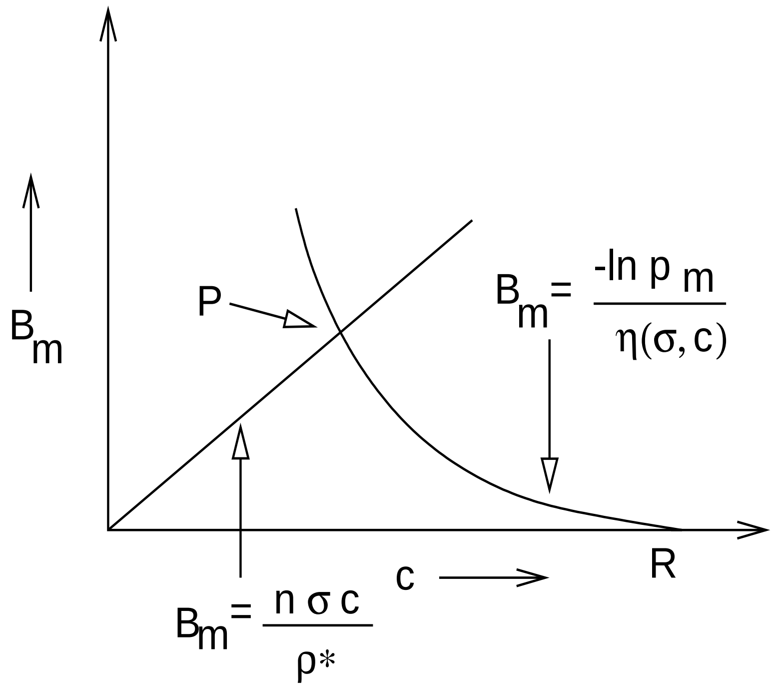

Figure 4.5: The extremal point for the approach followed in the proofs.

Proof:

Let a rate process $S(t)$ is fed into two shapers with parameters (σ, ρ^*) and $(\sigma+\Delta, \rho^*)$, $\Delta > 0$ with output rate processes $A_\sigma(t)$ and $A_{\sigma+\Delta}(t)$ respectively. $A_\sigma(t)$ and $A_{\sigma+\Delta}(t)$ are each fed to a multiplexer with capacity c and infinite buffer. Let $TB_\sigma(t)$ denote the content of the token bucket for the LB shaper with token bucket depth σ.

We shall first show that $\forall t > 0$, $A_\sigma(0,t) \leq A_{\sigma+\Delta}(0,t)$ where $A_{(.)}(0,t)$ is the cumulative amount of data (fluid) arrived to the multiplexer from the $(., \rho^*)$ shaper.

Let the two token buckets be full at $t = 0$, i.e., $TB_\sigma(0) = \sigma$ and $TB_{\sigma+\Delta}(0) = \sigma + \Delta$ and $S(t)$ is fed to both the leaky buckets. Let t_1 be defined as,

$$t_1 = \inf\{t > 0 : TB_\sigma(t) = 0\}$$

For $0 < t \leq t_1$, $A_\sigma(t) = A_{\sigma+\Delta}(t) = S(t)$ and hence $A_\sigma(0,t) = A_{\sigma+\Delta}(0,t)$.
Let t_2 be defined as, $t_2 = \inf\{t > t_1 : TB_\sigma(t) > 0\}$. Two cases may arise,

- <u>$TB_{\sigma+\Delta}(t) > 0$ for all $t_1 < t \leq t_2$:</u>

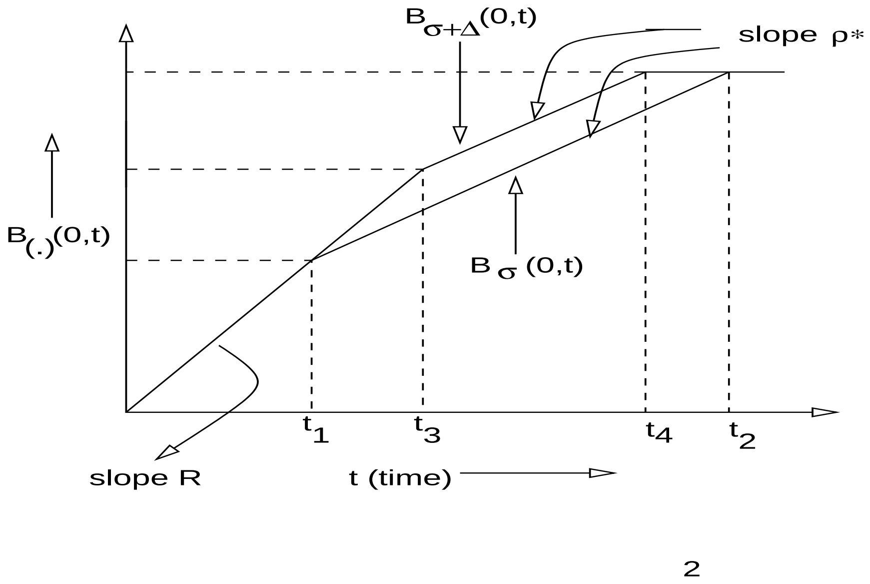

Figure 4.6: The cumulative output from (σ, ρ^*) and $(\sigma + \Delta, \rho^*)$ is bounded by $B_\sigma(0, t)$ and $B_{\sigma+\Delta}(0, t)$ respectively.

Thus, $A_{\sigma+\Delta}(t) = S(t)$ and $A_\sigma(t) = \rho^*$. Since,

$$\int_{t_1}^{t} S(t)dt \geq \rho^*(t - t_1)$$

thus,

$$A_{\sigma+\Delta}(t_1, t) = \int_{t_1}^{t} S(t)dt \geq \rho^*(t - t_1) = A_\sigma(t_1, t)$$

Thus,

$$A_{\sigma+\Delta}(0, t) \geq A_\sigma(0, t)$$

- $TB_{\sigma+\Delta}(t) = 0$ for some t, $t_1 < t < t_2$:

  Define t_3 as,

$$t_3 = \inf\{t_1 < t < t_2 : TB_{\sigma+\Delta}(t) = 0\}$$

  Observe that, $A_{\sigma+\Delta}(0, t_3) \geq A_\sigma(0, t_3)$ (by previous case discussion). Define t_4 as,

$$t_4 = \inf\{t_3 < t < t_4 : TB_{\sigma+\Delta}(t) > 0\}$$

  For $t_3 < t < t_4$, $TB_{\sigma+\Delta}(t) = TB_\sigma(t) = 0$ and hence, $A_{\sigma+\Delta}(t) = A_\sigma(t) = \rho^*$.

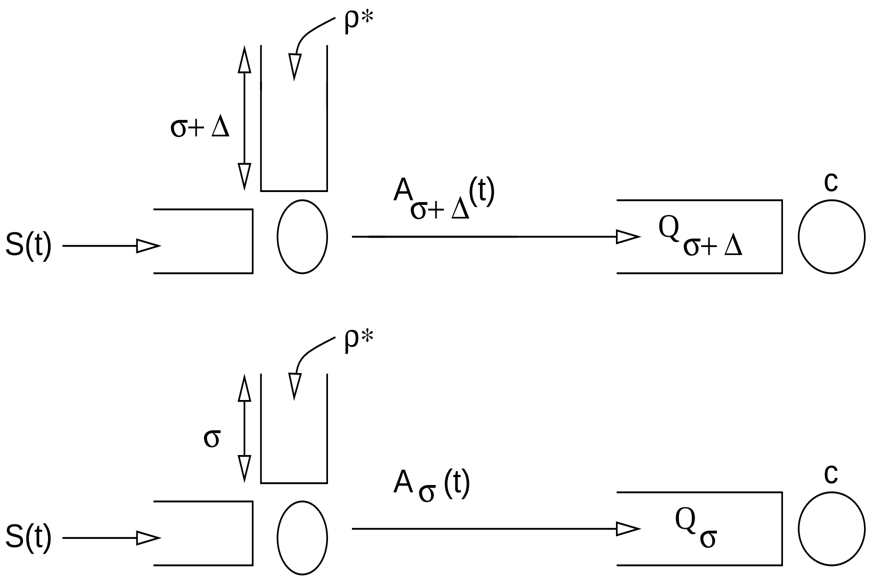

Figure 4.7: The arrangement for proving $Q_\sigma \leq Q_{\sigma+\Delta}$.

Thus, $A_{\sigma+\Delta}(t_3, t_4) = A_\sigma(t_3, t_4)$ and hence, $A_{\sigma+\Delta}(0, t) = A_{\sigma+\Delta}(0, t_3) + A_{\sigma+\Delta}(t_3, t) \geq A_\sigma(0, t_3) + A_{\sigma+\Delta}(t_3, t) = A_\sigma(0, t_3) + A_\sigma(t_3, t_4) = A_\sigma(0, t)$

At t_4, we are again back to a situation like that at t_1. Thus again we can argue along the same line and establish that $\forall\ t$ s.t. $0 \leq t \leq t_2$, $A_{\sigma+\Delta}(0, t) \geq A_\sigma(0, t)$. At t_2, $TB_\sigma(t) > 0$ and also $TB_{\sigma+\Delta}(t) > TB_\sigma(t)$. Thus again we are back to a situation like that for $t < t_1$.

Thus, we have seen that $A_\sigma(0, t) \leq A_{\sigma+\Delta}(0, t)\ \forall\ t > 0$. Also, Figure 4.6 shows the piecewise linear bounds $B_{\sigma+\Delta}(0, t)$ and $B_\sigma(0, t)$ on the process $A_{\sigma+\Delta}(0, t)$ and $A_\sigma(0, t)$ respectively. A_σ and $A_{\sigma+\Delta}$ are each fed to multiplexer with capacity c (See Figure 4.7). Let $Q_\sigma(t)$ and $Q_{\sigma+\Delta}(t)$ be the the queue lengths in the two (identical) multiplexers. We shall now prove that $Q_\sigma(t) \leq Q_{\sigma+\Delta}(t)\ \forall\ t > 0$. The approach followed is very simple. Whenever $A_\sigma = A_{\sigma+\Delta}$ the data coming to multiplexers will be coloured *red* and whenever $A_{\sigma+\Delta} > A_\sigma$, then the extra data coming to $Q_{\sigma+\Delta}$ will be coloured *blue*. Clearly the total queue length $Q_{\sigma+\Delta}$ is invariant with the service distribution. We will serve the *blue* data only when all of *red* data is served. Thus, due to the fact that $A_{\sigma+\Delta}(0, t) \geq A_\sigma(0, t)$, When all the red data has been served than either both the queues become empty simultaneously or $Q_{\sigma+\delta}$ has some *blue* data to be served. Thus $Q_\sigma \leq Q_{\sigma+\Delta} \forall\ t$. The distribution of Q_σ and $Q_{\sigma+\Delta}$ is the stationary queue length distribution. Now, for two random variable X_1 and

X_2, $X_1 \geq X_2$ implies that for a fixed B, $P(X_1 > B) \geq P(X_2 > B)$. As $Q_{\sigma+\Delta} \geq Q_\sigma$ thus $P(Q_{\sigma+\Delta} > B) \geq P(Q_\sigma > B)\ \forall B$. Thus

$$\frac{\ln P(Q_{\sigma+\Delta} > B)}{B} \geq \frac{\ln P(Q_\sigma > B)}{B}\ \forall B \tag{4.9}$$

$$\Rightarrow \lim_{B\to\infty} \frac{\ln P(Q_{\sigma+\Delta} > B)}{B} \geq \lim_{B\to\infty} \frac{\ln P(Q_\sigma > B)}{B}$$

$$\Rightarrow \eta(\sigma+\Delta, c) \leq \eta(\sigma, c)$$

as η is the negative of the asymptotic slope. Since this is true for any $\Delta > 0$, hence $\frac{\partial\eta(\sigma,c)}{\partial\sigma} \leq 0$, provided this derivative exists.

Theorem 4.3.2 *For a given σ, $\eta(\sigma, c)$ is non-decreasing with increasing c.*

Proof:

The above theorem has been proved to be true for a Markov-modulated fluid source in [19]. We will now prove it for a stationary and ergodic arrival process. From Equation 4.5

$$c(\eta) = \frac{\Gamma(\eta)}{\eta}$$

Now, from Lemma 4.3.1 $\Gamma(\eta)$ is non-negative, non-decreasing and convex. Also, at $\eta = 0$, $\Gamma(\eta) = 0$ as seen from the definition of $\Gamma(\eta)$ in Equation 4.4. Consider the following Lemma,

Lemma 4.3.2 *If $f(x)$ is non-decreasing and convex for $x > 0$ and $f(0) = 0$, then $\frac{f(x)}{x}$ is non-negative and non-decreasing.*

Proof:

Let,

$$g(x) = \frac{f(x)}{x}$$

Non-negativity is obvious. To show that $g(x)$ is non-decreasing, we need to show that for any $x_2 > x_1$,

$$\frac{f(x_1)}{x_1} \leq \frac{f(x_2)}{x_2}$$

Writing x_1 as the convex combination of x_2 i.e., $x_1 = \left(\frac{x_1}{x_2}\right) x_2 + \left(1 - \frac{x_1}{x_2}\right) 0$, by the convexity of $f(x)$ we have,

$$f\left(\left(\frac{x_1}{x_2}\right) x_2 + \left(1 - \frac{x_1}{x_2}\right) 0\right) \leq \left(\frac{x_1}{x_2}\right) f(x_2) + \left(1 - \frac{x_1}{x_2}\right) f(0)$$

Since, $f(0) = 0$, we get

$$f(x_1) \leq \frac{x_1}{x_2} f(x_2)$$

Hence, from Lemma 4.3.2, $\frac{\Gamma(\eta)}{\eta}$ is non-decreasing.

Remark: From now onwards in the proofs *whenever we are using derivatives we are assuming that the derivatives exists.*

Theorem 4.3.3 *The function* $B_{m_{(\sigma)}} \equiv B_m(\sigma, c(\sigma)) = \frac{n\sigma c(\sigma)}{\rho^*}$ *is non-decreasing in* σ, *i.e.* $\frac{dB_{m_{(\sigma)}}}{d\sigma} \geq 0$.

Proof:

We know, that $c(\sigma)$ is obtained as the solution to Equation 4.8 which is reproduced again below,

$$\frac{n\sigma c}{\rho^*} = \frac{-\ln p_m}{\eta(\sigma, c)} \tag{4.10}$$

Differentiating both the sides of Equation 4.10 w.r.t. σ and putting $a = \frac{-\rho^* \ln p_m}{n}$, we get, (for typographical simplicity we will write $c(\sigma)$ as c and $\eta(\sigma, c)$ as η)

$$\frac{d}{d\sigma}(\sigma c) = \frac{d}{d\sigma}\left(\frac{a}{\eta}\right)$$

$$\Rightarrow c + \sigma\frac{dc}{d\sigma} = \frac{-a}{\eta^2}\left(\frac{\partial\eta}{\partial c}\frac{dc}{d\sigma} + \frac{\partial\eta}{\partial\sigma}\right)$$

$$\Rightarrow \frac{dc}{d\sigma}\left(\sigma + \frac{a}{\eta^2}\frac{\partial\eta}{\partial c}\right) = -c - \frac{a}{\eta^2}\frac{\partial\eta}{\partial\sigma} \tag{4.11}$$

Substituting Equation 4.10 in Equation 4.11, we get,

$$\frac{dc}{d\sigma}\left(\sigma + \frac{a}{\eta^2}\frac{\partial\eta}{\partial c}\right) = \frac{-a}{\sigma\eta} - \frac{a}{\eta^2}\frac{\partial\eta}{\partial\sigma}$$

$$\Rightarrow \frac{dc}{d\sigma} = \frac{\frac{-a}{\eta}\left(\frac{1}{\sigma} + \frac{1}{\eta}\frac{\partial\eta}{\partial\sigma}\right)}{\left(\sigma + \frac{a}{\eta^2}\frac{\partial\eta}{\partial c}\right)} \tag{4.12}$$

For, $\frac{dB_{m_{(\sigma)}}}{d\sigma} \geq 0$, we require

$$\frac{dB_{m_{(\sigma)}}}{d\sigma} = \frac{n}{\rho}\left(\sigma\frac{dc}{d\sigma} + c\right) \geq 0$$

$$\Leftrightarrow \frac{dc}{d\sigma} \geq \frac{-c}{\sigma}$$

Substituting the value of $\frac{dc}{d\sigma}$ from Equation 4.12, we get,

$$\Leftrightarrow \frac{\frac{-a}{\eta}\left(\frac{1}{\sigma} + \frac{1}{\eta}\frac{\partial\eta}{\partial\sigma}\right)}{\left(\sigma + \frac{a}{\eta^2}\frac{\partial\eta}{\partial c}\right)} \geq \frac{-a}{\sigma^2\eta}$$

$$\Leftrightarrow \frac{\left(\frac{1}{\sigma} + \frac{1}{\eta}\frac{\partial\eta}{\partial\sigma}\right)}{\left(\sigma + \frac{a}{\eta^2}\frac{\partial\eta}{\partial c}\right)} \leq \frac{1}{\sigma^2}$$

Observe that since $\frac{\partial\eta}{\partial c} \geq 0$ by Theorem 4.3.2, we get

$$\Leftrightarrow \frac{1}{\sigma} + \frac{1}{\eta}\frac{\partial\eta}{\partial\sigma} \leq \frac{1}{\sigma} + \frac{a}{\sigma^2\eta^2}\frac{\partial\eta}{\partial c}$$

For $\eta > 0$,

$$\Leftrightarrow \eta\frac{\partial\eta}{\partial\sigma} \leq \frac{a}{\sigma^2}\frac{\partial\eta}{\partial c}$$

$$\Leftrightarrow \frac{\frac{\partial\eta}{\partial\sigma}}{\frac{\partial\eta}{\partial c}} \leq \frac{a}{\sigma^2\eta} = \frac{c}{\sigma}$$

$$\Leftrightarrow \sigma\frac{\partial\eta}{\partial\sigma} - c\frac{\partial\eta}{\partial c} \leq 0 \tag{4.13}$$

Observe that $\frac{\partial\eta}{\partial c} \geq 0$ by Theorem 4.3.2 and $\frac{\partial\eta}{\partial\sigma} \leq 0$ by Theorem 4.3.1, thus the condition required by Equation 4.13 is met.

Assumption 4.3.1 *For each $\sigma \in (0, T\rho^*)$, the c obtained as a solution to*

$$\lambda_\sigma(\frac{n\sigma c}{\rho^*}, c) = -\ln p_m$$

i.e., $c(\sigma)$ is unique.

Note: For the uniqueness of $c(\sigma)$ for each σ, by Equation 4.12 we get the following condition on the property of η,

$$\frac{\partial \eta}{\partial \sigma} \neq -\frac{\eta}{\sigma}$$

Theorem 4.3.4 *Assuming $c(\sigma)$ to be continuous, under the Assumption 4.3.1, $c(\sigma)$ is monotonically decreasing.*

A sketch of the Proof:

To prove we need two observations:

1. At $\sigma = 0$, $c(\sigma) = \rho^*$: This can be explained as follows.

   For lossless multiplexing, $c(0) = \rho^*$, which is the per source capacity at which the delay constraint line intersects the capacity-buffer tradeoff for lossless service. Also, as the lossy resource requirements are always less than the lossless requirements therefore for any loss in the mux the lossy tradeoff curve is always below the lossless tradeoff. For $\sigma = 0$, the (B_m, c) tradeoff for lossless and lossy is same, from ρ^* to R; i.e., the x-axis in Figure 4.1 from ρ^* to R. Let the output process $W(t)$ from the leaky bucket have mean rate $\bar{w}$ ($\bar{w} \leq \rho^*$ and $\bar{w} = \bar{r}$ if $\rho^* \geq \bar{r}$, where $\bar{r}$ is the mean rate of the source). Then for $\bar{w} \leq c \leq \rho^*$, the lossy curve is monotonically decreasing with being 0 at ρ^* and tending to ∞ as c approaches $\tilde{w}$. Thus the point where the delay curve intersects the lossless curve is still $(\rho^*, 0)$ for $\sigma = 0$. Thus for $\sigma = 0$, $c(\sigma) = \rho^*$.

2. For any $\sigma > 0$, $c(\sigma) < \rho^*$: Observe that for the lossless case, $\forall\ \sigma$, $c(\sigma) = \rho^*$, obtained as the point of intersection of the delay line and the lossless line. The lossless line is the bound on the capacity-buffer tradeoff. Let $B_{m_\sigma}(c)$ be the minimum value of total buffer required at the multiplexer for supporting n (σ, ρ^*) sources with per source capacity c with a CLR of p_m. For lossy case the capacity-buffer tradeoff is always less than the that for the lossless case. Thus,

$$B_{m_\sigma}(\rho^*) < n\sigma$$

Also,

$$B_{m_\sigma}(\rho^*) = \frac{n\sigma c_1}{\rho^*} \text{ for some } c_1 < \rho^*$$

Thus, since,

- $B_{m_\sigma}(c)$ is continuous on $[\bar{w}, \rho^*]$ and
- $B_{m_\sigma}(\rho^*) \neq B_{m_\sigma}(\bar{w})$

by the *Intermediate value theorem* [4] for functions continuous on closed intervals we have,

$$B_{m_\sigma}(c) = \frac{n\sigma c}{\rho^*} \text{ for some } c\text{, s.t. } \bar{w} < c < c_1 < \rho^*$$

Thus $\forall\ \sigma \in (0, T\rho^*]$, $c(\sigma) < \rho^*$.

To prove that $c(\sigma)$ is decreasing in σ, we provide a proof by *contradiction.* Let us consider two values of σ, σ_1 and σ_2 with $0 < \sigma_1 < \sigma_2$. Let us assume that $c(\sigma_1) < c(\sigma_2)$. Thus by the *Intermediate value theorem,* $\exists\ \sigma_3$, $\sigma_1 < \sigma_3 < \sigma_2$, s.t. $c(\sigma_3) = \frac{c(\sigma_1)+c(\sigma_2)}{2}$. Also since $c(\sigma_1) < \rho^*$ and $c(\sigma_2) < \rho^*$ and $c(0) = \rho^*$, again by the *Intermediate value theorem* $\exists\ \sigma_4$, $0 < \sigma_4 < \sigma_1$, s.t. $c(\sigma_4) = c(\sigma_3)$. But this violates the Assumption 4.3.1, *a contradiction.* Thus, we conclude that $c(\sigma_2) < c(\sigma_1)$. Since, this is true for any σ_1 and σ_2 with $\sigma_1 < \sigma_2$, hence, $c(\sigma)$ is monotonically decreasing.

Corollary 1 *Under the Assumption 4.3.1, the function* $B_{m_{(c)}} \equiv B_m(\sigma(c), c) = \frac{n\sigma(c)c}{\rho^*}$, *with* $\sigma(c)$ *being the inverse of the function* $c(\sigma)$, *is non-increasing, i.e.* $\frac{dB_{m_{(c)}}}{dc} \leq 0$.

Proof:

Observe that,

$$\left[\frac{dB_{m_{(c)}}}{dc}\right]_{(\sigma(c),c)} = \left[\frac{dB_{m_{(\sigma)}}}{d\sigma}\right]_{(\sigma(c),c)} \left[\frac{dc}{d\sigma}\right]_{(\sigma(c),c)}$$

$\frac{dB_{m_{(\sigma)}}}{d\sigma} \geq 0$ by Theorem 4.3.3 and $\frac{dc}{d\sigma} < 0$ by Theorem 4.3.4.

[4] If a function f is continuous on $[a, b]$ and $f(a) \neq f(b)$, then it assumes every value between $f(a)$ and $f(b)$.

It was conjectured in [41] that the curve $\{(B_m, c) : B_m = \frac{n\sigma c(\sigma)}{\rho^*}, c = c(\sigma), 0 \leq \sigma \leq T\rho^*\}$ is convex decreasing. By Corollary 1 we have established that it is decreasing.

We now provide an approach to prove the *convexity*. For the convexity we need, $\frac{d^2 B_m(\sigma(c),c)}{dc^2} > 0$. By the approach followed,

$$B_m(\sigma(c), c) = k\sigma(c)c$$

where $k = \frac{n}{\rho^*}$.

$$\frac{d^2 B_m(\sigma(c), c)}{dc^2} = k\left(c\frac{d^2\sigma}{dc^2} + 2\frac{d\sigma}{dc}\right)$$

Thus we need to show,

$$\frac{d^2\sigma}{dc^2} > \frac{-2}{c}\frac{d\sigma}{dc} \tag{4.14}$$

From Equation 4.12, we have $\frac{dc}{d\sigma}$ in terms of the first partial derivatives of η. We can similarly express $\frac{d^2\sigma}{dc^2}$ in terms of the (first and second) partial derivatives of η. Thus the requirement in Equation 4.14 can be boiled down to a condition involving η and its first and second partial derivatives.

