Satisfying End-to-End Quality of Service Requirements with End-to-End Performance Inference Technique

By

Benjamin Zhong Ming Feng

B.A. Sc, M.A. Sc

A thesis submitted to
The Faculty of Graduate Studies and Research
in partial fulfilment of
the degree requirements of
Doctor of Philosophy in Electrical Engineering

Ottawa-Carleton Institute for
Electrical and Computer Engineering

Department of Systems and Computer Engineering
Carleton University

Ottawa, Ontario, Canada

March 2009

Copyright ©

2009- Benjamin Zhong Ming Feng

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file   Votre référence
ISBN: 978-0-494-52053-6
Our file   Notre référence
ISBN: 978-0-494-52053-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

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

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■ ■ ■
Canada

Abstract

The proposed EPIT approach in this thesis is a hybrid of simulation and analytical methods. By treating a network as a black box, it scales well against complex network topologies. EPIT can predict the end-to-end QoS performance in a network under any traffic model (including LRD traffic models) that has a closed-form joint distribution function. In addition, EPIT has no constraints on the behaviour of the traffic itself.

The EPIT approach is effective, as it can project a wide range of probabilities of end-to-end overflow events accurately, with a fixed computation cost, in a network with multiple interconnecting nodes and multiple LRD input sources. Insights into the efficiency of EPIT are gained through the analysis on the efficiency of EPIT under a single-queue scenario. It is important to note that, even though this is a single-queue scenario, to the best of the author's knowledge, there has not been this type of analysis on efficiency in such scenario *with LRD input*. This is another contribution of this thesis. The analysis shows that, with a constant computation burden, the normalized variance of EPIT does not increase exponentially as the probability of overflow approaches zero asymptotically.

The resulting approach of Simulated Annealing (SA) integrated with EPIT (SA-EPIT) provides a traffic engineering solution, so that multiple end-to-end QoS requirements are satisfied while the network resource utilization is maximized. The value of this research is that, in fulfilling end-to-end QoS requirements while maximizing the network utilization, EPIT is an attractive solution to the deployment of applications such as QoS-aware VPN and Peer-to-Peer communications.

To my parents

Table of Contents

Table of Contents	3
List of Tables	6
List of Figures	7
Acronyms	9
Mathematical Notations	11
Chapter 1 Introduction	19
1.1 Problem	19
1.2 Solution	22
1.2.1 Contributions	22
1.2.2 Published Work	23
1.3 Application Examples	24
1.3.1 Virtual Private Network	25
1.3.2 Peer-to-Peer Network	25
Chapter 2 Background Knowledge	28
2.1 QoS Model	28
2.2 Overview of Importance Sampling and Importance Sampling Parameter Estimation	29
2.3 Introduction to Efficient Simulation Criteria	34
2.4 Fractional Gaussian Noise Random Process	37
Chapter 3 Related Work	43
3.1 Existing Analytical-based Analysis Approaches on QoS	43

3.1.1 Analytical Approaches Based On Short-range Dependent Traffic Models.....	44
3.1.2 Analytical Approach Based On Long-range Dependent Traffic Models	45
3.2 Existing Simulation-based Analysis Approaches on QoS	48
3.3 Existing IS Simulation Efficiency Analysis	50
Chapter 4 End-to-End Performance Inference Technique	54
4.1 EPIT Formulation	58
4.2 Advantages of EPIT	62
4.3 Simulated Annealing with EPIT	66
4.3.1 Heuristic Search Problem Formulation.....	67
4.3.2 Heuristic Search: Simulated Annealing.....	69
4.3.3 Simulated Annealing Parameters	71
Chapter 5 Analytical Efficiency Analysis	75
5.1 Change of Measure	76
5.2 IS Normalized Variance.....	81
5.2.1 Deriving IS Normalized Variance	83
5.2.2 Derived IS Normalized Variance Verification	87
5.2.3 IS Optimal Twist Amount.....	89
5.2.4 IS Simulation Behaviour under LRD input	94
5.3 IS Simulation Efficiency.....	100
5.4 EPIT Efficiency Analysis	107
5.4.1 Deriving EPIT Normalized Variance.....	107
5.4.2 Derived EPIT Normalized Variance Verification.....	110
5.4.3 Normalized Variance based Efficiency Analysis	112
Chapter 6 Simulation Experiments	115
6.1 Basic Network Model	115
6.2 Virtual Private Network.....	118
6.2.1 VPN Topology and Network Parameters	119

6.2.2 VPN Experiments	125
6.3 Peer-to-Peer Network.....	130
Chapter 7 Implementation	138
7.1 EPIT Implementation Architecture.....	140
7.2 End-to-end Loss Detection	142
7.3 Traffic Measurement.....	146
7.4 Hurst Parameter Measurement.....	151
7.5 Information Delivery Mechanism.....	153
7.6 Traffic Shaping Algorithms.....	157
7.6.1 Leaky Bucket	158
7.6.2 Token Bucket.....	159
Chapter 8 Conclusion	162
Chapter 9 Future Works	164
References	167

List of Tables

Table 6.1: NSFNET Network Routing Matrix Non-zero Entries	121
Table 6.2: NSFNET Network Parameters	123
Table 6.3: Network Link Parameters and Initial Link Utilization	125
Table 6.4: PE-PE Paths in Each Experiment, Their Initial Probability of Overflows, and their Overflow Probability Requirements	126
Table 6.5: EPIT Solution for VPN Experiments, Comprised of the Reduction Amounts for the Mean Rates of the 15 Traffic Sources	127
Table 6.6: New Overflow Probabilities after Reductions are Made to the Traffic Source Mean Rates	128
Table 6.7: New Link Utilizations for the SA-EPIT Solutions for VPN Experiments 1,2, and 3	129
Table 6.8: PE-to-PE Paths in Torrents 1, 2 and 3	133
Table 6.9: EPIT Solution for P2P Experiments, Comprised of the Reduction Amounts for the Mean Rates of the 15 Traffic Sources	133
Table 6.10: New Link Utilizations for the SA-EPIT Solutions of P2P Torrents 1,2, and 3	134
Table 6.11: Overflow Probabilities Before and After Reductions are Made	136
Table 7.1: Likelihood Ratio Table in Each Edge Node	147

List of Figures

Figure 2.1: Self-Similar Process and Poisson Process.....	38
Figure 3.1: Determining Norros Equation's Constant	45
Figure 4.1: Traditional End-to-end Path Performance Analysis.....	54
Figure 4.2: Multiple End-to-end Paths in an Arbitrary Multi-input Multi-node Network	55
Figure 4.3: Black Box View of the Network: Focusing on the End-to-end QoS of a Path Instead of the QoS of Individual Links in the Path.....	56
Figure 4.4: Multi-Source Multi-Queue Network	57
Figure 4.5: Black-Box view of a Multi-Source Multi-Queue Network.....	62
Figure 5.1: System Reformulation for the Change of Measure	77
Figure 5.2: Numerical Analysis of the Partial Autocorrelation Function and $\sigma^2_1(t)$	80
Figure 5.3: Analytical IS Normalized Variance and Simulation IS Normalized Variance	88
Figure 5.4: Simulation IS Normalized Variance and Simulation Replication Length	90
Figure 5.5: Analytical IS Optimal Twist Amount	92
Figure 5.6: Normalized Variance under Optimal Twist Amount and Non-optimal Twist Amount	93
Figure 5.7: IS Optimal Twist Amount vs. Effective Service Rate.....	95
Figure 5.8: IS Optimal Mean vs. Effective Service Rate.....	97
Figure 5.9: IS Optimal Twist Amount vs. Buffer Threshold Size	98
Figure 5.10: IS Optimal Twist Amount vs. Hurst Parameter	99
Figure 5.11: Analytical Probability and Simulation Probability Comparison.....	103
Figure 5.12: Analytical EPIT Normalized Variance and Simulation EPIT Normalized Variance	111

Figure 5.13: Analytical Probabilities of Overflow Based on Massoulie Equation and Simulation Probabilities of Overflow	112
Figure 6.1: VPN Topology for Experiment 1	119
Figure 6.2: The Core Provider Carrier Network Topology	120
Figure 6.3: Traces of the Three End-to-end Overflow Probabilities over the Course of the EPIT Optimization Process (Experiment 1)	126
Figure 6.4: Simulation Gain for VPN Experiment 1	129
Figure 6.5: P2P Torrent Configurations.....	131
Figure 6.6: Traces of the Aggregated Overflow Probability in Torrents, Over the Course of the SA-EPIT Optimization Process	135
Figure 6.7: P Trajectory for Torrent 1 as SA-EPIT Iterates Through Different Reduction Amounts to Bring P to TQ	136
Figure 7.1: EPIT Implementation Architecture	140
Figure 7.2: Taking Measurements Using a Jumping Window	149
Figure 7.3: Taking Measurements Using Sliding Window	150
Figure 7.4: Double Recursive Double Moving Window Measurement	150
Figure 7.5: Variance and Time Plot.....	152
Figure 7.6: SNMP Model for the EPIT Implementation	154
Figure 7.7: Managed Object	156
Figure 7.8: Leaky Bucket Implementation	158
Figure 7.9: Token Bucket Implementation	161

Acronyms

AS: Autonomous System

ATM: Asynchronous Transfer Mode

BGP: Border Gateway Protocol

CE: Customer Edge

CIM: Common Information Model

CSRC: Contributing SouRCe

DRDMW: Double Recursive Double Moving Window Measurement

FGN: Fractional Gaussian Noise

GCRA: Generic Cell Rate Algorithm

IP: Internet Protocol

IS: Importance Sampling

ISPE: Importance Sampling Parameter Estimation

LRD: Long Range Dependent traffic

MC: Monte Carlo

MIB: Management Information Base

MPLS: Multiprotocol Label Switching

NOC: Network Operations Center

NSFNET: National Science Foundation Network

P2P: Peer-to-peer

PDU: Protocol Data Units

PE: Provider Edge

QoS: Quality of Service

RTP: Real-time Transport Protocol

SLA: Service Level Agreements

SNMP: Simple Network Management Protocol

SRD: Short Range Dependent

TAP: Transport Access Point

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

WBEM: Web Based Enterprise Management

VPN: Virtual Private Network

Mathematical Notations

Math notations using the English alphabet

a^i : the amount of traffic arrived in time slot i .

A : a set containing samples of interest. It could be a set of sample-values for queue backlog such that the value will trigger an overflow; it could also be a set of sample traffic traces that will trigger an overflow.

b : buffer size threshold: if the queue backlog exceeds the buffer size threshold, a QoS related event would occur.

$b' := b^{2(1-H)}$, b' is used in the efficiency analysis

B_φ : an input to a system that depends on a rarity parameter φ

C_i : the cost of reducing one traffic unit per time unit from the mean rate of traffic stream i .

$C_{\max} := \max\{C_i : i = 1, \dots, S\}$

$C_{\min} := \min\{C_i : i = 1, \dots, S\}$.

$\vec{C}$: a covariance matrix that describes d -dimensional Gaussian vectors

C : a positive finite number

d^i : the amount of traffic departed at time slot i

D : assigned as the cost (penalty) for not fulfilling the QoS requirements in the objective function, during the simulated annealing heuristic search.

E : inner window size of Double Recursive Double Moving Windows algorithm

$F := \int \left(\frac{I_A(\vec{x}) f(\vec{x}, v)}{f(\vec{x}, v')} \right)^2 f(\vec{x}, v') d\vec{x}$, F is used in the efficiency analysis

$G_t := r(t-1) - \sum_{j=2}^t \phi_{t-1,j} r(t-j)$, use in generating the FGN process.

$g(M_k)$: a d -dimensional real valued function that converts a Markov chain into a real-value.

H : Hurst parameter, an indication of the burstiness of the LRD input process. In this thesis, it is the burstiness of the FGN traffic.

$I := -\lim_{\varphi \rightarrow \infty} \frac{1}{\varphi} \log(P_\varphi)$, used in the efficiency analysis

J : outer window size in Double Recursive Double Moving Windows algorithm

K : when estimating the Hurst parameter, a time series of length T is divided into K non-overlapping blocks.

$$L_{i,k,t} := \frac{\exp\left(\frac{-(x_{i,k}(t) - m_{i,k}(t))^2}{2\sigma^2_{(k)}(t)}\right)}{\exp\left(\frac{-(x_{i,k}(t) - m'_{i,k}(t))^2}{2\sigma^2_{(k)}(t)}\right)}$$

M : number of paths in the network whose end-to-end QoS must meet a target.

m^* : the twist amount. The difference between the value of the original statistical parameter v and the value of the biased statistical parameter v' .

m^*_{opt} : the optimal twist amount that minimizes the variance of an estimation.

M_k : a time homogenous Markov chain.

m : the time scale used to analyze LRD processes

N : number of replications; in finite-event horizon simulations, it also equals to the number of sample collected. The computation cost is often associated with N .

N_{MC} : Number of replications used for MC simulation

N_{EPIT} : Number of replications used for EPIT

P : the probability of some event occurring; in this thesis, the event is the event of buffer overflow on a single queue or the event of end-to-end buffer overflow on a path

$\hat{P}$: estimated value for P .

P^i : the overflow probability on the i -th node on an end-to-end path
 $\hat{P}_\varphi$: the estimation of a probability in which the rarity parameter is φ and as φ approaches infinity, P approaches zero
 Q_T : a random variable representing the queue backlog at time T .
 q_T : a sample value of random variable Q_T .
 r_i : the amount of reduction in traffic flow i .
 $\vec{r}$: the amounts of reduction in all ingress traffic flows in the network, $\vec{r} = [r_0, r_1, \dots, r_S]$
 $\vec{r}_k$: the amounts of reduction in all ingress traffic flows in the network on the k -th iteration of the simulated annealing search.
 $\vec{r}^*$: the amounts of reduction in all ingress traffic flows that satisfy all the QoS requirements and have the least amount of reduction cost in the objective function.
 $R := -\lim_{\varphi \rightarrow \infty} \frac{1}{\varphi} \log(F_\varphi)$, used in the efficiency analysis.
 S : number of traffic flows entering the network. In this thesis, it is the number of aggregated traffic flow entering the network at the network edge.
 $U_t := U_{t-1} - G_{t-1}^2 / U_{t-1}$, used to generate a FGN process
 X : a random variable
 x : a sample value of the random variable X
 $\vec{X}$: a random process. In this thesis, a random process is used to model a traffic flow.
 $\vec{X}_k$: a random process. In this thesis, a random process is used to model the k -th traffic flow entering the network.
 $\vec{x}$: a sample trace or a realization of a random process $\vec{X}$. In this thesis, it is a sample traffic trace. $\vec{x} = x(1), x(2), \dots, x(T)$
 $t_i := \lfloor T / K \rfloor (i-1) + 1$, used in the Double Recursive Double Moving Windows algorithm.
 T : length of a replication.
 T_s : when $t=T_s$, the probability of W_t greater than b is maximized.

T_{slot} : duration of a time slot

TQ : QoS target, a probability based QoS metric target.

TQ_i : the QoS target for end-to-end path i .

v : the original statistical parameter of a random process (in this thesis, it is the mean)

v' : the biased statistical parameter of a random process (in this thesis, it is the mean)

v_i : the mean rate of ingress traffic flow i . In this thesis, it is the aggregated traffic flow entering the core provider network through edge node i .

v'_i : the original/initial mean rate of ingress traffic flow i . In this thesis, it is the aggregated traffic flow entering the core provider network through edge node i .

v'_{opt} : the optimal mean for the biased distribution, based on the optimal twist amount

$$m^*_{opt}.$$

$\vec{v} = [v_1, v_2, \dots, v_s]$, the ingress traffic levels in a network.

$\vec{v}' = [v'_1, v'_2, \dots, v'_s]$, the original ingress traffic levels in a network.

$\vec{W}$: a random process call the work process,

$\vec{w}$: a sample trace of a work process $\vec{W}$.

W_t : a random variable representing the stochastic nature of a work process $\vec{W}$ at time slot t .

w_t : the value of sample trace $\vec{w}$ at time slot t . It is equal to the sum of all $x(i)$ minus μ , for $i=1,2,\dots,t$.

W : measurement window size

$\vec{Y}_k$: an i.i.d sequence of d -dimensional random vectors with a univariate distribution.

Math notations using Greek alphabet

α : the probability that a solution generated in an iteration of simulated annealing heuristic search process is accepted.

δ : computation time/cost for $L_{i,k,t}$.

ς : a quantity used in the bound provided by Norros for the estimation of overflow probability, it is independent of μ_e and q_T , see [30][31] for details.

$\theta_{i,j}$: the percentage of traffic received by node i that is sent to node j

Θ : is defined as a step size such that all reduction amount from the original traffic level must be an integer multiple of this quantity.

$\vec{\Lambda}$: network internal traffic flow matrix.

μ : the service rate of a single node.

μ_e : the effective service rate of a single node in which the service rate minuses the input traffic rate.

$\gamma := \frac{1-H}{H}(2H-1)$, a quantity in the formula that describes the overflow probability in a single node queue with a LRD input.

σ^2 : variance

$\sigma_\varphi^2 := \sigma^2 / \varphi^2$ where φ is a rarity parameter.

$\tau := \frac{\mu_e^{2H}}{2K^2(H)}$, used in the efficiency analysis.

$\tau_{m^*} := \frac{(\mu_e - m^*)^{2H}}{2K(H)^2}$, used in the efficiency analysis.

$\tau_{+m^*} := \frac{(\mu_e + m^*)^{2H}}{2K(H)^2}$, used in the efficiency analysis.

$\tau_{-m^*} = \frac{(2(\mu_e^{2H}) - (\mu_e - m^*)^{2H})}{2K^2(H)}$

$\tau_{*m^*} = \frac{(2((\mu_e + m^*)^{2H}) - \mu_e^{2H})}{2K^2(H)}$

$\tau_{1/m^*} := \frac{1}{2K^2(H)} \left(\frac{\mu_e^2}{\mu_e - m^*} \right)$, used in the efficiency analysis.

$\tau_{+1/m^*} := \frac{1}{2K^2(H)} \left(\frac{(\mu_e + m^*)^2}{\mu_e} \right)$, used in the efficiency analysis.

$\Gamma := \frac{K(H)^2}{b^{2(1-H)}}$, used in the efficiency analysis.

$\phi_{k,t}$: the t -th coefficient in an autoregressive process of order k .

$\phi_{k,k}$: partial autocorrelation function, regarded as a function of the lag k .

φ : a rarity parameter. As a rarity parameter approach infinity, a probability approaches zero.

$\psi(t) := \sum_{j=1}^t \phi_{t,j}$, used to describe an FGN process.

Ψ : a very complex function of H , see [33] for details.

Ω : a large number.

Mathematical functions named using English alphabet:

$$E_{f(v)}(g(x)) := \int g(x) f(x, v) dx$$

$$E_{f(v)}(g(\vec{x})) := \int g(\vec{x}) f(\vec{x}, v) d\vec{x}$$

$I_A(\vec{x})$: an indicator function. If sample trace $\vec{x}$ is in the set of sample traces in set A , then the indicator function will equal to one, and zero otherwise.

$I_A(q_T)$: an indicator function. If the sample value (q_T) is in set A , then the indicator function will equal to one, and zero otherwise.

$I_{A,i}$: an indicator function to monitor the event A occurring on end-to-end path i in the network.

$I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S})$: if traffic traces $\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}$ will trigger an end-to-end event in the network, then $I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}) = 1$ and zero otherwise.

$f(\vec{x}, v)$: the original probability distribution function of the input random process $\vec{X}$ with the original statistical parameter v (in this thesis, it is the mean).

$f(\vec{x}, v')$: the biased probability distribution function of the input random process $\vec{X}$ with the biased statistical parameter v' (in this thesis, it is the mean).

$f(x(1), x(2), \dots, x(T), v)$: same as $f(\vec{x}, v)$, where $\vec{x} = x(1), x(2), \dots, x(T)$

$f(x(1), x(2), \dots, x(T), v')$: same as $f(\vec{x}, v')$, where $\vec{x} = x(1), x(2), \dots, x(T)$

$K(H) := H^H (1 - H)^{1-H}$

$L(\vec{x}_i, v, v_0)$: likelihood ratio of traffic trace of random process $\vec{X}$ on the i -th replication.

$L(\vec{x}_{i,1}, v_1, v_{0_1}, \vec{x}_{i,2}, v_2, v_{0_2}, \dots, \vec{x}_{i,S}, v_S, v_{0_S})$: the likelihood ratio of random processes

$\vec{X}_1, \vec{X}_2, \dots, \vec{X}_S$ on the i -th replication.

$m(t)$: the conditional(marginal) mean of a FGN random process at time t .

$m_{i,k}(t)$: the conditional mean of $\vec{X}_k$ on the i -th iteration at time t .

$P\{\}$: the probability of the event in the curly bracket occurring.

$P(\vec{v})$: the QoS (described as a probability of QoS events in the network) given the traffic levels of the ingress traffic flows, $\vec{v} = [v_1, v_2, \dots, v_S]$.

$P_i(\vec{v})$: the QoS (described as probability of QoS events in the network) on path i given the traffic levels of the ingress traffic flows, $\vec{v} = [v_1, v_2, \dots, v_S]$

$p(q_T)$: the probability density function of the random variable queue length (Q_T) under the original input process distribution $f(\vec{x}, v)$.

$P'(q_T)$: the probability density function of the random variable queue length (Q_T) under the biased input process distribution $f(\vec{x}, v')$.

$r(t)$: autocorrelation function.

$R(t_i, m) := \max\{0, W(t_i, 1), \dots, W(t_i, m)\} - \min\{0, W(t_i, 1), \dots, W(t_i, m)\}$. This is used in the R/S analysis.

$s(q_T)$: any function in terms of the sample queue backlog value q_T .

$S(t_i, m)$: the sample variance of $x(t_i), \dots, x(t_i + m - 1)$. This is used in the R/S analysis.

$W(t_i, k) := \sum_{j=1}^k x(t_i + j - 1) - k \left(\frac{1}{m} \sum_{j=1}^k x(t_i + j - 1) \right)$. This is used in the R/S analysis.

$x(i)$: the value of sample trace $\vec{x}$ at time slot i . In this thesis, it is the amount of traffic received or sent in time slot i .

$x_{i,k}(t)$: the sample value of a sample trace of $\vec{X}_k$ on the i -th iteration at time t .

$x^{(m)}(k) := \frac{1}{m} (x(km - m + 1) + \dots + x(km))$, used to analyze the time series x in different time scales of m .