4.4 An Off-Line Algorithm for determining σ^*

In this section we shall provide an algorithm for determining σ^*. The algorithm is used for the verification of the properties of various functions that we have analytically studied in the previous section. Recall that the optimisation problem for determination of σ^*, that we formulated in Section 4.2, is

$$\min_{0\leq\sigma\leq T\rho^*} \gamma c(\sigma) + \frac{\beta\sigma c(\sigma)}{\rho^*} \tag{4.15}$$

where $c(\sigma)$ is the solution (for c) for each σ of the equation,

$$\lambda_{(\sigma,\rho^*)}(\frac{n\sigma c}{\rho^*}, c) = -\ln p_m \tag{4.16}$$

The solution of Equation 4.15 requires $c(\sigma)$ which is obtained by solving Equation 4.16 for c, for each $\sigma \in (0, T\rho^*)$. For a given σ, to solve Equation 4.16 we need to know the value of $\lambda_{(\sigma,\rho^*)}(\frac{n\sigma c}{\rho^*}, c)$, which is the CLR in a multiplexer with a buffer of size $\frac{n\sigma c}{\rho^*}$, arrival process $W(t)$ (the (σ, ρ^*) controlled $S(t)$), and a server of capacity c; see Figure 1.4. We are obtaining the value of $\lambda_{(\sigma,\rho^*)}(\frac{n\sigma c}{\rho^*}, c)$ through measurements over a finite time interval. Thus the values obtained are noisy. This puts us in the framework of stochastic approximation. Also, because of the very small values of CLR, $P(Q > \frac{n\sigma c}{\rho^*})$ is a rare event. We therefore need some speed-up techniques for getting a low variance measurement of this rare event. Observe that the above requirements of stochastic approximation and speed-up puts us in the framework of FSSAA. We will use FSSAA to determine $c(\sigma)$ and having obtained $c(\sigma)$, $\forall \sigma \in (0, T\rho^*)$ we shall have the value of our objective function (i.e., the cost) for each sigma. The optimisation is then done off-line. We shall now describe our *Off-Line Algorithm* which is simply an application of FSSAA developed in Section 3.5.

The Off-Line Algorithm

Determination of $c(\sigma)$:

Since $\forall \sigma \in (0, T\rho^*)$, $c(\sigma)$ is obtained by solving Equation 4.16 for c, thus our problem of determination of $c(\sigma)$ reduces to finding the root of the equation $f(c) = 0$, where

$$f(c) = -\theta_W(c)(\frac{n\sigma c}{\rho^*}) + \log P_{W,c}(Q = 0) - \log p_m \tag{4.17}$$

We shall employ FSSAA to solve $f(c) = 0$. Using the scaling factor s and virtual buffers B_1 and B_2, where B_1 and B_2 are obtained by scaling down $B_m(= \frac{n\sigma c}{\rho^*})$ we can express Equation 4.17 as

$$f(c) = -s\theta_{sW}(sc)(\frac{n\sigma c}{\rho^*}) + \log P_{W,c}(Q = 0) - \log p_m \tag{4.18}$$

where $\theta_{sW}(sc)$ is given by,

$$\theta_{sW}(sc) = \frac{-\log P_{sW,sc}(Q > B_2) + \log P_{sW,sc}(Q > B_1)}{B_2 - B_1}$$

Again assuming that our function and the measurement noise satisfy the requirements for convergence of the RM algorithm [5], our update at the $(k+1)$th iterate is given by,

$$c_{k+1} = c_k - a_k(f(c_k) + \epsilon_k) \tag{4.19}$$

with $\{a_k\}$ being the gain sequence and ϵ_k the noise at the kth measurement. We have used the same gain sequence as was used for the determination of ρ^* (Refer to Section 3.6.1).

Determination of σ^*:

Having obtained $c(\sigma)$, $\forall \sigma \in (0, T\rho^*)$, we get the value of the objective function ($= \gamma c(\sigma) + \beta \frac{\sigma c(\sigma)}{\rho^*}$) for each σ. Then σ^* is simply that $\sigma \in (0, T\rho^*)$ at which the cost is minimum i.e,

$$\sigma^* = arg \min_{\sigma \in (0, T\rho^*)} \left(\gamma c(\sigma) + \beta \frac{\sigma c(\sigma)}{\rho^*} \right)$$

4.5 Algorithm Implementation and Simulation Results

4.5.1 Simulation Setup

Our simulation setup consists of two blocks: a (σ, ρ^*, R) conformant source and a multiplexer unit (Figure 4.8). The multiplexer unit consists of three virtual multiplexers. One (call it *Bufferless-mux*) has buffer size of one to hold the packet in service and is served by a server of capacity c. The other two, have buffers of size B_1 and B_2, obtained by appropriate scaling (down) of $B_m (= \frac{n\sigma c}{\rho^*})$ and each have a server of capacity sc. We will call the multiplexers with buffers B_1 and B_2 as *Virtual-mux.1* and *Virtual-mux.2* respectively. The (σ, ρ^*, R) compatible source consists of a two state Markov modulated fluid, the output $S(t)$ of which is passed through a LB shaper. The output $W(t)$ of the LB shaper is then fed to the multiplexer unit.

$W(t)$ is scaled by s and the $sW(t)$ process is fed to *Virtual-mux.1* and *Virtual-mux.2*, and the process $W(t)$ is fed to the *Bufferless-mux*. This means that for each unit of data

[5] Refer to [39] and Section 3.6.1

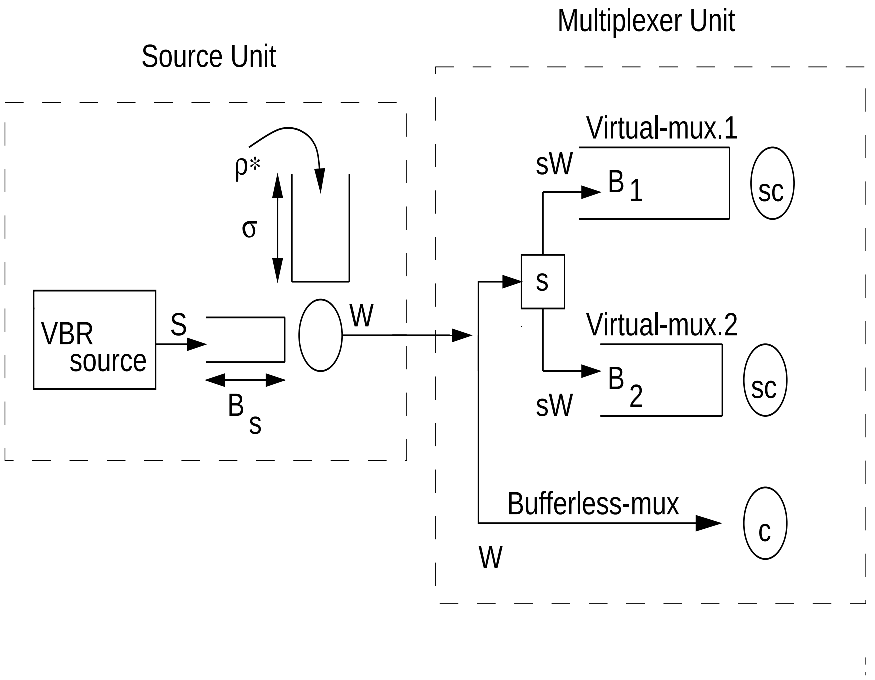

Figure 4.8: The Simulation Setup

arriving at the *Bufferless-mux* there are s units of data arriving at each of *Virtual-mux.1* and *Virtual-mux.2*.

We will find the cell loss occurring in the three multiplexers (*due to buffer overflow*), i.e., in *Virtual-mux.1 Virtual-mux.2* and *Bufferless-mux*. Then we calculate the CLR which is the same as $P(Q >$ Buffersize$)$ for each multiplexer; this is given by,

$$\text{CLR} = \frac{\text{total no. of cells lost in a simulation cycle due to buffer overflow}}{\text{total no. of cells arriving in a simulation cycle at the multiplexer}}$$

In the above expression for CLR the term *simulation cycle* denotes the length of the time the simulation is done for obtaining $f(c_k)$ at k^{th} iteration. It is the same as inter-update time in our simulations. The measurements obtained are then used to obtain the function $f(c)$ from Equation 4.18 which is then employed to get the iterates of c from Equation 4.19.

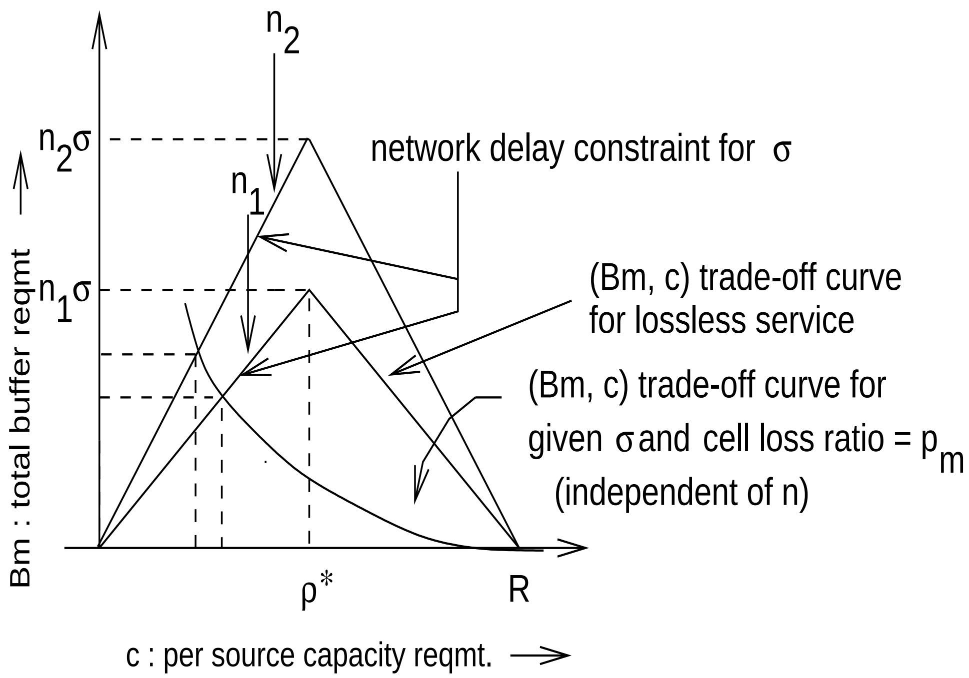

Figure 4.9: Network buffer (B_m) vs per source capacity (c) tradeoff curves at the multiplexer for a *given* σ and n_1 and n_2, with $n_2 > n_1$. Shown are the lossy tradeoff curve (which is independent of n) and, the delay constraint and lossless trade-off bound for n_1 and n_2.

4.5.2 Simulation Results and Discussions

The source model was the same as taken in Section 3.7.2 for determination of ρ^*. We take ρ^* value as 138, $J = 3$, $D = 4$ and $p_m = p_s = QVP = 10^{-5}$.

Here we present the simulation results and their possible interpretations.

Figure 4.10 shows the per source extremal capacity as a function of σ, i.e. the $c(\sigma)$ curve for $n = 50, 150$. The $c(\sigma)$ curve is found to be monotonically decreasing and convex.

Observe that, in Figure 4.10, as σ approaches 0, $c(\sigma)$ approaches $\rho^*(\simeq 138)$. This behaviour is explained in Theorem 4.3.4.

Also, as observed in Figure 4.10 with the increase in n for same σ, $c(\sigma)$ decreases. This behaviour can be explained follows. Observe that as n increases the slope of the delay constraint line, for same σ also increases as shown in Figure 4.9. Due to the monotonically decreasing nature of the *lossy* (B_m, c) tradeoff curve (this being independent of n) for a given σ and p_m, the point at which this lossy (B_m, c) tradeoff intersects the delay constraint has the property that $c_{n_1} > c_{n_2}$ for $n_1 < n_2$. Also by the approach followed in Proofs in Section 4.3.2,

$$\frac{n\sigma c}{\rho^*} = \frac{-\ln p_m}{\eta(\sigma, c)}$$

For fixed σ and substituting $k = \frac{-\ln p_m \rho^*}{\sigma}$, we get (taking n to be continuous)

$$c = \frac{k}{n\eta(c)} \Rightarrow \frac{dc}{dn} = k\left(-\frac{1}{n^2\eta(c)} - \frac{1}{n\eta^2(c)}\frac{\partial\eta}{\partial c}\frac{dc}{dn}\right)$$

$$\Rightarrow \frac{dc}{dn} = \frac{\frac{-k}{n^2\eta(c)}}{\left(1 + \frac{k}{n\eta^2(c)}\frac{\partial\eta}{\partial c}\right)} \tag{4.20}$$

Since $\frac{\partial\eta}{\partial c} > 0$ as proved in Theorem 4.3.2 and by the positivity of k and $\eta(c)$ we get $\frac{dc}{dn} < 0$ from Equation 4.20.

Figure 4.10 shows the per source extremal buffer, i.e. $\frac{B_m(\sigma)}{n}$ curve as a function of σ for $n = 50, 150$. The curve is found to be monotonically increasing. Also as n increases for a given σ, the total network buffer $B_m(\sigma, n)$ increases but the per source buffer $\frac{B_m(\sigma)}{n}$ decreases. Again going back to Figure 4.9, as n increases then, due to the increase in the slope of delay-constraint and the monotonically decreasing nature of the lossy (B_m, c) (this being independent of n) the extremal total network buffer $B_m(\sigma, n)$ increases. However, due to the convexity of the lossy (B_m, c) tradeoff, the per source extremal buffer $\frac{B_m(\sigma,n)}{n}$ decreases. Also, the per source extremal buffer $= \frac{\sigma c(\sigma,n)}{\rho^*}$. But, for fixed σ, $\frac{dc}{dn} < 0$, explaining the decrease in per source extremal buffer with n. In Figure 4.11 we show the extremal per source buffer-capacity tradeoff for the example source and for $n = 50$ and 150. The curve is found to be decreasing and convex. For this note that for a given $c(\sigma)$, the value of the per source extremal buffer decreases with increasing n.

***Cost Curves*: Per Source Linear Cost Function Vs σ**

We shall call the plots of $\gamma c(\sigma) + \beta\frac{B_m(\sigma)}{n}$, which can also be expressed as $\beta(\Delta c(\sigma) + \frac{B_m(\sigma)}{n})$ ($\Delta = \frac{\gamma}{\beta}$, as discussed in Section 4.2), vs σ as *Cost Curves*. Evidently the value of σ that minimizes $\beta(\Delta c(\sigma) + \frac{B_m(\sigma)}{n})$, i.e., σ^* is only dependent on Δ. *Cost Curves* for $n = 50, 150$

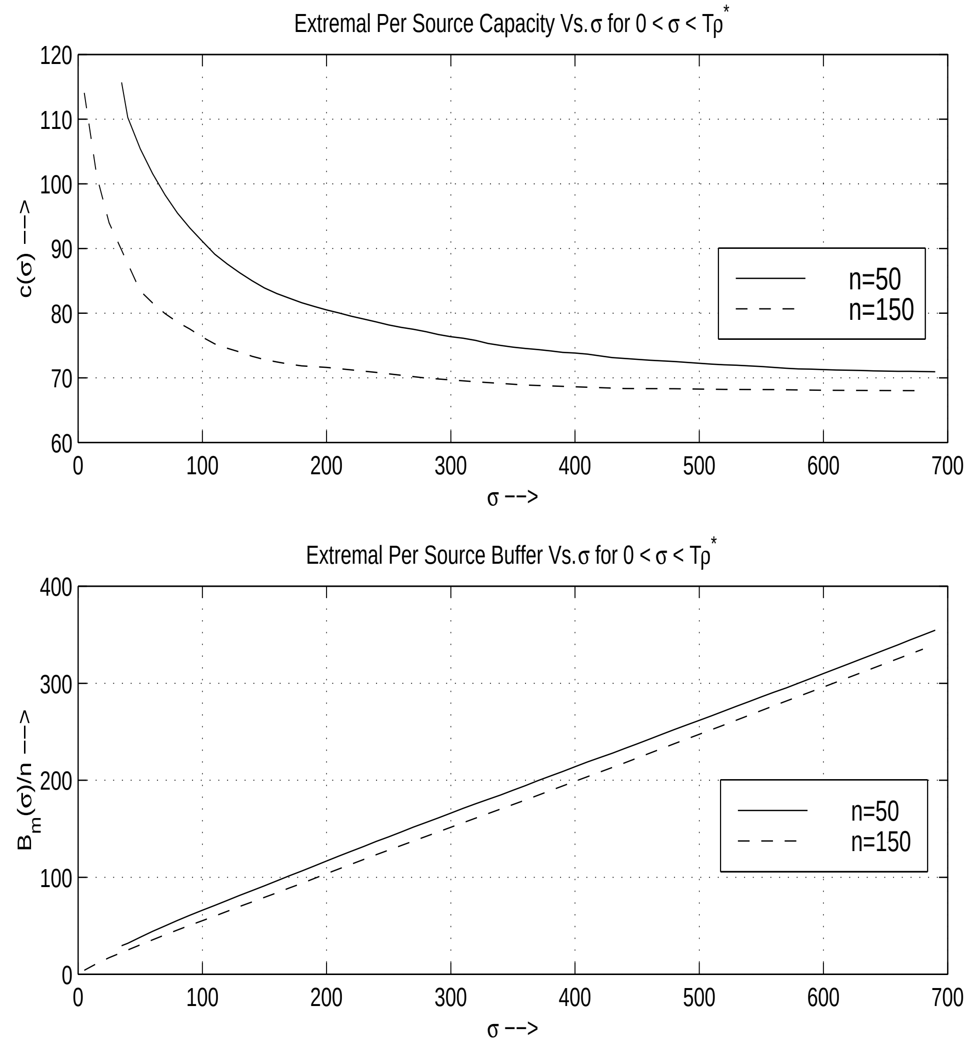

Figure 4.10: The graph of per source extremal capacity and per source extremal buffer required at the mux vs σ for lossy multiplexing for $n = 50$ and $n = 150$.

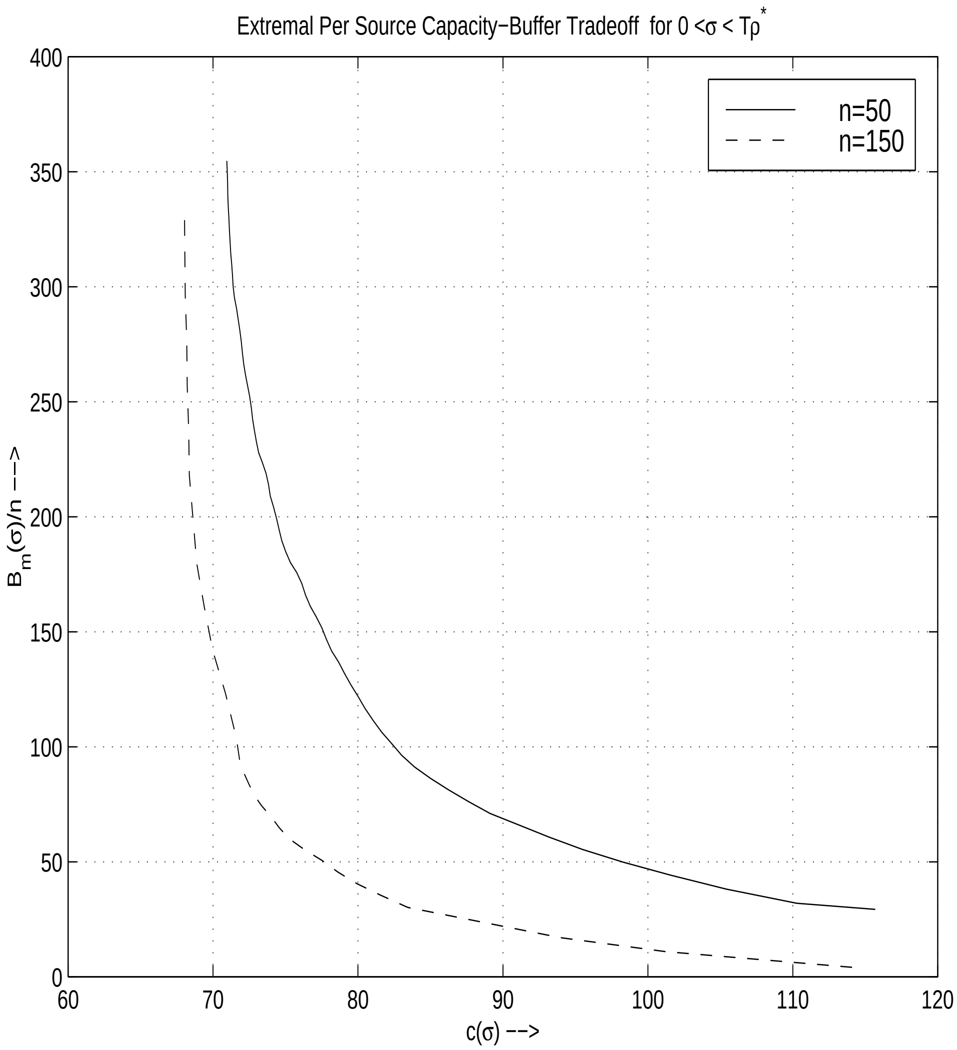

Figure 4.11: The per source extremal buffer-capacity tradeoff for lossy multiplexing for $n = 50$ and $n = 150$.

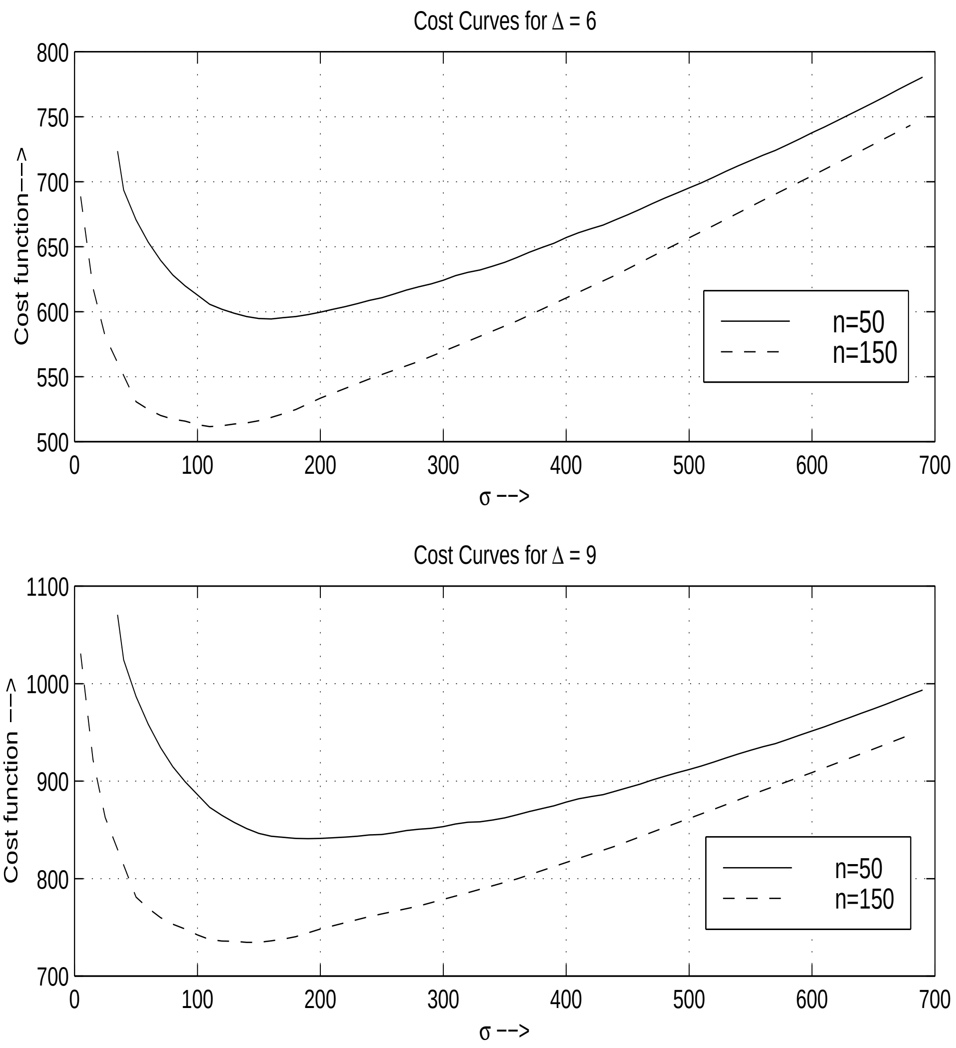

Figure 4.12: *Cost Curves* for $\Delta = 6$ and $\Delta = 9$ for $n = 50, 150$ showing the sensitivity of σ^* to Δ and n.

and two different values of Δ ($\Delta = 6, 9$) are shown in Figure 4.12; the source is the one we have used in the previous examples.

From the plots, we observe that as Δ increases for a given n, the minimum value of cost function and the corresponding value of σ^* increases. In our simulations, $\beta = 1$ always and so $\Delta = \gamma$. From the simulation results, the minimum values of cost function for $\Delta = 6$ and 9 are about 594 and 841 respectively for $n = 50$ and about 512 and 735 respectively for $n = 150$. Also for $\Delta = 6$ and 9, $\sigma^* = 160$ and 190 for $n = 50$ whereas $\sigma^* = 110$ and 140 for $n = 150$. The increase in the value of the minimum cost with the increase in Δ is because of the fact that at the minimum of the cost function (expressed as $\beta(\Delta c + b(c))$, with c being the extremal capacity and $b(c)$ the per source extremal buffer corresponding to this c [6], c is equal to c_Δ, where c_Δ solves the following equation ,

$$\left(\frac{db}{dc}\right)_{c_\Delta} = -\Delta \tag{4.21}$$

Thus, the minimum cost as a function of Δ is expressed as,

$$\text{MinimumCost}(\Delta) = \Delta c_\Delta + b(c_\Delta) \tag{4.22}$$

To study the sensitivity of the Minimum Cost to Δ, we differentiate both sides of Equation 4.22 with Δ. We have,

$$\frac{d\text{MinimumCost}(\Delta)}{d(\Delta)} = \Delta\frac{dc_\Delta}{d\Delta} + c_\Delta + \left(\frac{db}{dc}\right)_{c_\Delta}\frac{dc_\Delta}{d\Delta}$$

Substituting the value of $\left(\frac{db}{dc}\right)_{c_\Delta}$ from Equation 4.21 in above equation, we get,

$$\frac{d\text{MinimumCost}(\Delta)}{d(\Delta)} = \Delta\frac{dc_\Delta}{d\Delta} + c_\Delta + (-\Delta)\frac{dc_\Delta}{d\Delta}$$

$$\Rightarrow \frac{d\text{MinimumCost}(\Delta)}{d(\Delta)} = c_\Delta > 0$$

[6]The notations do not show c as $c(\sigma)$, for typographical simplicity.

This explains the increase in the *Minimum Cost* with increase in Δ.

Also note that, as Δ increases for a given n, the slope of the locus of the cost function on the buffer-capacity plane, i.e., $-\Delta$ becomes more negative. Thus the minimum is achieved at a smaller value of capacity and a larger value of buffer. But we know that as σ increases $c(\sigma)$ decreases and $B_m(\sigma)$ increases. This explains the increase in σ^* with increase in Δ.

From the plots we also observe that as n increases for a given Δ, the minimum cost and the corresponding value of σ^* decreases. From the simulation data we have, for $n = 50$ and 150, $\sigma^* = 160$ and 110 for $\Delta = 6$ and $\sigma^* = 190$ and 140 for $\Delta = 9$. We know that as n increases for a given σ, both c and b decreases. Thus, the *Cost Curve* for $n = 150$ is always below that for $n = 50$, explaining the decrease in *Minimum Cost* with increase in n.

We provide a comparison of the values of the shaping parameters σ^*, ρ^*, B_s^*, and the per source capacity c and buffer b, in Table 4.1, for the three approaches:

1. Peak rate allocation no shaping at the source
2. lossless multiplexing in the network
3. lossy multiplexing in the network

for our source model. The table shows the reduction in network resources possible by working with optimal shaper parameters.

	Approach	σ^*	ρ^*	B_s^*	c	$b_m(c)$
$\Delta = 6$	Peak rate	-	-	-	170	0
	Lossless	0	138	690	138	0
	Lossy ($n = 50$)	160	138	530	83.02	96.26
	Lossy ($n = 150$)	110	138	580	75.27	60.00
$\Delta = 9$	Approach	σ^*	ρ^*	B_s^*	c	$b_m(c)$
	Peak rate	-	-	-	170	0
	Lossless	0	138	690	138	0
	Lossy ($n = 50$)	190	138	500	81.03	111.58
	Lossy ($n = 150$)	140	138	550	73.35	74.41

Table 4.1: Optimal shaper parameters and and per source multiplexer capacity and buffer for various approaches. $\Delta = 6$ and 9.

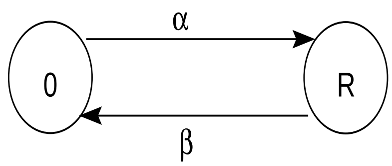

Figure 4.13: The two-state characterisation of the rate process at the input of the shaper

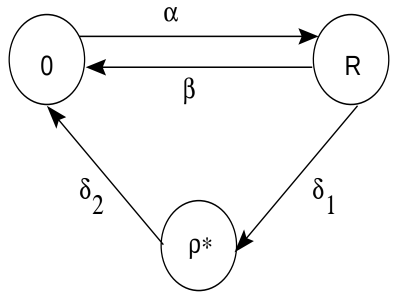

Figure 4.14: The approximate Markovian characterisation of the rate process at the output of the shaper

4.6 An Approximate Analytical Calculation of σ^*

In the previous section we basically used simulations to obtain $c(\sigma)$. In this section we study an approximate analysis for the calculation of $c(\sigma)$.

4.6.1 MMFP Characterisation of LB output

In this section we shall use the three state Markov modulated approximation for the output process of the leaky bucket the input to which is a two state Markov modulated fluid process (MMFP), to determine σ^*. The output characterisation is shown in [18]. Figure 4.13 is the two state MMFP source that is input to the LB, and Figure 4.14 is the approximate three state MMFP model for the output process of the LB. The input process $S(t)$ to the LB is characterised by the following matrices,

$$M_s = \begin{bmatrix} -\alpha & \alpha \\ \beta & -\beta \end{bmatrix}$$

and

$$\Lambda_s = \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}$$

where M_s is the transition rate matrix and Λ_s is the rate matrix of $S(t)$. The two diagonal elements of Λ_s are for the two possible rates of $S(t)$.

The output process $W(t)$ from the LB is approximately characterised by the following matrices,

$$M_w{}^{7} = \begin{bmatrix} -\alpha & \alpha & 0 \\ \beta & -(\beta+\delta_1) & \delta_1 \\ \delta_2 & 0 & -\delta_2 \end{bmatrix}$$

$$\Lambda_w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & \rho^* \end{bmatrix}$$

where,

$$\delta_1 = \frac{\exp(\Upsilon\sigma)\Upsilon(\rho^* - R)}{(1 - \exp(\Upsilon\sigma))}$$

$$\delta_2 = \frac{\rho^*\beta\Upsilon(\rho^* - R)}{\beta R - (\alpha+\beta)(R-\rho^*)\exp(\Upsilon B_s)}$$

$$\Upsilon = \frac{\rho^*(\alpha+\beta) - \alpha R}{\rho^*(\rho^* - R)}$$

$$B_s = T\rho^* - \sigma$$

M_w is the transition rate matrix and Λ_w is the rate matrix of $W(t)$. The three diagonal elements of Λ_w are for the three possible rates of $W(t)$.

[7] for details of the derivation of Refer [41]

4.6.2 Determination of $c(\sigma)$

We shall now employ the approximate characterisation of the LB output to approximately determine $c(\sigma)$. By the maximum eigen-value approach given in [19], we have the eigen-value problem (writing $z = -\eta$),

$$z\phi(\Lambda_w - cI) = \phi M_w$$

where ϕ is the eigen vector, and I is the identity matrix. Writing $g(z) = c$, the corresponding inverse eigen value problem is,

$$g(z)\phi = \phi A(z)$$

where,

$$A(z) = \Lambda_w - \frac{1}{z} M_w$$

That is, $g(z)$ is an eigenvalue value of the matrix $A(z)$ in which z is a parameter. The characteristic equation for the above (inverse) eigen-value problem is (writing $g(z)$ as g)

$$g^3 z^3 - g^2(z^3 A_1 + z^2 A_2) + g(z^3 A_3 + z^2 A_4 + zA_5) - z^2 A_6 - zA_7 = 0 \qquad (4.23)$$

where,

$$A_1 = (\rho^* + R)$$

$$A_2 = \delta_1 + \delta_2 + \alpha + \beta$$

$$A_3 = \rho^* R$$

$$A_4 = R(\alpha + \delta_2) + \rho^*(\delta_1 + \alpha + \beta)$$

$$A_5 = \delta_2(\alpha + \beta + \delta_1) + \alpha\delta_1$$

$$A_6 = R\alpha\rho^*$$

$$A_7 = \alpha(R\delta_2 + \delta_1\rho^*)$$

By a result in [19] we know that the maximum real eigen value of the essentially non-negative matrix $A(z)$ is the effective bandwidth required for a given z. Let this eigen-value be denoted by g_1. The function $g_1(z)$ has the following property

- $g_1(z) \downarrow z$ ($g_1(\eta) \uparrow \eta$).
- $\bar{w} \leq g_1(z) \leq R$, where $\bar{w}$ is the average rate of the shaper output process.
- Substituting $z = \frac{\ln p_m}{B_m}$ and solving for the maximum eigen value gives a unique g_1, which we take as the capacity required at a multiplexer with buffer B_m for achieving a CLR of p_m.

Observe that in our problem B_m and g_1 are related by the delay constraint, i.e. $B_m = \frac{n\sigma g_1}{\rho^*}$. Also,

$$z = \frac{\ln p_m}{B_m} \Rightarrow z = \frac{\ln p_m}{\frac{n\sigma g_1}{\rho^*}}$$

With $k = \frac{\ln p_m}{n\sigma}$, $z = \frac{k}{g_1}$. So, our required extremal capacity (for a given σ), is the maximal real root of Equation 4.23 with $z = \frac{k}{g_1}$. Substituting $z = \frac{k}{g}$ in Equation 4.23, we get,

$$g^2(k^3 - k^2A_2 + kA_5) + g(-k^3A_1 + k^2A_4 - kA_7) + (k^3A_3 - k^2A_6) = 0 \tag{4.24}$$

Thus the cubic equation in g reduces to quadratic. We now claim that the maximal real root of Equation 4.24, calling it g_1^* is the extremal capacity such that working with a capacity g_1^* and buffer $\frac{n\sigma g_1^*}{\rho^*}$ we are satisfying the CLR as well as delay requirements. To prove this claim consider the Figure 4.15. We need to show that the larger root of the quadratic is the intersection of $f(\eta)$ ($\eta = -z$) with $g_1(\eta)$, i.e., we need to show that $g_1^* > g_2^*$.

Suppose $g_2^* > g_1^* \Longrightarrow \eta_2 < \eta_1$ since $f(\eta) \downarrow \eta$. Now,

$$g_2(\eta_2) = f(\eta_2) = g_2^* > g_1^* = f(\eta_1) = g_1(\eta_1) > g_1(\eta_2)$$

as $g_1 \uparrow \eta$ (by the property of g_1). This *contradicts* the fact that $g_2(\eta) \leq g_1(\eta)\ \forall\ \eta$. Thus the maximum root of the quadratic in Equation 4.24 is the extremal capacity that will meet our dual requirements of loss and delay. The roots of the quadratic are

$$g_{1,2}^* = \frac{Y \pm \sqrt{Y^2 - 4XZ}}{2X}$$

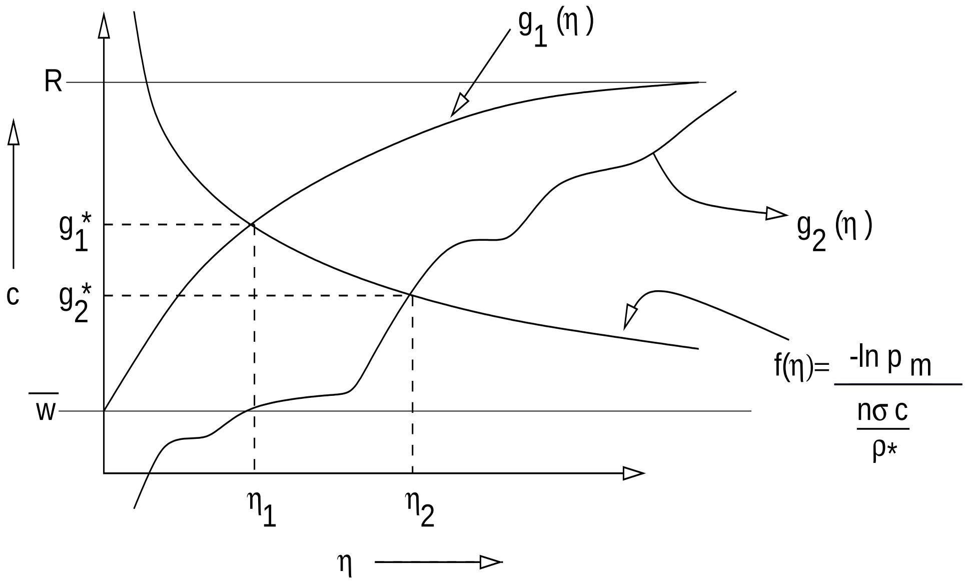

Figure 4.15: The Figure showing the characterisation of the maximum real root of Equation 4.24.

where,

$$X = k^3 - k^2 A_2 + k A_5$$

$$Y = -k^3 A_1 + k^2 A_4 - k A_7$$

$$Z = k^3 A_3 - k^2 A_6$$

Thus we shall get the extremal capacity $\forall\ \sigma \in [0, T\rho^*]$ by the maximum real root of Equation 4.24 for each σ. Thus we have the objective function value for each $\sigma \in [0, T\rho^*]$. We can then find σ^* analytically as that value of $\sigma \in [0, T\rho^*]$ where the objective function attains the minima.

4.6.3 Comparison of Analysis with Simulation

In this section we shall compare the values of the various functions, obtained by the simulations in Section 4.5.2 and by the approximate analysis in Section 4.6.2. The comparisons are shown in Figures 4.16, 4.17 and 4.18. From the figures we observe that the analytical curves matches the simulation results very well. Thus for VBR sources which are efficiently modeled as a two state on-off MMPP process (like telephone calls), knowing the peak rate

R, one can directly estimate the σ^* using the approximate analysis to sufficient degree of accuracy obviating the need for a measurement based estimation of σ^*. However, we have only shown the comparisons for a two state MMFP VBR source. The study of how good the approximation is for an n state MMFP VBR sources has to be done.

4.7 Conclusion

In this chapter we first recalled the characterisation of σ^* for Minimum Cost Lossy Multiplexing developed in [41]. We then provided the analytical proofs of various functions encountered in the formulation. We then developed an FSSAA based algorithm for off-line determination of σ^*. We have used the algorithm in a simulation for verifying properties of various functions (viz. $c(\sigma), B_m(\sigma), B_m(c(\sigma))$ and *Cost Curves*). This provides a background for our next work of developing an *On-Line algorithm* for σ^* determination in Chapter 5. We then used a three state MMFP characterisation of the LB output and *Perron Frobenius* eigen value theory to obtain an approximation to σ^* for two state Markov Modulated VBR sources. The analytical curves obtained from this approximate characterisation are then compared with simulations. The close agreement of the analysis (approximate) and simulations provides a basis for a close initial estimate of σ^* for a measurement based iterative algorithm.

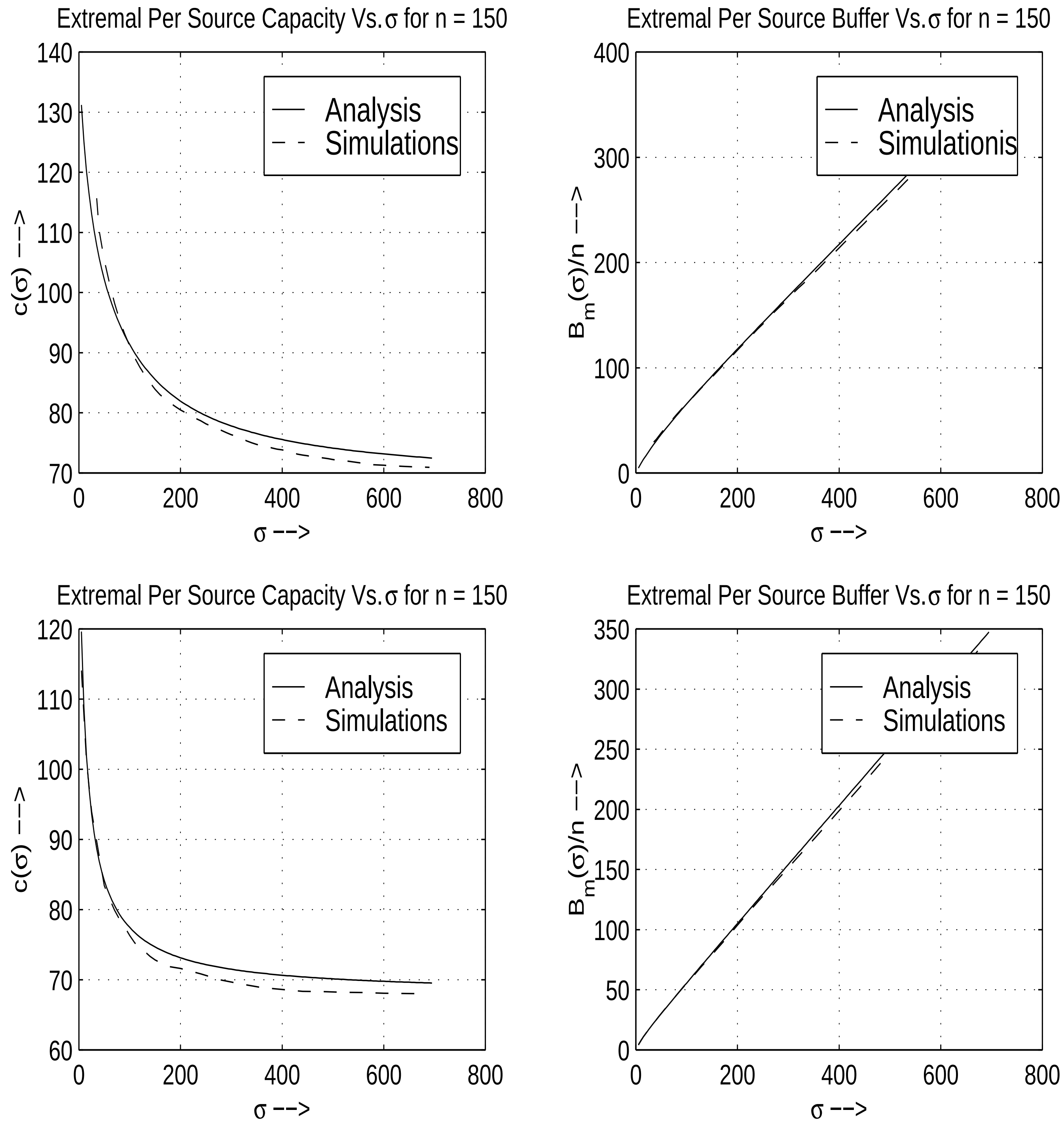

Figure 4.16: The comparison of approximate analysis with simulation of the per source extremal capacity Vs σ and the per source extremal buffer Vs σ curves for $n = 50$ and $n = 150$.

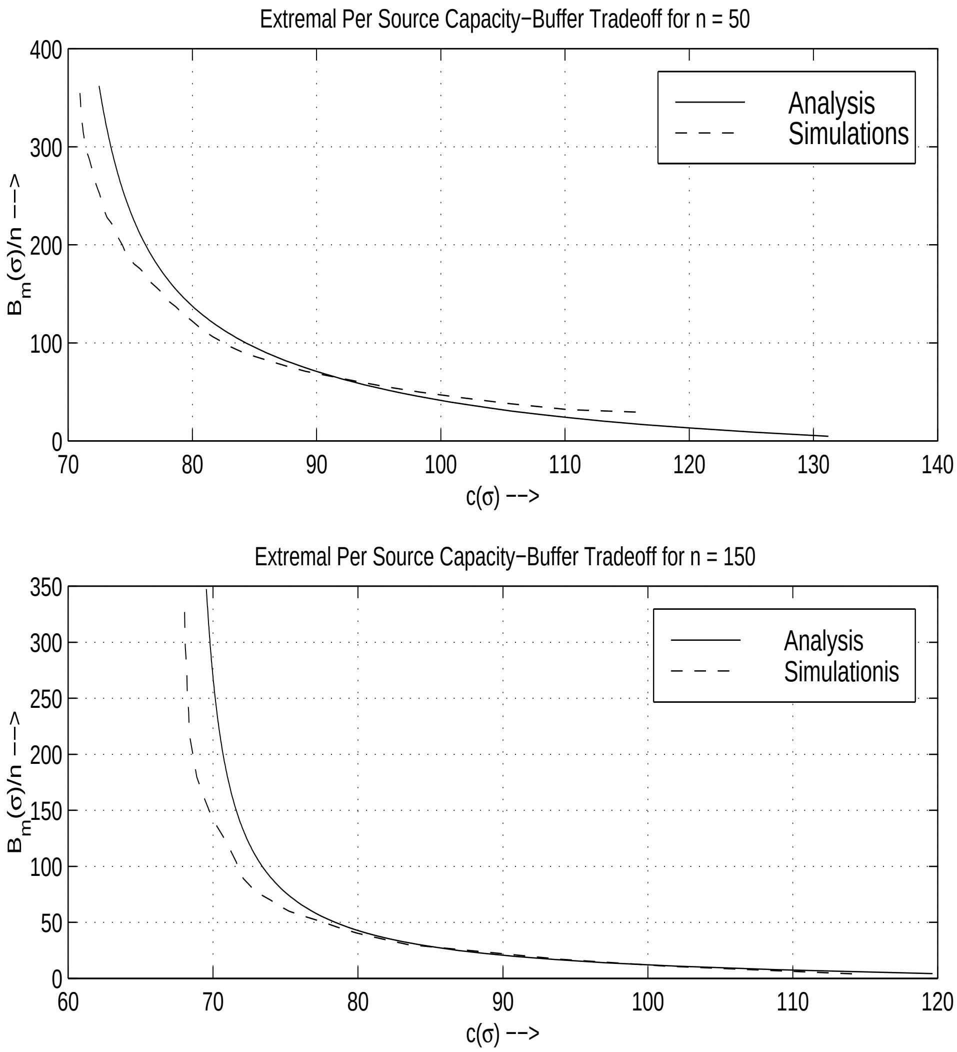

Figure 4.17: The comparison of approximate analysis with simulation of the per source extremal buffer-capacity tradeoff for $n = 50$ and $n = 150$.