Mathematical functions named using Greek alphabet

$\beta(k)$: acts as a decision control-value for the k -th iteration in the simulated annealing process. It controls the α in the heuristic search process.

$\Phi(\vec{r}_k)$: the total cost of the reduction amounts $\vec{r}_k$.

$\sigma^2(t)$: the conditional (marginal) variance of a FGN random process at time t .

$\sigma^2_1(t)$: the conditional (marginal) variance of a FGN random process, with a variance of one, at time t .

$\sigma^2_{(k)}(t)$: the conditional (marginal) variance of a FGN random process, at time t , for the k -th input FGN process.

$\sigma^2[\cdot]$: the variance of the estimated quantity in the bracket.

$\overline{\sigma^2}[\cdot]$: the normalized variance of the estimated quantity in the bracket.

$\varpi(T)$: a very complex equation used to simplify an expression, for more details, see equation (58)

Miscellaneous Notation

$\underset{b \rightarrow \infty}{=}$: the Left Hand Side of the operator equals the Right Hand Side of the operator as b approaches infinity.

Chapter 1 Introduction

Due to the increasing demand for transmitting video/voice and other real time applications over the Internet, end-to-end Quality of Service (QoS) guarantees from the service provider are becoming crucial.

1.1 Problem

Since network QoS generally degrades due to network congestion in times of high traffic volume, the problem of interest is how to achieve end-to-end QoS in the provider core network via traffic engineering.

Traffic engineering, in the context of this thesis, means that only the ingress traffic flows is controlled while the network topology, network node service rates, network node buffer sizes, are all fixed.

To over-provision a network, one can purchase more bandwidth and upgrade network equipment to drastically increase the service rates and buffer sizes of network nodes. To over-provision is to increase the network capacity to handle the worst-case traffic load for delivering a guaranteed QoS. However, the cost will be high and the resource utilization will be low during normal operation periods. It is more profitable and more efficient to provision the network accurately, without purchasing addition equipment and bandwidth.

The problem that this thesis attacks can be divided into two problem statements: 1) how to predict the QoS of an end-to-end network path accurately, given the ingress traffic level at the network edge, and; 2) how to determine the precise level of traffic so the QoS target is met.

Given the dynamic nature and the diverse nature of the traffic carried by the Internet today, the solution to be applied to the core network of a service provider has to be accurate, scalable, flexible, efficient and implementable, and has to maximize the network resource utilization:

- Accurate: The variance of the estimated QoS quantity, predicted using a solution method, must be small so the prediction is reliable and not “noisy”.
- Scalable: The solution must scale well against complex network topologies. This means that as the number of nodes and the number of interconnecting links increase, the approach has to remain tractable.
- Flexible: The solution must be able to handle many different types of traffic models while imposing no condition on the behaviour of the traffic. While there is no condition on the *behaviour* of the traffic, it is reasonable to assume that the traffic model has a closed-form function to describe the traffic.
- Efficient: The solution must be able to project accurately a wide range of values for the QoS metric of interest while the computation cost of the solution remains constant or does not increase exponentially.
- Implementable: Existing technologies can be utilized to realize the solution.

- Maximize utilization: This is mainly a requirement for the solution to problem statement two: how to determine the amount traffic should be allowed into the network so that QoS requirement can be satisfied. Clearly, decreasing the amount of traffic entering the network will improve the QoS in the network. Yet, at the same time, network resources may be wasted. Thus, the solution must maximize the network utilization while meeting the QoS requirement.

Since the problem deals with end-to-end QoS, the solution naturally has to work in the multi-queue context. Lastly, since much of the literature has shown that Internet traffic exhibits Long-Range Dependency (LRD) characteristics, the solution must also be able to handle LRD traffic model.

Existing analytical approaches for quantifying the QoS in a network either work only with the traditional traffic model (also called the Short-Range Dependent [SRD] model), are suitable only in single-queue scenarios, quickly become intractable as the number of nodes in the network increases, or impose restrictions on the traffic behaviour (e.g. burstiness restriction). A major difficulty with the analytical analysis is the strong correlation within the LRD processes.

Existing simulation approaches that analyze the QoS in a network, such as the direct simulation approach (also called the Monte Carlo simulation) are inefficient because of the computational cost required for an acceptable variance of the estimation. The computational cost approaches infinity as the quantity being estimated approaches zero. Some work has been done to reduce the computation cost (e.g. using Importance

Sampling [IS]), but it was done for the single-queue simulation. More details are provided in Chapter 2.

1.2 Solution

The proposed End-to-end Performance Inference Technique (EPIT) in this thesis is a solution to the first problem statement. Because EPIT is a hybrid of simulation analysis and analytical analysis, EPIT overcame many of the obstacles faced by the existing analytical and simulation-based approaches.

The proposed Simulated Annealing coupled EPIT (SA-EPIT) approach in this thesis, is a solution to the second problem statement: how to determine the amount of ingress traffic that should be allowed into the network.

1.2.1 Contributions

The contributions of this thesis are listed as the following:

1. Proposed an Importance Sampling Parameter Estimation(ISPE) [1] based end-to-end traffic engineering architecture: EPIT. While there has been much work done on the topic of traffic engineering, no work has used the ISPE in traffic engineering in multi-queue scenarios with LRD traffic model. By employing ISPE in traffic engineering, EPIT is accurate, scalable, and flexible.
2. Based on a unique property of the Fractional Gaussian Noise (FGN) distribution, an innovative change of measure is used for the analytical analysis of EPIT and Importance Sampling. This change of measure has never been done with FGN.

3. Derived the optimal twist amount of IS simulation in a single node with FGN input and constant service rate. Similar work has been done under traditional input traffic model, but not under LRD traffic model (FGN is LRD).
4. Proved that IS simulation is efficient in a single node with FGN input and constant service rate. Similar work has been done under traditional input traffic model, but not under LRD traffic model (FGN is LRD).
5. Showed that the normalized variance of EPIT, with respect to the twist amount m^* , in a single node with FGN input, grows as $O(m^{*2H})$ while the overflow probability in a single node, with LRD input, decreases as $O(m^{*1-2H} \exp(-m^{*2H}))$. This in turn shows that EPIT remains efficient for a long range of projection.
6. Combined simulated-annealing search algorithm with likelihood ratio, SA-EPIT, to enable black-box-based traffic engineering.
7. Proposed an implementation to realize the proposed EPIT approach.

1.2.2 Published Work

The following are the publications of the author, related to the work of this thesis:

- B. Feng, C. Huang, M. Devetsikiotis "FISTE: A Black Box Approach for End-to-End QoS Management" ACM Transactions on Modeling and Computer Simulation (TOMACS) 2007 (accepted).
- B. Feng, C. Huang, M. Devetsikiotis "An End-to-End Performance Inference Technique for Peer-to-Peer Networks", IEEE Globecom'08, New Orleans, Dec. 2008. Pages(s) 1463-1467

- B. Feng, C. Huang, M. Devetsikiotis, and P. Rabinovitch, "Provisioning End-to-End Quality of Service with Fast Importance Sampling Based Traffic Engineering" In Proc. of 19th International Teletraffic Congress, Beijing, Aug. 2005. Page(s) 839 - 848.
- B. Feng, C. Huang, M. Devetsikiotis, and Y. Champoux, "Mapping Bandwidth to Quality of Service: An Importance Sampling Based Traffic Engineering Approach," in Proceedings of IEEE ICC 2005, Seoul, May 2005, Page(s):110 – 114.
- B. Feng, C. Huang, M. Devetsikiotis, and Y. Champoux, "Network Performance Optimization using Importance Sampling Theory," in Proceedings of IEEE Globecom'04, Dallas, Nov. 2004, Page(s):327 – 330.

To explain further the importance of answering the question of how to achieve end-to-end QoS in a network, some real network applications are presented next.

1.3 Application Examples

While the proposed approach in this thesis is applicable to many types of networks, this thesis has chosen overlay networks as application examples for the proposed approach, due to the popularity of overlay networks such as VPN and P2P networks. An overlay network is a virtual network built on top of another network, usually an IP network. Nodes in the overlay can be treated as being connected by virtual or logical links, each of which corresponds to an end-to-end path. Each path may go through many physical links in the underlying network. All the hosts in a Peer-to-Peer (P2P) network form an overlay network and exchange information through the overlay. Similar to P2P networks, a

Virtual Private Network (VPN) service provider provides virtual point-to-point links between customer sites for the overlay VPNs. In other words, the forwarding path between sites is not implemented in a hop-by-hop manner, but rather, the underlying network is used as a “cut-through” to other customer sites on the other end of the network.

1.3.1 Virtual Private Network

VPN customers usually have MPLS (Multi Protocol Label Switching [2]) tunnels dedicated to them. If the SLAs for those MPLS tunnels do not include any statistical QoS metrics (e.g. packet loss probability), the end-to-end tunnel/connection has to be greatly over-provisioned to deliver deterministic performance guarantees for the user traffic that is stochastic in nature.

Over-provisioning means more bandwidth allocated or purchased, which in turn means higher cost. New VPN techniques, such as the RANGE model [3][4], typically allow VPN users to explore statistical multiplexing within the core network. Provisioning of statistical QoS guarantees can lead to a more efficient utilization (high utilization) of network resources than the worst-case-based QoS provisioning.

1.3.2 Peer-to-Peer Network

For voice/video applications that are based on Peer-to-Peer (P2P) models, ensuring the end-to-end Quality of Service (QoS) is crucial, especially if users are paying fees.

While many free P2P applications settle for the best effort service model, other big P2P players do pay attention to their QoS. For example, the popular real-time voice chat

software Skype is based on the P2P model. Clearly, the end-to-end QoS is extremely important for audio conversations, especially since Skype charges a fee for calls to landlines and cell phones (while calls over the Internet are free). Thus, Skype does require QoS support from the underlying provider network by leasing the correct amounts of bandwidths to link their distributed super nodes [5]. Skype can always lease more bandwidth from the provider (over-provisioning) to ensure QoS, but that leads to higher financial cost. There is currently much research interest in delivering QoS in P2P networks [6] [7].

The voice quality of Skype usually deteriorates during peak hours caused by packet losses or long latencies due to high traffic volume. As a result, it is useful to have an approach that can predict the QoS in the network with respect to the ingress traffic levels.

Again, over-provisioning the end-to-end paths that connect the super nodes in the Skype system can deliver guaranteed network performance. However, because many applications (voice/video) can typically tolerate the loss or delay of a small fraction of the traffic, provisioning of statistical QoS guarantees can lead to a more efficient utilization of network resources than worst-case QoS provisioning. Over-provisioning often results in more bandwidth and/or equipment purchase that lead to higher cost and less revenue. This thesis focuses on the ingress-traffic control aspect.

Clearly, less traffic in the network always leads to less congestion and delay, yet it can also lead to less revenue: each refused voice call translates to a loss in fees. Thus, the goal is again to achieve minimum traffic reduction while satisfying QoS requirement(s), like in the VPN case.

The next chapters shall provide some background information to help explain EPIT, efficiency analysis, the QoS model, and the traffic model used in this thesis. This background information will also help clarify the related work in this field in Chapter 3.

Chapter 2 Background Knowledge

This chapter provides the background knowledge for the material presented in this thesis. It defines the QoS model used in this thesis and the other materials that are intended for those not familiar with the concept of Importance Sampling simulation, simulation efficiency, and the Fractional Gaussian Noise random process used to model traffic in this thesis.

2.1 QoS Model

The QoS in a network can be measured from stochastic events in the network, such as packet loss events, time-out events, etc. For example, packet loss probability is an important QoS metric. In a congested network, end-to-end packet loss is mainly caused by excessive delay or buffer overflow in one of the nodes (routers and switches) along the end-to-end path. In both cases, the cause is that the queue length (backlog) has grown too large: a queue length exceeding a threshold can lead either to buffer overflow or long queue waiting time resulting in time out. Thus, the term “probability of packet loss” is sometimes replaced with the term “probability of overflow” in this thesis. The term “probability of overflow” may seem more descriptive when discussing queue backlog exceeding a threshold. Another popular QoS metric is jitter. Jitter is essentially the variations in the delay of the arriving traffic. There is usually a threshold where the jitter

becomes intolerable for certain real-time applications. This will be discussed in detail in section 4.1.

Most SLAs between the service provider and the customer specify the QoS in terms of time-average measures (e.g. Mbyte/second, packets/second,..). Those measures can still be described by events: There is usually a mechanism either on the provider side or on the customer side to measure and calculate the time-average based QoS performance. SLA usually specifies a threshold level that should not be crossed. An event can be defined as the instance in which the particular threshold is violated by the detected time-average measure.

The end-to-end overflow probability on a path is the probability that the queue length of *any one* of the node, along the path, exceeding a threshold. The probability of overflow is used only as an example in this thesis. As it will be demonstrated in Chapter 4, any QoS metric, that can be defined by or measured by probability based on random events in the network, can be predicted by EPIT.

2.2 Overview of Importance Sampling and Importance Sampling Parameter Estimation

EPIT is based on the concept of Importance Sampling (IS). Consider a random variable X and let $f(x, \nu)$ be the true distribution of X , where ν is a statistical parameter of the distribution such as the mean or variance. If one wishes to estimate $P = \Pr\{X \in A\}$, P can be estimated via direct simulation (also known as the Monte Carlo method):

$$\hat{P} = \sum_{i=1}^N I_A(x_i) / N \quad (1)$$

where x_i is an independent sample of X and $I_A(\cdot)$ is an indicator function that equals one if x_i is in the set A .

Importance Sampling translates the true probability density $f(x, v)$ to a simulation/biased distribution $f(x, v')$ in order to generate more events and weights these events afterwards with a likelihood ratio:

$$L(x, v, v') = \frac{f(x, v)}{f(x, v')} \quad (2)$$

It should be noted that IS does not impose any restriction on the stochastic behaviour of the distribution of the random variable X . As long as X has a closed-form probability density function $f(x, v)$ and a likelihood ratio can be calculated, IS should work. (The likelihood ratio must be supported by a condition that if $f(x, v) \neq 0$ then $f(x, v') \neq 0$.)

Intuitively, $f(x, v')$ will make the event $X \in A$ occur more frequently, thus, N does not have to be large for an accurate estimation of P , if P is small. Mathematically:

$$P = \int I_A(x) \frac{f(x, v)}{f(x, v')} f(x, v') dx := E_{f(v')} (I_A(x) L(x, v, v')) \quad (3)$$

The variance of the estimator is

$$\sigma^2 \left[\hat{P} \right] = \frac{1}{N} \left(\int \left(\frac{I_A(x) f(x, v)}{f(x, v')} \right)^2 f(x, v') dx - P^2 \right) \quad (4)$$

Equation (3) can be implemented as

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N I_A(x_i) L(x_i, v, v') \quad (5)$$

where x_i follows the distribution of $f(x, v')$ instead of $f(x, v)$.

If instead of a random variable X , a random process $\vec{X}$ is employed. Equations (3) to (5) become the following, respectively:

$$P = \int I_A(\vec{x}) \frac{f(\vec{x}, v)}{f(\vec{x}, v')} f(\vec{x}, v') d\vec{x} = E_{f(v')} (I_A(\vec{x}) L(\vec{x}, v, v')) \quad (6)$$

$$\sigma^2 \left[\hat{P} \right] = \frac{1}{N} \left(\int \left(\frac{I_A(\vec{x}) f(\vec{x}, v)}{f(\vec{x}, v')} \right)^2 f(\vec{x}, v') d\vec{x} - P^2 \right) \quad (7)$$

$$\hat{P} = \frac{1}{N} \sum_{i=1}^N I_A(\vec{x}_i) L(\vec{x}_i, v, v') \quad (8)$$

where $\vec{x}_i$ is an independent sample trace (sample realization) of the random process $\vec{X}$ and v remains a statistical parameter (such as the mean or variance) of the distribution of $\vec{X}$.

Let

$$F = \int \left(\frac{I_A(\vec{x}) f(\vec{x}, v)}{f(\vec{x}, v')} \right)^2 f(\vec{x}, v') d\vec{x} = E_{f(v')} (I_A(\vec{X}) L(\vec{X}, v, v')^2) \quad (9)$$

where $L(\vec{x}, v, v')$ is still the likelihood ratio and equals to $\frac{f(\vec{x}, v)}{f(\vec{x}, v')}$. Now, Equation (7) can

be rewritten as

$$\sigma^2 \left[\hat{P} \right] = \frac{1}{N} (F - P^2) \quad (10)$$

Thus, the impact of the choice-of biased distribution is completely represented by F in Equation (10).

While it is theoretically possible to reduce $\sigma^2 \left[\hat{P} \right]$ to zero, it is impossible to implement, since that would require the knowledge of P , which is the value we are trying to measure [8].

A large variance means a large confidence interval, which in turn implies that the estimated result $\hat{P}$ is inaccurate, unreliable, or “noisy”. The question is how large of a variance is large? A variance of 10^{-3} may seem small, but if the probability being estimated is in the order of 10^{-6} , the variance of 10^{-3} is actually very large.

To remedy this situation, a *normalized variance* is used. It is the variance of the estimated probability from the simulation, divided by that probability squared:

$$\overline{\sigma^2}[\hat{P}] = \frac{\int \left(\frac{I_A(\vec{x}) f(\vec{x}, v)}{f(\vec{x}, v')} \right)^2 f(\vec{x}, v') d\vec{x} - P^2}{P^2} = \frac{1}{N} \left(\frac{F}{P^2} - 1 \right) \quad (11)$$

By combining the variance and the probability, one can examine the underlying characteristic of small variances. Thus, this thesis is more focused on the normalized variance rather than the variance.

N in Equations (4), (5), (7), and (8) represents the number of sample values or sample traces to be collected. While the larger the N is, the smaller the variance, but it will also be more time consuming. This leads to the efficiency discussion in the next section.

This thesis uses Mean-value Modification (MM) biasing, in which v is the mean of the distribution $f(\vec{x}, v)$. The amount of change is defined as the twist amount, m^* , in which $|v' - v| = m^*$. It will be come clear in later sections that MM biasing fits in perfectly with modeling the traffic flows as random processes and using v to represent the mean rate of the traffic flow.

After an overview of IS, ISPE (Importance Sampling Parameter Estimation), shall be explained. The best way to explain ISPE is to contrast its working with that of IS.

To summarize IS, in order to find P with respect to a fixed v , a biased v' is used in a simulation run to find P with reduced simulation time. One can vary v' , so the variance given in Equation (7) is minimized. However, instead of fixing v and varying v' , one can fix v' and vary v . Thus, P can be predicted as a function of the parameter v using Equation (8). In summary, based on one simulation run under $v=v'$, $P(v)$ can be predicted for any other value of v . This is essential Importance Sampling Parameter Estimation. To explain this intuitively, see Equation (8): N is number of samples collect during the simulation run so it is fixed. The result of the indicator function I_A is also recorded during the simulation run so it is fixed. The value of v' is fixed in the likelihood ratio. Thus, the only variable left is v . Therefore, $\hat{P}$ is a function of v .

From an application perspective, IS is used to estimate a fixed probability while ISPE tries to estimate the probability as a dynamic quantity with respect to some input parameter.

IS and ISPE both have a critical issue: How to assess the efficiency of the approach. Equation (7) clearly shows that as P becomes smaller, the variance becomes relatively larger. The question is whether one can find an efficient simulation distribution (biasing distribution) to compensate for a very small P so the variance does not increase exponentially. If ISPE is not efficient, one must utilize a large number of samples (large N , hence huge computation cost and long simulation running time) to bring down the variance, which is clearly inefficient.

2.3 Introduction to Efficient Simulation Criteria

The efficiency criteria deal with the case when the probability P becoming very small. This is important to consider because it is anticipated that the future Internet will likely have much better QoS (for example, lower packet loss probability) so the requirements of mission-critical applications can be achieved. The question then becomes that as the probability becomes smaller, must N increase in order to keep the variance of its estimation down, which in turn keeps the estimation reliable?

Direct simulation, also known as the Monte Carlo (MC) simulation [1], described in equation (1), can estimate the event occurrence probability in a system. However, in order to obtain a confidence interval of a desired accuracy for a rare event probability, the MC simulation time tends to infinity as the rare event probability asymptotically approaches zero. Importance Sampling (IS) is able to reduce the variance of the estimation, thus reducing the required simulation time, by modifying the stochastic distribution of the system input process so that events occur more frequently. Those events are re-weighted afterwards for the probability calculation. (This was discussed in detail in the previous section).

With longer simulation time, more samples can be collected to reduce the variance and the normalized variance; yet, the increased simulation effort means the Monte Carlo simulation approach will cease to be efficient eventually, as the event probability becomes smaller. In contrast, the normalized variance of an *asymptotically efficient* simulation approach will remain constant. As a result, the simulation effort (described in N) can remain constant.

To analyze the simulation efficiency of IS, the existing definition of the simulation efficiency needs to be explained first.

A rarity parameter, φ , is a parameter that influences a probability P_φ such that: as $\varphi \rightarrow \infty$, $P_\varphi \rightarrow 0$.

[9] states that a probability of large deviation decays exponentially as the rarity parameter φ tends to ∞ . [10] has shown cases where P_φ would decay slower than exponential, if P_φ is the probability of data loss (due to overflow) in a queue, fed by LRD input, and φ is the buffer size threshold (note that [10] is focused on overflow probability and not simulation efficiency).

Since P_φ vanishes at a rate with respect to φ , the simulation effort, in terms of N , has to increase to maintain a certain estimator precision with Monte Carlo simulation.

Yet, if the F term (denoted as F_φ in Equation [11]), which represents the impact the chosen IS simulation distribution has in the variance, can decrease as fast or close to that of P , with respect to the rarity parameter φ , the variance will not increase significantly. As a result, no excessive simulation effort will be required, and N can stay constant.

Mathematically, if the following limits exist

$$\lim_{\varphi \rightarrow \infty} \frac{1}{\varphi} \log(F_\varphi) = -R \quad (12)$$

$$\lim_{\varphi \rightarrow \infty} \frac{1}{\varphi} \log(P_\varphi) = -I \quad (13)$$

then the simulation method is called asymptotically efficient if $R=2I$.