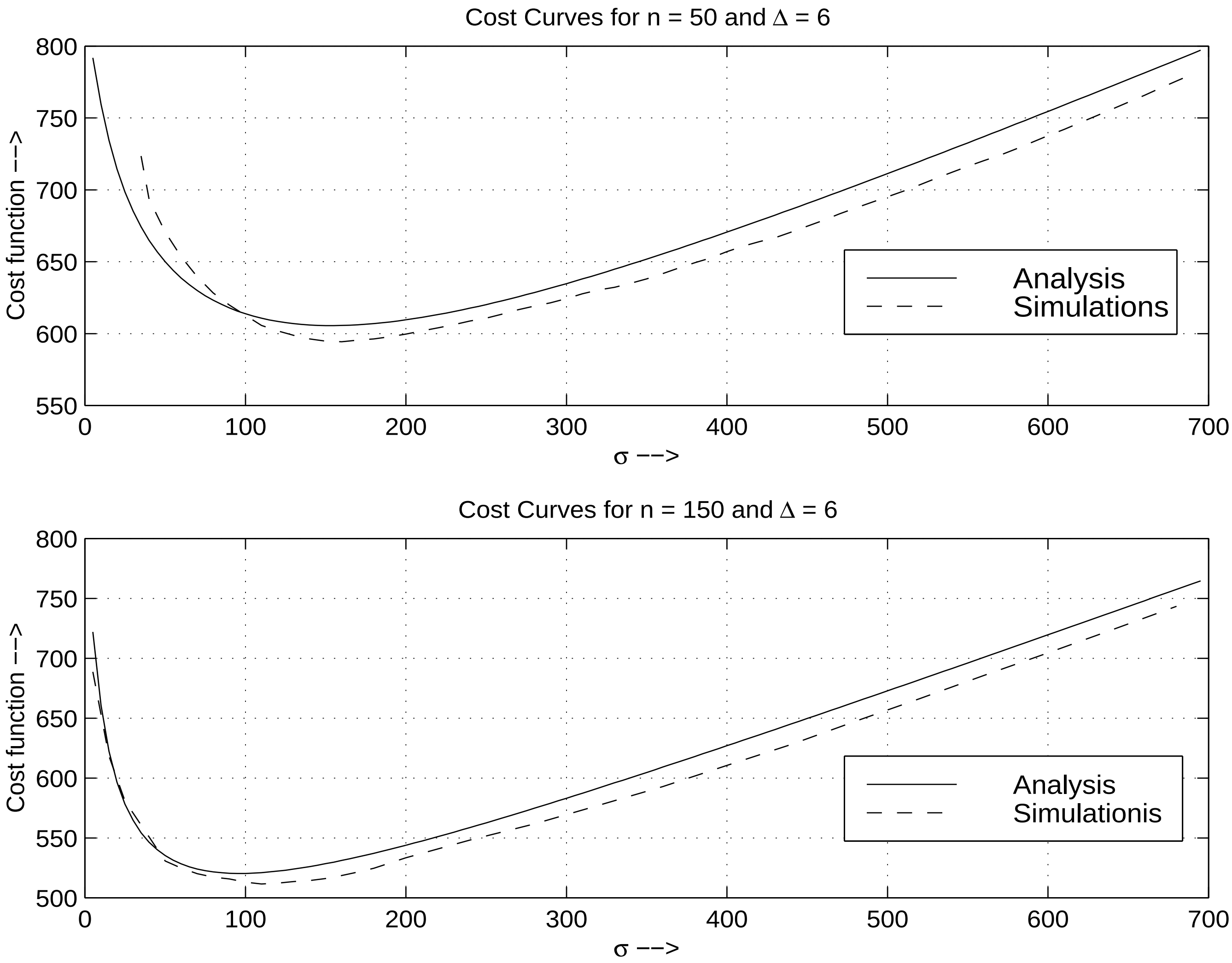

Figure 4.18: The comparison of approximate analysis with simulation for $\Delta = 6$ of the *Cost Curves* for $n = 50$ and $n = 150$.

Chapter 5

Measurement Based Estimation of Optimal Bucket Size

5.1 Introduction

Recall that we have two options, either to work with *lossless* or with *lossy* multiplexing at the network node. We have already determined in Chapter 3 that for the lossless case the optimal parameter values are ρ^* ($\bar{r} < \rho^* < R$) and $\sigma^* = 0$, which corresponds to a per source network-node capacity requirement of $c = \rho^*$, and per source network buffer requirement of $b_m = 0$. For the lossy case (where the CLR at the network node is of the order of the QoS Violation Probability(QVP) [1]) we have analytically proved and subsequently observed by simulations in Chapter 4 that, by using $\sigma > 0$, we may achieve a reduced value of network cost needing $c < \rho^*$ and $b_m > 0$. Thus we have two options:

1. *Solution 1:* To keep $\sigma^* = 0$[2] and $\rho = \rho^*$ (corresponding to minimal cost for lossless multiplexing).

[1] For the definition of QVP refer to Section 3.2

[2] Note that this value is for a fluid source. In practice, when packets are transported, σ has to be at least one packet worth.

2. *Solution 2:* To keep $\rho = \rho^*$, and determine a $\sigma^* > 0$ by running an *on-line algorithm* at the network node for lossy multiplexing. However, for an algorithm to be useful it should have the feature of fast convergence to σ^*.

In the previous chapter we used an example to study the formulation for obtaining an optimal value of $\sigma = \sigma^*$ that minimises the multiplexer buffer-bandwidth cost (a linear tradeoff) for lossy multiplexing. We used simulations to explore the nature of $c(\sigma)$, $B_m(\sigma)$ and the extremal buffer-capacity tradeoff curves. There we have used an *off-line* approach to obtain σ^*. We first obtained the value of $c(\sigma)$ for all σ in $(0, T\rho^*)$. Then we proceeded to obtain the argument of the minimum of the cost function, i.e., σ^*, using the values of $c(\sigma)$. In this chapter, we consider the lossy multiplexing option and propose a general, amalgamated and practical algorithm for *on-line estimation* of the optimal sustainable parameter σ^*. Our algorithm is

- general as it is devoid of any source modeling or apriori knowledge of traffic characteristics.
- amalgamated as it includes the features of several simulation optimization techniques like *Ordinal Optimization, Discrete Stochastic Optimization* and *Continuous Stochastic Optimization.*
- practical because "it fulfills the engineering requirements for a useful and satisficing [3] solution without any overzealous claim on the degree of optimality".

The chapter is organized as follows. Section 5.2 recalls our optimization problem and introduces our solution approach. Section 5.3 introduces the general concepts of ordinal optimization as applied to our problem. Section 5.4 provides an introduction to discrete stochastic optimization, particularly to Andradottir's discrete stochastic optimization algorithm, a variant of which we shall develop and use in our algorithm. Section 5.5 describes

[3] Satisficing is the process of developing a satisfactory but not necessarily optimum solution to a problem(cf. pp. 4149, Volume 6, *Systems and Control Encyclopedia: Theory, Technology, Applications, Ed. Madan G. Singh*)

our algorithm with complete details. Section 5.6 provides the simulation set-up and ithe simulation results. Finally in the appendix we developed a theorem for tandem arrangement of leaky buckets which is helpful in the implementation of our proposed algorithm.

5.2 The Optimisation Problem and the Overall Solution Approach

Our optimization problem was

$$\min_{0 \leq \sigma \leq T\rho^*} f(\sigma) = \gamma c(\sigma) + \frac{\beta \sigma c(\sigma)}{\rho^*} \tag{5.1}$$

where for each σ, $c(\sigma)$ solves,

$$\lambda_{(\sigma,\rho^*)}(\frac{n\sigma c}{\rho^*}, c) = -\ln p_m \tag{5.2}$$

The following are the chief features of this optimisation problem:

1. The search space (which for our optimisation problem is the set $\{\sigma : 0 \leq \sigma \leq T\rho^*\}$) has discrete parameter values; as σ denotes the token bucket depth in terms of cells.

2. The search space has cardinality $\lfloor T\rho^* \rfloor + 1$ The size of the search space can be excessively large for any efficient real-time estimation of σ^*. For our example source, with $T = 5$ and $\rho^* = 138$, the size is 691 ($\lfloor T\rho^* \rfloor = 690 + 1$ (for $\sigma = 0$)).

3. For each feasible value of σ, evaluation of the objective function requires the solution of an implicit equation (Equation 5.2).

In the *off-line technique* we obtained $c(\sigma)$ for each σ, $0 \leq \sigma \leq \lfloor T\rho^* \rfloor$ from Equation 5.2 using a modification of a continuous stochastic approximation called FSSAA. The optimisation problem can be solved by finding the value of $c(\sigma)$ for each σ (which in turn will give us the corresponding value of objective function) and then selecting the argument of the minimum value of objective function as our σ^*. But this would not be feasible because of large magnitude of the search space, for real time solutions where decisions have to be made on the fly. Hence, we need to reduce the search space.

We propose to use the *Ordinal Optimisation* technique for search space reduction. After reducing our search space, we use a discrete stochastic optimisation technique to optimise Equation 5.1 over this reduced search space. The use of discrete stochastic optimisation gives an on-line feature to our algorithm where each "good" iterate value is fed back to the source and the source changes its token bucket depth dynamically. At the mth iterate of the discrete stochastic optimisation algorithm (σ_m) we need an estimate [4] of the objective function at σ_m. We use the continuous stochastic approximation, FSSA to obtain an estimate of $c(\sigma_m)$ which in turn will give us an estimate of $f(\sigma_m)$ (See Equation 5.1) at the mth iterate. Thus, the FSSA is used as an "inner loop" in the discrete stochastic optimisation.

Thus, we divide our proposed algorithm into two phases:

1. *Phase I:* Search space reduction and obtaining a "good" initial estimate
2. *Phase II:* Fine tuning of the estimated optimal σ towards σ^*

Thus, in brief, we shall reduce the search space of σ in *Phase I* so that for *Phase II* we have a reduced search space and an initial approximation that is close to σ^*. We follow the approach of ordinal optimization in *Phase I.* Having obtained a reduced search space at the end of the first phase and a good initial approximation σ_0 of σ^*, we proceed to *Phase II* to fine tune our estimate of optimal σ towards σ^*. This phase involves the techniques of discrete and continuous stochastic optimization.

To further reduce the search space and in the spirit of providing a *"satisficing"* solution (defined earlier) we also propose to reduce the granularity of our search space. Define d as the *granularity parameter* of our search space, i.e. d is the minimum *distance* (since our σ values are integers and hence distance is the eucledian metric in one dimension over set $\mathcal{I}$) between successive values of σ in the search space. Thus, the larger the value of d, the coarser is the

[4] here estimate of a function means the noise corrupted value of the function

granularity of our search space. Observe that for $d = 1$ our search space, Θ, has cardinality of $\lfloor T\rho^* \rfloor + 1$, i.e. $\mid \Theta \mid = \lfloor T\rho^* \rfloor + 1$ and for $d = 10$, $\mid \Theta \mid = \lfloor \frac{T\rho^*}{d} \rfloor + 1$ (for $\sigma = \lfloor T\rho^* \rfloor$) $+ 1$ (for $\sigma = 0$). An important remark to be made here is that *the proposed solution of reducing the granularity for reducing the search space which ultimately will lead to faster σ^* estimation is acceptable only when the the structure of the objective function ensures that even with the reduced granularity the values of our objective function at successive points, with inter-distance d does not have sharp differences.* Again there is a tradeoff in going for increased precision in estimation of σ^*(corresponding to working with $d = 1$ in the extreme case) and the relatively fast convergence to some sub/near-optimal solution (corresponding to working with $d > 1$).

With these ingredients in mind we shall now concentrate on the theory behind our approach in the next two sections.

5.3 Ordinal Optimization

The traditional simulation optimization techniques serve the objective of finding an *optimizer* or at least some *near-optimal* solution. Refer to [21] and [5] for an excellent survey and critical evaluation of various prevalent Simulation Optimization Techniques. The precise determination of the optimal solution is time-consuming and computationally intensive. As an alternative, Ordinal Optimization was first proposed by Ho et. al in [57]. The idea behind ordinal optimization is basically that *relatively less computer time is required for ranking alternatives* [5]*, according to the values the objective function takes at these alternatives, than for estimating the objective function values precisely at the different alternatives.* They have claimed by observations that the relative ranking among the design samples is immune to even large estimation noise.

Ordinal optimization departs from the traditional approaches in two ways: ([57])

[5] *For example if our optimisation problem aim to minimise (maximise) the objective function then the alternative at which the objective function achieves the minima (maxima) will be ranked first and the other alternatives are ranked in the order of the increasing (decreasing) values of objective function at those alternatives*

1. The approach aims to find good, better or best designs by attempting to find designs that belong to the top n percentile of all designs.

2. The interest is in determining whether a given design is better than another rather than accurately estimating quantities such as gradients and performance values, that is, the approach primarily deals with **ordinal** rather than **cardinal optimization**.

The next subsection introduces the terminology related to ordinal optimization.

5.3.1 Terminology related to Ordinal Optimization [[57]]

1. **Good enough subset:** is defined as the subset of the search space in which the members satisfy some design criteria set forth by the designer.

2. **Selected subset:** is defined as the subset of the search space in which the members are picked out by the designer using certain evaluation scheme. **Every optimization problem can, in principle, be conceived as the goal of "matching" a selected subset with the good enough subset.** As a matter of fact, most traditional ways of describing an optimization problem ask for the good enough subset and the selected subset both to be singleton, i.e., to pick out one design (selected singleton) with target at the true optimum (good enough singleton) in the design space. Notice that should the designer insist on having both the good enough subset and the selected subset as *singletons*, there is a high probability of mismatches between the two sets.

   In ordinal optimization, proposed by Ho et al, the basic idea is the **softening of the membership criteria for the good enough subset and the selected subset while maintaining reasonable "matching" outcomes with efficiency and confidence**. This can be achieved by

   - good enough subset can have design samples as the top n percentile of the design space. This is called **goal softening**.

   - the selected subset can have more than one design sample from the set of representative samples. The increase in the member of the selected subset secures with higher confidence certain degree of "matching", or **alignments**, between the selected subset and good enough subset. The degree of "matching" is called the **alignment level**, and the confidence of achieving a certain alignment level is called the **alignment probability**.

5.3.2 Ordinal optimization as applied to our problem

Our representative set consists of $(\lfloor T\rho^* \rfloor + 1)$ design samples (i.e., σ) ranging from 0 to $\lfloor T\rho^* \rfloor$. We will denote our representative set by Θ, and with $\mid \Theta \mid = N = \lfloor T\rho^* \rfloor + 1$. The designs are $\sigma_i \in \Theta, i = 1, \ldots, N$, with $\sigma_i = i - 1$ and the corresponding costs are $f_i \equiv f(\sigma_i)$. Observe that, in the discussion in this section we are working with $d = 1$. For $d > 1$ an analogous discussion can be done with

$$\mid \Theta \mid = N = \lfloor \frac{T\rho^*}{d} \rfloor + 1(\text{ for } \sigma = \lfloor T\rho^* \rfloor) + 1(\text{ for } \sigma = 0)$$

$$\sigma_i = \begin{cases} 0 & \text{for } i = 1 \\ i * d & \text{for } 0 < i < N \\ \lfloor T\rho^* \rfloor & \text{for } i = N \end{cases}$$

Selection of good enough subset

A good enough subset $G(\Theta)$ as defined before contains the top n percentile of the design samples. Thus, $|G(\Theta)| = \lceil N * n\% \rceil = g$. The **ordered performances** (note that we are considering a minimising problem) $C_{[i]}, i = 1, \ldots, N$, are obtained by re-ordering the f_i's such that

$$f_{[1]} \leq f_{[2]} \leq \ldots \leq f_{[N]},$$

and the corresponding ordered design samples are

$$\sigma_{[1]}, \sigma_{[2]}, \ldots, \sigma_{[N]},$$

where $f_{[i]} = f(\sigma_{[i]})$. Observe that the group $\{\sigma_{[1]}, \sigma_{[2]}, \ldots, \sigma_{[N]}\}$ is a *permutation* of the designs $\sigma_1, \sigma_2, \ldots, \sigma_N$ and $\sigma_{[1]}$ is the desired optimum. Thus,

$$G(\Theta) \equiv \{\sigma_{[1]}, \sigma_{[2]}, \ldots, \sigma_{[g]}\}$$

Selection of the selected subset

In the selection of a selected subset we aim to pick out a subset $S(\Theta)$, based on certain rule, from Θ so as to achieve a reasonable degree of "matching" with $G(\Theta)$. In other words, we are interested in the **alignment probability** of $\{G \bigcap S\}$ defined by

$$P_A = Prob\{|G \bigcap S| \geq k\}$$

where k is the alignment level. It is obvious that $1 \leq k \leq min(|G|, |S|)$, and we let $P_A = 1$ when $k = 0$. We shall now consider a subset selection rule called *Blind Picking (BP)* [6]. for selecting the elements of S.

Blind Picking Rule: Blind picking involves the selection of S from the representative set Θ:

- randomly
- without replacement
- without comparison, thus it doesn't require any *performance evaluation.*

P_A for the blind picking rule is

$$P_A(k, s, g|BP) = Prob\{|G \bigcap S| \geq k\} = \sum_{i=k}^{min(g,s)} \frac{\begin{bmatrix} g \\ i \end{bmatrix} \begin{bmatrix} N-g \\ s-i \end{bmatrix}}{\begin{bmatrix} N \\ s \end{bmatrix}} \tag{5.3}$$

[6]Refer to [57]

The above is the hypergeometric distribution. For a given P_A and k, Equation 5.3 gives

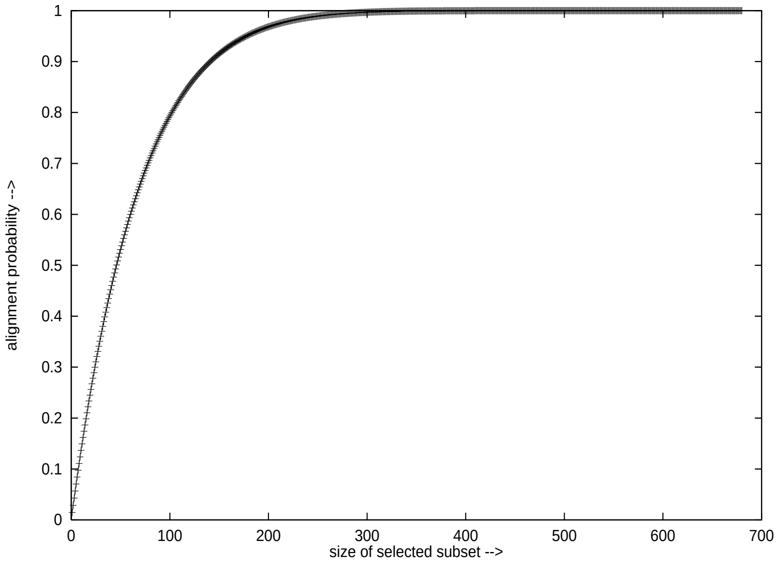

Figure 5.1: The curve showing the subset sizes required (s) for different alignment probabilities (P_A) by BP rule with $k = 1, g = 10, d = 1, \mid \Theta \mid = 691$

us the subset size s. For example, Figure 5.1 and Figure 5.2 shows the subset sizes (s) versus the alignment probability(P_A) curve for $d = 1, g = 10, \mid \Theta \mid = 691, k = 1$ and for $d = 10, g = 10, \mid \Theta \mid = 70, k = 1$ respectively. From these Figures we can obtain the P_A achievable when working with a given s. Obviously as s increases, P_A increases. However, we observe that even when working with $s = 141$ for $\mid \Theta \mid = 691$ case and with $s = 14$ for $\mid \Theta \mid = 70$ case we can achieve a $P_A \geq 0.90$. This means that we have only to optimize over a search space of size s and still we can say that the σ value from this set at which our cost function attains a minimum is, with probability greater than 0.90, one among the top $g(= 10)$ σ.

5.4 Discrete Stochastic Optimization

5.4.1 Motivation for using discrete stochastic optimization

Observe that the value of the objective function for different σ are obtained through measurements and hence are associated with measurement noise. This brings us in the framework of

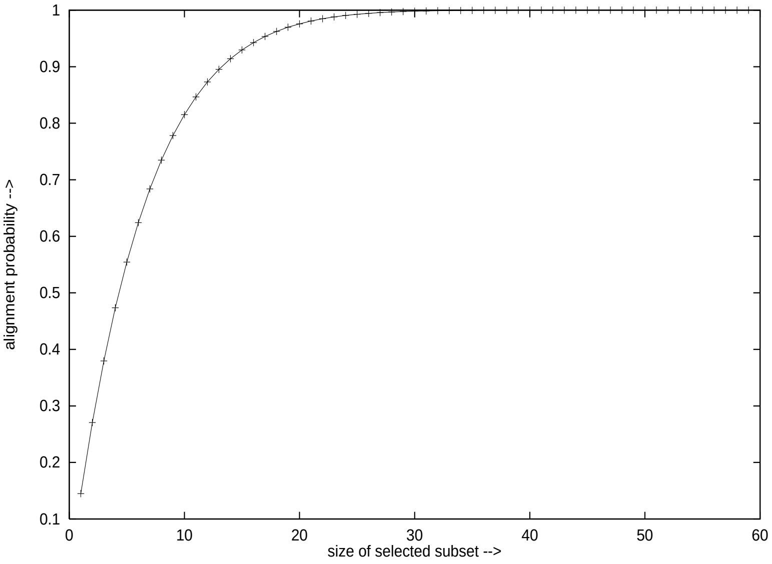

Figure 5.2: The curve showing the subset sizes required (s) for different alignment probabilities(P_A) by BP rule with $k = 1, g = 10, d = 10, \mid \Theta \mid = 70$

stochastic approximation. Also since the parameter values are discrete we can't work with the pervasive continuous stochastic approximation techniques and we need to work with a discrete stochastic optimization algorithm.

Here we digress to mention that we previously tried to apply the continuous stochastic approximation techniques to solve our discrete stochastic optimization problem. We basically worked with continuous σ and the set-up was analogous to the discrete set-up. However the method was not efficient as it suffered from the problem of extremely slow convergence which made it infeasible for real-time parameter estimation where decisions have to be made on the fly.

5.4.2 A Discrete Stochastic Optimization Algorithm

The discrete stochastic optimization algorithms are concerned with solving the optimization problems when the feasible region Θ is discrete. In the research community of stochastic optimization, continuous stochastic optimization has received much attention but the discrete parameter case has received very little attention. Yet, many important simulation optimization problems dealing with control and optimization of stochastic networks in general and communications in particular such as the problem of applying simulations to determine the optimal buffer allocation or a problem like ours require an optimization over discrete decision parameters. Recently some algorithms have been proposed to solve the problem of discrete stochastic optimization like [4], [23], [56] etc. Also see [5] for an excellent review and bibliography-list.

We now proceed to describe a discrete stochastic optimization algorithm proposed in [4], a variant of which we will develop and use in *Phase II* of our algorithm. Let F_σ denote a random variable whose distribution is parameterised by a parameter $\sigma \in \Theta$. For each σ we can observe a sample from the distribution of F_σ. In this framework we want to solve the optimization problem,

$$\sigma^* = arg \min_{\sigma \epsilon \Theta} EF_\sigma \tag{5.4}$$

where E is the expectation operator. Since the objective function is defined on a discrete set, an estimation of a "good" direction to move cannot be based on gradient information (as in the *Keifer-Wolfowitz algorithm*) as the objective function is discontinuous. Observe that our feasible region Θ is $\{\sigma_1, \sigma_2, \ldots, \sigma_{N-1}, \sigma_N\}$. Thus we can associate each $\sigma \in \Theta$ with a unique positive integer $i \in \mathcal{N}$, where $\mathcal{N}$ is the set of positive integers restricted to Θ i.e., $\mathcal{N} = \{1, 2, \ldots, \mid \Theta \mid\}$. For the ease of explaining the algorithm we shall speak in terms of i [7]. At each feasible point we have three points to go in the next move, up, down or to stay at the current point (i.e., at each i we can either go to $i+1$ or to $i-1$ or stay at i)

For this framework, a discrete stochastic optimization algorithm was proposed in [4] by Andradottir under the following assumption

Assumption 5.4.1 *For each $i \in \mathcal{N}$ there exists random variables F_i and F_i^+ such that $P(F_i^+ > F_i) > 0$, $P(F_i^+ < F_i) > 0$ and,*

$$f(i+1) > f(i) \Longrightarrow P(F_i^+ > F_i) > P(F_i^+ < F_i),$$

$$f(i+1) < f(i) \Longrightarrow P(F_i^+ > F_i) < P(F_i^+ < F_i),$$

[7] for example when we say that our iterate is i it actually means that the iterate is σ_i and similarly in all other contexts

$$f(i+1) = f(i+1) \Longrightarrow P(F_i^+ > F_i) = P(F_i^+ < F_i).$$

Thus, if in a given iteration, we have decided to compare the two points i and $i+1$, then we will obtain observations of the random variables F_i and F_i^+. If $F_i < F_i^+$, then by Assumption 5.4.1, we have reasons to believe that $f(i) < f(i+1)$, and hence we have reason to believe that i is a better estimate than $i+1$ and so we let i be our next point. If $F_i > F_i^+$, then by Assumption 5.4.1, we have reasons to believe that $f(i) > f(i+1)$ and hence we have reasons to believe that $i+1$ is a better estimate than i and so we let $i+1$ be our next point. Similarly if $F_i = F_i^+$, then we have reasons to believe that both the points i nd $i+1$ are equally good and so we remain at i.

The algorithm goes as follows

- Let at the kth iterate the current feasible point be i_k. Then take randomly either $i_k + 1$ or $i_k - 1$ to be the possible next point. Also, at each iterate, let i_k^* be the current estimate of the optimum, where i_k^* is the most frequently visited i value by the sequence $\{i_k\}$ till the kth iterate.

- Now by conducting an experiment or a simulation at i_k and its chosen neighbour we will get an estimate of the value of the objective function at the two points, i.e. an observation of each of the random variables F_{i_k} and $F_{i_k}^+$. Then we determine the point amongst the two (i_k and i_k+1 or i_k and i_k-1) at which the objective function is likely to have a lower value.

- We will let the next iterate to be the "better" of the two. For example, if we have decided to compare i_k and i_k+1 then,

$$i_{k+1} = \begin{cases} i_k + 1 & \text{if } F_{i_k} > F_{i_k}^+ \\ i_k & \text{if } F_{i_k} < F_{i_k}^+ \end{cases}$$

Under the guise of Assumption 5.4.1 and also assuming that $\forall k$ the observations of F_{i_k}, $F_{i_k}^+$ are independent of observations of $F_{i_0}, F_{i_0}^+, F_{i_1}, F_{i_1}^+, \ldots, F_{i_{k-1}}, F_{i_{k-1}}^+$, Andradottir has proved that the random walk $\{i_k\}$ is positive recurrent and the sequence $\{i_k^*\}$ converges almost surely to an element of $\mathcal{S}$, where $\mathcal{S}$ is the set of local minima of $f(i)(= EF_i)$. We shall henceforth call the above algorithm as ADSOA (Andradottir's' Discrete Stochastic Optimization Algorithm).

5.4.3 Our problem and ADSOA

Observe that at each iterate of ADSOA we need to generate an observation of the random variable F_i. In our problem we have

$$EF_i = \gamma C(\sigma_i) + \beta \frac{n\sigma_i C(\sigma_i)}{\rho^*}$$

where $C(\sigma_i)$ solves, for given σ_i the equation

$$\lambda_{(\sigma_i,\rho^*)}(\frac{n\sigma_i c}{\rho^*}, c) = -\ln p_m \tag{5.5}$$

Observe that F_i is random as $C(\sigma_i)$ is obtained from 5.5 via simulations. Thus,

$$F_i = \gamma C_{\sigma_i} + \beta \frac{n\sigma_i C_{\sigma_i}}{\rho^*}$$

where C_{σ_i} is an estimate of $C(\sigma_i)$. We shall use the *Fast Simulation Based Stochastic Approximation Algorithm(FSSAA)* for getting the estimate C_{σ_i} of $C(\sigma_i)$. For the complete theory behind *FSSAA* please refer to [16]. Note that because of the fact that our estimates of F_i are obtained from an iterate of FSSAA for σ_i there are two peculiar features of our algorithm which makes it unsuitable for a direct application of ADSOA,

1. The observations of F_{i_k} and $F_{i_k}^+$ are not independent of $F_{i_0}, F_{i_0}^+, F_{i_1}, F_{i_1}^+, \ldots, F_{i_{k-1}}, F_{i_{k-1}}^+$.

2. At the kth iterate we need to obtain an estimate of the random variables F_{i_k} and $F_{i_k}^+$ by updating the FSSAA associated with σ_{i_k} and σ_{i_k+1} or σ_{i_k-1}. Note that an update for the FSSAA associated with σ_i can be triggered at the kth iterate in the following conditions,

(a) If at the kth iterate the sequence $\{i_k\}$ is at i, i.e. $i_k = i$.

(b) If $i_k = i-1$ and for the $k+1$th iterate we have decided to compare the estimates of objective function at $i-1$ and i.

(c) If $i_k = i+1$ and for the $k+1$th iterate we have decided to compare the estimates of objective function at $i+1$ and i.

Let us denote by $F_{(i,k)}$ as the estimate of F_i at the kth iterate of F_i. Due to the w.p. 1 convergence of FSSAA, given any $i \in \mathcal{N}$, for each $\epsilon > 0$, $\exists\ K_{i,\epsilon} < \infty$ s.t. $\forall k > K_{i,\epsilon}$, $\mid F(i,k) - F(i,K_{i,\epsilon}) \mid < \epsilon$ w.p. 1.

Let us define,

$$\epsilon(a,b) = \mid EF_a - EF_b \mid$$

Thus, due to the with probability (w.p.) 1 convergence of FSSAA, $\forall k > \max\{K_{i,\epsilon(i,i+1)}, K_{i+1,\epsilon(i,i+1)}\}$, either one of the two will hold w.p. 1,

- $P\{F_i > F_{i+1}\} = 0$ and $P\{F_i < F_{i+1}\} = 1$. This will happen if $EF_i < EF_{i+1}$.
- $P\{F_i > F_{i+1}\} = 1$ and $P\{F_i < F_{i+1}\} = 0$. This will happen if $EF_i > EF_{i+1}$.

Thus the random variables F_{i_k} and $F^+_{i_k}$ in our problem will satisfy the following property $\forall k$,

$$P\{F^+_{i_k} > F_{i_k}\} \geq 0 \; and \; P\{F^+_{i_k} < F_{i_k}\} \geq 0 \text{ w.p.1}$$

Observe that the Assumption 5.4.1 of ADSOA fails because the positivity of $P\{F^+_{i_k} > F_{i_k}\}$ and $P\{F^+_{i_k} < F_{i_k}\}$ is replaced by non-negativity.

We claim that inspite of this fact, we can apply ADSOA and our sequence $\{i^*_k, F_{i^*_k}\}$ will converge to $\{i^*, EF_{i^*}\}$ w.p. 1, with i^* being the true optimum in $\mathcal{N}$. The following discussion establishes the truth of this claim. Let us consider some $k = k_1$. Without loss of generality we will take the case where $i_{k_1} = i$ and we are comparing the objective function at i and $i+1$. Also, assume that $EF_{i-1} < EF_i < EF_{i+1}$. The complementary case of $i-1$ and i can be proved similarly. Suppose $k_1 = \max\{K_{i,\epsilon(i,i+1)}, K_{i+1,\epsilon(i,i+1)}\}$. Then $i_{k_1+1} = i$ w.p. 1. For $k > k_1$, as long as we will compare the objective function values at i and $i+1$ we will always get $i_{k+1} = i$ w.p. 1. Let $k_2 = \inf\{k >$

k_1, decided to compare i and $i-1$ at k }. If $k_2 \geq \max\{K_{i,\epsilon(i,i-1)}, K_{i-1,\epsilon(i,i-1)}\}$ then $i_{k_2+1} = i-1$ w.p. 1 else $i_{k_2+1} = i$ or $i-1$. Let us define K^* as,

$$K^* = \max_{i,i+1 \in N} K_{i,\epsilon(i,i+1)}$$

Also, since $\forall\ i,\ K_{i,\epsilon(i,i+1)} < \infty$, hence $K^* < \infty$. This means that $\forall k > K^*$, $\{i_k\}$ will be such that $EF_{i_k} < EF_{i_{k-1}}$ w.p. 1. and hence $\{i_k\}$ will converge to $\{i^*\}$ w.p. 1. A little thinking will clear that $\{i_k^*\}$ will also converge to $\{i^*\}$ w.p. 1.

5.5 The On-Line Optimization Algorithm

Our algorithm has two phases:

1. Search Space Reduction

2. Fine Tuning

5.5.1 *Phase I:* Search Space Reduction

The following is the sequence of steps to be executed in this phase:

1. *Determination of the size of good enough subset*, i.e. $\mid G(\Theta) \mid = \lceil N * n\% \rceil = g$.

2. *Selection of the selected subset*, $S(\Theta)$ for a given alignment probability, P_A, alignment level k and a given selection rule such as BP.

3. *Determination of ranking in the selected subset S*. Observe that, S is written as a function of Θ because a subset is selected from the elements of set Θ, i.e. $S \subset \Theta$. Also, by the philosophy of ordinal optimization, the relative ranking amongst different competing parameter values reach stability much earlier than the time required to evaluate the exact value of the objective function. Thus we shall, run the simulations (FSSAA) in parallel for all the members of S, for a sufficient no of iterates (note that

the FSSAAs for different σ are synchronous in the sense that the update epochs for all FSSAA is same) such that the relative ranks reach stability and then we will find the argument of the minimum objective function from S, call it as σ_0, i.e.,

$$\sigma_0 = arg \min_{\sigma \epsilon S} EF_\sigma$$

σ_0 is then used as an initial estimate of σ^*. Observe that there is a potential problem in applying this approach,

Definition 5.5.1 *Due to the very approach of Ordinal Optimization, there is always a small but positive probability ($= 1 - P_A$) that σ_0 may not belong to the set G at all. If this event occurs we shall say that our σ_0 is a* victim of P_A. *Now, if we start our second phase with the initial approximate as σ_0, then our estimation of σ^* at the end of the second phase may not even be a member of G (in the worst case). We propose the following solution.*

Solution: We assume that a unique σ^* exists for our optimization problem and there are no local minima. Let $g = 10$ then as seen in Figure 5.2 if we work with $s = 17$ we get $P_A \geq 0.95$. This means that there is 95% chance that at least one element of S will be among the top 10 designs. Assume that in the course of simulation we obtain the true ranking amongst these s samples of S. Due to the underlying assumption in the theory of ordinal optimization it is known that the number of iterates of FSSAA require for achieving stable ranking amongst various σ is much lesser than the number of iterates required for achieving the true convergence of FSSAA. Assume that we achieve stable ranking at the end of mth iterate of FSSAA and that the values of cost function at the end of this mth iterate be denoted by $F_\sigma \forall \sigma \in S$. Let σ_0 have the first rank (i.e. $F_{\sigma_0} \leq F_\sigma \forall \sigma \in S$) among all the samples S. Now we will give a very simple check to find whether this σ_0 is a victim of the P_A or not.
Let $G(\Theta)$ be $\{\sigma_{[1]}, \sigma_{[2]}, \ldots, \sigma_{[g]}\}$. Observe that because we are working with a granularity of d the consecutive σ values have a fixed difference of d. Also, because of the assumption of unique minima the maximum difference between the members of G is

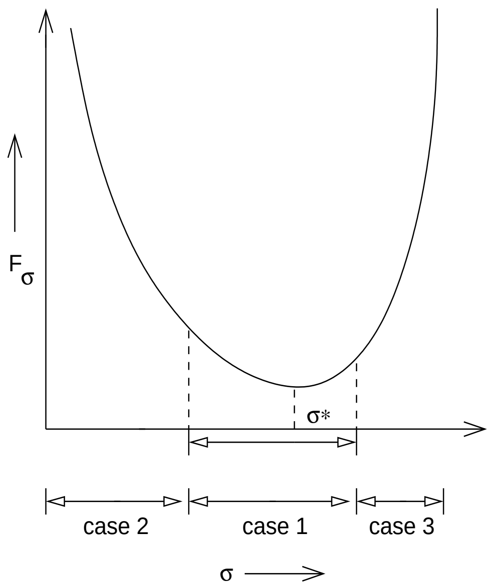

Figure 5.3: The three possible cases with σ^* being the true optima.

$d * g$. We shall call this the *range* of G.

- Look [8] at F_{σ_1} and F_{σ_2} where $\sigma_1 = \max(\sigma_0 - d * g, 0)$ and $\sigma_2 = \min(\sigma_0 + d * g, T\rho^*)$.
- Compare F_{σ_1} and F_{σ_2} with F_{σ_0}. One of the following three conditions may arise (see Figure 5.3)
  - **Case 1:** (σ_0 lies in the region marked as case 1 in Figure 5.3) $F_{\sigma_1} > F_{\sigma_0}$ and $F_{\sigma_2} > F_{\sigma_0}$

    This means that F_{σ_0} is not a victim of P_A and $\sigma_0 \in G$, since if σ_0 was a victim then either $\sigma_0 + d * g$ or $\sigma_0 - d * g$ will yield a smaller F.
  - **Case 2:** (σ_0 lies in the region marked as case 2 in Figure 5.3) $F_{\sigma_1} > F_{\sigma_0}$ and $F_{\sigma_2} < F_{\sigma_0}$

    This means that in the worst case σ_0 is less than the smallest member of G. Note that here the term *smallest* is in the sense of the actual values of σ and not of F_σ. Now, we again start *Phase I* with $S = S_{new}$ given as

$$S_{new} = S(\Theta_{new}) \ with \ \Theta_{new} = \Theta \setminus \{\sigma : \sigma < \sigma_0\}$$

[8] an important remark to be made here is that because of the fact that we have achieved stable ranking, for any σ_1 and σ_2 belonging to S, we can say that, $F_{\sigma_1} < F_{\sigma_2}$ hopefully implies that $EF_{\sigma_1} < EF_{\sigma_2}$

- **Case 3:** (σ_0 lies in the region marked as case 3 in Figure 5.3) $F_{\sigma_1} < F_{\sigma_0}$ and $F_{\sigma_2} > F_{\sigma_0}$
  This means that in the worst case σ_0 is greater that the largest member of G. Now we again start *Phase I* with $S = S_{new}$ given as

$$S_{new} = S(\Theta_{new}) \; with \; \Theta_{new} = \Theta \setminus \{\sigma : \sigma > \sigma_0\}$$

At the end of every *Phase I* we can check for the above problem and if the problem exists then the proposed solution (of restarting *Phase I* with S_{new}) can be executed . Observe that, this solution procedure guarantees that we shall be able to find a S such that the σ_0 obtained from it is in G. This assertion shall be made clear in the following discussion. Let us initially start *Phase I* with $\Theta = \Theta_{new,0}$. Observe that if an error (here the term error means that our σ_0 obtained after *Phase I* is a victim of P_A) occurs and is detected, we shall gain execute *Phase I* but with $\Theta = \Theta_{new,1}$. Also due to the above procedure $\Theta_{new,1} \subseteq \Theta_{new,0}$. More generally, the sequence $\{\Theta_{new,i}\}$ has the property that $\Theta_{new,i} \subseteq \Theta_{new,i+1}$ $\forall i \geq 0$. The equality is due to the fact the the σ_0 obtained at the ith execution of *Phase I* is the smallest member of $\Theta_{new,i}$ and a problem like **Case 2** arises or because σ_0 is the largest member of $\Theta_{new,i}$ and a problem like **Case 3** arises. In both the cases $\Theta_{new,i+1} = \Theta_{new,i}$. In such a case we are stuck. But, observe that because of BP rule, at the $i+1$th execution when we will generate $S(\Theta_{new,i+1})$ then there is always a positive probability that σ_0 that was obtained in the ith execution is not the member of $S(\Theta_{new,i+1})$. Thus, at $i+1$th iterate there is a non-zero probability that $\Theta_{new,i+1} \subset \Theta_{new,i}$. Equivalently the sequence $\{N_i(=\mid \Theta_{new,i} \mid)\}$ is non-increasing [9]. Note that from the BP rule it is clear that as N_i decreases s also decreases for a given P_A and g. The minimum value of s can be 1. For $s = 1$, the minimum value of N for a given P_A, g and k is :

$$N = \frac{g}{P_A}$$

[9] For any two sets A and B, $A \subseteq B \Rightarrow \mid A \mid \leq \mid B \mid$.

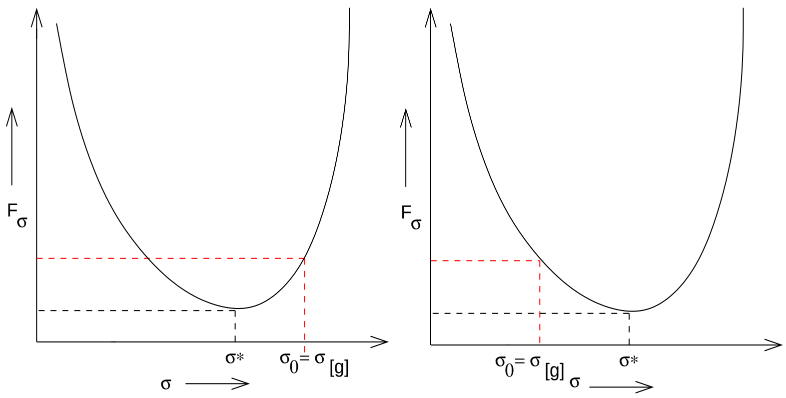

Figure 5.4: The two possible cases

Since, N has to be integer valued and hence for $N = \lceil \frac{g}{P_A} \rceil$, and with $s = 1$, the alignment probability is lesser than P_A and for $N = \lfloor \frac{g}{P_A} \rfloor$, it is greater than P_A. Thus, for $g \leq N \leq \lceil \frac{g}{P_A} \rceil$, even for $s = 1$ (minimum value of s), P_A starts increasing and for $N = g$, $P_A = 1$, thus guaranteeing that w.p. 1 $\sigma_0 \in G$.

Recall that for each $\sigma \in \Theta$ we have associated a unique natural number i ($1 \leq i \leq N$). Henceforth, we will again talk in terms of i as we did in Section 5.4.2. Let σ_0 (initial estimate of σ^* at the start of *Phase II*) correspond to i_0. Observe that, it may happen(in the worst case) that the σ_0 obtained is the worst member of G, i.e. $\sigma_0 = \sigma_{[g]}$. Two worst cases may arise,

- (Refer to Figure 5.4) When we shall have to narrow our search space to,

$$\mathcal{N}_r = \{(i_0 - g + 1), (i_0 - g + 2), \ldots, i_0\}$$

- (Refer to Figure 5.4) When we shall have to narrow our search space to,

$$\mathcal{N}_r = \{i_0, (i_0 + 1), \ldots, (i_0 + g - 1)\}$$

Taking into consideration the possibility of either of the worst case we propose to have our search space for the second phase as,

$$\mathcal{N}_r = \{(i_0 - g + 1), (i_0 - g + 2), \ldots, i_0, \ldots, (i_0 + g - 2), (i_0 + g - 1)\}$$

i.e, $\mid N_r \mid = 2g + 1$.

5.5.2 *Phase II:* Fine Tuning

In *Phase II* we will follow the ADSOA mentioned in Section 5.4.3. At the kth iterate the estimates of F_{i_k} and $F_{i_k}^+$ are are obtained by updating the FSSAA associated with i_k and $i_k + 1$ (or $i_k - 1$). The details of the steps to be executed are given below but first we will describe the new notation used.

$N_m(i)$ is the number of times the sequence $\{i_m\}$ has visited state i during the first m iterations of the algorithm

i_m^* is the state value that the sequence $\{i_m\}$ has visited most often during the first m iterations of the algorithm

$g(i)$ is the no. of iterations of the FSSAA associated with i

$a_{i,g(i)}$ is the value of the gain coefficient associated with i for the $(g(i) + 1)$th iterate of the FSSAA

$H_{i,g(i)}$ is the observation of C_i at the $g(i)$th iterate of FSSAA for i

$\varepsilon_{i,g(i)}$ is the simulation noise associated with the $(g(i)+1)$th measurement interval of FSSAA for i

1. Starting point is (i_0, C_0), where C_0 is the initial estimate of capacity for all $i \in \mathcal{N}_r$ at $N_m(i) = 1$. Let $N_0(i_0) = 1$ and $N_0(i) = 0$ for all $i \in \mathcal{N}_r$, $i \neq i_0$. Let $m = 0$ and $i_m^* = i_0$. We also have $C_{m,N_m(i_0)}(i_0) = C_0$.

2. Generate a Bernoulli random variable J_m such that,

$$J_m = \begin{cases} 1 & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases}$$

Now do,

- if $J_m = 1$ and $i_m \neq (i_0 + g - 1)$, then go to Step 3
- if $J_m = 1$ and $i_m = (i_0 + g - 1)$ then put $i_{m+1} = i_m$ and goto Step 5
- if $J_m = 0$ and $i_m \neq 1$, then go to Step 4
- if $J_m = 0$ and $i_m = 1$, then put $i_{m+1} = i_m$ and go to Step 5

3. Generate an observation R_m of F_{i_m} and an observation R_m^+ of $F_{i_m}^+$ with i_m being the value of the parameter at the mth iterate. Also, to obtain R_m we need to have an observation H_{i_m} of C_{i_m} because,

$$R_m = \gamma H_{i_m} + \beta \frac{n\sigma_{i_m} H_{i_m}}{\rho^*}$$

We shall obtain H_{i_m} by FSSAA. Thus the FSSAA associated with i_m undergoes an updation i.e.,

$$H_{i_m,g(i_m)+1} = H_{i_m,g(i_m)} + a_{i_m,g(i_m)} \left(Z(H_{i_m,g(i_m)}) + \epsilon_{i_m,g(i_m)} \right)$$

and modify $g(i_m)$ as,

$$g(i_m) = g(i_m) + 1$$

and then let,

$$H_{i_m} = H_{i_m,g(i_m)}$$

Similarly for R_m^+ we need to have an observation H_{i_m+1} of C_{i_m+1} which shall again be estimated using FSSA as,

$$H_{i_m+1,g(i_m+1)+1} = H_{i_m+1,g(i_m+1)} + a_{i_m+1,g(i_m+1)} \left(Z(H_{i_m+1,g(i_m+1)}) + \epsilon_{i_m+1,g(i_m+1)} \right)$$

again,

$$g(i_m + 1) = g(i_m + 1) + 1$$

and then let,

$$H_{i_m+1} = H_{i_m+1, g(i_m+1)+1}$$

and R_m^+ is obtained by,

$$R_m^+ = \gamma H_{i_m+1} + \beta \frac{n\sigma_{i_m+1} H_{i_m+1}}{\rho^*}$$

Now do,

- if $R_m^+ < R_m$ then let $i_{m+1} = i_m + 1$ and go to Step 5
- else if $R_m^+ \geq R_m$ then let $i_{m+1} = i_m$ and go to Step 5

4. Generate an observation R_m of F_{i_m-1} and an observation R_m^+ of $F_{i_m-1}^+$. Again, for each observation R_m of F_{i_m-1} and R_m^+ of F_{i_m} we need to have an observation H_{i_m-1} of C_{i_m-1} and an observation $H_{(i_m-1)+1}$ of $C_{(i_m-1)+1}$ respectively. Again H_{i_m-1} and $H_{(i_m-1)+1}$ are obtained by FSSAA as in the previous step and the corresponding observations of R_m and R_m^+ are obtained.

   Now do,

   - if $R_m^+ > R_m$ then let $i_{m+1} = i_m - 1$ and go to Step 5
   - else if $R_m^+ \leq R_m$ then let $i_{m+1} = i_m$ and go to Step 5

5. Let $m = m + 1$, $N_m(i_m) = N_{m-1}(i_m) + 1$ and $N_m(i) = N_{m-1}(i)$ for all $i \epsilon \mathcal{N}_r$ and $i \neq i_m$. Now do,

   - if $N_m(i_m) > N_m(i_{m-1}^*)$, then let $i_m^* = i_m$ and go to Step 2
   - else let $i_m^* = i_{m-1}^*$ and go to Step 2

REMARKS:

1. It is clearly not necessary to store the sequences $\{i_m\}$, $\{i_m^*\}$, and J_m, as in each iteration it is possible to overwrite the earlier values as they will not be used again by the algorithm.

2. However for each $i \in \mathcal{N}_r$ we need to maintain three counters $\{N_m(i)\}$, $\{H_{i,g(i)}\}$ and $\{g(i)\}$. If the cardinality of set $\mathcal{N}_r$ is very large than this may require a sufficient memory at the network node(remember that our σ^* determination algorithm is running at the network node). One possibility (proposed in [4]) is to choose a number of counters to maintain at the beginning, and then increasing the number of counters whenever the algorithm reaches a state that does not have a counter associated with it.

3. The choice of stopping criteria is left to the user. In fact we can run the algorithm continuously without stopping it during the life of the connection and periodically update the value of σ^* until convergence takes place.

5.6 Algorithm Implementation and Simulation Results

Figure 5.5 shows in top-down fashion the motivation for using different techniques during the course of our algorithm.