If Equation (9) is substituted into $R=2I$ and the outcome is expanded:

$$\begin{aligned}
\lim_{\varphi \rightarrow \infty} \frac{1}{\varphi} \log \left(E_{f(v')} \left(I_A(\vec{X}) L(\vec{X}, v, v')^2 \right) \right) &= 2 \frac{1}{\varphi} \log(P_\varphi) \\
\lim_{\varphi \rightarrow \infty} \frac{\log \left(E_{f(v')} \left(I_A(\vec{X}) L(\vec{X}, v, v')^2 \right) \right)}{\log(P_\varphi)} &= 2
\end{aligned} \tag{14}$$

Intuitively, I is the logarithmic decay rate of the probability P_φ with respect to the rarity parameter φ . F_φ is the logarithmic decay rate of the influence of the IS simulation of the chosen biasing distribution in the variance. The idea is for the biasing distribution to “compensate” for the decreasing P_φ , thus keeping the variance low. In summary, the IS biasing is asymptotically efficient if the simulation effort required to obtain a specified estimator precision does not grow with φ .

As stated previously, this simulation efficiency analysis has been performed on IS. However, none of those studies was done under LRD inputs. This thesis uses a LRD random process, the Fractional Gaussian Noise (FGN) process, as the input process to a system where IS is used. While this is not for the sake of explaining EPIT, this analysis is done as a contribution to this field in order to fill in the gap. More information on the FGN process will be provided in the next section.

The previously-discussed simulation efficiency criteria is for IS, but such criteria does not exist for ISPE, which EPIT is based on. However, one can show that EPIT is efficient by following the spirit of the criteria for IS: with the computational cost (N) fixed, EPIT is efficient if the normalized variance of EPIT grows at a rate that is moderate compared to the rate of decrease on the probability, thus the variance does not increase exponentially.

A closed-form analysis of a multi-queue network under FGN is very difficult because non-Gaussian distributions lose their original statistical distribution at the output of a queue. This thesis settled for proving the efficiency of EPIT in a single FGN source – a single node network. Yet, to the best of the author’s knowledge, this is still something that has not been done in this field. The lack of such analysis is due to the strong correlation within the LRD processes that makes the analysis difficult. This thesis is able to overcome this obstacle with an innovative change of measure technique.

The motivation for the thesis to focus on LRD input model is that real Internet traffic traces have shown LRD properties [11][12].

2.4 Fractional Gaussian Noise Random Process

The input traffic model used in this thesis is a Self-similar random process with LRD characteristics called the Fractional Gaussian Noise (FGN) process. It is important to note that this thesis is focused on the core network where user traffic has been aggregated. As stated by the central limit theorem, the aggregation of a large number of independent flows will exhibit a Gaussian property; therefore, the Fractional Gaussian Noise process is a good model to use in this thesis.

Before the Fraction Gaussian Noise (FGN) [13][14][15][16] random process is explained, the concept of self-similarity and Long-range Dependency need to be explained first.

A random process exhibits self-similarity characteristics if it has strong correlations across multiple time scales.

Let $x(t)$ be a time series of $t=1,2,3,\dots$. Create a number of new time series by averaging the original series with non-overlapping blocks of size m .

$$x^{(m)}(k) = \frac{1}{m} (x(km - m + 1) + \dots + x(km)) \quad (15)$$

For $m=1,2,3$, etc, the series $\vec{x}^{(m)} := x^{(m)}(1), x^{(m)}(2), \dots$, essentially represent the behaviour of $x(t)$ in different time scales ($m=1,2,3,\dots$). By plotting $\log\left(\sigma^2\left[\vec{x}^{(m)}\right]\right)$ vs. $\log(m)$, one can see that a Poisson process will have a slope of -1 while a self-similar process will have a slope flatter than -1 [17]. ($\sigma^2[\cdot]$ is defined as the variance of the data series in the bracket)

Figure 2.1: Self-Similar Process and Poisson Process

FGN is a Self-Similar process with a LRD structure. LRD processes exhibit dependencies over a long range of time scales (consistent with a Self-Similar process exhibiting strong correlation in multiple time scale), while SRD processes have autocorrelation functions that decay fast enough to be summable. The autocorrelation function of a FGN process is [18]:

$$r(t) = \frac{(t+1)^{2H} - 2t^{2H} + (t-1)^{2H}}{2} \quad (16)$$

where $t=1,2,3\ldots\infty$ and H is the Hurst parameter (an indicator of the burstiness of the random process).

The FGN process is chosen to model the core network traffic in this thesis because, based on the central limit theorem, the aggregation of a large number of independent flows in the core network will exhibit Gaussian property, and studies have shown that Internet traffic shows characteristics of self-similarity. It is important to emphasize that FGN process is not the only random process EPIT can handle. As long as a random process (whose probability density function can be used to calculate the likelihood ratio) can model the traffic, EPIT shall work. FGN is chosen as a sample traffic model in this thesis.

It is very difficult to get traffic traces in the core network from the major Internet Service Providers (ISP). Even if the traces were obtained, the mean rate of the traffic traces would be fixed, while SA-EPIT (to be introduced later) needs to adjust the mean rate to satisfy the end-to-end QoS requirement. Thus, the traffic is generated according to the FGN distribution in the simulation experiments of this thesis.

Due to the LRD structure, it is difficult to generate independent samples within one replication while Equation (5) requires independent samples. The technique this work used was to generate multiple independent replications. Each replication was treated as one sample upon its completion. This is called finite-horizon simulation. Since each replication is one sample, N , in Equation (5) is the number of replications used in the

simulation. It will be shown later that the computation time of EPIT scales linearly with respect to N .

The generation of FGN traces in each replication uses the Hosking procedure [19]. This procedure is based on an autoregressive process of order p , $AR(p)$, defined in [18]. The order p intuitively refers to how far back one has to go, in a time series, to generate the next element in the series. For example,

$$z(t) = \eta_1 z(t-1) + \eta_2 z(t-2) + \dots + \eta_p z(t-p) \quad (17)$$

The partial autocorrelation function [18] is a construct which exploits the fact that an $AR(p)$ process has an autocorrelation function which, by its very nature, can be described in terms of p non-zero functions of the autocorrelations.

Denote by $\phi_{k,t}$, the t -th coefficient in an autoregressive process of order k , so that $\phi_{k,k}$ is the last coefficient. $\phi_{k,t}$ satisfies the following set of equations:

$$r(t) = \phi_{k,1} r(t-1) + \dots + \phi_{k,k-1} r(t-k+1) + \phi_{k,k} r(t-k) \quad (18)$$

The quantity $\phi_{k,k}$, regarded as a function of the lag k , is called the partial autocorrelation function. The partial autocorrelation function shall be used later in the efficiency analysis. The following are how the sample traces are generated using the partial autocorrelation functions [20]:

$$\phi_{t,j} = \phi_{t-1,j} - \phi_{t,t} \phi_{t-1,t-j+1} \quad (19)$$

$$\phi_{t,t} = G_t / U_t \quad (20)$$

where $j=2, \dots, t-1$ and $G_t = r(t-1) - \sum_{j=2}^t \phi_{t-1,j} r(t-j)$ and $U_t = U_{t-1} - G_{t-1}^2 / U_{t-1}$.

Note that $\phi_{t,j}$ depends only on the value of H ($0 < H < 1$), thus, it can be pre-computed for any given H value.

Define $m(t)$ as the marginal (conditional) mean, at time t , of the FGN process $\vec{X}$ whose mean is v : $m(t) = v + \sum_{j=1}^{t-1} \phi_{tj} (x(t-j) - v)$. Define $\sigma^2(t)$ as the marginal (conditional) variance of $\vec{X}$ at time t . With the partial correlation functions of $\vec{X}$ being $\phi_{t,j}$, then $\sigma^2(t) = \sigma^2 \prod_{j=2}^t (1 - \phi_{jj}^2) = \sigma^2 \sigma_1^2(t)$, where $\sigma_1^2(t)$ is the marginal variance of a FGN process with a variance of one. Let $\psi(t) = \sum_{j=1}^t \phi_{t,j}$, and m^* be the twist amount and $m'(t)$ be the biased marginal (conditional) mean at time t , of the biased FGN process $\vec{X}'$ whose mean is v' : $m'(t) = v' + \sum_{j=1}^{t-1} \phi_{tj} (x'(t-j) - v')$. To conclude, a random sample of $\vec{X}$ at time t , containing “flow” intensity values (arrivals), is generated as $norm(m(t), \sigma^2(t))$. Thus, the likelihood ratio becomes:

$$\frac{f(x(1), x(2), \dots, x(T), v)}{f(x(1), x(2), \dots, x(T), v')} = \prod_{t=1}^T \frac{\exp\left(-\frac{(x(t) - m(t))^2}{2\sigma^2(t)}\right)}{\exp\left(-\frac{(x(t) - m'(t))^2}{2\sigma^2(t)}\right)} \quad (21)$$

Note that $(\vec{X} - v)$ and $(\vec{X}' - v')$ both follow the zero mean FGN distribution. This fact and Equation (21) will be very useful in the efficiency analysis of EPIT in Chapter 5.

As this thesis is focused on traffic engineering, and not traffic modeling, it is assumed that the traffic model will be given, and the parameters (the Hurst parameter, the

mean, the variance, etc) of the traffic distribution have been estimated correctly. Chapter 7 will provide some existing approaches to measure those parameters but will not go into too much detail.

While the input traffic model can be described by several parameters (mean, variance, Hurst parameter, etc), in this thesis, traffic engineering will only adjust the mean of the traffic. Thus, the QoS depends only on the mean rate of the ingress traffic. There will be more detailed explanation of this topic later.

With the background knowledge presented in this chapter, the next chapter will discuss the related work that has already been done in this field.

Chapter 3 Related Work

This chapter shall outline some of the work already done in network QoS analysis and efficiency analysis. It will be demonstrated that existing QoS analysis approaches either only work with SRD traffic model, or imposes behaviour restriction on the traffic. The existing analyses on efficiency were done only under SRD input.

3.1 Existing Analytical-based Analysis Approaches on QoS

The traffic carried by the Internet has shown a vast diversity that poses a significant challenge to all analytical techniques. Extensive measurements of real traffic data [11] [12] have led to the conclusion that Internet traffic and Ethernet traffic cannot be sufficiently represented by traditional models, but instead possess long-range dependent (LRD) characteristics that can be more accurately matched by self-similar models. It has been determined that traditional stochastic models which exhibit short-range dependent (SRD) greatly underestimate the burstiness of the Internet traffic, which results in the underestimation of the probability of buffer overflow in a network. Thus, the LRD traffic model remains a popular research topic today [21] [22] [23].

While the speed of backbone links and routers have increased dramatically in the recent years, one has to keep in mind that the LRD model was introduced to capture the behaviour of aggregated traffic flows rather than individual traffic flows. With more

traffic flows, the aggregated flow will exhibit stronger self-similarity since aggregating streams typically intensifies the self-similarity (“burstiness”) instead of smoothing it [11]. This implies that the more one aggregate flows, the more accurate the LRD model is.

Some of the related work in congestion (and other behaviours) of a network is based on SRD traffic models, while others can handle LRD traffic models.

3.1.1 Analytical Approaches Based On Short-range Dependent Traffic Models

One of the most notable analytical techniques developed in the early 80s is the Matrix-Geometric solution [24]. However, as the number of nodes in a network becomes larger and user traffic becomes ever more complex, this approach quickly becomes intractable. While [25] provides a product-form solution so that each node can be analyzed independently, unfortunately, it is only applicable to specific Markovian traffic models. In the early 90s, large deviation techniques were successfully applied to a single queue under Markovian traffic models [26] [27].

The authors in [28] studied the blocking probability on a single network link instead of an end-to-end path, and the paper utilizes traditional queuing models such as the $M/G/n-0$ queue for modeling the admission control system and $M/G/\infty$ for the capacity over-provisioning system.

The authors in [29] developed a network calculus that is the stochastic equivalent of the existing deterministic network calculus. The paper demonstrated that, under some conditions, the end-to-end performance of a network could be calculated based on some simple rules. However, this calculus can only work for specific types of traffic (e.g., SRD traffic) and specific types of queues (e.g., FIFO queues).

3.1.2 Analytical Approach Based On Long-range Dependent Traffic Models

There have been some analytical queueing results reported in this area, mainly on a single node. The lack of progress on multi-queue system appears to be because of the strong correlation within the LRD processes .

Papers [30] [31] provided asymptotic expressions for the steady-state waiting time in a single node under LRD and self-similar processes. Norros [32] also provided an approximation for the probability of a queue backlog exceeding a limit, under a LRD input:

$$P\{Q_T > q_T\} = \varsigma \exp\left(-\tau q_T^{2(1-H)}\right) \quad (22)$$

where Q_T is the queue backlog at time T , H is the Hurst parameter of the input traffic

flow, $K(H) = H^H (1-H)^{1-H}$, $\tau = \frac{\mu_e^{2H}}{2K^2(H)}$, μ_e is the effective service rate of this setup in

that it is the difference between the constant service rate (μ) of the queue and the mean arrival rate (ν), and ς is some quantity independent of μ_e and q_T .

In order to use the Norros' equation, one has to find the constant ς :

Figure 3.1: Determining Norros Equation's Constant

Since a constant is being calculated, one has only to find the difference between the simulation result and the Norros' result in a single scenario, and this difference will give ς :

$$\varsigma = \frac{P_{SIM}(Q_T > q_T)}{\exp(-\tau q_T^{2(1-H)})} \quad (23)$$

If the simulation implementation is correct, one only has to calculate ς once, and it will work for any other scenarios. This is indeed the case as it is demonstrated in Figure 5.13.

Massoulie [33] provided a more complex function for $P(Q_T > q_T)$:

$$P\{Q_T > q_T\} = \frac{\Psi}{\mu_e^{2H-1}} \frac{1}{q_T^\gamma} \exp(-\tau q_T^{2(1-H)}) \quad (24)$$

where $\gamma = \frac{1-H}{H}(2H-1)$ and Ψ is a very complex function of H , see [33] for details.

Equation (24) shows that the loss probability (due to backlog exceeding a threshold) decreases faster than exponentially with increasing effective service rate, μ_e , and it closely resembles a Weibull-function. This information shall be important in the efficiency analysis (Chapter 5). To avoid any confusion, keep in mind that [10] deals with the case of increasing buffer size threshold instead of the increasing service rate, thus [10] observed a different behaviour for P .

While Norros and Massoulie have provided analytical equations to describe the probability of queue backlog exceeding a limit, under LRD input, the equations are only applicable in the single-queue scenario. While they cannot help with achieving end-to-end QoS in a network, they will be useful in the efficiency analysis (Chapter 5).

Another study was presented in [34]: based on the many-source-asymptotic technique [35], [34] demonstrated that a network can be decomposed into a simplified network in which all the nodes with large capacity have been eliminated. However, it does not completely solve the problem because the question on how to estimate the performance of those remaining nodes is still open.

For analytical queueing results used in multiple-node scenarios, they usually impose restrictions on the ingress traffic flows (burstiness restrictions [36] or interval traffic arrival restrictions [37]) or have certain network topology constraints (such as feed-forward network only [38] [39]).

The end-to-end network diagnoses proposed in [39] determine the loss rate on individual network links from: 1) measured end-to-end loss rates on the different network paths in the network and 2) information on which paths are sharing a particular link. The feasibility of this approach again depends on the topology of the network. In certain topologies, the loss rate of certain links in the network cannot be calculated using this approach.

[40] predicts the characteristics of congestion events under a self-similar traffic model. The characteristics of interest include the duration of the congestion and the occurrence frequency of the congestion. However, the analysis conducted is limited to single queue networks.

In summary, the existing analytical approaches all have their limitations: restrictions on the topology, dependency on certain traffic models, or applicable only to single queue

networks. Instead, EPIT addresses all those issues. Next, some of the existing simulation approaches are discussed briefly.

3.2 Existing Simulation-based Analysis Approaches on QoS

Network simulation has been a long time ally in improving the performance of a network. By first creating and implementing a network model, direct simulation can be used to estimate the QoS in network under different network parameter settings. The shortcoming of the direct simulation, in terms of efficiency, has been discussed already.

[20] is the first to combine Importance Sampling technique with LRD traffic models and was initially achieved with FGN processes and later generalized to a more general class of Self-Similar processes with arbitrary marginal distributions [41]. However, those previous studies were done for a single queue rather than a multiple-queue network. Those studies are done for IS and not ISPE. Lastly, the FGN processes in those papers all have a mean of zero and a variance of one. In order to handle arbitrary mean and variance values, the likelihood ratio of IS and ISPE had to be re-derived in this thesis.

The Importance Sampling concept has been applied to general systems before, such as the regenerative method under traditional SRD models (e.g. Markov chain). A system is regenerative if a particular system state (as an example, the state is the queue backlog) returns to state 0 (queue backlog=0) infinitely often. Upon reaching that state, the stochastic evolution of the system state becomes independent from the past, and has the same distribution as if it were started in state 0 [42]. The period between state 0 is denoted as a cycle. [43] [44] [45] used importance sampling in queue systems with this regeneration method.

With a regenerative system, the events of buffer overflow in different cycles are independent from each other. Importance sampling biases the arrival rate to a point that it exceeds the service rate so the system becomes unstable and overflow occurrence is guaranteed. In such a situation, the biasing is reduced drastically so the queue backlog is quickly reduced to zero (system returns to state 0). This is essentially how IS speeds up the simulation. However, this approach only works with tradition models, and not LRD models.

The optimal biasing (twist) amount has been derived [41] [43] for the general case of d -dimensional Gaussian vectors, described by a covariance matrix $\vec{C}$, using large deviation theorems. However, this approach can only be used to numerically calculate the optimal twist under certain conditions due to the complex covariance matrix, and the approach does not provide a close-form relationship between the normalized variance and various parameters. Thus, the shortcomings of this solution are:

- 1) The solution is in a Euclid dot-product form of several matrixes, including the inverse of $\vec{C}$. Therefore, it is only useful for calculating numerical values. As the number of elements in the covariance matrix is the number of samples collected squared, the computation effort to find the optimal twist amount shall increase very fast with respect to the number samples.
- 2) The matrix-form solution makes it difficult to observe the trend of the optimal twist amount and the normalized variance as a function of other parameters in the simulation.

For simulations in this thesis (see Chapter 6), 500 samples of the arrival process are used for each replication. If it is increased to 1000 samples, the computational burden of

the solution proposed in [41] will be even greater. In contrast, the solution in this thesis is able to characterize the optimal twist amount explicitly in a closed-form function with system parameters, in the absence of any large matrix. In a brief note, instead of estimating normalized variance using input samples, the estimation uses output samples (more in section 5.1).

Based on the closed-form equation, one can observe (graphically) and study the trend of the optimal twist amount with respect to different system parameters, such as the queue size threshold and the service rate.

The next section shall cover the simulation efficiency work already done for IS.

3.3 Existing IS Simulation Efficiency Analysis

The simulation efficiency of IS has been examined in the early literature [9] [46] [47]. The analyses in those works are carried out under system inputs that are modeled as i.i.d sums, Markov Sums (from homogeneous Markov chain), and independent Gaussian noise samples. The correlation, within a short-range dependent (SRD) random process, e.g., Markov chain, between values at different times decreases rapidly (exponentially) as the time lag increases. In contrast, long-range dependent (LRD) random processes, e.g., Fractional Gaussian Noise (FGN) processes, exhibit self-similar characteristics across different time scales, and its auto-covariance function decays slower than exponential. Studies have shown that Internet traffic exhibits long-range dependence [11] [12].

[9] introduced the issue of asymptotic efficiency for IS simulation. The paper states that large deviation probability vanishes with an exponential rate with respect to the rarity parameter, under short-range dependent models. [10] later successfully applied large

deviation to scenarios where the decay is slower than exponential: the loss probability in a queue with an increasing buffer threshold size, under long-range dependent arrival models. In this case, the probability decays following a Weibull-function, which is slower than exponential.

A decaying probability means more simulation effort unless the IS influence of the biasing in the variance decays at the same rate, so the variance will not grow. Experiments were performed on exponentially twisting (biasing) where the input to the system is $B_\varphi = \frac{1}{\varphi} \sum_{k=1}^{\varphi} \vec{Y}_k$. $\{\vec{Y}_k\}$ is an i.i.d sequence of d -dimensional random vectors with a univariate distribution. For the Markov sums input in the paper, the definition of B_φ remains the same except $\vec{Y}_k = g(M_k)$, where $\{M_k\}$ is a time homogenous Markov chain, and $g(\cdot)$ is a d -dimensional real valued function. For the system with Small Gaussian Noise Inputs, B_φ is a d -dimensional Gaussian vector with a covariance matrix $\sigma_\varphi^2 C$ where $\sigma_\varphi^2 = \sigma_0^2 / \varphi^2$ and C is positive finite. Simulation efficiency analysis were conducted on those IS simulations.

[46] expanded on the works of Sadowsky to five different IS biasing techniques: variance modification (VM), mean value modification (MM), shift of absolute values (SAV), replacing the Gaussian distribution with Gaussian tail (GT), and Rayleigh tail distribution (RT). The system input used in the paper is a d -dimensional zero mean Gaussian (row) vector with covariance matrix $\vec{C} = \sigma^2 I$ where I is the identity matrix.

The paper provided the sufficient condition for the asymptotic efficiency of these techniques.

[47] studied the efficiencies of IS from the accuracy perspective of the IS estimators. The efficiency criteria are described in terms of bounded relative error (BRE), bounded normal approximation (BNA) and asymptotic optimality (AO).

The bounded relative error is the square root of the normalized variance of IS: if there exists a constant c such that

$$\lim_{\varphi \rightarrow \infty} \sqrt{\sigma^2 \left[\hat{P}_\varphi \right]} \leq c < \infty \quad (25)$$

where φ is the rarity parameter.

The bounded normal approximation is expressed as:

$$\frac{E_{f(v')} \left(\left(I_A(X) L(X, v, v') - \hat{P} \right)^3 \right)}{\sigma^3 \left(\hat{P} \right)} \leq c < \infty \quad (26)$$

And finally the asymptotic optimality is:

$$\lim_{\varphi \rightarrow \infty} \frac{\ln E_{f(v')} \left(I_A(X)^2 L(X, v, v')^2 \right)}{\ln \left(\hat{P}_\varphi \right)} = 2 \quad (27)$$

As one can see, the asymptotic optimality condition is very similar to the previous simulation efficiency criteria, see Equation (14). The dependencies of these three efficiency conditions were studied in the papers. Again, the system input used in the paper is i.i.d random variables.

Observe that the simulation efficiency analysis of IS under LRD input is missing from existing research, thus, section 5.3 of this thesis shall fill in that void. On another note, to the best of the author's knowledge, the normalized variances for IS simulation, under LRD input, have not been analytically derived, nor has the normalized variance of ISPE. All those are done in Chapter 5, as additional contributions of this thesis.

Chapter 2 and Chapter 3 have been about existing knowledge and existing work related to this thesis. From Chapter 4 onward, the materials are the original work of the author. The implementation chapter, Chapter 7, contains an architecture proposed by the author to implement EPIT, but mainly deals with existing technologies that can be used to implement EPIT. This work was not included in this chapter because it might distract the readers from existing work related to the theme of this thesis.

Chapter 4 End-to-End Performance Inference Technique

For many of the Internet applications today, it is the end-to-end QoS that is the most crucial, rather than the QoS on individual links along the end-to-end path.

Figure 4.1: Traditional End-to-end Path Performance Analysis

Traditionally, the analysis for an end-to-end path begins at the arrivals (a^1, a^2 , etc) and the departures (d^1, d^2 , etc) of the individual nodes on the path as shown in Figure 4.1. Based on the arrival process, the departure process, the buffer size, and the service capacity of the node, one can determine the backlog, delay, burstiness in the individual nodes. The end-to-end performance is then calculated as the product form or summation form of each node's performance [25][29]

The challenge arises from the fact that traffic flows that are not Markovian lose their original statistical characterizations at the output of a queue. Thus, researchers [29] [48] have used envelopes or bounds to describe the output processes. The performance measures such as backlogs and delays are then provided also as bounds.

If one is given a QoS target, for example, an end-to-end loss probability P of 10^{-6} , how does one provision the loss probabilities, P^1 , P^2 , etc, of each individual nodes? One trivial solution is that, assume there are 10 nodes in total, each individual node must not exceed $10^{-6}/10=10^{-7}$ in terms of packet loss probability. Clearly, budgeting the requirement equally among the nodes is not fair, since some nodes have more traffic load than others, while some nodes have less service capacity than others. The problem is made even more complicated if a node is shared by several end-to-end paths in which each path has a different end-to-end QoS requirement, as shown in Figure 4.2.

Figure 4.2: Multiple End-to-end Paths in an Arbitrary Multi-input Multi-node Network

It is a more efficient approach to analyze the QoS of an end-to-end path as a whole, instead of analyzing the performance of individual links (nodes) on the path. In other words, a black box approach that can perform the analysis not based on the buffer size, link rate (service rate), backlog, and delay of individual links but rather treat the end-to-end path as a single entity, as shown in Figure 4.3, is desired.

**Figure 4.3: Black Box View of the Network: Focusing on the End-to-end QoS of a Path
Instead of the QoS of Individual Links in the Path**

The black box view is not the same as treating the entire network as a single queue. There is only one observable behaviour from a single queue, yet in the black box view, one can observe the behaviour of several end-to-end paths. In the case that a single queue is fed by several traffic sources, all sources will have some degree of influence in the backlog or the delay in the queue. In the black box view, specific traffic sources can have minimal or zero influence on specific end-to-end paths if the traffic sources are disconnected or are “very far” from the specific end-to-end paths.

By considering a very sophisticated and arbitrary queuing system, illustrated in Figure 4.4, the problem of analyzing the end-to-end performance in the network is now extremely challenging, since non-Markovian traffic flows lose their original marginal distribution at the output of a queue. This problem is made even more complicated by introducing LRD input into the scenario in which there are strong correlations across multiple nodes making it difficult to decompose a network.

Figure 4.4: Multi-Source Multi-Queue Network

Each host in Figure 4.4 acts both as a traffic source and as a traffic sink, and there are S sources in total. The traffic flows from the traffic sources are the aggregated traffic that is entering the provider's core network, and they are modeled as independent random processes $\vec{X}_1, \vec{X}_2, \dots, \vec{X}_S$ with joint density distributions of $f(\vec{x}, v'_1), f(\vec{x}, v'_2), \dots, f(\vec{x}, v'_S)$, where $v'_1, v'_2, \dots, v'_S$ are the mean rates (bandwidths) of the respective traffic flows. Let $\vec{v}' = [v'_1, v'_2, \dots, v'_S]$. For FIFO queue behaviour, each flow arriving at the queue will be facing the same average loss probability $\hat{P}(\vec{v})$, which will be defined in Section 4.1.

Assume the end-to-end path of interest is between host 1 and host 6, as highlighted (there could also be multiple paths that require attention). The path spans several queues.

In the network considered in this thesis, the FGN random process is used to model the ingress traffic flows generated from the traffic sources in the network. A FGN process is completely described by the mean, the variance, and the Hurst Parameter. To model a more realistic network, the mean, variance, and the Hurst parameter of each traffic source are set to be different from each other in this thesis. For every queue in the system, the buffer sizes and the service rates are not identical. The number of connections from a queue to other queues and the overall topology are variable as well.

As stated previously, the QoS in the network is achieved by varying the amount of traffic load entering the network, thus, only the mean (or mean rate) of each traffic source is varied and the variance and the Hurst parameter are fixed. This is why the joint density distributions $f(\vec{x}, v'_1), f(\vec{x}, v'_2), \dots, f(\vec{x}, v'_s)$ are only functions of the mean rates of the traffic.

Because ISPE has not been applied to traffic engineering or in scenarios of multiple-queue multiple-sources, one of the major contributions of this thesis is to adapt ISPE in this new area of application by creating the new definitions for the event of interest, the event indicator function, and the likelihood ratio.

4.1 EPIT Formulation

Let the set $\mathcal{A}$ be a set containing all the realizations (traffic traces) from $\vec{X}_1, \vec{X}_2, \dots, \vec{X}_s$ such that they will trigger one or more events along the end-to-end path if they were

injected into the network together. The event of interest thus is no longer an event occurring on a single node/link on the end-to-end path, but an end-to-end event: an example is that, if the backlog of any node along the end-to-end path has exceeded its threshold, the path as a whole is considered to have had an end-to-end overflow event. The event indicator function then becomes $I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S})$, since the end-to-end event on an end-to-end path depends (with various degree) on all the traffic flow that enters the network.

$\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}$ are traffic traces of the traffic flows $\vec{X}_1, \vec{X}_2, \dots, \vec{X}_S$ of the i -th replication. As stated before, using the finite-horizon method, each replication will yield a sample, in this case, a possible end-to-end event, which in turn is used to calculate the probability of interest (loss probability). This means that if traffic traces $\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}$ trigger an end-to-end event in the network, then $I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}) = 1$ and zero otherwise.

The end-to-end event of interest can be the end-to-end overflow event that was defined in section 2.1; or an “end-to-end jitter” related event or an “end-to-end delay” related event. If the measured jitter/delay of a particular connection/session exceeds a specific threshold (as if backlog in a queue exceeds a threshold), it can also be considered as an end-to-end event.

Let $\vec{v} = [v_1, v_2, \dots, v_S]$ be some other arbitrary traffic level setting while $\vec{v}' = [v'_1, v'_2, \dots, v'_S]$ remains the original traffic level in the network. The likelihood ratio, for replication i , becomes a function of both $\vec{v}$ and $\vec{v}'$:

$$L(\vec{x}_{i,1}, v_1, v'_1, \vec{x}_{i,2}, v_2, v'_2, \dots, \vec{x}_{i,S}, v_S, v'_S) = \frac{f(\vec{x}_{i,1}, v_1, \dots, \vec{x}_{i,S}, v_S)}{f(\vec{x}_{i,1}, v'_1, \dots, \vec{x}_{i,S}, v'_S)} \quad (28)$$

where $f(\vec{x}_{i,1}, v_1, \dots, \vec{x}_{i,S}, v_S)$ is the joint probability density function of all the traffic processes going into the network.

As this thesis studies the core network of service providers in which the aggregated traffic entering the edge nodes that are located in different geographical locations, it is reasonable to assume that each aggregated ingress traffic flow is independent from each other. Thus, equation (28) can be re-written as:

$$L(\vec{x}_{i,1}, v_1, v'_1, \vec{x}_{i,2}, v_2, v'_2, \dots, \vec{x}_{i,S}, v_S, v'_S) = \prod_{k=1}^S \frac{f(\vec{x}_{i,k}, v_k)}{f(\vec{x}_{i,k}, v'_k)} = \prod_{k=1}^S L(\vec{x}_{i,k}, v_k, v'_k) \quad (29)$$

Now, one can estimate $P(v_1, v_2, \dots, v_S)$ as:

$$\hat{P}(v_1, v_2, \dots, v_S) = \frac{1}{N} \sum_{i=1}^N \left(I_A \left(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S} \right) \prod_{k=1}^S L(\vec{x}_{i,k}, v_k, v'_k) \right) \quad (30)$$

where a simulation run consists of N independent simulation replications and traffic traces, $\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}$ were generated in every replication according to the distributions $f(\vec{x}, v'_1), f(\vec{x}, v'_2), \dots, f(\vec{x}, v'_S)$, respectively. (In traffic engineering, it is assumed that the distribution of the traffic is known or can be estimated via measurements)

Since $\vec{x}_{i,k}$ is a sample trace of a random process,

$$f(\vec{x}_{i,k}, v_k) = f(x_{i,k}(1), x_{i,k}(2), \dots, x_{i,k}(T), v_k) \quad (31)$$

where $x_{i,k}(t)$ is the sample value of $\vec{X}_k$ in the i -th iteration at time t .

Since $\vec{X}_k$ follows the distribution of an FGN process, one uses equation (21) and equation (31) to find :

$$L(\vec{x}_{i,k}, v_k, v'_k) = \prod_{t=1}^T L_{i,k,t} \quad (32)$$

where

$$L_{i,k,t} = \frac{\exp\left(\frac{-(x_{i,k}(t) - m_{i,k}(t))^2}{2\sigma^2_{(k)}(t)}\right)}{\exp\left(\frac{-(x_{i,k}(t) - m'_{i,k}(t))^2}{2\sigma^2_{(k)}(t)}\right)} \quad (33)$$

$m'_{i,k}(t)$ is related to v'_k , and $m_{i,k}(t)$ is related to v_k . For the definitions of all the quantities in equation (33), please refer to section 2.4.

By substituting equation (32) into equation (30), one has:

$$\hat{P}(v_1, v_2, \dots, v_S) = \frac{1}{N} \sum_{i=1}^N \left(I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}) \prod_{k=1}^S \prod_{t=1}^T L_{i,k,t} \right) \quad (34)$$

Equation (34) will be used shortly to find the computation cost of EPIT.

Equation (30) essentially describes EPIT. From equation (30), one can see that EPIT is a technique that combines simulation with analytical analysis. The end-to-end event indicator function $I_A(\cdot)$ is observed in each replication during the simulation run, while the likelihood ratio is calculated (the analytical analysis portion of this approach) based on the probability density function of the traffic process. To summarize, based on one simulation run that observes the end-to-end QoS related events on the end-to-end path in the network under the initial traffic level $\vec{v} = \vec{v}'$, EPIT calculates (or infers) the statistical QoS performance (based on the occurrence probability of the QoS event) under arbitrary

level $\vec{v}$ *without* more simulations. In other words, Equation (30) essentially provides a mapping between the ingress traffic levels to the end-to-end QoS in the network. This is why the proposed approach is named as the End-to-end Performance Inference Technique.

4.2 Advantages of EPIT

Figure 4.5: Black-Box view of a Multi-Source Multi-Queue Network

EPIT has the following advantages:

- **Black-box:** With the indicator function I_A capturing the end-to-end events in the network, EPIT analyzes, as shown in equation (30), the performance of the end-to-end path (P) as a whole is analyzed instead of analyzing the performance of the individual links of the path, thus providing a black-box view to the traffic engineers.

The ingress traffic streams (or their mean rates) are the inputs to the black box, and the end-to-end QoS event Indicator is the output from the black box, as shown in Figure 4.5. Equation (30) shows that the internal details of the network, such as the individual service rate, buffer size threshold, interconnectivity of the nodes is not used in the inference of the end-to-end QoS performance. Intuitively, the internal details of the network are “taken care of” in the observed $I_A(.)$ during the simulation run. It is important to emphasize again that only one simulation run (but with multiple replications) is used while the prediction of the QoS, with respect to different amounts of ingress traffic, is done analytically. EPIT essentially treats the target network as a black box, as it uses the end-to-end QoS event in the network rather than the behaviour of individual queues.

- Scalable: With a black box approach, the complexity of the network does not have effects on the computational complexity of EPIT. Assume the computation time for all arithmetic operations is the same, and assume the computation time for each $L_{i,k,t}$ in equation (34) is δ . The total computation time for EPIT is therefore:

$$TotalTime = S \times N \times T \times \delta \quad (35)$$

Clearly, the computation time scales linearly with respect to the number of traffic sources (S) in the network, the number of replications (N) used in the initial simulation run, and the length (T) of the replication. Since the number of traffic sources in the network model is determined by how many sources are in the actual network and the run length of the replication depends on when the system (the network) will enter steady state, the number of replications (N) is variable and

essentially describes the computational cost of EPIT. (N will be used in the efficiency analysis of EPIT in Chapter 5.) Since EPIT scales linearly with respect to the number of traffic source in the network, it is independent of the number of nodes in the network, and is not concerned with how nodes are interconnected; thus, EPIT scales well against complex networks.

- Flexible: Equation (30) shows that EPIT can handle any traffic model as long as the probability density function of the traffic is known for the likelihood ratio calculation. Thus, EPIT does not rely on bounds or envelopes of the input traffic to perform its analysis, nor does it impose restrictions on the stochastic property of the traffic. EPIT is compatible with both LRD traffic and SRD traffic.
- Time Saving: EPIT requires only one simulation run to predict a wide range of QoS values. With the MC technique, a simulation run is needed for each value of QoS. The time-saving becomes especially important when one is searching the correct traffic level that can meet a certain target QoS (more on this in section 4.3).
- Lastly, since EPIT treats the network as a black box, it does not assume there is only a single point of congestion in the network. There could be multiple congested queues (hot spots) in the network, yet that is irrelevant, since the simulation captured end-to-end events and the distributions of the ingress traffic are all that EPIT requires.

So far, one can see that EPIT has answered the first problem statement presented at the beginning of thesis: How to predict the end-to-end QoS in a network accurately as a function of the ingress traffic level of the network. It has been shown that EPIT is scalable and flexible. The accuracy of the EPIT approach will be reported in Chapter 6.

The efficiency of the EPIT approach is analyzed in Chapter 5 and Chapter 6. The implementation of EPIT is presented in Chapter 7.

For the initial simulation run in EPIT, the rule of thumb stated in [1] is used in choosing the number of traces (N) to generate for each source (N is essentially the number of replications): $2\sqrt{\hat{P}/N} < \hat{P}/2$, based on the sampled P . Note that while the probability target may be small (10^{-6}), the initial probability P will not be so small. Thus, N does not have to be big to keep the variance at a reasonable level.

The probability predicted by EPIT is an estimation, and the uncertainty is reflected in its variance:

$$\sigma^2 \left[\hat{P}(\vec{v}) \right] = \frac{1}{N} \frac{\sum_{i=1}^N \left(I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S}) \prod_{k=1}^S (L(\vec{x}_{i,k}, v_k, v'_k)^2 - \hat{P}(\vec{v})^2) \right)}{N-1} \quad (36)$$

To take the uncertainty of the prediction into consideration, the upper bound of the 95% confidence interval of the calculated probability is used (this bound will be used in the heuristic search algorithm discussed in the next section):

$$\hat{P}(\vec{v}) + z_{0.025} \sqrt{\sigma^2 \left[\hat{P}(\vec{v}) \right] / N} \quad (37)$$

In order to have a good estimate of the variance for the estimator, samples must be independent. Samples within one traffic trace are highly correlated due to its LRD nature. In order to have independent samples, one has to generate multiple independent traces with multiple replications and treat each trace as one sample only. As stated before, this is the finite-horizon technique. As a result, each sample of $I_A(\vec{x}_{i,1}, \vec{x}_{i,2}, \dots, \vec{x}_{i,S})$ from each

replication will be independent from each other. Thus, $\hat{P}(\vec{v})$ is estimated from N independent samples of $I_A(.)$ and given N is large, according to the central limit theorem, a confidence interval can be calculated for $\hat{P}(\vec{v})$. The simulated-annealing search algorithm in the following section shall use the upper bound of this confidence interval.

The second problem statement presented earlier in thesis is how to achieve a target QoS by adjusting the ingress traffic rates while maximizing the network utilization. A heuristic search algorithm, simulated annealing, will be integrated with EPIT in the next section to answer this question.

4.3 Simulated Annealing with EPIT

From the traffic engineering perspective, $\vec{v}$ is current (or initial) ingress traffic level of the network. If the QoS on a critical end-to-end path (“critical” means that the QoS of the path requires attention) exceeds its requirement, the network administrator may want to control the traffic entering the network via traffic shaping or admission control. By using EPIT, the network administrator can foresee the performance of end-to-end path for any arbitrary input traffic levels $\vec{v}$. Thus, EPIT shall be used, in combination with simulated annealing to achieve the End-to-End QoS target in a network.

The key traffic engineering constraint in this work is to satisfy a certain QoS target, TQ : such that $P(v_1, v_2, \dots, v_S) \leq TQ$. Through traffic shaping or call admission control, $v_1, v_2, \dots, v_S$ can be adapted so that the given QoS target is met. However, excessive traffic shaping or call rejection can lead directly to customer dissatisfaction and (or) loss in

revenue. For this reason, another traffic-engineering goal in this work is to maximize the traffic flows and the network resource utilization. The Simulated Annealing (SA) coupled with EPIT approach proposed in this thesis can offer a traffic engineering solution that maximizes the ingress traffic while achieving the expected QoS in the network.

The basic idea is to set the traffic mean rates high enough initially so that the end-to-end QoS $P(\vec{v})$ is much larger than the target QoS, TQ . This allows us to reduce the simulation time significantly by making rare events more likely to happen. One can then estimate the required mean rates by reducing the initial mean rates $(\vec{v})$ to create new “twisted” mean rates $(\vec{v})$, where $\vec{v} = \vec{v}' - \vec{r}$. This is done gradually until the target QoS is met: $P(\vec{v}) \leq TQ$.

The trivial solution is to let $\vec{r} = \vec{v}'$ which will leave the network devoid of any traffic. There will be no congestion in the network, but this is clearly not acceptable. One must allow as much traffic in the network as possible to maximize the network utilization and in the same time satisfy the QoS constraints. Thus, the problem becomes a Non-Linear Programming problem.

4.3.1 Heuristic Search Problem Formulation

The objective function:

$$\min(\Phi(\vec{r})) = \min\left(\sum_{i=1}^S C_i r_i + D \sum_{i=1}^M [P_i(\vec{v}' - \vec{r}) - TQ_i]^+\right) \quad (38)$$

($[x]^+ = x$ if $x \geq 0$, and $[x]^+ = 0$ if $x < 0$)
subjected to

$$P_i(\vec{v}' - \vec{r}) \leq TQ_i \text{ for } i=1, 2, \dots, M \quad (39)$$

And the potential solution space is:

$$\vec{r} = [r_1, r_2, \dots, r_s] \text{ where } 0 \leq r_1 \leq v'_1, 0 \leq r_2 \leq v'_2, \dots, 0 \leq r_s \leq v'_s \quad (40)$$

D is assigned as the cost (penalty) for not fulfilling the QoS requirements in the objective function. It forces the heuristic search (will be discussed shortly) process to converge to a solution that can satisfy the QoS constraints.

As stated before, aside from achieving the QoS goals, one also wants to maximize the network utilization by maximizing the traffic flow. Since the amounts of traffic reduction are used as the solution space for the heuristic search algorithm, and since the final traffic amounts applied to the network to achieve the QoS target is equal to the initial traffic amounts minus the reduction amounts, to maximize traffic flows, the reduction amounts should be minimized. Thus, in the objective function, C_i is assigned as the cost of reducing one traffic unit per time unit from the mean rate of customer traffic stream i . This allows the providers to assign different costs to the different customers' traffic according to criticalness and price.

There is naturally more than one end-to-end route/path in the network that requires a QoS guarantee. Therefore, assume there are M such paths and M QoS targets ($TQ_1, TQ_2, \dots, TQ_M$) for those paths.

If there are M paths that need QoS supports, the simulation run simply uses several indicator functions $I_{A,1}, I_{A,2}, \dots, I_{A,M}$ to simultaneously observe the end-to-end events on the respective path 1, path 2, ..., path M . The same equation, Equation (30), is simultaneously applied for each path, and the computation time for each path will be the

same, since S and N are constant. Clearly, the total computational cost scales linearly with respect to M .

Note that the range for $\vec{r}$ is very large since each r_i can be any positive real number less or equal to v'_i . To reduce this space, a quota Θ is defined as a step size such that all r_i must be an integer multiple of Θ . The exact value of Θ is to be determined by the traffic engineers as to how precise they want their solution to be. Equations (38)-(40) pose an NP-hard problem and a heuristic search algorithm will be used to solve it.

4.3.2 Heuristic Search: Simulated Annealing

While there are many heuristic search techniques, such as Simulated Annealing, Genetic Search, Tabu Search, etc., that can be used in non-linear programming problems, Simulated Annealing (SA) [49] is selected as the heuristic search algorithm for this problem. This decision is based on the fact that SA is simple, fast, and applicable to scenarios where one does not have much knowledge about the problem (e.g., the network) itself.

Vector $\vec{r}$ starts as a zero vector (i.e., no reduction anywhere). In the k -th iteration, SA modifies the solution from the previous iteration $(\vec{r}_{k-1})$ by randomly picking a position i (this is the same as randomly pick a traffic source i in the network to reduce its traffic rate) in $\vec{r}_{k-1}$ and modifies it by an amount of Θ to create a new solution $\vec{r}_k$, for the current iteration k .

The quantity r_i is allowed to decrease (as long as it remains positive) at times because there will be situations in which it is “cheaper” (in terms of the total cost in the objective

function) to reduce one traffic source than to reduce another, while their respective effects on the QoS are similar.

It was stated earlier that the cost D in equation (38) is to bring $P_i(\vec{v}' - \vec{r})$ closer to TQ_i from iteration to iteration, while the cost C in equation (38) keeps the traffic reduction low. This mirrors the stated traffic-engineering goal of meeting QoS constraints while maximizing the traffic flow in the network. Among all the solutions of $\vec{r}$ in which $P_i(\vec{v}' - \vec{r}) \leq TQ_i$, for $i=1, \dots, M$, the one with the lowest total cost is the best solution, denoted as $\vec{r}^*$.