5.6.1 Simulation Set-up

Phase I:

Figure 5.7 shows the simulation set-up that we need for *Phase I* of our optimization algorithm. The VBR source is smoothed through the LB shaper with $\sigma = T\rho^*$ and $\rho = \rho^*$. The shaper output is the arrival process to the multiplexer with buffer size $b_m = T\rho^*$ and server of capacity $c = \rho^*$. which simulates the network node. Simultaneously the same arrival process also acts as an input process for each of the $\mid S \mid$ parallely operating LB shapers, with parameters $\{\sigma_i^s, T\rho^* - \sigma_i^s\}$ where $\sigma_i^s \in S \; \forall i \; \{1, 2, \ldots, \mid S \mid\}$. For each i the output of the

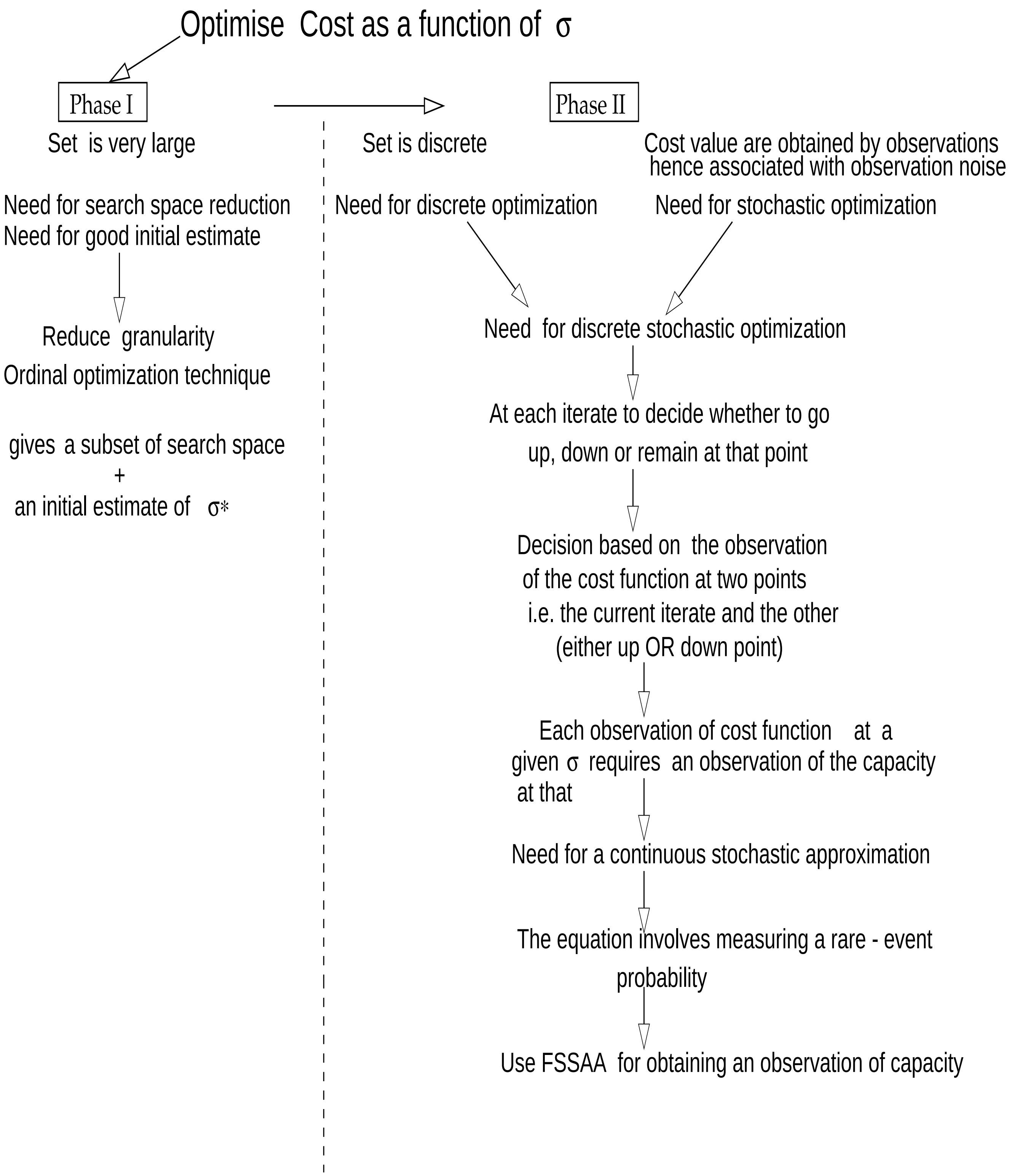

Figure 5.5: A top-down representation of our optimisation problem depicting the motivation for using different techniques during the course of finding a solution

$\{\sigma_i^s, T\rho^* - \sigma_i^s\}$ shaper acts as the input process for the FSSAA [10] for that σ_i^s, i.e., $FSSAA_{\sigma_i^s}$. The need for such an arrangement can be appreciated by the following line of reasoning

- If in the given setting the token bucket depth of the LB shaper at the source has value σ_L and the token arrival rate is ρ^* then the arrival stream to the network node is (σ_L, ρ^*) complaint.

- The FSSAA for each σ_i^s requires an arrival process which is (σ_i^s, ρ^*) complaint.

- There is thus a need to obtain a (σ_i^s, ρ) complaint arrival process from the arrival process to the network node, which is (σ_L, ρ^*) complaint.

- Theorem 5A.1 in Appendix 5A provides an arrangement for generating a (σ_2, ρ^*) complaint arrival process from a (σ_1, ρ^*) for any $\sigma_1 > \sigma_2$.
  Significance of Theorem 5A.1:
  Suppose we want to generate the arrival process W_{σ_2} of a (σ_2, ρ^*, R) source from an arrival process W_{σ_1} of a (σ_1, ρ^*, R) source, with $\sigma_2 < \sigma_1$. Then Theorem 5A.1 says that the two arrangements shown in Figure 5.6 are equivalent. Thus, we can gener-

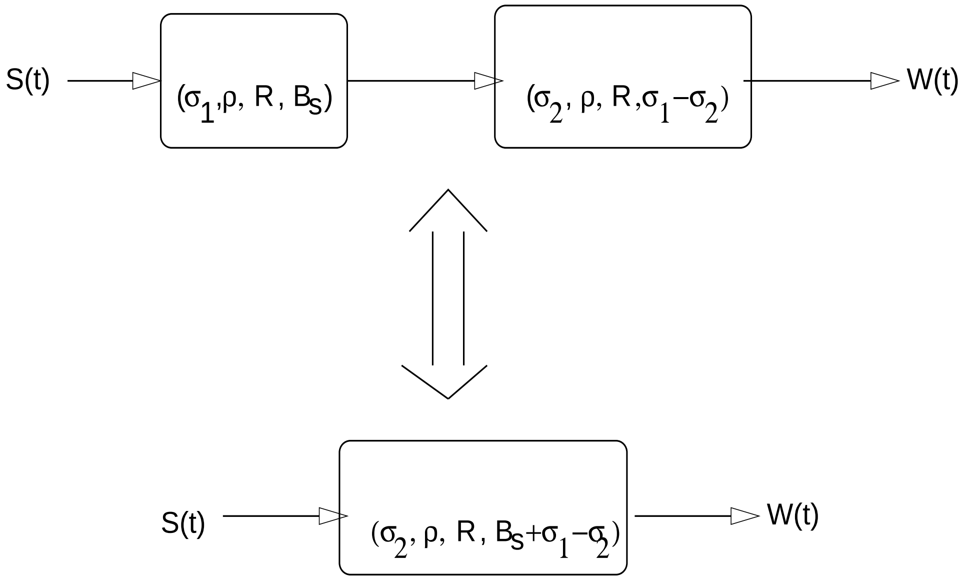

Figure 5.6: Equivalent Leaky Bucket Arrangements

[10] For the details of the simulation set-up for FSSAA refer to [16]

ate a (σ_2, ρ^*, B_s) arrival process at the multiplexer by feeding the (σ_1, ρ^*, B_s) input process(which is the output process from the LB shaper at the source end) to the multiplexer to a $(\sigma_2, \rho^*, \sigma_1 - \sigma_2)$ LB shaper at the network node.

- Observe that each of the block marked with dashed lines shown in Figure 5.7 is similar to the arrangement shown in Figure 5.6 with $\sigma_L = \sigma_1$ and $\sigma_i^s = \sigma_2$

- Note that because we are working with $d = 10$ and BP rule for selecting S, $\sigma_i^s \ \forall i \ \in S$ can take values from $\{0, 10, 20, \ldots, 680, 690\}$. The maximum element of S can be 690 $(= T\rho^*)$.

- We will keep σ_L to be $T\rho^*$. By doing this we can always generate the arrival process for any $\sigma_i^s \in S$ by an arrangement similar to the dashed blocks in Figure 5.7 since $\forall\ i\ \sigma_i^s \leq T\rho^*$.

Since, at the beginning of the *Phase I* we have no knowledge of the cost curve so we keep $\sigma_L = T\rho^*$ at the LB shaper at the source with $\rho = \rho^*$. We also demand for a $b_m = T\rho^*$ and $c = \rho^*$. This is the maximum possible value of per source capacity and buffer for lossless multiplexing. An argument can be raised that as the minimum cost for lossless multiplexing is achieved at $c = \rho^*$ and $b_m = \sigma = 0$, aren't we unfair to the user by charging a higher cost from it initially (by working with $b_m = T\rho^*$) during *Phase I* of our algorithm. We claim that,

- Though it is true that the source shall be initially charged at a higher cost than that possible with a *lossless multiplexing* but we assume that our σ determination algorithm converges in a much smaller time as compared to the lifetime of the call. Hence, *the average cost during the call lifetime for lossy multiplexing is expected to be less than that for lossless multiplexing.*

- Also a source needs to demand initially the largest value possible value of capacity, i.e., ρ^* and the largest possible buffer value i.e., $T\rho^*$ that it may require during its lifetime [11]. We shall later see that this guarantees that during the course of running the σ^* identification algorithm, whenever the σ_L value is updated to an improved estimate of σ^*, which is conveyed to the source by the network node, the source can *successfully* renegotiate for new values of c and b_m with the network node (as these are always down renegotiations and hence are always successful).

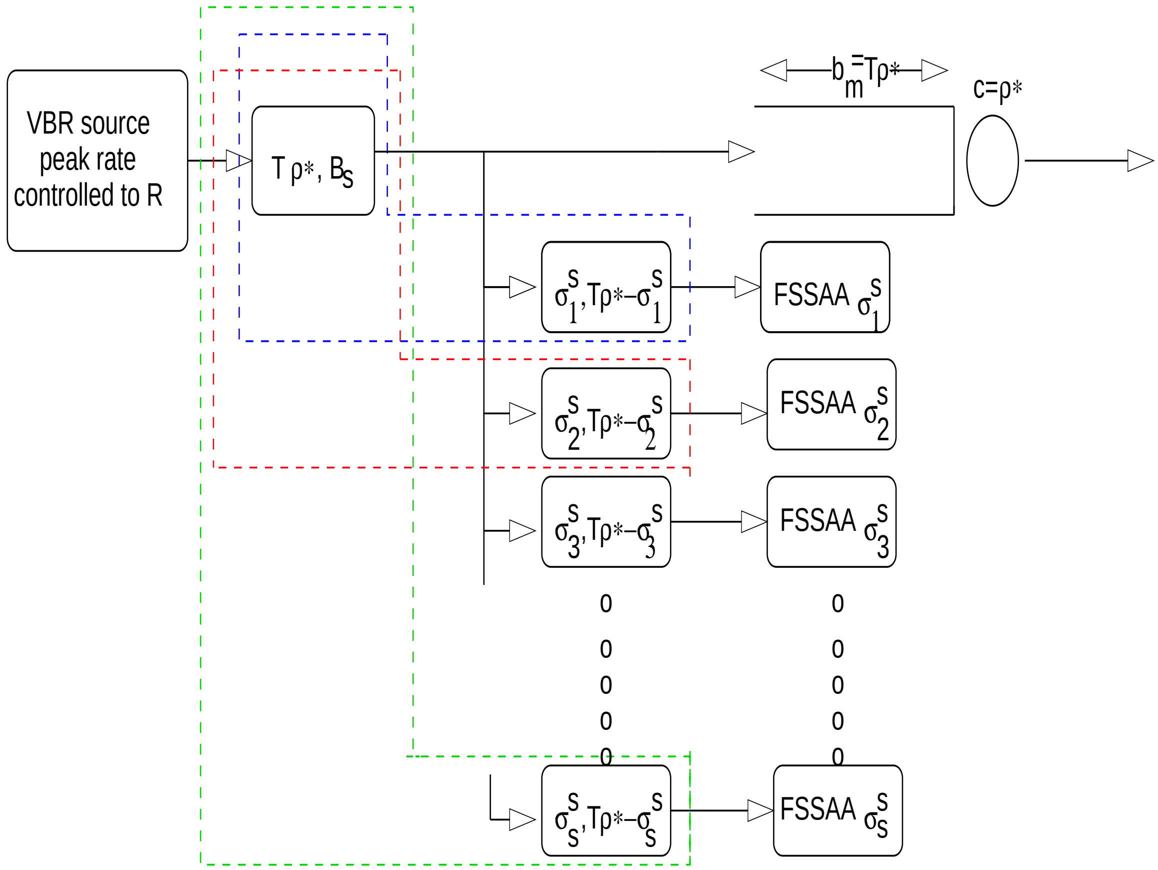

Figure 5.7: Block diagram of the simulation set-up for *Phase I* of our algorithm

Phase II:

Figure 5.8 shows the simulation set-up required for *Phase II* of our algorithm. The important observations in Figure 5.8 are:

[11]Here the term lifetime means the period during which the source in a particular stationary state and so the ρ^* is unchanged

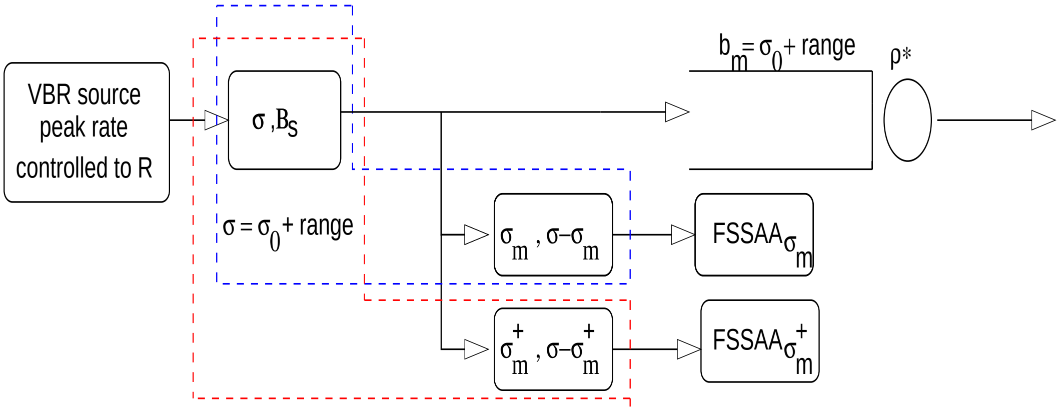

Figure 5.8: Block diagram of the Simulation Set-up for the *Phase II* of our algorithm at the mth iterate. Here σ_m^+ can be either $\sigma_m + 1$ or $\sigma_m - 1$

- We have now kept the value of σ of the LB shaper at the source to be $\sigma_0 + range$. This is because this corresponds to the maximum possible σ value (note that in the second phase our search space is $\{\sigma_0 - range, \sigma_0 + range\}$).
- Also, we keep the value of $c = \rho^*$ and $b_m = \sigma + range$. Since this is the maximum possible values of buffer and capacity that can be possible in the second stage of our algorithm and hence we have again guaranteed that our probability of successful renegotiation is 1.

5.6.2 Simulation Results

In the simulations, the source model was the same as taken in Section 3.7.2 for determination of ρ^* and again employed in Section 4.5.2 for the simulations with the *off-line* algorithm. The LB controller has a token arrival rate of ρ^* ($\simeq 138$) obtained by the our proposed on-line algorithm for ρ^* is Section 3.7.2. We have done the simulations for $n = 150$. Observe that, the average source rate ($\bar{r}_s$) is $\left(\frac{\frac{5}{3}}{\frac{5}{3}+\frac{5}{2}}\right) 170 = 68$ cells/unit time. We use the same value of $J(=3)$ and $D(=4)$ for the gain sequence $\{a_{i,g(i)}\}$ used in FSSAA (See Section 3.6.1) for each i. We work with granularity $d = 10$ and $\Delta = 6$.

***Phase I* Results:**

1. With $d = 10$, the search space $\{\sigma : 0 \leq \sigma \leq 690\}$ reduces to the set $\Theta = \{0, 10, 20, \ldots, 680, 690\}$. Thus, $\mid \Theta \mid = 70$.

2. We took the size of our *good enough set* as 10, i.e., $\mid G(\Theta) \mid = g = 10$.

3. Applying BP rule for subset selection (Refer to Figure 5.2) for a $P_A = 0.93$ and $k = 1$ we get the size of selected subset i.e., $\mid S \mid = s = 15$. We now blindly select (for the meaning of blind selection refer to Section 5.3.2) 15 elements from Θ. This yielded the elements of S as,

$$S = \{40, 140, 280, 80, 90, 370, 400, 340, 190, 600, 660, 430, 420, 50, 550\}$$

4. We now determine the relative ranking among the elements of S. The ranking is obtained by running several parallel FSSAA (one for each element of S) for a sufficient time-interval(till the rankings stabilise). The rank dynamics can be visualised using a *rank dynamics table*. The table shows the relative ranking of various samples at the sampling points. For S used in our simulations the rank dynamics table is given in Table 5.1.

   The time interval between sampling points is 2000. We observed that the rankings have stabilized after 5 sampling points. This is because the ranks remain unchanged after 4th sampling point.

5. The value of the sample occupying the first rank called σ_0 is found to be 140. Note that by *off-line* simulations we have $\sigma^* = 110$. Since $\mid \sigma^* - \sigma_0 \mid < d * g = 100$ hence, our σ_0 is not a *victim of* P_A. Thus we skipped the test for checking whether $sigma_0$ is a victim or not. Observe that this will mostly be the case due to the very high value of P_A. However in the absence of any initial guess/rough estimate of σ^*, the rigorous approach would be to conduct the test after rank stabilisation. If σ_1 and σ_2 (needed in the test) are in S then the test can be done instantly. Otherwise FSSAA is run for σ_1 and σ_2 (till rank stabilisation) and test done after that. With $\sigma_0 = 140$ as the initial

σ	*s*ampling point				
	1st	2nd	3rd	4th	5th
40(6)	1	2	5	6	6
50(5)	2	6	6	5	5
80(3)	3	5	4	3	3
90(2)	4	4	3	2	2
140(1)	5	3	1	1	1
190(4)	6	1	2	4	4
280(7)	7	7	7	7	7
340(8)	8	8	8	8	8
370(9)	9	9	9	9	9
400(10)	10	10	10	10	10
420(11)	12	11	11	11	11
430(12)	11	12	12	12	12
550(13)	13	13	13	13	13
600(14)	14	14	14	14	14
660(15)	15	15	15	15	15

Table 5.1: The rank dynamics table for our example. For each σ the corresponding ranks obtained at sampling points are shown. The term in () brackets gives the correct relative rank for the corresponding σ

estimate we proceed to *Phase II*.

Phase II **Results:**

Starting with $\sigma_0 = 140$ and $c_0 = 72.47$ (obtained at the 4th sampling point of first phase) we run *Phase II* of our algorithm. We did the simulations for measurement interval of 1000 time units. We know the value of $\sigma^* = 110$ and $c(\sigma^*) = 75.26$ from our results of *Of-Line Simulations*. After 13 iterations our proposed *On-Line Algorithm* gives $\sigma^* = 110$ and $c(\sigma^*) = 75.75$.

Figure 5.9 and Figure 5.10 respectively show the successive values of σ_m^* and σ_m on the *y-axis* at the mth iterate with the iterate no. on the *x -axis* for the measurement interval of 1000. We observed that at the 13th iterate in *Phase II* the value of σ_m^* has stabilized and $\forall m > 13$, σ_m^* remains unchanged. We also observed that the sequence $\{\sigma_m\}$ also converges to σ_m^*. This requires further explanation

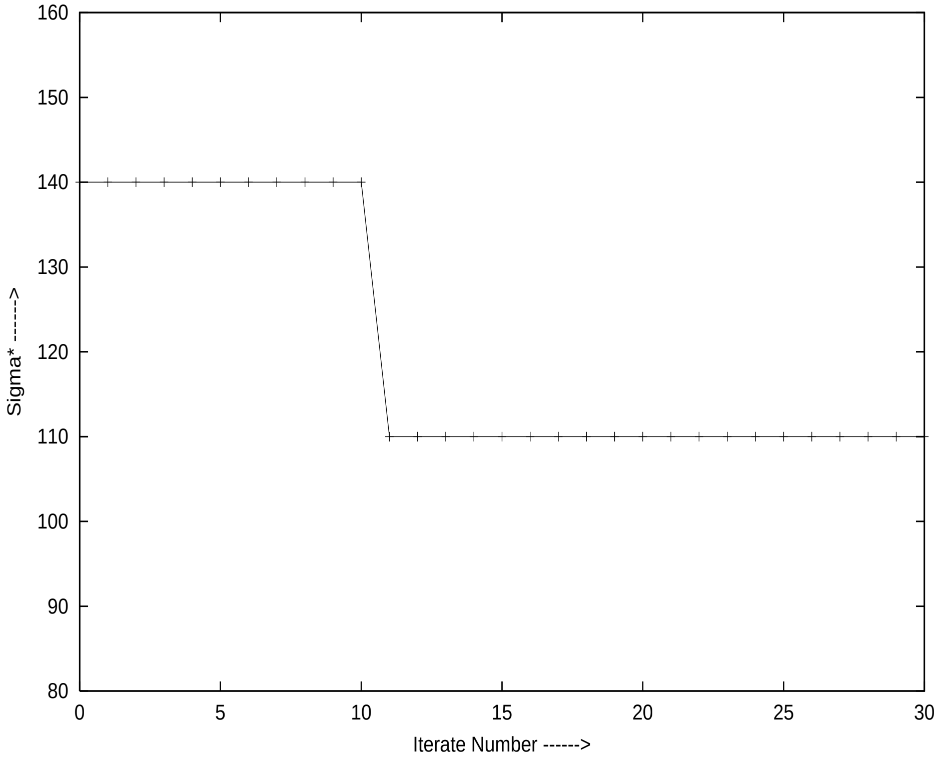

Figure 5.9: The values of σ_m^* at the mth iterate obtained in the *Phase II* of our Algorithm. *off-line technique* gives a value of $\sigma^* = 110$.

Thus, with *Phase I* requiring 5 updates (with 2000 as the time interval between updates) of FSSAA and *Phase II* requiring 13 updates (with 1000 as the time interval) for a time unit of 10 ms, *Phase I* requires 100 secs and *Phase II* requires 130 secs. Thus our σ^* estimation requires about 210 secs of real time, which is acceptable for a video source.