Since $\sum_{i=1}^M [P_i(\vec{v}' - \vec{r}) - TQ_i]^+ = 0$ for all those solutions that have satisfied all the QoS constraints, the one with the lowest total cost is the one with the lowest reduction cost. With C_j 's fixed for all $\vec{r}$'s, $\vec{r}^*$ with its lowest reduction cost will require the least amount of reduction in traffic. Thus, $\vec{r}^*$, while meeting the QoS targets, will offer the maximal network utilization, since it allows the maximum amount of traffic to enter the network.

SA accepts or rejects a potential solution $\vec{r}$ in each iteration of the optimization process as the follows: if the total cost $\Phi(\vec{r}_k)$ of the k -th iteration is smaller than the previous total cost $\Phi(\vec{r}_{k-1})$, $\vec{r}_k$ is always accepted, i.e. $\vec{r}_{k+1}$ will be generated based on $\vec{r}_k$. However, for a potential solution that increases the total cost, it is accepted with

probability $\exp[\beta(k)(\Phi(\vec{r}_{k-1}) - \Phi(\vec{r}_k))]$. Thus, a potential solution is always accepted with the probability α :

$$\alpha = \min\{1, \exp[\beta(k)(\Phi(\vec{r}_{k-1}) - \Phi(\vec{r}_k))]\} \quad (41)$$

where $\beta(k) \geq 0$ and $\beta(k)$ acts as a decision control-value for the k -th iteration. The following section discusses the details of $\beta(k)$.

4.3.3 Simulated Annealing Parameters

This section discusses the implementation of the SA process in detail. One can skip this section if one is already familiar with SA or is more interested in how EPIT is used with SA.

SA requires [49] that α starts with a value close to one and proceeds toward zero after K iterations have passed. This is to make sure that the heuristic search process will not get stuck in a local minimum in early iterations, while ensuring that the search proceeds toward an optimal (minimum cost) solution after the K -th iteration. While SA has been studied extensively, $\beta(k)$ essentially has to be redefined for each individual optimization problem. By defining $\beta(k) = \vartheta\beta(k-1)$, the values of ϑ and $\beta(0)$ have to be found, so the SA demands are met.

To make $\alpha = 1$ at the first iteration, and $\alpha = 0$ at the K -th iteration, use the following facts to find $\beta(0)$ and ϑ : Since all probabilities have a value between zero and one, the absolute difference between two probability values is also between zero and one (this fact is described in equation [42]). It is also true that only one mean rate is changed in each

iteration, and each reduction amount must be an integer multiple of Θ (this is described in equation [43]).

$$0 \leq \left| P_i(\vec{v}' - \vec{r}_{k-1}) - P_i(\vec{v}' - \vec{r}_k) \right| \leq 1 \quad (42)$$

$$\left| \sum_{i=1}^S (r_{i,k-1} - r_{i,k}) \right| = \Theta \quad (43)$$

$$\Phi(\vec{r}_{k-1}) - \Phi(\vec{r}_k) = \sum_{i=1}^S C_i (r_{i,k-1} - r_{i,k}) + D \sum_{i=1}^M \left[P_i(\vec{v}' - \vec{r}_k) - P_i(\vec{v}' - \vec{r}_{k-1}) \right] \quad (44)$$

therefore,

$$\Theta C_{\min} \leq \left| \Phi(\vec{r}_{k-1}) - \Phi(\vec{r}_k) \right| \leq \Theta C_{\max} + D * M \quad (45)$$

where $C_{\max} = \max\{C_i : i = 1, \dots, S\}$ and $C_{\min} = \min\{C_i : i = 1, \dots, S\}$.

Since $\exp(0) = 1$, $\exp(-20) \approx 0$ and $\beta(K) = g^K \beta(0)$, therefore

$$\beta(0) = \frac{1}{\Omega \Theta C_{\min}} \quad (46)$$

$$g = \sqrt[K]{-20 \frac{\Omega \Theta C_{\min}}{\Theta C_{\max} + D * M}} \quad (47)$$

where Ω is a large number.

So far the discussion has been about solving a Non-Linear Program problem, how does EPIT fit into this?

The issue that was not addressed in section 4.3.1 and section 4.3.2 is how to evaluate $P(\vec{v}' - \vec{r})$ for the solution $\vec{r}$ in each iteration. An analytical result is not available for a multi-queue network under the self-similar traffic model, as stated in section 3.1. One could use a direct simulation run (the Monte Carlo simulation) or a direct measurement to find $P(\vec{v}' - \vec{r})$ in each iteration. As the $P(\vec{v}' - \vec{r})$ value becomes smaller, efficiency becomes

a problem because the time to measure each $P(\vec{v}' - \vec{r})$ becomes longer and longer. With one simulation run from EPIT under $\vec{v}'$, EPIT can determine $P(\vec{v}' - \vec{r})$ for any values of $\vec{r}$ with *no additional simulation runs*. EPIT is clearly the better choice. (As stated in the last section, to take the uncertainty of the EPIT prediction into account, the upper 95% confidence interval of the estimated $P(\vec{v}' - \vec{r})$ is used in the heuristic search process.)

To summarize, SA-EPIT answered the second problem statement: how does one find the appropriate traffic levels so the end-to-end QoS is achieved while maximizing network utilization.

Simulation in Chapter 6 verify that the proposed approaches can deliver their promises: EPIT can predict an end-to-end performance measure P given the ingress traffic level $\vec{v}$, and SA-EPIT can find a best solution $\vec{r}^*$ so that $P(\vec{v}' - \vec{r}^*) \leq TQ$ for $i=1,2,...M$ and $\vec{r}^*$ minimizes the traffic reduction.

As stated before, every iteration in the heuristic search process requires an EPIT prediction, and the probability of interest keeps getting smaller. Can the computational cost of the EPIT prediction on the shrinking probability stay the same, or will the cost increase exponentially and eventually bring everything to a halt because each iterations takes too long to find the probability? To answer this question, one essentially has to perform an efficiency analysis on EPIT: with a fixed computation cost, will the normalized variances of the EPIT prediction increase faster than the rate of decreasing probability as the probability approaches zero? In other words, will the EPIT prediction remain accurate, with a fixed computation cost, as the overflow probability decreases?

The efficiency analysis will be split into two portions: 1) the next section shall examine the asymptotic (as the probability approaches zero) efficiency of EPIT analytically. 2) Simulation in Chapter 6 shall show that EPIT is efficient, since its prediction can project a long range of values for the probability (even when the probability is extremely small) with a fixed computation cost.

Chapter 5 Analytical Efficiency Analysis

This chapter is focused on the efficiency of the EPIT approach: with a constant computation cost, will the variance of the EPIT estimation increase exponentially as the probability measure becomes asymptotically small, thus rendering the EPIT prediction un-usable? or, will the variance of EPIT increase at a rate that is moderate compared to the fast decreasing of the probability?

As explained previously, the computational cost of EPIT mainly depends on the number of replications (N) used in the initial simulation run. For the analysis in this chapter, the N is fixed (which in turn implies the computational cost is fixed) for all the derivations and discussions.

We gain insight into the analytical-based efficiency analysis of EPIT through an analytical efficiency analysis in a single-queue scenario (single node, single LRD input source). It is worthwhile to note that, even though this is a simple scenario, there has not been any type of analytical efficiency analysis done on such scenario *with LRD input*. This is one of the contributions of this thesis. The multiple-queue scenario (multiple nodes, multiple LRD input sources) efficiency analysis is covered in Chapter 6 , where the efficiency of EPIT is demonstrated via simulation.

It was stated that variance is an indication on the how “noisy”/reliable a prediction is and the efficiency of a prediction approach, such as EPIT, essentially depends on how the variance changes (the rate of change) as the probability quantity being estimated becomes smaller. For this reason, the first step in the analysis is to derive an equation to characterize the normalized variance of EPIT under LRD input. The second step is to show that the normalized variance, described by an equation derived in the first step, grows at a speed that is slower than or close to that of the decreasing probability,

Because EPIT comes from the concept of ISPE that is in turn derived from the concept of IS, this thesis shall provide an efficiency analysis for IS under LRD input model simply because it has not been done before. By filling in this gap, this becomes another contribution of this thesis. The efficiency criteria (described in section 3.3) has already been defined for IS, although it was not applied for IS under LRD input. This is the criteria that will be used in this chapter to show that IS is efficient under LRD input. Since such a criteria does not exist for EPIT, after deriving the normalized variance of EPIT under LRD input, the work will show that the rate of increase of the normalized variance is moderate to that of the fast-decreasing probability. This follows the basic concept of the existing efficiency criteria for IS.

5.1 Change of Measure

The desired integral in Equation (11) (which describes the normalized variance) is hard to evaluate because of the form of the indicator function $I_A\left(\vec{x}\right)$ in Equation (11) is unknown, given the complex distribution function of the FGN input (see section 2.4). To

get around this problem, the indicator function $I_A(\bullet)$ is redefined and in turn, a different distribution (measure) will be used.

Figure 5.1: System Reformulation for the Change of Measure

Let Q_T be the random variable representing the queue length at time T . In a system formulation, $\vec{x}$ is a sample trace of $\vec{X}$, and $x(1), x(2), \dots, x(T)$ are the values of $\vec{x}$ at time 1, 2, ..., T . Since the input FGN process is too cumbersome for the variance analysis, the analysis shall be performed at the output of the system: the queue length at time T , Q_T .

$I_A(q_T)$ is redefined as an indicator function that is one if the queue length (backlog) at time T , q_T , is greater than the threshold b and zero otherwise. Note that the variable of the indicator function, formally, is a vector sample of an input random process and the variable now is the queue length.

$$P\left\{I_A\left(\vec{X}\right)=1\right\}=P\{I_A(Q_T)=1\} \quad (48)$$

Intuitively, $I_A\left(\vec{x}\right)$ and $I_A(q_T)$ are indicator functions for the same event: the event of queue backlog exceeding a threshold, which is driven by the input process.

For the rest of the chapter, the measure of queue size on the right-hand side of equation (48) is used instead of the measure of the input processes for the calculation of the normalized variances. This is called a change of measure. However, it should be noted that the normalized variances of the estimators for the two measures are not necessarily equal. Theorem 4.3.1 in [43] states that the two normalized variances will be equal if the following condition is satisfied almost surely:

$$\frac{f(x(1), x(2), \dots, x(T), v)}{f(x(1), x(2), \dots, x(T), v')} = s(q_T) \quad (49)$$

where s is some function and T is the length of a replication.

Mathematically, two random variables X and Y are equal *almost surely* if and only if, the probability that they are different is zero:

$$P(X \neq Y) = 0 \quad (50)$$

The following paragraphs will show that the change of measure is asymptotically accurate. This will be done in two steps: First, the relationship between $\vec{x}$ and q_T is developed; subsequently, the relationship that satisfies Equation (49) is developed.

Define $\vec{W}$ as the total workload process and μ as the constant service rate of a single node:

$$w_t = \sum_{i=1}^t (x(i) - \mu) = \left(\sum_{i=1}^t x(i) \right) - t\mu \quad (51)$$

Based on the Lindley equation,

$$P\{Q_T > b\} = P\left\{\max_{0 \leq t \leq T} W_t > b\right\} > \max_{0 \leq t \leq T} P\{W_t > b\} \quad (52)$$

Reference [10] shows that the following is in fact logarithmically accurate for a very large value of b . This can be explained with the intuitive principle that “rare events occur only in the most probable way” [50].

$$P\{Q_T > b\} = P\left\{\max_{0 \leq t \leq T} W_t > b\right\} \sim \max_{0 \leq t \leq T} P\{W_t > b\} = P\{W_{T_s} > b\} \quad (53)$$

Let $t=T_s$ be the value that maximizes the probability $P(W_t > b)$. In brief, there exists a value $t=T_s$ such that

$$P_{W,\infty} \stackrel{\Delta}{=} \max_{t \geq 0} P\{W_t > b\} \cong P\{W_{T_s} > b\} \quad (54)$$

Thus loosely speaking, T_s is the time when the queue enters steady state and

$$P\{Q_\infty > b\} \underset{b \rightarrow \infty}{=} P\{W_{T_s} > b\} \quad (55)$$

From [50] and [41], the LHS and RHS of Equation (55) are asymptotically equal as b approaches infinity. In this thesis, the “ $\underset{b \rightarrow \infty}{=}$ ” sign is used to denote the asymptotic equality.

According to [41], T_s is a function of b . Asymptotically, as b approaches infinity, T_s also approaches infinity. As T is the length of a simulation replication, T_s cannot be larger than T for a simulation with finite horizon. Therefore, we have $T_s = T$ due to the monotonic increasing when $T_s < T$. Therefore, we have the following equation asymptotically almost surely for the event $q_T \geq b$.

$$q_T \underset{b \rightarrow \infty}{=} w_{T_s} = \sum_{i=1}^{T_s} x(i) - \mu T_s \quad (56)$$

Now, let us move on to the second step. Since $v=v'-m^*$, then Equation (21) can be rewritten as:

$$\begin{aligned}
\frac{f(x(1), x(2), \dots, x(T), v' - m^*)}{f(x(1), x(2), \dots, x(T), v')} &= \prod_{t=1}^T \frac{\exp\left(\frac{-(x(t) - m(t))^2}{2\sigma^2(t)}\right)}{\exp\left(\frac{-(x(t) - m'(t))^2}{2\sigma^2(t)}\right)} \\
&= \prod_{t=1}^T \frac{\exp\left(\frac{-(x(t) - (m'(t) - (1 - \psi(t))m^*))^2}{2\sigma^2(t)}\right)}{\exp\left(\frac{-(x(t) - m'(t))^2}{2\sigma^2(t)}\right)}
\end{aligned} \tag{57}$$

Through some arithmetic manipulation and let

$$\varpi(T) = \frac{m^*}{\sigma^2} \sum_{t=1}^T \frac{x(t)\psi(t)}{\sigma^2_1(t)} - \sum_{t=1}^T \frac{(-2m'(t) + (1 - \psi(t))m^*)(1 - \psi(t))m^*}{2\sigma^2(t)} \tag{58}$$

Then:

$$\frac{f(x(1), x(2), \dots, x(T), v' - m^*)}{f(x(1), x(2), \dots, x(T), v')} = \exp\left(-\frac{m^*}{\sigma^2} \sum_{t=1}^T \frac{x(t)}{\sigma^2_1(t)} + \varpi(T)\right) \tag{59}$$

Since $\sigma^2_1(t)$ solely depends on the partial autocorrelation function ϕ_{ij} and the partial autocorrelation function depends solely on the Hurst parameter (H), one can simply use numerical analysis to observe the behaviour of $\sigma^2_1(t)$ with respect to t (note that $0.5 < H < 1$ for LRD processes):

Figure 5.2: Numerical Analysis of the Partial Autocorrelation Function and $\sigma^2_1(t)$

Regardless of the value of H , Figure 5.2 shows that $\sigma^2_1(t)$ quickly becomes a constant. Based on this observation, for a large T , one can rewrite Equation (59) as:

$$\frac{f(x(1), x(2), \dots, x(T), v' - m^*)}{f(x(1), x(2), \dots, x(T), v')} \underset{b \rightarrow \infty}{=} \exp\left(-\frac{m^*}{\sigma^2_1 \sigma^2} \sum_{t=1}^T x(t) + \varpi(T)\right) \quad (60)$$

Asymptotically (as b approaches infinity), $T=T_s$, thus employing Equation (56), we have:

$$\begin{aligned} \frac{f(x(1), x(2), \dots, x(T), v' - m^*)}{f(x(1), x(2), \dots, x(T), v')} &\underset{b \rightarrow \infty}{=} \exp\left(-\frac{m^*}{\sigma^2_1 \sigma^2} (q_T + \mu T) + \varpi(T)\right) \\ &= s(q_T) \end{aligned} \quad (61)$$

Q.E.D

Now, through a change of measure (distribution), the normalized variance, based on Equation (11), can be written as

$$N \overline{\sigma^2} \left[\hat{P} \right] = \frac{\int_b^\infty I_A(q_T) \frac{p(q_T)^2}{p'(q_T)} dq_T}{P^2} - 1 \quad (62)$$

Where $p(q_T)$ is the probability density function of queue length under the true input process and $p'(q_T)$ is the probability density function of queue length distribution under the biased (simulated) input process.

5.2 IS Normalized Variance

Let $\hat{P}$ be the estimated probability of a queue backlog exceeding a buffer size threshold (b). This section derives the equation to characterize the normalized variance of the

estimated probability, $\overline{\sigma^2} \left[\hat{P} \right]$, through an asymptotic approximation: If there is a function $h(b)$ such that

$$\lim_{b \rightarrow \infty} h(b) \overline{\sigma^2} \left[\hat{P} \right] = \text{constant} \quad (63)$$

The normalized variance can then be approximated by simply dividing $h(b)$:

$$\overline{\sigma^2} \left[\hat{P} \right] \approx \frac{\text{constant}}{h(b)} \quad (64)$$

The asymptotic approximation technique is a popular technique, and through simulation experiments (see Figure 5.3), it will be demonstrated that the asymptotic approximation is actually very close to the actual normalized variance observed from simulations, even if b is not close to infinity.

This work has determined that for $h(b) = \Psi b^{-r} \exp(-\tau_{m^*} b^{2(1-H)})$, where $\tau_{m^*} = \frac{(\mu_e - m^*)^{2H}}{2K(H)^2}$, the following is true:

Theorem 1:

$$\lim_{b \rightarrow \infty} \Psi b^{-r} \exp(-\tau_{m^*} b^{2(1-H)}) \left(N \overline{\sigma^2} \left[\hat{P} \right] + 1 \right) = \frac{(\mu_e^{2H})^2}{(\mu_e - m^*)(2\mu_e^{2H} - (\mu_e - m^*)^{2H})} \quad (65)$$

where μ_e is the effective service rate of the queue (service rate minus the mean arrival rate), N is number of simulation replications used in the estimator, and Ψ is a very complex function of H , see [33] for details. (γ and $K(H)$ have been already defined in section 3.1.2). Keep in mind that Equation (65) is for the scenario in which a single node is fed with a FGN arrival process.

With this theorem, one can derive analytical equations to characterize the normalized variance of IS. With a well-defined equation available, this work can analyze the behaviour of IS simulation in terms of normalized variances and determine analytically, the optimal twist (biasing) amount that will minimize the normalized variance.

As stated by [11] [12] [51], the Hurst parameter value for LRD (self-similar) arrival is between 0.5 and 1, while the Hurst value for SRD arrival is between 0 and 0.5. This fact will be very useful in the subsequent derivation.

5.2.1 Deriving IS Normalized Variance

The proof of Theorem 1 is quite involved and uses a change of measure, L'Hôpital's rule, and some properties of integration:

To clarify any potential confusion between b and Q_T : b is defined as the capacity threshold while the event of interest is the event that the queue occupancy Q_T exceeding b . To find the distribution of Q_T , one can take the derivation of one minus Equation (24), which describes $P(Q_T > q_T)$, with respect to q_T :

$$p(q_T) = \frac{\Psi}{\mu_e^{2H-1}} \exp\left(-\tau q_T^{2(1-H)}\right) \left(\gamma q_T^{-\gamma-1} + \tau 2(1-H) q_T^{2(1-H)-\gamma-1} \right) \quad (66)$$

The distribution of the queue length under *biased* input process is then:

$$p'(q_T) = \frac{\Psi}{(\mu_e - m^*)^{2H-1}} \exp\left(-\tau_{m^*} q_T^{2(1-H)}\right) \left(\gamma q_T^{-\gamma-1} + \tau_{m^*} 2(1-H) q_T^{2(1-H)-\gamma-1} \right) \quad (67)$$

As a reminder, the IS biasing used in this thesis is Mean-value Modification, in which v is the mean of the input process. By biasing the input process with a twisted amount of m^* , it essentially reduces the effective service rate by m^* (since the effective service rate

is the service rate of the node minus the input process mean rate), as reflected in equation (67).

Since $I_A(q_T)$ is one if $q_T > b$ and zero otherwise, it is obvious that based on equation (62):

$$N\overline{\sigma^2}\left[\hat{P}\right] = \frac{\int I_A(q_T) \frac{p(q_T)^2}{p'(q_T)} dq_T}{P^2} - 1 = \frac{\int_b^\infty \frac{p(q_T)^2}{p'(q_T)} dq_T}{P^2} - 1 \quad (68)$$

where $\overline{\sigma^2}[\cdot]$ is the normalized variance of the estimation in the bracket.

The proof of *Theorem 1* begins at the left-hand side of Equation (65) of *Theorem 1*:

$$\lim_{b \rightarrow \infty} \Psi b^{-\gamma} \exp(-\tau_{m*} b^{2(1-H)}) \left(N\overline{\sigma^2}\left[\hat{P}\right] + 1 \right) \quad (69)$$

A straightforward substitution of Equation (68) into Equation (69) gives:

$$\lim_{b \rightarrow \infty} \Psi b^{-\gamma} \exp(-\tau_{m*} b^{2(1-H)}) \left(\frac{\int_b^\infty \frac{p(q_T)^2}{p'(q_T)} dq_T}{P^2} \right) \quad (70)$$

The integral in Equation (70) is hard to evaluate, but if one can show that:

$$\int_b^\infty \frac{p(q_T)^2}{p'(q_T)} dq_T \rightarrow 0 \text{ as } b \rightarrow \infty \quad (71)$$

and

$$\lim_{b \rightarrow \infty} \frac{b^\gamma}{\Psi} \exp(\tau_{m*} b^{2(1-H)}) P^2 \rightarrow 0 \text{ as } b \rightarrow \infty \quad (72)$$

then by L'Hôpital's rule, Equation (70) becomes:

$$\lim_{b \rightarrow \infty} \frac{\frac{d}{db} \int_b^{\infty} \frac{p(q_T)^2}{p'(q_T)} dq_T}{\frac{d}{db} \left[\frac{b^\gamma}{\Psi} \exp(\tau_{m^*} b^{2(1-H)}) p^2 \right]} \quad (73)$$

One can then simplify Equation (73) further:

$$\lim_{b \rightarrow \infty} \frac{-\frac{p(b)^2}{p'(b)}}{\frac{d}{db} \left[\frac{b^\gamma}{\Psi} \exp(\tau_{m^*} b^{2(1-H)}) p^2 \right]} \quad (74)$$

To prove Theorem 1 based on Equation (74), one must first show that Equation (72)

is true. Substitute Equation (24) into the left-hand side of Equation (72) to obtain:

$$\begin{aligned} & \lim_{b \rightarrow \infty} \frac{b^\gamma}{\Psi} \exp(\tau_{m^*} b^{2(1-H)}) \left[\frac{\Psi}{b^\gamma (\mu_e)^{2H-1}} \exp(-\tau b^{2(1-H)}) \right]^2 \\ &= \lim_{b \rightarrow \infty} \frac{\Psi}{b^\gamma \mu_e^{4H-2}} \exp(-\tau_{m^*} b^{2(1-H)}) = 0 \end{aligned} \quad (75)$$

where $\tau_{m^*} = \frac{(2(\mu_e^{2H}) - (\mu_e - m^*)^{2H})}{2K^2(H)}$.