5.7 Conclusion

In this chapter, we have devised an *On-line Stochastic Optimization Algorithm* for determining σ^*. Observe that the search space of σ^* is discrete and may be very large. Thus without a "good" initial approximate of σ^*, any discrete stochastic optimization algorithm will be futile for real-time implementation as the convergence time might be "too much". To work with large search spaces we propose two solutions in *Phase I* of our algorithm, granularity reduction and ordinal optimization. This results in a thinning down of search space from

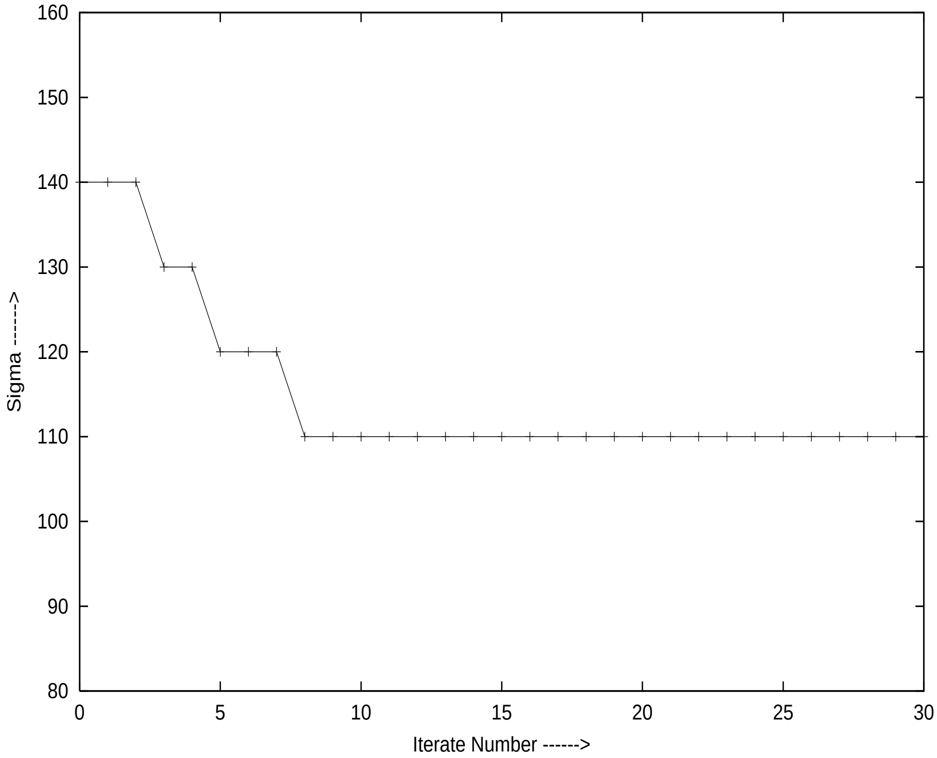

Figure 5.10: The values of σ_m at the mth iterate obtained in *Phase II* of our Algorithm. An important feature to be observed is that the sequence $\{i_m\}(i_m = \sigma_m)$ also converges to σ^* as opposed to ADSOA where only $\{i^*_m\}$ converges and $\{i_m\}$ is a positive recurrent random walk.

cardinality $\lceil T\rho^* \rceil$ to $\lceil \frac{T\rho^*}{d} \rceil + 2$ due to granularity and further reduction in search space from $\lceil \frac{T\rho^*}{d} \rceil + 2$ to $\mid S \mid$ due to subset selection in ordinal optimization. Then we obtained an initial estimate, σ_0 of σ^* from the elements of set S. Having obtained a "good" initial estimate of σ^* we then proceeded to get the exact σ^* in *Phase II* by working with a combination of a discrete stochastic optimization and a fast simulation based continuous stochastic approximation algorithm(FSSAA) that we developed in Chapter 3. Simulations were done for the example source. The σ^* obtained by our proposed algorithm is same as that obtained through off-line measurements in Chapter 4.

Appendix 5A

Here we will develop a result that will be helpful in the practical implementation of our proposed algorithm. In the following, $\{S(t), t \geq 0\}$ denotes the rate process of a fluid source.

Theorem 5A.1 *For a peak rate controlled source, i.e., $S(t) \leq R$, the tandem arrangement of two fluid leaky bucket shapers L_1 and L_2 with parameters (σ_1, ρ, B_s) and $(\sigma_2, \rho, \sigma_1 - \sigma_2)$ with $\sigma_1 > \sigma_2$ is equivalent to a single leaky bucket shaper L with parameter $(\sigma_2, \rho, B_s + \sigma_1 - \sigma_2)$. The equivalence is in the sense that the output rate process $W(t)$ is the same for the two arrangements for $t \geq t_0$ if in either case the leaky buckets are full at t_0.*

Proof:

Figure 5.11 shows the single shaper L and Figure 5.12 shows the tandem arrangement of L_1 and L_2 that we shall be using in our proof. Let $W_1(t)$ denote the output of L and $W_2(t)$

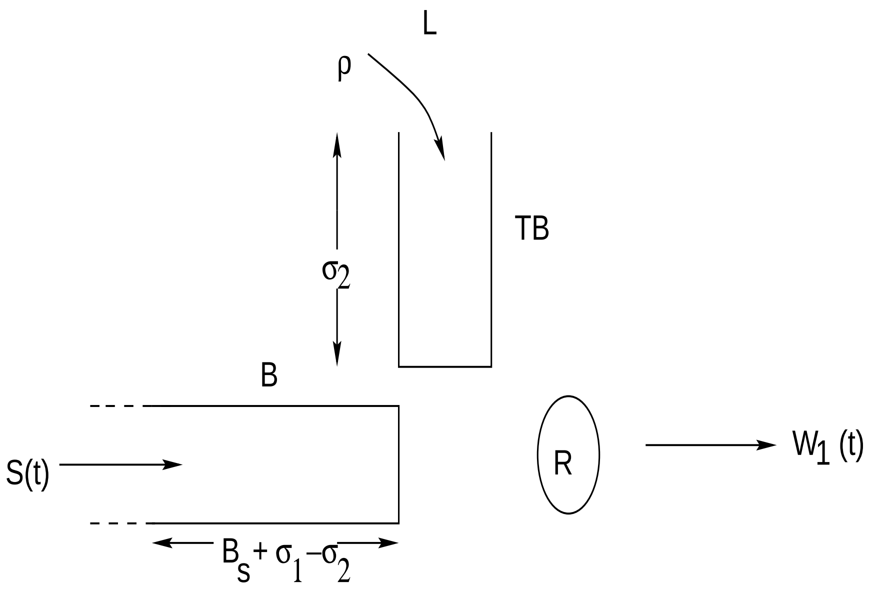

Figure 5.11: Single leaky bucket shaper

denote the output of the tandem arrangement of L_1 and L_2, henceforth called $L_1 + L_2$. For $L_1 + L_2$, let $TB_i(t)$ and $Q_{B_i}(t)$ denote the content of the token bucket TB_i and of the buffer B_i, associated with L_i at time t respectively. For L, let the corresponding quantities be

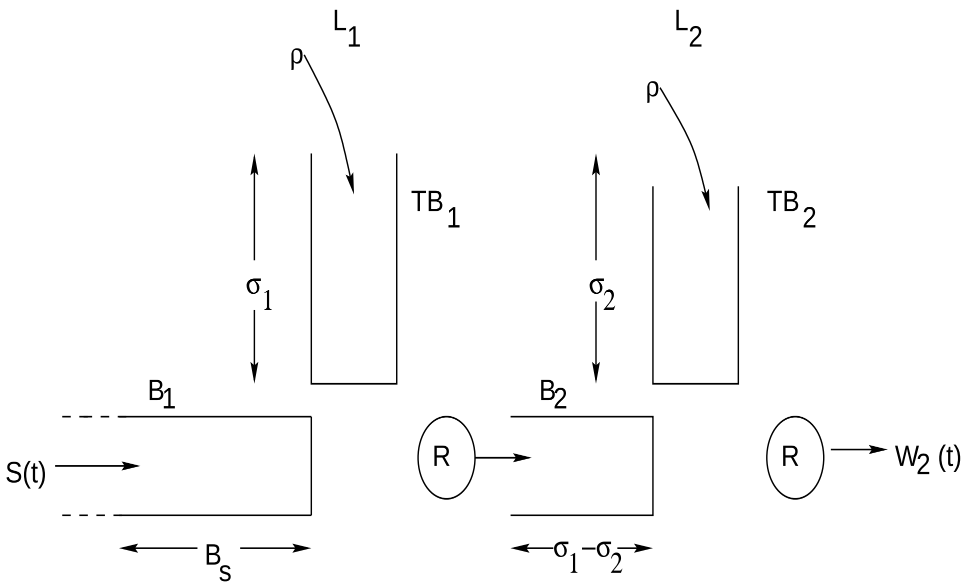

Figure 5.12: Tandem arrangement of two leaky bucket shapers

denoted by $TB(t)$ and $Q_B(t)$. We need to show that for the same arrival process $S(t)$ to L and $L_1 + L_2$, for $t \geq t_0$, $W_1(t) = W_2(t)$ given that, $TB_1(t_0) = \sigma_1$, $TB(t_0) = TB_2(t_0) = \sigma_2$ (and hence, $Q_B(t_0) = Q_{B_1}(t_0) = Q_{B_2}(t_0) = 0$.)

Let,

$$t_{11} = \inf\{t > t_0 : Q_B(t) > 0\}$$

and,

$$t_{12} = \inf\{t > t_0 : Q_{B_2}(t) > 0\}$$

Observe that for L, $W_1(t) = S(t)$ as $TB(t) > 0$ for $0 \leq t < t_{11}$. Similarly, for $L_1 + L_2$, $W_2(t) = S(t)$ for $0 \leq t < t_{12}$ as $TB_1(t) > 0$ and $TB_2(t) > 0$. Also as the depth of TB is same as TB_2 thus $t_{11} = t_{12} := t_1$. Thus $W_1(t) = W_2(t)$ for $t_0 \leq t < t_1$ Also, $TB(t_1) = TB_2(t_1) = 0$ and $TB_1(t_1) = \sigma_1 - \sigma_2$.

Let,

$$u_{11} = \inf\{t > t_1 : TB(t) > 0\}$$

and,

$$u_{12} = \inf\{t > t_1 : TB_2(t) > 0\}$$

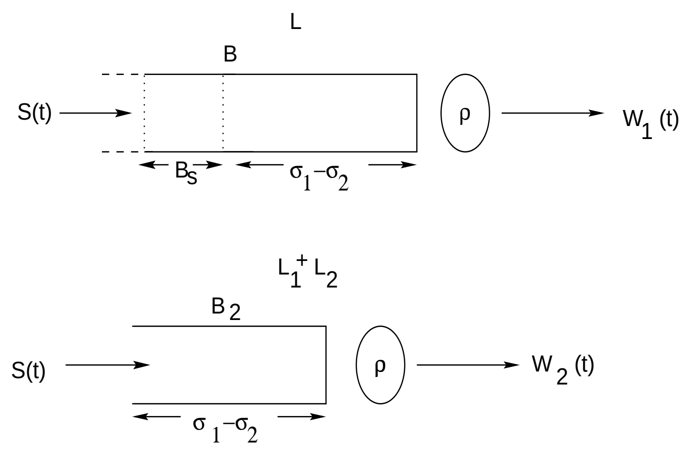

Figure 5.13: The equivalent arrangement for L and $L_1 + L_2$ for $t \geq t_1$

We shall now show that $u_{11} = u_{12} = u_1$ and thus, for $t_1 \leq t < u_1$, $W_1(t) = W_2(t) = \rho$. Figure 5.13 shows the equivalent arrangement for L and $L_1 + L_2$ for $t \geq t_1$. From the figure for L, $W_1(t) = \rho$, $Q_B(t) \leq \sigma_1 - \sigma_2$ and B starts filling and for $L_1 + L_2$, $W_2(t) = \rho$, $Q_{B_2}(t) \leq \sigma_1 - \sigma_2$ and B_2 starts filling. This situation will either continue upto u_{12} or will terminate at a time v_1, s.t.

$$v_1 = \inf\{t : t_1 < t < u_{12}, TB_1(t) = 0\}$$

Also, because of the definition of v_1, for any time t such that, $t_1 < t \leq v_1$,

$$TB_1(t_1) + (\text{amount of token fluid injected in } TB_1 \text{ from } t_1 \text{ to } t) \geq \int_{t_1}^{t} S(t)dt$$

$$(\sigma_1 - \sigma_2) + \rho(t - t_1) \geq \int_{t_1}^{t} S(t)dt \tag{5A-1}$$

with equality at v_1. Also, from Figure 5.13,

$$\int_{t_1}^{t} S(t)dt = \rho(t - t_1) + Q_{B_2}(t) \tag{5A-2}$$

By Equation 5A-1 and Equation 5A-2,

$$(\sigma_1 - \sigma_2) + \rho(t - t_1) \geq \rho(t - t_1) + Q_{B_2}(t)$$

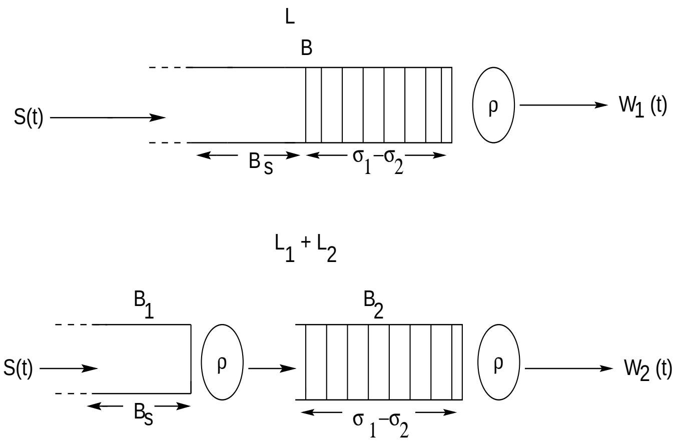

Figure 5.14: The equivalent arrangement for L and $L_1 + L_2$ for time t when $v_1 \leq t < w$

Thus, $Q_{B_2}(t) < \sigma_1 - \sigma_2$ for $t_1 < t < v_1$ and $Q_{B_2}(v_1) = \sigma_1 - \sigma_2$. Since the size of B_2 is $(\sigma_1 - \sigma_2)$ hence no fluid loss can occur in B_2 till v_1.

Similarly for L, from Figure 5.13 for $t_1 < t \leq v_1$, we have,

$$\rho(t - t_1) + Q_B(t) = \int_{t_1}^{t} S(t)dt$$

From Equation 5A-2 we get $Q_B(t) = Q_{B_2}(t)$. Thus $Q_B(v_1) = Q_{B_2}(v_1) = \sigma_1 - \sigma_2$. Since the size of B is $B_s + \sigma_1 - \sigma_2$ hence, no fluid loss will occur in L till v_1.

Define,

$$w_1 = \inf\{t > v_1 : TB_1(t) > 0\}$$

For $v_1 \leq t < w_1$, the equivalent arrangements of L and $L_1 + L_2$ are shown in Figure 5.14 with $Q_B(t) \geq \sigma_1 - \sigma_2$, $Q_{B_1}(t) > 0$ and $Q_{B_2}(t) = \sigma_1 - \sigma_2$. For $v_1 \leq t < w_1$, $TB_1(t) = TB_2(t) = 0$. Thus the arrival rate to B_2 is ρ and $W_2(t) = \rho$ and hence there shall be no fluid loss in B_2 till time w. $Q_{B_2}(t)$ will remain constant at $\sigma_1 - \sigma_2$ and B_1 will start filling at rate $(S(t) - \rho)^+$. Because for any t, $S(t) \in [0, R]$, thus either $\forall t$ s.t. $t < w_1$, $Q_{B_1}(t) \leq B_s$ (hence no QVP [12]violation) or there is a time x_1, s.t.

$$x_1 = \inf\{t < w_1 : Q_{B_1}(t) > B_s\}$$

[12]said to occur when the queue length in B_1 exceeds B_s

also define

$$y_1 = \inf\{x_1 < t < w_1 : 0 < Q_{B_1}(t) \leq B_s\}$$

For $x_1 < t < y_1$, the arrangement is equivalent to Figure 5.14 with $Q_{B_1}(t) > B_s$, $Q_{B_2}(t) = \sigma_1 - \sigma_2$ and $Q_B(t) > B_s + \sigma_1 - \sigma_2$. From the above discussion for $x_1 < t < y_1$, the total fluid in $L_1 + L_2$ is $Q_{B_1}(t) + Q_{B_2}(t)(> B_s + \sigma_1 - \sigma_2)$. Let us define two processes $Q_B^1(t)$ and $Q_B^2(t)$ s.t. $\forall\ t \geq t_0$,

$$Q_B^1(t) = Q_{B_2}(t) \text{ and } Q_B^2(t) = Q_{B_1}(t)$$

From the above discussion it is clear that at any $t \geq t_0$ $Q_B(t) = Q_B^1(t) + Q_B^2(t)$.
Observe that, because of the fact that the source can reduce its rate we can associate a time y_1(defined above) with x_1. Thus, for $x_1 < t < y_1$ there will be QVP violation in $L_1 + L_2$ and also in L, with x_1 being the starting and y_1 the ending epoch of violation for both. After y_1, we again have a situation like Figure 5.13. Hence either this situation will continue upto w_1 or again there is a time x_2 s.t.,

$$x_2 = \inf\{y_1 < t < w_1 : Q_{B_1}(t) > B_s\}$$

associated with a y_2. Continuing in this way, the sequence $\{(x_i, y_i)\}$ will finally terminate [13] in w_1. At w_1 we have, $Q_{B_1}(w_1) = 0, Q_{B_2}(w_1) = \sigma_1 - \sigma_2$ and TB_1 just starts filling. We are again in a situation like Figure 5.13. This situation will either continue upto u_{12} or will terminate in v_2 s.t.,

$$v_2 = \inf\{t : w_1 < t < u_{12}, TB_1(t) = 0\}$$

with an associated w_2 defined as,

$$w_2 = \inf\{t : t > v_2, TB_1(t) > 0\}$$

Continuing in this way, the sequence $\{(v_i, w_i)\}$ will finally terminate in u_{12}. At u_{12}, we have $QB_1(u_{12}) = QB_2(u_{12}) = 0, TB_1(u_{12}) = \sigma_1 - \sigma_2$ and TB_2 starts filling. Also, at u_{12}, $Q_B(t) = Q_B^1(t) + Q_B^2(t) = 0$ and TB will also start filling. But this is precisely the definition

[13] This will happen in a finite time because $S(t) \in [0, R]$ and the assumption that call duration is finite

of u_{11}. Thus $u_{11} = u_{12} = u_1$. After u_1, we can again have a t_2 with its corresponding u_2. For that we can again prove that for $u_1 \leq t < t_2$, $W_1(t) = W_2(t) = S(t)$ and for $t_2 \leq t < u_2$, $W_1(t) = W_2(t) = \rho$ along the same line of arguments. Continuing in this way , we can show that the sequence $\{(t_i, u_i)\}$ has the property that for $u_i \leq t < t_{i+1}$, $W_1(t) = W_2(t) = S(t)$ and for $t_{i+1} \leq t < u_{i+1}$, $W_1(t) = W_2(t) = \rho$. Thus $\forall\ t \geq t_0$, $W_1(t) = W_2(t)$.

Thus, we find that for the same $S(t)$, the output rate process of L and $L_1 + L_2$ and the QoS violation epochs are always the same. Thus the two arrangements are equivalent.

Chapter 6

Related Issues

6.1 Introduction

The ability to dynamically estimate and then renegotiate resources with the network is of paramount importance for efficient utilization of resources in this age of unpredictable multimedia traffic. While our on-line algorithms provide means for robust estimation of the shaper parameters at the call set-up time, the strategy for renegotiation of shaper parameters during the call duration is not clearly stated. In this chapter we shall first design the renegotiation strategies and then shall present a call admission control algorithm based on the optimal parameter estimation algorithms that we have developed in previous chapters.

The results which we have so far presented in our work deals with a subset of the larger set of the problems dealing with providing end-to-end delay guarantees for communication networks carrying multimedia traffic. The network can be single-hop (intra-company PBX or video conferencing) or multi-hop (WAN etc.) and the traffic can be homogeneous (e.g., all English (American/Indian) speaking voice calls with same coding scheme) or heterogenous (e.g., different audio and video). It should be noted *the very aim of any modeling is to gain sufficient insights into the corresponding real world problems. To make any analysis tractable and computationally feasible assumptions are inherent. As the work starts showing results assumptions are gradually pruned down so that the ultimate model is a highly close match of* *the real world model.* In this spirit, we have worked with,

- homogeneous sources
- shaper+multiplexer delay (one hop network)
- linear cost function
- inherently peak rate controlled sources

Having established a characterization of ρ^* and σ^* within the framework of above constraints we are now ready to attack a general and more *real-world like* problem where one or more of the above constraints are not valid. In this chapter we have also looked into the possibility of expanding the horizon of our algorithms to these general *real-world like* situations. We have discussed the feasibility of applying our algorithms to heterogeneous sources and with non-linear cost functions The discussion is preliminary and will provide useful foundation for further work in our problem.

6.2 Renegotiation Issues

Multimedia applications having inherently nonstationary behavior like MPEG coded audio/video etc. cannot be expected to have same arrival statistics during the entire lifetime of the application. If we will overlook this behavior and characterize our traffic with static traffic parameters (estimated at the time of call set-up) then there can be severe over/under utilization of resources as the call progresses. This is an important problem faced in any on-line measurement based call admission control technique as here the source parameters are estimated by making measurements over some finite time window assuming that the arrival process is stationary. But the assumption of stationarity is not true for multimedia applications and hence the extrapolated values of ρ^* and σ^* not be true for the entire duration of the connection.

We aim to solve this problem by dynamic parameter estimation and subsequent resource renegotiation. The discussion following is for homogeneous sources. Renegotiation can be initiated by any one or more of the following events:

- A new call requests connection: The call will need to estimate its ρ^* value. In case of statistically identical sources the ρ^* value of the new source will be the same as ρ^* of other sources. For these sources, the network could explicitly convey the ρ^* value (network can store the ρ^* value of already established connections) to the new source or the source could estimate it on-line. However there is a need to trigger the σ^* estimation algorithm at the network node as σ^* is dependent on n.

- The change detection window of ρ^* running at the source detects a change: Thus there is a need to restart the ρ^* determination algorithm at the source end and as σ^* is a function of ρ^* thus the σ^* determination algorithm at the network end has also to be subsequently retriggered.

We will now describe the ρ^* and σ^* estimation techniques with renegotiations.

6.2.1 ρ^* estimation

The steps for ρ^* estimation with renegotiations are :

- The new call will initially be accepted at its Peak Rate and the ρ^* determination algorithm starts running at the source.

- Let T_ρ be the measurement period for successive updates of ρ. Now we have two alternatives, either to convey the new iterate of ρ at each updation or at some multiple of T_ρ. If we convey at the end of each successive T_ρ, then we are having an efficient utilization of resources but will be requiring excessive signalling. On the other hand conveying at some multiple of T_ρ will obviously require less signalling overhead but will result in inefficient resource utilization. Thus there is a tradeoff between resource utilization and signalling cost.

- Let T_r denote the renegotiation interval (either equal to T_ρ or some multiple of T_ρ) and

T_{r_k} be the kth conveying epoch. We can choose a threshold value, say χ. If at the kth conveying epoch, i.e. at T_{r_k}, $\mid \rho_k - \rho_{k-1} \mid > \chi$, then renegotiate, otherwise not.

- It is preferable to renegotiate "down" in the event of periodic update of ρ as "up" renegotiations are prone to rejection. This will be a major concern in algorithms which are iterative in nature. Because we are working with measurement based estimation, due to measurement errors, at some of the iterates the estimates may be underestimate (until the algorithm converges).

- However as seen from our plots, in Chapter 3 Figure 3.14, for fixed measurement interval, our ρ^* determination algorithm has a graceful monotonic degradation to the true ρ^* and hence we can start renegotiations even at the end of every T_ρ (as all are "down" renegotiations) without the fear of "up" renegotiations later on.

6.2.2 σ^* estimation

σ^* estimation algorithm has to be triggered if a new source arrives or the change detection window of ρ^* detects a change and subsequently a new value of ρ^* is estimated.

- At the end of ρ^* estimation the final estimate of ρ^* is conveyed to the network and and the σ^* estimation algorithm starts at the network.

- One possible variant is to run the ρ^* and σ^* algorithm parallely. In fact we can run the first phase of σ^* algorithm in parallel with the ρ^* algorithm. Successive updates of ρ shall be conveyed to the shaper at the source and the shapers(virtual) at the network node (implemented for σ^* determination) which then modify their token arrival rate value.

- Observe that if we shall start the first phase of σ^* algorithm at the end of the ρ^* algorithm then the total time required for optimal parameter estimation is = time required for ρ^* estimation + time required for first phase of σ^* + time required for second phase of σ^*. However, if we start the ρ^* and first phase of σ^* simultaneously then due to the overlapping of time, the total time required for the end of ρ^* estimation and the first phase of σ^* is relatively less than that required when they are done in succession.

6.3 A Measurement Based Call Admission Control Algorithm (MBCAC) based on ρ^* and σ^*

In our proposed technique for on-line estimation of optimal LB parameters we are running the ρ^* determination algorithm at the source end and the σ^* determination algorithm at the network end. Suppose n sources are currently admitted by the network and the $(n+1)$th source comes. The new source will estimate ρ^* by running the ρ determination algorithm at the source end. Also as noted earlier the σ^* determination algorithm will be triggered at the network end.

After having obtained a set of optimal leaky bucket parameters ρ^* and σ^* for the first time the parameters are negotiated with the network. Then two things may happen:

- Negotiation is successful which implies that the network can provide the optimal Capacity $(c(\rho^*, \sigma^*))$ and optimal Buffer $(b(\rho^*, \sigma^*)$ to all the $(n+1)$ homogeneous sources i.e.,

  $$(n+1)c(\sigma^*) \leq C_l \text{ and } (n+1)b(\sigma^*) \leq B$$

  where, C_l and B are the total link Capacity and the total node Buffer. Thus the $(n+1)th$ source is admitted by the network with LB shaper parameters as ρ^* and σ^*.