The limit in Equation (75) is zero since $\gamma = \frac{1-H}{H}(2H-1) > 0$ and $2(1-H) > 0$ for

$0.5 < H < 1$. As a result, both b^γ and $b^{2(1-H)}$ approaches infinity as b approaches infinity.

With Equation (72) proven to be true, we still have to prove the condition in Equation

(71). One first has to find $\frac{p(q_T)^2}{p'(q_T)}$. By substituting equations (66) and (67) into $\frac{p(q_T)^2}{p'(q_T)}$,

one obtains:

$$\frac{p(q_T)^2}{p'(q_T)} = \frac{\exp(-2\tau q_T^{2(1-H)}) \left(\gamma q_T^{-\gamma-1} + \tau 2(1-H) q_T^{2(1-H)-\gamma-1} \right)^2 \left(\frac{\Psi}{u_e^{2H-1}} \right)^2}{\exp(-\tau_{m^*} q_T^{2(1-H)}) \left(\gamma q_T^{-\gamma-1} + \tau_{m^*} 2(1-H) q_T^{2(1-H)-\gamma-1} \right) \frac{\Psi}{(u_e - m^*)^{2H-1}}} \quad (76)$$

Since the case of interest is when q_T approaches infinity: $\lim_{q_T \rightarrow \infty} \frac{p(q_T)^2}{p'(q_T)}$, therefore, due

to the fact that $\gamma = \frac{1-H}{H}(2H-1) > 0$ for $0.5 < H < 1$, as a result, $\gamma q_T^{-\gamma-1} \rightarrow 0$ as $q_T \rightarrow \infty$.

This enables the further simplification of Equation (76):

$$\frac{p(q_T)^2}{p'(q_T)} = 2(1-H)\Psi \tau_{1/m^*} \exp(-\tau_{m^*} q_T^{2(1-H)}) q_T^{\frac{1}{H}-2} \quad (77)$$

$$\text{where } \tau_{1/m^*} = \frac{1}{2K^2(H)} \left(\frac{\mu_e^2}{\mu_e - m^*} \right).$$

Clearly, $\frac{p(q_T)^2}{p'(q_T)}$ is a function that goes to zero faster than the Weibull function. As

the Weibull function is integrable, therefore the integral in Equation (71) goes to zero with increasing b .

Equation (74) can now be used. Substitute Equation (77) into Equation (74), Equation (74) becomes:

$$\begin{aligned} & \lim_{b \rightarrow \infty} \frac{-2(1-H)\Psi \tau_{1/m^*} \exp(-\tau_{m^*} b^{2(1-H)}) b^{\frac{1}{H}-2}}{\frac{d}{db} \left[\frac{b^\gamma}{\Psi} \exp(\tau_{m^*} b^{2(1-H)}) P^2 \right]} \\ &= \frac{(\mu_e^{2H})^2}{(\mu_e - m^*) (2(\mu_e^{2H}) - (\mu_e - m^*)^{2H})} \\ &= \text{Right hand side of Equation (65) in Theorem 1.} \end{aligned}$$

Q.E.D.

To summarize, Theorem 1 is proved using a change of measure, L'Hôpital's rule and integration properties to simplify the difficult integral in Equation (4).

From Theorem 1, one can obtain a closed-form formula for the approximation of the normalized variance of IS, for a node with a constant service rate and a single FGN arrival process:

$$\overline{\sigma^2} \left[\hat{P} \right] \approx \frac{1}{N} \left[\frac{\mu_e^{2H} b^\gamma \tau \exp(\tau_{m^*} b^{2(1-H)})}{(\mu_e - m^*) \tau_{-m^*} \Psi} - 1 \right] \quad (78)$$

With an analytical equation defined for the normalized variance, one can perform analysis on the behaviour of IS simulations under different simulation conditions (section 5.2.4) and calculate the optimal twist amount for IS under LRD input (section 5.2.3).

5.2.2 Derived IS Normalized Variance Verification

By comparing the normalized variance calculated with the analytical Equation (78) and the normalized variance of the IS simulation, one can verify the derived equation if the results agree with each other.

Figure 5.3 clearly shows that the analytical results are almost a perfect match with the simulation results (the simulation uses a fluid traffic model and conducted three different experiments with different parameter values), thus, Equation (78) is an accurate characterization of the IS normalized variance. Notice that the twist amount in the figure stops at the effective service rate of the experiment. As explained before, theoretically [52], if the twist amount exceeds the effective service rate, then the queue becomes

unstable and the probability of the queue occupancy surpassing the threshold becomes one.

Figure 5.3: Analytical IS Normalized Variance and Simulation IS Normalized Variance

However, in actual simulations, it is impossible for each simulation replication to be infinity long enough to counteract the effect of the warm-up period [52]. This is why one might observe that there are small differences between the analytical results and the simulation results.

Graphically, both the simulation results and the analytical results have a “bowl” shape (consistent with observed normalized variance in other IS simulation experiments [41] [53]) with the difference being that the “bowl” of the simulation results has a larger width than that of the analytical results. Figure 5.4 shows that as the length of a replication approaches infinity, the width of the “bowl” becomes smaller, and thus, closer to the analytical results under steady state.

Thus, as the simulation replication time approaches infinity, one can expect that the normalized variance from the simulation will match the analytical normalized variance completely.

5.2.3 IS Optimal Twist Amount

Intuitively, for IS simulation, as one increases the mean of the simulation (biased) distribution, more events will occur and therefore decrease the normalized variance. However, at the same time, the simulation distribution is moving further away from the original distribution, thus creating more uncertainty and increasing the normalized variance. Consequently, these two opposing forces result in a “bowl” shaped variance with an optimal twist amount at the bottom of the “bowl” [53] [41]. This can also be seen in Figure 5.3 and Figure 5.4.

Figure 5.4: Simulation IS Normalized Variance and Simulation Replication Length

One can determine the optimal twist value that will minimize the normalized variance by taking the derivative of equation (78) with respect to the twisted amount m^* , and find the minimum. The mechanics of algebraic manipulation are straightforward and will be skipped here. The optimal twist amount m^*_{opt} is:

$$m^*_{opt} = \mu_e - \left[\mu_e^{2H} + \left(1 + \frac{1}{2H}\right)\Gamma - \mu_e^{2H} \sqrt{1 + \frac{\left(2 - \frac{1}{H}\right)\Gamma}{\mu_e^{2H}} + \frac{\left(\frac{1}{8H} + \frac{1}{16H^2}\right)^2}{\mu_e^{4H}} \Gamma^2} \right]^{\frac{1}{2H}} \quad (79)$$

Note that for high H values, the bottom of this “bowl” in Figure 5.5 is very flat. Therefore, when choosing a simulation distribution for IS, it is not necessary to find the exactly optimal twist value. As long as the twist amount is close to the optimal value, the normalized variance will be small. This is further demonstrated in the Figure 5.6.

Shown in Figure 5.6, the normalized variance of IS estimation under optimal twist amount is always the lowest: its data line is at the bottom. As H becomes large, the data lines starts to overlap each other indicating that a small deviation with respect to the optimal twist amount will not degrade the normalized variance significantly. This agrees with the observation in Figure 5.5 that the “bowl” has a flatter bottom for high H values.

Figure 5.5: Analytical IS Optimal Twist Amount

Figure 5.6: Normalized Variance under Optimal Twist Amount and Non-optimal Twist Amount

5.2.4 IS Simulation Behaviour under LRD input

Many other interesting properties of IS simulation under a LRD input can be observed with the help of equations (78) and (79): for example, how does the service rate or the backlog threshold of the queue in the IS simulation affect the optimal twist value?

As the effective service rate $\mu_e \rightarrow \infty$, Equation (79) becomes

$$m_{opt}^* \approx \mu_e - \left[\mu_e^{2H} + \left(1 + \frac{1}{2H} \right) \Gamma - \mu_e^{2H} \right]^{\frac{1}{2H}} \quad (80)$$

$$m_{opt}^* \approx \mu_e \quad (81)$$

From Equation (81), the optimal twist amount m_{opt}^* approaches the effective service rate μ_e . This is demonstrated graphically in Figure 5.7.

Figure 5.7 clearly shows that as the effective service rate increases, the relationship between the optimal twist amount and the effective service rate becomes linear. The explanation is that as the effective service rate increase (the difference between the service rate and the arrival rate increases), the event of queue backlog exceeding the threshold becomes rarer. Thus, the biasing, represented in the optimal twist amount, must “catch up” to the ever-decreasing occurrences of events by making them occur more often. Thus, the two quantities eventually have a linear relationship.

Figure 5.7: IS Optimal Twist Amount vs. Effective Service Rate

Let v'_{opt} be the optimal mean for the simulation distribution that uses the optimal twist amount: $v'_{opt} = v + m^*_{opt}$ (v is the mean of the original distribution). Since the effective service rate is the service rate minus the arrival rate ($\mu_e = \mu - v$):

$$\begin{aligned}
v'_{opt} &= \mu - \mu_e + m^*_{opt} \\
&= \mu - \mu_e + \mu_e - \left[\mu_e^{2H} + \left(1 + \frac{1}{2H}\right)\Gamma - \mu_e^{2H} \sqrt{1 + \frac{\left(2 - \frac{1}{H}\right)\Gamma}{\mu_e^{2H}} + \frac{\left(\frac{1}{8H} + \frac{1}{16H^2}\right)^2}{\mu_e^{4H}} \Gamma^2} \right]
\end{aligned} \tag{82}$$

As $\mu_e \rightarrow \infty$, equation (82) becomes

$$\begin{aligned}
v'_{opt} &= \lim_{u \rightarrow \infty} v_{0_{opt}} = \mu - \mu_e + \mu_e - \left[\mu_e^{2H} + \left(1 + \frac{1}{2H}\right)\Gamma - \mu_e^{2H} \right] \\
v'_{opt} &= \mu - \left(1 + \frac{1}{2H}\right)\Gamma
\end{aligned} \tag{83}$$

where $\Gamma = \frac{K(H)^2}{b^{2(1-H)}}$.

One can conclude that as the effective service rate μ_e increases (the input rate decreases or the node service rate increases), the mean of the optimal simulation distribution eventually “saturates” (stops changing with respect to μ_e) after the event becomes extremely rare. This is demonstrated graphically in Figure 5.8.

Figure 5.8 shows the optimal mean for the simulation distribution decreases first, and then stops after the effective service rate becomes very large. This is understandable since as the events become extremely rare, decreasing the mean of the optimal simulation distribution by an arbitrarily small amount can move the simulation distribution a little closer to the true distribution, but at the same time, fewer events would occur. Therefore, the optimal mean stays at this “equilibrium point”.

Figure 5.8: IS Optimal Mean vs. Effective Service Rate

Figure 5.9: IS Optimal Twist Amount vs. Buffer Threshold Size

Keep in mind that the optimal mean does not guarantee a low variance; it only provides a “best among the bad” variance. The variance under optimal simulation distribution can still be very large, as shown in Figure 5.6.

Note that as $b \rightarrow \infty$, $\Gamma = \frac{K(H)^2}{b^{2(1-H)}} \rightarrow 0$, then Equation (80) becomes:

$$m^*_{opt} \approx \mu_e - \left[\mu_e^{2H} - \mu_e^{2H} \right]^{\frac{1}{2H}}$$

$$m^*_{opt} \approx \mu_e$$

Thus, the optimal twist amount is no longer affected by b . This is demonstrated in Figure 5.9.

Clearly, as H approaches one, the lines in Figure 5.9 become “straighter”. This is expected since $\Gamma = \frac{K(H)^2}{b^{2(1-H)}} = K(H)^2$ if $H=1$. Thus, Γ becomes independent of b .

The following figure shows the changes in optimal twist amount as the arrival becomes burstier or more LRD (H increases).

Figure 5.10: IS Optimal Twist Amount vs. Hurst Parameter

Figure 5.10 shows how the optimal twist amount changes with respect to the Hurst parameter. It is understandable that if the input process was burstier, then more backlog

exceeding threshold events would occur; therefore, not too much biasing/twisting is needed. Thus, one can see that the optimal twist amount decreases with respect to the Hurst parameter.

This concludes the analysis on the normalized variance of IS simulation. The next section covers the simulation efficiency of IS, and one can see that the efficiency criteria relates closely to the variances.

5.3 IS Simulation Efficiency

To show that IS is efficient under LRD input in a single node scenario, the simulation efficiency criteria stated by [9] is used (this was introduced in section 3.3). A rarity parameter must be determined first. Clearly, the buffer size threshold (b) is a rarity parameter: the larger b becomes, the less likely that the queue occupancy will exceed the threshold b . In other words, as b approaches infinity, the probability that the queue length exceeds b approaches zero.

With the Monte Carlo simulation, the effort to obtain confidence intervals of desired accuracy for a rare event probability grows to infinity as the event becomes rarer. For an asymptotically optimal importance-sampling estimator, the F in Equation (11) should decrease while P decreases, so N does not have to increase to keep the normalized variance low. This implies that the simulation cost stays the same. The approach can then be viewed as efficient. Thus, in the remaining of the analysis, N is fixed in all the equations.

As stated before, the IS biasing used in this thesis is the Mean-value Modification biasing in which the mean of the true distribution and the mean of the simulation distribution differ by a twisted amount (m^*). Therefore, the condition under which the IS

simulation will be efficient, under LRD input, is defined in terms of the value the twist amount must take on so the IS simulation satisfies the efficient criteria (see equations (84) and (85)).

The rarity parameter chosen in this simulation efficiency analysis is $b' = b^{2(1-H)}$. Clearly, as $b \rightarrow \infty$, $b' \rightarrow \infty$, since $0.5 < H < 1$ for self-similar (LDR) random processes; and as $b' \rightarrow \infty$, $P \rightarrow 0$. Thus, the efficiency criteria becomes

$$\lim_{b' \rightarrow \infty} \frac{1}{b'} \log(F_{b'}) = -R \quad (84)$$

$$\lim_{b' \rightarrow \infty} \frac{1}{b'} \log(P_{b'}) = -I \quad (85)$$

The IS formulation is asymptotically efficient if $R=2I$.

Clearly, under MM biasing, not all twist amounts will yield an efficient simulation, especially as b' approaches infinity (making the event rarer). The next theorem shows under what condition IS will be efficient.

Theorem 2:

$R=2I$, if the twist amount (m^*) equals to the effective service rate (μ_e).

As it will be shown, $m^* = \mu_e$ is the optimal twist amount that was calculated in Equation (79) (in section 5.2.3) when b approaches infinity. This matches results from two different perspectives: the efficiency perspective and the minimal variance perspective. There will be more discussion on this topic later.

To explicitly calculate R (if the limit exists), a change of measure is performed on Equation (9) just like for Equation (4). This is possible since the condition that was proved in the section 5.1 still holds true in this case. However, L'Hôpital's rule is not

utilized here because of the logarithmic function in R : taking the derivative of

$\log\left(\int_b^\infty \frac{p(q_T)^2}{q(q_T)} dq_T\right)$ will not remove the integration operation, as it will be shown below:

$$F_{b'} = \int \left[I_{A_{b'}}\left(\vec{x}\right) \frac{f\left(\vec{x}, v\right)}{f\left(\vec{x}, v'\right)} \right]^2 f\left(\vec{x}, v'\right) d\vec{x} = \int \left[I_A(q_T) \frac{p(q_T)}{p'(q_T)} \right]^2 p'(q_T) dq_T \quad (86)$$

Since $I_A(\bullet)$ is an indicator function that is one if queue length q_T is greater than the threshold b and zero otherwise, therefore:

$$F_{b'} = \int \frac{I_A(q_T) p(q_T)^2}{p'(q_T)} dq_T \Leftrightarrow F_{b'} = \int_b^\infty \frac{p(q_T)^2}{p'(q_T)} dq_T \quad (87)$$

The left hand side of Equation (84) becomes:

$$\lim_{b \rightarrow \infty} \frac{\log\left(\int_b^\infty \frac{p(q_T)^2}{p'(q_T)} dq_T\right)}{b^{2(1-H)}} \quad (88)$$

Taking the derivative in the numerator and denominator in equation (88) clearly will not remove the integration operation, and it only will make the equation more complicated.

Without L'Hôpital's rule, one has to integrate $\frac{p(q_T)^2}{p'(q_T)}$ and then perform the logarithmic function on the result. However, $\frac{p(q_T)^2}{p'(q_T)}$ based on Massoulie's Equation (24) is too complex for integration, and in contrast, the probability equation provided by Norros [32] [50], Equation (22), provides the same results as the Massoulie's equation, and it is simpler and integration is possible. [50] has stated that Equation (22) is in fact

Figure 5.11: Analytical Probability and Simulation Probability Comparison

logarithmically accurate for large b , shown by Duffield and O'Connell [10], with techniques of the theory of large deviation. Since this analysis is interested in the case of b approaching infinity, Norros' equation fits the need very well. Furthermore, as demonstrated in Figure 5.11, via simulation, the result of Equation (22) is still very close to that of the Massoulié's equation for small b values:

Clearly, the results according to Norros and Massoulié are very close; therefore, one is able to replace the Massoulié equation with the Norros equation so integration is possible.

Norros' Equation yields:

$$P\{Q_T > q_T\} = 1 - \varsigma \exp(-\tau q_T^{2(1-H)}) \quad (89)$$

where ς is some quantity independent of μ_e and q_T [32] [50]. The approach to find ς has been discussed in section 3.1.2.

By taking the derivative of Equation (89) with respect to q_T , the pdf of the queue length under the original input is:

$$p(q_T) = \varsigma(2(1-H)\tau)q_T^{2(1-H)-1} \exp(-\tau q_T^{2(1-H)}) \quad (90)$$

The pdf of the queue length under biased input is then:

$$p'(q_T) = \varsigma(2(1-H)\tau_{m^*})q_T^{2(1-H)-1} \exp(-\tau_{m^*} q_T^{2(1-H)}) \quad (91)$$

$\frac{p(q_T)^2}{p'(q_T)}$ becomes:

$$\frac{p^2(q_T)}{p'(q_T)} = 2\varsigma(1-H)\tau_{1/m^*}(q_T^{2(1-H)-1}) \exp(-\tau_{m^*} q_T^{2(1-H)}) \quad (92)$$

Integrating Equation (92) from b to infinity and using some arithmetic simplifications:

$$R = -\lim_{b \rightarrow \infty} \frac{1}{b^{2(1-H)}} \log \frac{\zeta \left(\frac{(\mu_e^{2H})^2}{(\mu_e - m^*)^{2H}} \right)}{2K^2(H)\tau_{-m^*}} + \frac{1}{b^{2(1-H)}} \log [\exp(-\tau_{-m^*} b^{2(1-H)})] \quad (93)$$

$$R = \tau_{-m^*} = \frac{2\mu_e^{2H} - (\mu_e - m^*)^{2H}}{2K^2(H)}$$

After obtaining R , I must be explicitly determined (if the limit exists). To do this, substitute Equation (22) into Equation (85):

$$I = -\lim_{b \rightarrow \infty} \frac{1}{b^{2(1-H)}} \log [\zeta \exp(-\tau b^{2(1-H)})] \quad (94)$$

$$= \lim_{b \rightarrow \infty} -\frac{1}{b^{2(1-H)}} \log(\zeta) + \tau$$

$$= \tau = \frac{\mu_e^{2H}}{2K^2(H)} \quad (95)$$

To determine the condition so that IS is asymptotically efficient under the FGN input, a twist amount m^* has to be found, such that $R=2I$. Observe that in the case of the twisted amount (m^*) equals the effective service rate (μ_e), R becomes

$$R = \frac{2\mu_e^{2H}}{2K(H)^2}$$

$$= 2I$$

Q.E.D.

This completes the proof of Theorem 2. Theorem 2 implies that, under the condition that the twist amount m^* of the IS simulation equals to μ_e , IS simulation under self-similar (LRD) arrival, is asymptotically efficient.

Going back to the equation for the optimal twist amount, Equation (79), examine what happens to the value of m^*_{opt} as b approaches infinity. Since $\Gamma = \frac{K(H)^2}{b^{2(1-H)}}$, as b approaches to infinity, Γ becomes zero. As a result, the optimal twist amount becomes:

$$m^*_{opt} = \mu_e - \left[\mu_e^{2H} - \mu_e^{2H} \right]^{\frac{1}{2H}} = \mu_e \quad (96)$$

Equation (96) shows that the optimal twist amount under an asymptotically large b is the same as the asymptotic simulation efficiency condition that was derived in Theorem 2. Intuitively, when the probability is extremely small, one wants to use the optimal twist amount to minimize the normalized variance so the simulation stays efficient.

Meeting the efficiency criteria means that F_b , representing the IS biasing in the variance, decays at a rate equal to that of P^2 , as the rarity parameter approaches infinity. Because of this compensation, the variance will not increase and, in turn, one does not need more replications (does not have to increase N) to estimate P .

The intuition behind this result is that as the queue backlog threshold approaches infinite, the event inevitably becomes extremely rare; thus, one has to bias the original distribution as much as one can so that more events can occur, and μ_e is the maximum value the twist amount can take on.

The next section discusses the efficiency of EPIT, which is based on ISPE. Because there is no existing efficiency criteria for ISPE, the thesis shall demonstrate that ISPE is efficient under LRD input in a manner that is consistent with the existing efficiency criteria for IS: first, the normalized variance of EPIT is derived, just like in this section. Then, with a fixed computation cost (N), it will be shown that the normalized variance of

EPIT grows with a speed that is moderate compared to the rate of decrease on the probability. Thus, the variance will not increase exponentially.

5.4 EPIT Efficiency Analysis

Just like the IS efficiency analysis, the efficiency analysis of EPIT is studied on a single node fed a FGN arrival process. In order to determine whether EPIT is efficient under LRD input, one first has to derive the normalized variance of EPIT.

This work has found that for $h(b) = \Psi b^{-\gamma} \exp(-\tau b^{2(1-H)})$, the following is true:

Theorem 3

$$\lim_{b \rightarrow \infty} \Psi b^{-\gamma} \exp(-\tau b^{2(1-H)}) \left(N \overline{\sigma^2} \left[\hat{P} \right] + 1 \right) = \frac{((\mu_e + m^*)^{2H})^2}{\mu_e (2(\mu_e + m^*)^{2H} - \mu_e^{2H})} \quad (97)$$

$\overline{\sigma^2} \left[\hat{P} \right]$ in this case is the normalized variance of the prediction made by EPIT (in

Theorem 2, this was the normalized variance of the prediction made by IS). Furthermore,

$h(b)$ in *Theorem 3* is different from $h(b)$ in *Theorem 2*.

5.4.1 Deriving EPIT Normalized Variance

With *Theorem 3*, one can derive the analytical equation to characterize the normalized variance of EPIT in the single node scenario. With a well-defined equation available, one can analyze the behaviour of EPIT simulation in terms of normalized variances. Specifically, with respect to a decreasing ingress-traffic rate (ingress traffic rate = original ingress traffic rate – m^*), we observe the rate of change of the normalized

variance in comparison to the rate of change of the probability, to see if the EPIT prediction can remain accurate for a long range of probabilities and thus remain efficient.

It is straightforward to derive the distribution of Q_T under the original input distribution ($p(q_T)$) and under the biased distribution ($p'(q_T)$), by using the probability formula provided by Massoulie and Simonian [33]. It is similar to that which was used in the IS case but with a *key* difference: in IS, ν' (the mean of the biased input distribution) is varied and ν (the mean of the original input distribution) is fixed. In EPIT, which is based on ISPE, ν' is fixed and ν is varied (the difference between the two means is the twisted amount (m^*)). For this reason, the $p(q_T)$ and $p'(q_T)$ shall be different from that which was derived in section 5.2, even though both cases use the Massoulie equation.

$$p'(q_T) = \frac{\Psi}{\mu_e^{2H-1}} \exp\left(-\tau q_T^{2(1-H)}\right) \left(\gamma q_T^{-\gamma-1} + \tau 2(1-H) q_T^{2(1-H)-\gamma-1} \right) \quad (98)$$

$$p(q_T) = \frac{\Psi}{(\mu_e + m^*)^{2H-1}} \exp\left(-\tau_{+m^*} q_T^{2(1-H)}\right) \left(\gamma q_T^{-\gamma-1} + \tau_{+m^*} 2(1-H) q_T^{2(1-H)-\gamma-1} \right) \quad (99)$$

where $\tau_{+m^*} = \frac{(\mu_e + m^*)^{2H}}{2K^2(H)}$.

Note that increasing the effective service rate μ_e by the twist amount (m^*) is the same as reducing the ingress traffic rate by m^* (*the IS case in section 5.2 increases the traffic rate by m^**), since the effective service rate equals to the service rate minus the ingress traffic rate. This fits into the framework that EPIT predicts the QoS in the network as a function of different ingress traffic rates.

The rest of the proof of Theorem 3 follows the same step as the proof for Theorem 1: determine that both the numerator and denominator of the left-hand side of Equation (97)

approach zero as b approaches infinity, then apply L'Hôpital's rule to simplify the equation and the removal of the integration operation. Thus, the detailed steps will be omitted here:

$$\begin{aligned} & \lim_{b \rightarrow \infty} \Psi b^{-\gamma} \exp(-\tau b^{2(1-H)}) \left(N \sigma^2 \left[\frac{\Lambda}{P} \right] + 1 \right) \\ &= \lim_{b \rightarrow \infty} \frac{-\frac{p(b)^2}{p'(b)}}{\frac{d}{db} (\Psi b^\gamma \exp(\tau b^{2(1-H)}) P^2)} \end{aligned} \quad (100)$$

With

$$\frac{p(q_T)^2}{p'(q_T)} = 2(1-H) \Psi \tau_{+1/m^*} \exp(-\tau_{*m^*} q_T^{2(1-H)}) q_T^{\frac{1}{H}-2} \quad (101)$$

$$\text{where } \tau_{+1/m^*} = \frac{1}{2K^2(H)} \left(\frac{(\mu_e + m^*)^2}{\mu_e} \right), \tau_{*m^*} = \frac{(2(\mu_e + m^*)^{2H}) - \mu_e^{2H}}{2K^2(H)}.$$

Equation (100) becomes:

$$\lim_{b \rightarrow \infty} \frac{-2(1-H) \Psi \tau_{+1/m^*} \exp(-\tau_{*m^*} b^{2(1-H)}) b^{\frac{1}{H}-2}}{\frac{d}{db} \left[\frac{b^\gamma}{\Psi} \exp(\tau b^{2(1-H)}) P^2 \right]} \quad (102)$$

By substituting P with equation (24) and some arithmetic manipulations, equation (102) becomes:

$$\frac{((\mu_e + m^*)^{2H})^2}{\mu_e (2(\mu_e + m^*)^{2H} - \mu_e^{2H})}$$

= Right-hand side of Equation (97) of Theorem 3.

Q.E.D.

From Theorem 3, one can obtain a closed-form formula approximating the normalized variances of EPIT, for a single node with a constant service rate and a single FGN arrival process:

$$\overline{\sigma^2} \left[\hat{P} \right] \approx \frac{1}{N} \left[\frac{\left[(\mu_e + m^*)^{2H} \right]^2 b^r \exp(\tau b^{2(1-H)})}{\mu_e 2K^2(H) \tau_{*m^*} \Psi} - 1 \right] \quad (103)$$

5.4.2 Derived EPIT Normalized Variance Verification

Simulation experiments are conducted to verify the analytical normalized variance in Equation (103), for single node system with LRD input, as in Figure 5.12. The figure clearly shows that analytical normalized variance from derived Equation (103), through asymptotic approximation, agrees with the simulation-measured normalized variance, even when the queue backlog threshold (b) is not close to infinity.

Figure 5.12: Analytical EPIT Normalized Variance and Simulation EPIT Normalized Variance

5.4.3 Normalized Variance based Efficiency Analysis

To analyze the efficiency of EPIT, one also has to observe how the overflow probability decreases with respect to decreasing ingress-traffic rate. Decreasing the ingress-traffic rate by m^* is the same as increasing the effective service rate μ_e with m^* . Thus, one can view each μ_e in equation (24) replaced with $\mu_e + m^*$. Figure 5.13 plots the overflow probability based on the Massoulie equation, Equation (24), and the overflow probability observed from simulation.

Figure 5.13 shows that the simulation result matches that of Equation (24).

Thus, by observing Equation (24), with μ_e replaced with $\mu_e + m^*$, we have the following corollary:

Corollary 1: The overflow probability in a single node, with LRD input, decreases as $O(m^{*1-2H} \exp(-m^{*2H}))$.

Figure 5.13: Analytical Probabilities of Overflow Based on Massoulie Equation and Simulation Probabilities of Overflow

By examining Equation (103) for the normalized variance, we have the following corollary:

Corollary 2 The normalized variance of EPIT in a single serve queue, with LRD input, grows as $O(m^{*2H})$.

From the two corollaries above, it should be noted that the normalized variance of EPIT grows with a speed that is moderate compared to the rate of decrease on the probability of overflow, with respect to decreasing ingress traffic rate (represented in m^*). One can see this intuitively by comparing Figure 5.12 and Figure 5.13 (note that the probability in Figure 5.13 are in logarithmic scale). This provides the range for EPIT (ISPE) applications for the prediction to remain accurate as shown in EPIT in Chapter 6. This also demonstrates that EPIT is efficient, under LRD input, for a wide range of probabilities.

The next chapter shall be the simulation experiments used to verify the effectiveness of EPIT in some sample applications, such as the VPN and P2P networks. If the process of showing the prediction of EPIT on end-to-end QoS quantity is accurate, there will be two conclusions from those experiments: 1) EPIT is effective in predicting the end-to-end overflow probability in a complex network, and the solution provided by SA-EPIT does satisfy the end-to-end QoS requirement. 2) With a fixed computation cost (the number of simulation replications is fixed), the EPIT remains accurate for a wide range of probabilities (even those probabilities that are very small). This implies the variance of the EPIT prediction has remained small. For this reason, this work concludes that EPIT is

efficient, under LRD input, in a multiple-queue network (EPIT was analytically proven efficient in this section in a single node network.)

Chapter 6 Simulation Experiments

This chapter reports on the extensive set of simulations conducted to validate the proposed approach. The simulations in this section allow us to investigate the effectiveness of EPIT and SA-EPIT under two different environments: the VPN network and the P2P network. As a way to validate the approach, prediction results from EPIT are compared with simulation results using the Monte Carlo approach. In the end, the traffic engineering solution proposed by SA-EPIT is implemented in the network, and the new end-to-end QoS is measured to see if the predefined QoS target is met.

From another perspective, while it is impossible to make the probability asymptotically small in simulation, work in this chapter has decreased the original overflow probability by *three* orders of magnitude (achieved by reducing the ingress traffic rates), to see if the EPIT prediction remains accurate (i.e., if the variance remains small) with a fixed number of replications ($N=50,000$). If that is true, EPIT is efficient for a wide range of probabilities in multi-queue systems with LRD input.

6.1 Basic Network Model

A fluid traffic model based simulator is used in this thesis. With the fluid model, the amount of traffic arrived and served in each discrete time interval is treated as real numbers (the amount of fluid entering or being served), which makes the buffer

occupancy a continuous random variable. This model is frequently used [54] [55]. [56] has shown that the largest difference between the fluid approximation and the discrete (packet) model, in terms of the departure time of a packet, is one maximum packet length divided by the service rate of the node. In terms of the amount of traffic served, the difference is simply the max packet length. Given the high link capacities of networks today, typically in Gigabit per second range or Megabit per second range, the impact the fluid traffic model has on the result is therefore negligible.

The simulator for this thesis was built using Matlab and Microsoft Visual C++ 6.0. The simulator consists of multiple nodes with configurable topology and multiple traffic sources with adjustable parameters. The queue behaviour of each node in the network is modeled by the Lindley equation.

$$Q_t = (Q_{t-1} + x(t) - \mu)^+ \quad (104)$$

Q_t is the queue length or the backlog at time interval/slot t , $x(t)$ is the amount of traffic arrived in time slot t , and μ is the service rate of the queue, i.e. link capacity. The amount of traffic (measured in traffic units) that enters the network from a traffic source, in one time unit, follows a FGN distribution over a sequence of time units.

While it is tempting to use real traffic traces in the simulation, but because this work is focused on the aggregated traffic flows in a core network, it is very difficult to obtain core network traffic traces from carriers. Even if such a trace is obtained, SA-EPIT has to modify the mean rate of the traffic to achieve QoS target but the mean in a recorded

traffic trace is fixed. For this reason, a traffic generator in the simulator will generate the ingress traffic flows.

As these theses is focused on the traffic engineering aspect of the problem, and not concentrate on the traffic modeling aspect, it is assumed that a traffic model is given, and parameters that describe the traffic have been estimated correctly and can be safely used by EPIT to calculate the likelihood ratio. Chapter 7 outlines some existing approaches to estimate the parameters of FGN traffic, but it does not go into too much detail since they are out of the scope of this thesis.

In the simulation, an event occurs when the backlog Q_t exceeds a threshold (may or may not be the buffer size) at the end of a replication. The event can also be the case that the end-to-end delay exceeds a threshold, or end-to-end jitter exceeds a threshold. If any of the nodes on an end-to-end path has an event, the path as a whole is considered to have had an end-to-end event. Each replication is sampled on whether an event occurred, and these samples are used to determine the end-to-end QoS based on the probability of those events occurring.

To verify that the Hurst parameter of the generated LRD traffic flow, in the simulation, is actually the Hurst parameter value one desires, one can apply any of the Hurst parameter measurement approaches mentioned in section 7.4 to the traffic traces generated in the simulation. This thesis has indirectly verified that the actual Hurst parameter of the generated traffic matches the values set by the user by comparing the simulation queuing results and analytical queuing results, as shown in Figure 5.13. Had

there been a significant discrepancy between the intended Hurst parameter and the actual Hurst parameter, the results would not have matched.

The length of each replication in the simulation run has to be longer than the warm up period so the system will be in steady state at the end of the replication. The warm up period is calculated using results from [41]. It then verified by repeating the simulation run several times with different lengths larger than the calculated length from [41], to make sure the system is in steady state (i.e. the probability of overflow does not vary drastically).

6.2 Virtual Private Network

To formulate a VPN network as a traffic-engineering problem, let TQ be a statistical QoS constraint for one or multiple end-to-end paths used by the VPN. $v_1, v_2, \dots, v_S$ could represent the bandwidths the VPN service provider contracted to its customers and thus the amount of aggregated traffic expected from the customer VPN site. EPIT can be used to predict the statistical QoS performance P based on $v_1, v_2, \dots, v_S$ and SA-EPIT can offer a traffic engineering solution so that, for all the end-to-end paths of interest, $P(v_1, v_2, \dots, v_S) \leq TQ$ while the traffic entering the network is maximized.

6.2.1 VPN Topology and Network Parameters

Figure 6.1: VPN Topology for Experiment 1

Shown in Figure 6.1, each CE (Customer Edge, $CE_{vpn\#,site\#}$) is connected to a PE (Provider Edge) of the VPN provider. The PE-PE paths are to provide secure and reliable connectivity among the customer sites ($CE_{1,1}$ to $CE_{1,2}$, $CE_{2,1}$ to $CE_{2,2}$, $CE_{3,1}$ to $CE_{3,2}$). Since this thesis is focused on the performance of the network, the security issues of VPN will not be covered here.

Figure 6.2 shows the VPN network in Figure 6.1 is built on top of a core provider network consists of only backbone routers. This core provider network is a complicated network based on a real network topology: the National Science Foundation Network (NSFNET). This topology is frequently used in network simulations.

Figure 6.2: The Core Provider Carrier Network Topology

The neighbouring node-connections are modeled as bi-directional links. The CE nodes act as traffic sources/sinks that send and receive traffic from the PE nodes. Its traffic is the aggregated traffic from its attached VPN site. The other traffic sources models the non-VPN traffic that is passing through the publicly shared core provider network. It is therefore reasonable to assume that each traffic source in this model is independent of others.

The internal network traffic flows can be identified by the routing matrix

$$\vec{\Lambda} = \begin{bmatrix} \theta_{0,0} & \theta_{0,1} & \dots & \theta_{0,14} \\ \theta_{1,0} & \theta_{1,1} & \dots & \theta_{1,14} \\ \dots & \dots & \dots & \dots \\ \theta_{14,1} & \theta_{14,1} & \dots & \theta_{14,14} \end{bmatrix} \text{ where } \theta_{i,j} \text{ is the percentage of traffic received by node } i$$

that is sent to node j . The actual assignments of the non-zero elements are shown in Table 6.1

Table 6.1: NSFNET Network Routing Matrix Non-zero Entries

i	j	$\theta_{i,j}$	I	j	$\theta_{i,j}$	i	j	$\theta_{i,j}$
1	1	0.32	6	5	0.26	10	14	0.08
1	2	0.19	6	6	0.43	10	15	0.18
1	3	0.19	6	8	0.31	11	7	0.3
1	8	0.3	7	3	0.21	11	10	0.24
2	1	0.2	7	5	0.19	11	11	0.45
2	2	0.37	7	7	0.24	12	9	0.24
2	3	0.15	7	11	0.16	12	10	0.31
2	4	0.29	7	13	0.19	12	12	0.31
3	1	0.19	8	1	0.26	12	13	0.14
3	2	0.14	8	6	0.15	13	7	0.32
3	3	0.35	8	8	0.25	13	12	0.19
3	7	0.33	8	10	0.34	13	13	0.41
4	2	0.2	9	4	0.32	13	14	0.08
4	4	0.28	9	9	0.32	14	9	0.17
4	5	0.24	9	12	0.26	14	10	0.21
4	9	0.28	9	14	0.11	14	13	0.1
5	4	0.24	10	8	0.27	14	14	0.52
5	5	0.31	10	10	0.19	15	10	0.48
5	6	0.22	10	11	0.1	15	15	0.52
5	7	0.22	10	12	0.19			

Each value in Table 6.1 is determined as follow:

1. Using shortest-path routing to establish a path from every node to every other node.
2. The percentage of traffic going out from a node to one of its neighbouring nodes equals the number of paths using the link connected to that neighbour node, divided by the total number of paths goes through the original node. This can be seen as a “fairness” mechanism.

In Table 6.1, each node has a portion of its own traffic going to itself. That is the sink traffic of that node, i.e., the amount of traffic leaving the network through that node.

The values for the network parameters listed in Table 6.2 are generic values: their units are “traffic unit” or “traffic unit/time unit”. The generic “traffic unit per time unit” could be adapted to any traffic volume units such as millions-of-packets-per-second or giga-bytes-per-minute under different application scenarios.

With ever-greater processing power, the data link capacities and router speeds change from year to year. There is a huge diversity in terms of link capacities and buffer sizes in real networks. It is hard to decide what typical link capacities and buffer sizes are. As a result, in the simulation setup, the value of the parameters is selected randomly to remove human bias. As for the ranges for the value of each parameter, they are meaningful only in the relative sense. This does not compromise the generality of the proposed approach. In fact, there is no reason why the proposed approach will not work under certain realistic parameters. It should be noted that the topology chosen for the simulations came from a real network, i.e. the NSF network. It is widely adopted as a typical topology for network study.

The specific values of the network parameters can be configured by the network engineer, and EPIT would still work given a different set of settings.

It was stated earlier that it is a black-box approach. To avoid any bias in the network setup, the network parameters (shown in Table 6.2) are generated randomly according to uniform distributions (Hurst Parameter [0.5 to 0.9], Mean Rate [1 to 10], Variance [0 to 3]) and exponential distributions (mean Buffers size is 500 and mean service rate is 10).

Table 6.2: NSFNET Network Parameters

Node	Buffer Size (traffic unit)	Service Rate (traffic unit/time unit)	Hurst Parameter of Attached Traffic Source	Mean Rate of Attached Traffic Source(traffic unit/time unit)	Variance of Attached Traffic Source
0	375	9.0	0.9	1.8	0.1
1	415	10.0	0.56	5.1	2.2
2	915	16.4	0.74	9.47	4.1
3	252	7.5	0.69	1.37	0.6
4	516	10.5	0.7	4.47	1
5	317	7.6	0.86	3.52	0.16
6	639	16.0	0.92	5.27	0.2
7	898	17.9	0.59	8.0	3.8
8	432	6.7	0.83	2.47	0.25
9	612	23.8	0.57	7.97	2.5
10	598	4.6	0.67	2.47	0.5
11	212	11.6	0.72	3.97	0.3
12	391	8.9	0.79	3.8	1.0
13	576	3.7	0.81	2.67	0.2
14	646	8.3	0.63	5.87	1.8

One may question whether the proposed approach is still a black-box approach, given that the buffer sizes, service rates, and internal structure in the simulation model have to be specified. The reason a simulation model has to be built is that there is not a real network at our disposal. Carriers do not have such a problem. The purpose of traffic engineering for a carrier is to provision an existing network. The simulation step can be replaced by an initial measurement through the real network. Under this situation, one only has to measure the end-to-end overflow events in the network and does not have to know the buffer sizes, service rates, and the internal structures of the network. In contrast, without a real network at our disposal, a simulated network must be built initially. Thus, the buffer sizes, service rates, and internal structure have to be known. Once the

simulated network is built, the buffer sizes, service rates, and internal structure will not be used by EPIT to project the overflow probability with respect to different input traffic rates. Thus, EPIT is a black-box approach.

The service rate in Table 6.2 represents the aggregated outgoing link rate for that router. In Table 6.7, each link in the network is indexed by the two nodes (routers) attached to it, and the maximum capacity on each link is determined by how fast the source router of the link can transmit (or process) traffic.

As stated previously, EPIT (based on the IS concept) is designed for rare event simulations. This means that the overflow probabilities should be low. According to their measurement, [57] and [58] stated that the overall packet loss probability in the current Internet ranges from 2.7% to 5.2%. [59] stated that their measurement reports a packet loss of 0.36% in the Internet. It is anticipated that future Internet will likely have much lower packet loss probability so that better QoS can be achieved to satisfy the requirements of mission-critical applications.

Therefore, the target loss probabilities in the examples in this chapter were set much lower than the ones in the current Internet. In order to make rare events happen more often for reducing simulation time, the initial simulation parameters were set in such a way that the loss probabilities are much higher than their targets while at the same time being close to the level of the current Internet (see Table 6.4). The corresponding link utilizations, as shown in Table 6.3, are also close to the level of the current Internet. It will be shown that one can achieve the target QoS with EPIT, which is much better than the QoS of the current Internet, with only a minor reduction in the link utilizations.

It is easy to use a different set of parameters so that the overflow probabilities and the link utilizations are higher. EPIT and SA-EPIT would still work because, as stated before, EPIT views the network as a black box.

Table 6.3: Network Link Parameters and Initial Link Utilization

src	dst	Link load	Link capacity	Link Util.	src	dst	Link load	Link capacity	Link Util.
7	0	2.0800	4.6540	0.4469	8	11	0.6422	1.7420	0.3687
1	3	1.4790	2.9000	0.5100	0	2	0.3420	1.7100	0.2000
7	5	1.2000	2.6850	0.4469	2	6	3.1251	5.4120	0.5774
11	12	0.5558	1.6240	0.3422	2	1	1.3258	2.2960	0.5774
11	9	1.2307	3.5960	0.3422	6	4	1.0013	3.0400	0.3294
6	12	1.0013	3.0400	0.3294	4	5	0.9834	2.3100	0.4257
9	13	0.6376	1.9040	0.3349	9	7	2.1519	6.4260	0.3349
2	0	1.7993	3.1160	0.5774	14	9	2.8176	3.9840	0.7072
5	7	1.0912	2.3560	0.4632	0	7	0.5400	2.7000	0.2000
9	11	1.5143	4.5220	0.3349	13	8	0.4539	0.6290	0.7216
11	8	0.9528	2.7840	0.3422	12	13	0.3040	0.7120	0.4270
12	6	1.2160	2.8480	0.4270	7	9	2.7200	6.0860	0.4469
8	13	0.2717	0.7370	0.3687	9	10	0.7970	2.3800	0.3349
5	4	0.9152	1.9760	0.4632	10	6	0.7410	1.3800	0.5370
0	1	0.3420	1.7100	0.2000	3	4	0.3288	1.8000	0.1827
1	0	1.0200	2.0000	0.5100	1	2	0.7650	1.5000	0.5100
3	8	0.3836	2.1000	0.1827	3	1	0.2740	1.5000	0.1827
4	6	0.9834	2.3100	0.4257	6	10	0.8432	2.5600	0.3294
8	3	0.7904	2.1440	0.3687	6	2	1.1067	3.3600	0.3294
4	3	1.0728	2.5200	0.4257	10	9	0.5928	1.1040	0.5370
9	14	1.4346	4.2840	0.3349	13	12	0.2670	0.3700	0.7216
13	9	0.5607	0.7770	0.7216	12	11	0.7220	1.6910	0.4270

**Overall
Average 0.4161**

6.2.2 VPN Experiments

The PE-PE paths in Figure 6.2 covers only part of the network. In order to verify that the SA-EPIT solution can indeed satisfy multiple end-to-end performance requirements, two other experiments are conducted in which the PE-PE paths are more spread out. Each experiment involves three paths and three different overflow probability requirements.