- Negotiation is unsuccessful i.e. [1],

  $$(n+1)c(\sigma^*) \geq C_l \text{ and/or } (n+1)b(\sigma^*) \geq B$$

[1] in the succeeding discussion we shall suppress ρ^* as it is same for all sources due to homogeneity

Now, what to do?

Two Options

- reject the connection request.
- the network can convey a suboptimal value of σ to all the $(n+1)$ sources. By suboptimality of σ we mean a $\tilde{\sigma}$ such that

  1. $0 \leq \tilde{\sigma} \leq T\rho^*$
  2. $\tilde{\sigma} \neq \sigma^* \Rightarrow Cost(\tilde{\sigma}) > Cost(\sigma^*)$
  3. $(n+1)c(\tilde{\sigma}) \leq C_l$ and $(n+1)b(\tilde{\sigma}) \leq B$

  There is a tradeoff between going for $\tilde{\sigma}$ and rejecting the connection. As with $\tilde{\sigma}$ the "cost" is more as compared to that with σ^*. Note that $\tilde{\sigma}$ may or may not exist. It may happen that there exists no $\tilde{\sigma}$ such that it satisfies all the three properties above. For each σ assume that $c(\sigma)$ is unique. Let $\frac{C_l}{n} = c(\sigma_{C_l})$ for some σ_{C_l}, $0 < \sigma_{C_l} < T\rho^*$ and $B = B_m(\sigma_B)$ for some σ_B, $0 < \sigma_B < T\rho^*$. Observe that $\tilde{\sigma}$ will exists only if,

  $$c^{-1}(\frac{C_l}{n}) < B_m^{-1}(B) \Rightarrow \sigma_{C_l} < \sigma_B$$

  Thus there is a reduction in the feasible values of $\tilde{\sigma}$ from $[0, T\rho^*] \setminus \{\sigma^*\}$ to $[\min(0, \sigma_C), \min(T\rho^*, \sigma_B)]$

The problem of how to choose $\tilde{\sigma}$ is a subject of further study.

Observe that negotiation can even be unsuccessful if the call is initially accepted at its peak rate, R and if the value of Δ is such that the cost can only decrease by working with capacity less than R and a nonzero buffer. In this case, the value of ρ^* and σ^* is such that $c(\sigma^*) < R$ and $b(\sigma^*) > 0$. This will happen because though $(n+1)c(\sigma^*) < C_l$ but it may happen that $(n+1)b(\sigma^*) > B$ as we know that with the increase in n, c decreases but $B_m(= (n+1)b(\sigma^*)$ increases. In such a case either the network can enforce $\tilde{\sigma}$ (if it exists) as the σ value for all the $(n+1)$ sources or keep the capacity of the $(n+1)$th source at R (this results in a high "cost" for the $(n+1)$th source).

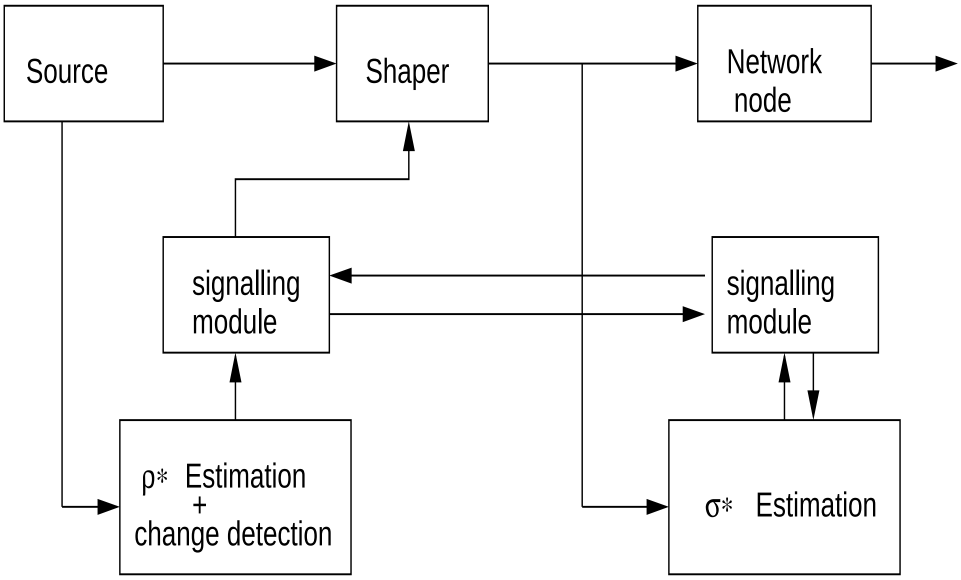

Figure 6.1: The complete block diagram for implementing the ρ^* and σ^* determination Algorithm

6.4 An Overall Block Diagram for Implementing our Proposed ρ^* and σ^* Algorithms

Figure 6.1 shows the set-up required for implementing our proposed algorithms. Initially the shaper parameters are kept at some conservative value (like $\rho = R$ and $\sigma = 0$) and simultaneously the ρ^* source traffic is also fed to the ρ^* estimation module. Successive estimates of ρ^* are conveyed to the shaper through the signalling module at the end of each conveying interval T_c. Once the ρ^* algorithm converges the signalling module at the source conveys the value to its counterpart at the network end and the σ^* estimation algorithm starts running there. At the end of the σ^* estimation the value is conveyed to the shaper at the source end to update its value accordingly. From now on the Shaper has parameters (ρ^*, σ^*) until the occurrence of the need for renegotiation (either because of change detection at the source or the arrival of a new connection request).

6.5 Non-Homogeneous Sources

In our whole analysis we have assumed sources to be homogeneous. This is not true for multimedia applications where heterogeneity is inherent in sources.

Let us consider two sources with (*peak rate, delay bound, QVP*) as (R_1, T_1, q_1) and (R_2, T_2, q_2). We shall now formulate the Minimum Cost Lossless Multiplexing (MCLM) problem for the case of two heterogeneous classes. Working on the same lines as that for homogeneous classes, we get,

$$\min_{b_1, B_1, \sigma_1, \rho_1} c_1 = \max\left(\rho_1, R_1 - \frac{b_1}{\sigma_1}(R_1 - \rho_1)\right)$$

$$\min_{b_2, B_2, \sigma_2, \rho_2} c_2 = \max\left(\rho_2, R_2 - \frac{b_2}{\sigma_2}(R_2 - \rho_2)\right)$$

Subject to: multiplexer delay constraint:

$$\frac{b_1 + b_1}{c_1 + c_2} \leq \min\{\left(T_1 - \frac{B_1 - \sigma_1}{\rho_1}\right), \left(T_2 - \frac{B_2 - \sigma_2}{\rho_2}\right)\}$$

shaper delay constraint and delay QVP:

$$\frac{B_1 - \sigma_1}{\rho_1} \leq T_1 \text{ and } B_1 = g_{q_1}(\rho_1)$$

$$\frac{B_2 - \sigma_2}{\rho_2} \leq T_2 \text{ and } B_2 = g_{q_2}(\rho_2)$$

and by definition,

$$0 \leq \sigma_1 \leq B_1 \text{ and } 0 \leq \sigma_2 \leq B_2$$

Though the problem is formulated but the solution is not evident due to common queues for both classes.

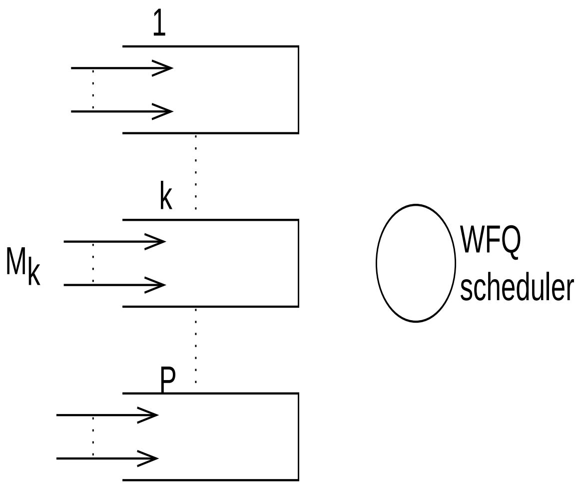

Figure 6.2: Per homogeneous class based queueing at the multiplexer with P homogeneous classes and M_k being the number of sessions in the kth homogeneous class.

6.5.1 Per Class Queueing:

However our scheme for homogeneous sources can readily be extended to non-homogeneous sources by considering a separate shared queue for each homogeneous class. In fact we are asking for *per homogeneous class based queueing*. A possible implementation is shown in Figure 6.2. Here there are say P homogeneous classes(for audio sources this could mean each class has a different coding rate). Let there be M_k members in the kth homogeneous class. We can find the optimal ρ^* and σ^* value for each homogeneous class. However because of other class queues and finite (non zero) packet lengths, the shaper+multiplexer delay (T) is modified as,

$$T' = T - \frac{L_{max}}{C_l}$$

where,

$$L_{max} = \text{ maximum packet length over all the sessions over all the homogeneous classes}$$

$$C_l = \text{ total link bandwidth}$$

6.6 Non-Linear Cost functions

Observe that in our analysis and calculations we have taken a cost function which is linear in buffer and capacity. However due to the shape of our feasible region shown in Figure 4.2 we claim that for the class of "cost" functions having some common properties the minima again lies on the *locus of extremal points.*

Theorem 6.1 *Let $\mathcal{C}$ be a family of positive cost functions $\{C(c,b)\}$ with the following properties,*

1. $\frac{\partial C}{\partial c} > 0$ *and* $\frac{\partial C}{\partial b} > 0$
2. $C(c=0,b) \neq 0$ *and* $C(c,b=0) \neq 0$

Then the minima of each member of $\mathcal{C}$ lies on the locus of extremal points.

Proof

Consider our feasible region shown shaded in Figure 4.2. For lossless case, by Theorem 3.3.1 we have, the minimum feasible value of $c = \rho^*$ and $b = \sigma$, where $0 \leq \sigma \leq T\rho^*$. Since our cost function is s.t. $\frac{\partial C}{\partial b} > 0$, hence the minima is again at $c = \rho^*$ and $b = 0$. For lossy case, we again claim that it is sufficient to look at extremal points only. The reason is that for any point say (c_1, b_1) which is not extremal , we can find an extremal point say (c_0, b_0) s.t. the capacity requirement of this extremal point is the same as that of the non-extremal point, i.e., $c_0 = c_1$ but the buffer requirement is smaller. And since our function is s.t. $\frac{\partial C}{\partial b} > 0$, the non-extremal points are *inadmissible* in the sense that we will have a *strict* decrease in cost by working at extremal point (due to reduced buffer but same capacity as non-extremal point.)

Examples of such functions are linear functions (as we have taken), exponential cost functions like $e^{\gamma c+\beta b}$ etc. Thus, our domain of cost functions has been expanded.

Chapter 7

Conclusion and Future Research Directions

In this thesis, our mail goal was to design measurement based on-line algorithms that could be implemented at the switches and/or at the source for computing optimal LB shaper parameters (σ^* and ρ^*) for sessions through the link. The optimality was in the sense of minimising the network resource requirements (switch buffers + link bandwidth) while simultaneously guaranteeing the shaping+multiplexing (single hop network) delay. In pursuing this goal the following contributions were made

In Chapter 3 we provided the formulation for *lossless multiplexing* for characterizing ρ^*. We also provided a thorough proof for the optimality of ρ^* and pointed out that ρ^* could be written as a solution of a certain implicit equation. The characterisation was initiated in [41]. We completed it and validated the assumptions made. We then developed a measurement based algorithm for on-line estimation of ρ^*. In pursuing this goal we developed a fast simulation algorithm (we called it *FSSAA*) for quick on-line determination of the minimum capacity required for supporting a given session with a given loss probability at a multiplexer with a given buffer size. FSSAA was then applied for developing the algorithm for estimation of ρ^*. The algorithm tracks ρ^* for stationary as well as piecewise-stationary (nonstationary sources with stationary regimes) sources. Simulations are shown to validate the claims of the algorithm.

In Chapter 4 we provide the formulation for *lossy multiplexing* for characterizing σ^*. as the solution of a certain constrained non-linear optimisation problem. The characterisation was proposed in [41]. We analytically established the properties of various functions involved in the characterisation. We then developed an *Off-Line* measurement based algorithm for verifying the function properties that we analytically established. Simulations for the example source validate the analytical conclusions. We next proceeded to develop an analytical calculation of σ^* using the three state MMFP characterisation of the LB output and the *Perron Frobenius* eigen value theory. The analytical curves obtained from this approximate characterisation closely match with those obtained earlier by simulations.

In Chapter 5 we developed a measurement based algorithm for on-line estimation of σ^*. The algorithm combines the feature of discrete and continuous stochastic optimization techniques. The algorithm is found to converge to the off-line estimated σ^* for our example source.

In Chapter 6 we discussed implementation issues for our proposed algorithms and devised a call admission control algorithm with renegotiations. We also expanded the domain of our cost functions and discussed an approach for working with non-homogeneous sources.

7.1 Future Research Directions

A) In Chapter 3 the gain coefficient for ρ^* estimation algorithm at the kth iterate was of the form $\frac{R}{(k+J)(-\ln q)D}$. The terms J and D controlled the convergence rate of the algorithm. We have worked with $J = 3$ and $D = 4$ in our simulations. However, there is a need for developing a theoretical basis for finding J and D for faster convergence. Also, a method for determining the optimum length of the measurement interval has to be devised.

B) The approximate analytical calculation of σ^* using the Markovian characterisation for the output of the shaper (in Section 4.6) is done for a two-state MMFP source. The effectiveness of using the approximate model for arbitrary VBR sources (like n state MMFP) is to be shown.

C) In *Phase II* of the on-line algorithm for σ^*, we are updating the FSSAA associated with a particular σ only when the algorithm visits that σ or one of its immediate neighbour. Another approach can be to run the FSSAA for every σ in the search space for *Phase II* parallely and to rank the σs at each update epoch. At the kth iterate we will take σ_k^* (i.e., the optimum σ at the kth iterate) as the highest ranked (for which the associated cost is minimum at the kth iterate) σ at the kth update. It can be easily shown that due to w.p. 1 convergence of FSSAA, the sequence $\{\sigma_k^*, F_{\sigma_k^*}\}$ will also converge to $\{\sigma^*, EF_{\sigma_k^*}\}$ w.p. 1. A comparative convergence study of the two approaches will be interesting.

D) Our network model consists of homogeneous sources with a single network node. The model has to be extended to include more complex and practical models with non-homogeneous sources and multinode networks.

E) Simulations for more general and arbitrary source models has to be done. Also experiments with real video and audio traces for studying the effectiveness of our algorithms is a necessary work.

Bibliography

[1] A.A.Borokov. *Asymptotic Methods in Queueing Theory.* Wiley, 1984.

[2] Santosh P. Abraham. *Asynchronous Distributed Rate Control Algorithms for Best-Effort Sessions in Integrated Services Networks with Minimum Rate Guarantees* PhD Thesis. Indian Institute of Science, India, 1998.

[3] Santosh P. Abraham and Anurag Kumar. A new approach for asynchronous distributed elastic sessions in integrated packet networks. *IEEE Transactions on Networking*, submitted.

[4] Sigrun Andradottir. A method for discrete stochastic optimization. *Management Science*, 41:1946–1961, 1995.

[5] Sigrun Andradottir. *Chapter 9, "Simulation Optimization", Handbook of Simulation, Editor Jerry Banks*, volume 2. John Wiley and Sons, Inc., 1998.

[6] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system. *The Bell System Technical Journal*, 61(8):1871–1894, October 1982.

[7] Anwar I. Elwalid, Debasis Mitra, T.V.Lakshman, Daniel Heyman and Alan Weiss. Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing. *IEEE Journal on Selected Areas in Communications*, 13(6):1004–1016, August 1995.

[8] Alex Aussem. Call admission control in ATM networks with the random neural network. *IEEE International Conference on Neural Networks*, VI:2482–2487, June 1994.

[9] Arthur Berger. Performance analysis of a rate-control throttle where tokens and job queue. *IEEE Journal on Selected Areas in Communications*, 9(2):165–170, Feb 1991.

[10] Gagan L. Choudhury, David M. Lucantoni, and Ward Whitt. On the effective bandwidths for admission control in ATM networks. *Proc. 14th Int'l Teletraffic Congress (ITC'94)*, pages 411–420, 1994.

[11] C. Courcoubetis, G. Kesidis, A. Ridder, J. Walrand, and R. Weber. Admission control and routing in ATM networks using inferences from measured buffer occupancy. *IEEE Transactions on Communications*, 43:1778–1784, April 1995.

[12] G. de Veciana and J. Walrand. Effective bandwidths: Call admission, traffic policing & filtering for ATM networks. *Queueing Systems (QUESTA)*, 20:37–59, 1995.

[13] Gustavo de Veciana. Leaky buckets and optimal self-tuning rate control. In *GLOBECOM*, 1994.

[14] A. Dembo and O. Zeitouni. *Large deviations techniques and applications*. 2nd ed. Springer, Application of Mathematics, vol. 38, 1998.

[15] Yoan Shin Du-Hern Lee and Young-Han Kim. A neural network based ATM call admission controller for multiple service classes with different qos. *IEEE International Conference on Neural Networks*, IV:2142–2147, June 1996.

[16] Parijat Dube and Anurag Kumar. Measurement based selection of token buffer size for leaky bucket controlled sources: A simulation study. Technical report, Indian Institute of Science, November 1998.

[17] Parijat Dube and Anurag Kumar. An algorithm for on-line estimation of optimal token bucket depth for leaky bucket controlled sources. Technical report, Indian Institute of Science, September 1999.

[18] Anwar I. Elwalid and Debasis Mitra. Analysis and design of rate based congestion control of high speed networks, i: stochastic fluid models, access regulatiom. *Queueing Systems*, 9:29–64, 1991.

[19] Anwar I. Elwalid and Debasis 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.

[20] Anwar I. Elwalid, Debasis Mitra, and Robert H. 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.

[21] Michael C. Fu. Optimization via simualtion: A review. *Annals of Operations Research*, 53:199–247, 1994.

[22] Erol Gelenbe, Xiaowen Mang, and Raif Onvural. Diffusion based statistical call admission control in ATM. *Performance Evaluation*, 2(27):411–436, 1996.

[23] S.B. Gelfand and S.K. Mitter. Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications*, 62:49–62, 1989.

[24] R. J. Gibbens and P. J. Hunt. Effective bandwidths for the multi-type UAS channel. *Queueing Systems*, 9:17–28, 1991.

[25] R. J. Gibbens and F. P. Kelly. Measurement-based connection admission control. In J. Labetoulle and J. W. Roberts, editors, *ITC-14*, pages 879–888. Elsevier Science Publishers, 1994.

[26] R. J. Gibbens, F. P. Kelly, and P. B. Key. A decision-theoritic approach to call admission control in ATM networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1101–1115, August 1995.

[27] Matthais Grossglauser and David Tse. A framework for robust measurement-based admission control. *Proc. ACM SIGCOMM97*, September 1997.

[28] Roch Guerin, Hamid Ahmadi, and Mahmoud Naghshinch. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, 1991.

[29] Fabrice Guillemin, Catherine Rosenberg, and Josee Mignault. On characterizing an ATM source via sustainable cell rate traffic descriptor. In *INFOCOM*, 1995.

[30] Atsushi Hiramatsu. ATM communications network control by neural networks. *IEEE Transactions on Neural Networks*, 1(1):122–130, March 1990.

[31] Atsushi Hiramatsu. ATM call admission control using a neural network trained with a virtual output buffer method. *IEEE International Conference on Neural Networks*, VI:3611–3616, June 1994.

[32] Ivy Hsu and Jean Walrand. Dynamic bandwidth allocation for ATM switches. *Journal of Applied Probability*, pages 758–771, Sept 1996.

[33] Po L. Tien Jen M. Hah and Maria C. Yuang. Neural network based call admission control in ATM networks with heterogeneous arrivals. *Computer Communications*, 20:732–740, 1997.

[34] F. P. Kelly. On tariffs, policing and admission control for multiservice networks. *Operations Research Letters*, 15:1–9, 1994.

[35] S. Keshav. *An Engineering Approach to Computer Networking.* Addison Wesley, 1998.

[36] George Kesidis, Jean Walrand, and Cheng-Shang Chang. Effective bandwidths for multiclass markov fluids and other ATM sources. *IEEE/ACM Transactions on Networking*, 1(4):424–428, August 1993.

[37] Hans Kroner, Thomas Renger, and Ralf Knobling. Performance modelling of an adaptive cac strategy for ATM networks. In J. Labetoulle and J. W. Roberts, editors, *ITC-14*, pages 1077–1088. Elsevier Science Publishers, 1994.

[38] Anurag Kumar. Adaptive load control of the central processor in a distributed system with a star topology. *IEEE Transactions on Computers*, 38(11):1502–1512, November 1989.

[39] Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, 1978.

[40] Brian L. Mark and Gopalakrishnan Ramamurthy. Real-time estimation and dynamic renegotiation of UPC parameters for arbitrary traffic sources in ATM networks. *IEEE/ACM Transactions on Networking*, 6(6):811–827, December 1998.

[41] Natwar Modani. *Measurement Based Optimal Source Shaping and Connection Admission Control in ATM Networks* Masters' Thesis. Indian Institute of Science, India, 1998.

[42] Robert J.T. Morris and Behrokh Samadhi. Neural network control of communications systems. *IEEE Transactions on Neural Networks*, 5(4):639–650, July 1994.

[43] N. Ogino and Y. Wakahara. Application of neural network in ATM call admission control based on cell transfer state monitoring with dynamic threshold. *IEICE Transactions on Communications*, E78(B4):465–475, 1995.

[44] Nilesh R. Pastagia. *An Approach for Measurement Based Admission Control for End-to-End Delay Guarantees in Packet Networks* Masters' Thesis *(under preparation)*. Indian Institute of Science, India, 2000.

[45] Harry G. Perros and Khaled M. Elsayed. Call Admission Control Schemes: A Review. *IEEE Communications Magazine*, pages 82–91, November 1996.

[46] Francesco Lo Presti, Zhi-Li Zhang, Jim Kurose, and Don Towsley. Source time scale and optimal buffer/bandwidth trade-off for regulated traffic in an ATM node. In *INFOCOM*, 1997.

[47] Kiran M. Rege. Equivalent bandwidth and related admission criteria for ATM systems: A performance study. Technical report, AT&T Bell Laboratories, September 1993.

[48] S.Amarnath and Anurag Kumar. A new technique for link utilization estimation in packet data networks using SNMP variables. In *GLOBECOM*, 1997.

[49] Ness B. Shroff and Mischa Schwartz. Improved loss calculations at an ATM multiplexer. *IEEE/ACM Transactions on Networking*, 6(4):411–421, August 1998.

[50] Khosrow Sohraby. Heavy traffic multiplexing behavior of highly-bursty heterogeneous sources and their admission control in high-speed networks. In *GLOBECOM*, 1992.

[51] Khosrow Sohraby. On the theory on general on-off sources with applications in high-speed networks. In *INFOCOM*, 1993.

[52] D. Tse and M. Grossglauser. Measurement-based call admission control: Analysis and simulation. *Proc. IEEE Infocom '97*, April 1997.

[53] Qinglin Wang and S. Frost. Efficient estimation of cell blocking probability for ATM systems. *IEEE/ACM Transactions on Networking*, 4(2):230–235, April 1993.

[54] Paul P. White. RSVP and Integrated Services in the Internet: A Tutorial. *IEEE Communications Magazine*, pages 100–106, May 1997.

[55] Indra Widjaja and Franco Callegati. A call admission control for multiservice networks with declarable and enforceable traffic parameters. In J. Labetoulle and J. W. Roberts, editors, *ITC-14*, pages 213–222. Elsevier Science Publishers, 1994.

[56] D. Yan and H. Mukai. Stochastic discrete optimization. *SIAM Journal on Control and Optimization*, 30:594–612, 1992.

[57] R.S. Sreenivas Y.C.Ho and P. Vakili. Ordinal optimization of DEDS. *Discrete Event Dynamic Systems*, 2:61–88, 1992.

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