The paths selected in each experiment have different degrees of initial congestion, as shown in Table 6.4.

Table 6.4: PE-PE Paths in Each Experiment, Their Initial Probability of Overflows, and their Overflow Probability Requirements

Experiment	PE-PE Paths in Experiment	Nodes in the PE-PE Path	Initial Prob. of Overflow (log scale)	Target Prob. Of Overflow (log scale)
1	1	9,11	-3.08	-6.0
	2	3,4	-3.35	-5.0
	3	13,8,4,3	-2.6198	-4.0
2	4	4,3,8	-3.017	-5.25
	5	14,9,11	-2.65	-4
	6	7,9,11,12	-2.17	-3.5
3	7	10,9,11,8	-2.8601	-4.5
	8	11,8,3,4	-2.8665	-3.75
	9	4,3,1,0	-2.4365	-3.5

Figure 6.3: Traces of the Three End-to-end Overflow Probabilities over the Course of the EPIT Optimization Process (Experiment 1)

Figure 6.3 plots the decrease of the three overflow probabilities (in experiment 1) as SA-EPIT searches for a solution that meets all the overflow targets in the experiment while keeping the reduction cost to a minimum. The target overflow probabilities are randomly chosen to make the task more challenging. Figure 6.3 also shows the

corresponding measured overflow probabilities, via MC simulation, for verifying the accuracy of the EPIT predictions.

Figure 6.3 clearly shows that the EPIT prediction agrees with the MC measurement, thus verifying the effectiveness of EPIT.

Due to the nature of Simulated Annealing, the overflow probabilities drop slowly at first and pick up speed gradually, as depicted in Figure 6.3.

Table 6.5: EPIT Solution for VPN Experiments, Comprised of the Reduction Amounts for the Mean Rates of the 15 Traffic Sources

Exp\Src	0	1	2	3	4	5	6	7
1	0.02	0.09	0.02	0.13	0.05	0.14	0.04	0.04
2	0.05	0.16	0.18	0.04	0.07	0.01	0.01	0.02
3	0.02	0.14	0.08	0.07	0.07	0.05	0.07	0.08
Exp\Src	8	9	10	11	12	13	14	
1	0.1	0.2	0.03	0.12	0.12	0.12	0.1	
2	0.12	0.08	0.11	0.14	0.04	0.02	0.03	
3	0.02	0.18	0.14	0.04	0.06	0.11	0.03	

Table 6.5 shows that, in all three solutions, the reduction amount of each source varies in a wide range. This is expected since some traffic sources will have more influence on the buffer-overflows of a selected path than others will. The fact that all the traffic mean rates are modified implies that the overflow probabilities of the PE-PE paths chosen are affected by all the traffic sources in the network. Thus, these paths are chosen appropriately since the problem will be oversimplified if the behaviours of these paths are affected by only a few traffic sources.

Table 6.6 clearly shows that the solutions provided by SA-EPIT have reduced the overflow probability from 10^{-3} to 10^{-6} (see Table 6.4), yet the sacrifice in utilization is

almost negligible (Table 6.7): original average utilization is 0.4161, new utilizations under the SA-EPIT solutions: 0.4061, 0.4085, and 0.40727. Clearly, the utilization is compromised while meeting QoS targets, but the compromise is small enough to be neglect-able.

Results in Table 6.6 validate SA-EPIT since all overflow requirements have been satisfied by using solutions provided by SA-EPIT. Since EPIT prediction has remained accurate (i.e. the variance has remained small), with a fixed computation cost, as the probability has changed by three orders of magnitude, one can claim that EPIT is efficient through this VPN simulation example.

Table 6.6: New Overflow Probabilities after Reductions are Made to the Traffic Source Mean Rates

Experiment	PE-PE Paths in Experiment	Target (log scale)	Probability of Overflow (log scale) After Reductions
1	1	-6.0	-6.1089
	2	-5.0	-5.0365
	3	-4.0	-4.0422
2	4	-5.25	-5.2505
	5	-4.0	-4.0081
	6	-3.5	-3.5581
3	7	-4.5	-4.5857
	8	-3.75	-3.7602
	9	-3.5	-3.6094

Next, the simulation time saving of EPIT, expressed as $\frac{N_{MC} - N_{EPIT}}{N_{EPIT}} * 100\%$, shall be

demonstrated. N_{MC} was chosen according to the rule of thumb stated in Chapter 4, and since EPIT requires only one simulation run, N_{EPIT} is fixed. For those experiments, 50,000 replications are used for the EPIT initial simulation run.

Figure 6.4: Simulation Gain for VPN Experiment 1

Table 6.7: New Link Utilizations for the SA-EPIT Solutions for VPN Experiments 1,2, and 3

src	dst	Link Util. 1	Link Util. 2	Link Util. 3	Src	dst	Link Util. 1	Link Util. 2	Link Util. 3
7	0	0.445	0.446	0.443	8	11	0.354	0.351	0.366
1	3	0.501	0.494	0.496	0	2	0.198	0.194	0.198
7	5	0.445	0.446	0.443	2	6	0.576	0.567	0.573
11	12	0.332	0.330	0.339	2	1	0.576	0.567	0.573
11	9	0.332	0.330	0.339	6	4	0.327	0.329	0.325
6	12	0.327	0.329	0.325	4	5	0.421	0.419	0.419
9	13	0.327	0.332	0.327	9	7	0.327	0.332	0.327
2	0	0.576	0.567	0.573	14	9	0.695	0.704	0.704
5	7	0.445	0.462	0.457	0	7	0.198	0.194	0.198
9	11	0.327	0.332	0.327	13	8	0.689	0.716	0.692
11	8	0.332	0.330	0.339	12	13	0.414	0.423	0.420
12	6	0.414	0.423	0.420	7	9	0.445	0.446	0.443
8	13	0.354	0.351	0.366	9	10	0.327	0.332	0.327
5	4	0.445	0.462	0.457	10	6	0.530	0.513	0.507
0	1	0.198	0.194	0.198	3	4	0.165	0.177	0.173
1	0	0.501	0.494	0.496	1	2	0.501	0.494	0.496
3	8	0.165	0.177	0.173	3	1	0.165	0.177	0.173
4	6	0.421	0.419	0.419	6	10	0.327	0.329	0.325
8	3	0.354	0.351	0.366	6	2	0.327	0.329	0.325
4	3	0.421	0.419	0.419	10	9	0.530	0.513	0.507
9	14	0.327	0.332	0.327	13	12	0.689	0.716	0.692
13	9	0.689	0.716	0.692	12	11	0.414	0.423	0.420
Average							0.406	0.409	0.407

Figure 6.4 clearly demonstrates the significant simulation gain of EPIT with respect to number of simulation replications required. The gain goes up and down because the probability of overflow goes up and down during the heuristic search process.

6.3 Peer-to-Peer Network

The formulation of a P2P network as a traffic-engineering problem is similar to that of VPN, except that a P2P network is typically more interested in the aggregate performance of the overall P2P network rather than the individual performance of one or multiple paths.

In the P2P network model that will be used in the simulation analysis, v_i could represent the aggregated traffic from a number of peers instead of a single peer. The aggregated traffic from peer group to peer group (“peer group” shall be defined shortly) is modeled as FGN process.

Fractional Gaussian Noise (FGN) process is a self-similar process that exhibits Gaussian characteristics. Since P2P traffic accounts for a huge portion of the Internet traffic (up to 50%) and since the Internet has shown self-similar (Long-range dependent) characteristics, one can also assume that aggregated P2P traffic is also self-similar. Furthermore, the central limit theorem guarantees that the aggregation of large number of independent flows will exhibit Gaussian properties. [12] has determined that SRD input model underestimates the burstiness of the internet traffic, which results in the underestimation of the overflow probability in a network. Therefore, using the LRD input (the FGN traffic model) can be viewed as the worst-case scenario

Figure 6.5: P2P Torrent Configurations

The topology of the provider network is still the NSFNET topology (see Figure 6.5) with the same background traffic setup as the previous section.

In the following simulation model, peers are divided into groups according to their geographical locations. Thus, from the core provider network's point of view, each peer in the same peer group will be connected to the same Provider Edge (PE) router. Figure 6.5 shows the configurations of peering arrangements for torrent 1, torrent 2 and torrent 3 embedded in the core network.

For each torrent setup, peers in any peer group could be communicating with any peer in the other peer groups. The activity could be seeding/leeching, or it could be voice or video (e.g., Skype). To provide QoS to those applications, the aggregated performance of the end-to-end paths (PE to PE paths stated in Table 6.8) connecting the peer groups has to meet the QoS target as a whole since one cannot know which end-to-end path the peer user will use.

The end-to-end overflow probability, P , (aggregated) currently exceeds the QoS target TQ (see Table 6.11). Combining EPIT with Simulated Annealing (SA-EPIT) brings P down to TQ and minimizes traffic reduction by trying different reduction amounts in the sources for each iteration. This is depicted in Figure 6.6. Note that minimizing traffic flow reduction means maximizing traffic flow, thus maximizing link utilizations in the network. The reduction of P2P traffic can be accomplished via traffic shaping (P2P traffic can be isolated via port-based analysis of the traffic) or call admission control.

Table 6.8: PE-to-PE Paths in Torrents 1, 2 and 3

Torrent	Peer Group	Peer Group PE Node	PE-PE Paths used to connect all peer groups
1	1	Node 1	"1-3-8-13","13-12-11","10-9-13"
	2	Node 11	
	3	Node 10	
	4	Node 13	
2	1	Node 3	"3-8-13-9","9-10","11-9"
	2	Node 10	
	3	Node 11	
	4	Node 9	
3	1	Node 2	"2-0","0-1-3-8","9-7-0"
	2	Node 8	
	3	Node 9	
	4	Node 0	

One sees that in Figure 6.6, the overflow probabilities of all the user groups decrease slowly at first then picks up speed later, which is expected for the Simulated Annealing process.

Table 6.9: EPIT Solution for P2P Experiments, Comprised of the Reduction Amounts for the Mean Rates of the 15 Traffic Sources

Exp\Src	0	1	2	3	4	5	6	7
1	0.01	0.12	0	0.14	0.07	0.05	0.01	0
2	0.01	0.14	0.13	0.16	0.04	0.08	0.04	0.19
3	0.02	0.09	0.02	0.13	0.05	0.14	0.04	0.04
Exp\Src	8	9	10	11	12	13	14	
1	0.15	0.09	0.02	0.14	0.01	0.05	0.01	
2	0.17	0.15	0.13	0.1	0.03	0.1	0	
3	0.1	0.2	0.03	0.12	0.12	0.12	0.1	

Table 6.10 exhibits the link utilizations in the network after the QoS target of the torrents has been achieved. Note that SA-EPIT is used to satisfy each QoS target for the three torrents, but not at the same time.

Table 6.10: New Link Utilizations for the SA-EPIT Solutions of P2P Torrents 1,2, and 3

src	dst	Link Util. 1	Link Util. 2	Link Util. 3	Src	dst	Link Util. 1	Link Util. 2	Link Util. 3
7	0	0.447	0.436	0.445	8	11	0.346	0.343	0.354
1	3	0.498	0.496	0.501	0	2	0.199	0.199	0.198
7	5	0.447	0.436	0.445	2	6	0.577	0.570	0.576
11	12	0.330	0.334	0.332	2	1	0.577	0.570	0.576
11	9	0.330	0.334	0.332	6	4	0.329	0.327	0.327
6	12	0.329	0.327	0.327	4	5	0.419	0.422	0.421
9	13	0.331	0.329	0.327	9	7	0.331	0.329	0.327
2	0	0.577	0.570	0.576	14	9	0.706	0.707	0.695
5	7	0.457	0.453	0.445	0	7	0.199	0.199	0.198
9	11	0.331	0.329	0.327	13	8	0.708	0.695	0.689
11	8	0.330	0.334	0.332	12	13	0.426	0.424	0.414
12	6	0.426	0.424	0.414	7	9	0.447	0.436	0.445
8	13	0.346	0.343	0.354	9	10	0.331	0.329	0.327
5	4	0.457	0.453	0.445	10	6	0.533	0.509	0.530
0	1	0.199	0.199	0.198	3	4	0.164	0.161	0.165
1	0	0.498	0.496	0.501	1	2	0.498	0.496	0.501
3	8	0.164	0.161	0.165	3	1	0.164	0.161	0.165
4	6	0.419	0.422	0.421	6	10	0.329	0.327	0.327
8	3	0.346	0.343	0.354	6	2	0.329	0.327	0.327
4	3	0.419	0.422	0.421	10	9	0.533	0.509	0.530
9	14	0.331	0.329	0.327	13	12	0.708	0.695	0.689
13	9	0.708	0.695	0.689	12	11	0.426	0.424	0.414
Average							0.4091	0.4049	0.4061

Table 6.10 clearly shows that the solution provided by SA-EPIT (listed in Table 6.9) has reduced the overflow probability by almost one hundred times, yet the sacrifice in utilization is almost negligible: the original average link utilization in the network is 0.4161; the new average link utilization in the network after the overflow probability reduction is 0.4061.

Figure 6.6: Traces of the Aggregated Overflow Probability in Torrents, Over the Course of the SA-EPIT Optimization Process

For each prediction made by EPIT with respect to a new set of reduction amounts, a direct simulation (MC simulation) is performed to measure the new P . Those measurements are graphed in the solid data line of Figure 6.7. Figure 6.7 clearly shows that EPIT predictions match the measurements. Keep in mind that the number of replications used remains the same, yet the variance of the EPIT prediction remains small, as the probability decreases drastically (note that Figure 6.7 is in logarithmic scale). One can claim the variance is small since the EPIT prediction remains accurate. This process is repeated on torrent 2 and 3 with similar results.

Figure 6.7: P Trajectory for Torrent 1 as SA-EPIT Iterates Through Different Reduction Amounts to Bring P to TQ .

At the end of the iteration process, SA-EPIT finds the reduction amount solution that has the lowest reduction cost among all other solutions that meet the QoS requirement.

Table 6.11: Overflow Probabilities Before and After Reductions are Made

Torrent	Initial Prob. Of Overflow	Target (log scale)	Probability of Overflow (log scale) After Reductions
1	-2.2612	-4	-4.084970
2	-2.5003	-4.25	-4.267819
3	-1.9512	-3.5	-3.52670

As shown in Table 6.11, the new overflow probabilities on those end-to-end paths used by P2P torrents all meet their respective targets. Note that some probabilities have decreased almost 100 times that of the original probability, and the number of replications used remains the same; yet the EPIT prediction is still accurate. Thus, we deduce that the variance of the EPIT estimation has remained small, and subsequently EPIT is efficient for a wide range of probability values in a multiple queue and multiple LRD input network.

The next chapter would give an overview of how EPIT can be implemented in a real network.

Chapter 7 Implementation

With the ever-increasing size, complexity and dynamic nature in networks today, adaptive and self-management functionality in service provider networks are essential. Old methods of building networks and services, based on individual device management, have reached their limits in today's networking environment. The idea of autonomous management of communication networks has become dominant.

An autonomous management system consists of distributed traffic measurement, traffic analysis, network performance analysis capabilities, and dynamic configuration capabilities to automatically act on the monitored information. To realize this goal, devices in the network, such as routers, switches, servers and dedicated metering devices, have to work together as a whole instead of being managed separately.

To realize an autonomous network management architecture, the following components are required:

- 1) Network performance measurement component: this component determines the current QoS of the network.
- 2) Network traffic measurement: this component meters the amount of traffic entering the network and calculates some of the parameters of the traffic: mean, variance, etc.

- 3) Measurement information delivery component: this component delivers performance measurement information and traffic measurement information to some destination(s) where this information can be processed and used.
- 4) Decision-making component: based on the received performance and traffic information, this component decides whether any adjustment(s) should be made to the network. If so, this component decides what specific adjustment(s) is to be made.
- 5) Adjustment information delivery component: this component delivers adjustment(s) derived by the decision-making component to destinations where those adjustment(s) can be made. This component can be the same component that delivers performance and traffic information to the decision-making component.
- 6) Adjustment enforcement component: once received the adjustment information, the job of this component is to carry out the adjustment(s).

All the components essentially form a feedback loop: measure, decide, act, and measure again.

The purpose of this chapter is to show how existing technologies can be used to implement those components for the EPIT architecture. In this sense, EPIT utilizes these techniques to enable a more autonomous approach to management, based on adaptation and self-management. Some parts of implementation are more challenging than others are, and they will be discussed.

7.1 EPIT Implementation Architecture

Based on the components listed previously, the following architecture is proposed to implement EPIT.

Figure 7.1: EPIT Implementation Architecture

Components in the EPIT implementation include:

- 1) On each provider edge node, there is an end-to-end loss detection mechanism. It can discover if any traffic was lost along an end-to-end path in the network. This mechanism monitors all end-to-end paths whose end-to-QoS are of interest. Given the strong-correlation property of long-range dependent traffic, it is difficult to obtain independent loss events from measurements (this problem will be

discussed in section 7.2). This is essentially the network performance measurement component.

- 2) A traffic measurement mechanism is also at each provider edge node. It measures the aggregated traffic flow entering the network at that node. The mean, variance, Hurst parameter (measuring the Hurst parameter is a very challenging task and there are several approaches), and the likelihood ratio are calculated. This is the network traffic measurement component.
- 3) Simple Network Management Protocol (SNMP) will be used to deliver the performance and traffic measurements to a decision-making server.
- 4) The EPIT server calculates the end-to-end loss probability based on information received. If the loss probability exceeds the threshold, an adjustment is needed. SA-EPIT is then used to project an end-to-end loss probability that is acceptable, as a function of the admission rate of the ingress traffic flows at the edge nodes. The EPIT server is the decision-making component.
- 5) The adjustments from the EPIT server is delivered by SNMP back to the edge node to be enforced.
- 6) The traffic shapers at the network edge nodes use the received adjustment information to regulate the mean rate of the traffic allowed into the network. The shaper will shape only the mean rate of the ingress aggregated traffic flow and not the burstiness of the traffic flow. This is the adjustment enforcement mechanism.

With the help of those mechanisms, EPIT can manage a network efficiently so a certain QoS is achieved, and the entire process is automatic.

As demonstrated in Figure 7.1, all information is collected at the edge of the network. This is consistent with the black-box aspect of the EPIT approach.

Each of the following sections shall discuss in detail one of the components of the implementation.

7.2 End-to-end Loss Detection

For LRD traffic that exhibit strong correlation, EPIT so far has relied on independent replications to get independent end-to-end loss events in order to determine the loss probability. There are several possible solutions to address this issue in implementation (Hurst parameter is a measurement of the degree of correlation within the traffic and is a separate topic covered in section 7.4):

- 1) On-Line Simulation: use a simulation that imports the latest monitored traffic patterns at the network edge nodes, where the simulation is run on-line in a network operations center (NOC). This is similar to the approach proposed in [64].

During the simulation run, the simulated traffic takes on the measured traffic parameters, and N independent replications are executed. SA-EPIT is then performed to find the best solution to achieve the QoS target. The simulation then automatically sends the proposed solution back to the network edge nodes, and those edge nodes reconfigure their traffic shaping mechanisms according to the proposed solution to achieve the target QoS.

- 2) Correlation measurement: In order to create “independent replications” in a real-network, one can divide the time axis into non-overlapping windows (see Figure

7.2 in section 7.3). To determine the end of one window and the beginning of another window, the monitored traffic flows at the network edge are treated as the sample traces of random processes. The autocorrelation of the process is continuously calculated as new traffic arrives. Once the auto-correlation falls below and remains below a threshold, for example, 0.1, for a predefined period of time, the correlation can be assumed to have decayed enough that the samples arriving afterwards are independent from samples collected at the beginning of the window.

It is understandable that some traffic flow will have a stronger self-similar characteristic, and thus its autocorrelation will decay at a slower rate, resulting in a longer window length. Therefore, to accommodate all traffic flows, EPIT will use the longest window length measured among all traffic flows,

- 3) Carriers are usually interested in a particular time-scale, at which time they will perform traffic engineering on their networks. It is assumed that the event/traffic outside this time scale (or time window) will be independent from each other.

Thus, these windows can be used as the independent replications.

Regardless of which approach is implemented, at the end of each time-window/replication, the EPIT server will poll each edge node to check if an end-to-end loss event has been detected. After a pre-defined number of time-windows have passed, the end-to-end loss probability is calculated by the EPIT server.

Next, we shall discuss in detail on how end-to-end loss can be detected.

This thesis has stated that EPIT is used in scenarios in which traffic flows are aggregated at the edge of the provider network. TCP trunking [60] encapsulates customer site traffic in a way that all the traffic from a source to a destination looks like one flow or trunk. In other words, at the transportation layer, all the packets over the trunk appear to have the same source and destination addresses. Thus, multiple customer packets might be encapsulated in a single transport packet to reduce header overheads.

The loss of a transport packet along an end-to-end path would result in the loss of multiple customer packets with potentially undesirable effects on the customers' TCP connections. Transport access points (TAPs) usually use TCP for the transport trunks. The customer packets would be encapsulated in a transport packet with both an IP header and a TCP header. The sequence number in the TCP header can be used to detect if an end-to-end loss event has occurred: if there is a jump in the received sequence number or a timeout occurs. The implementation details are left to the network engineer.

Many Internet applications today are based on voice/video traffic. Real-time Transport Protocol (RTP) [61] is a popular choice to provide end-to-end delivery for data (voice/video) with real-time characteristics. In the case of a bandwidth discrepancy between customers/users (some have high-bandwidth while others have low-bandwidth), a mixer may be placed near the low-bandwidth customers. This mixer resynchronizes incoming audio packets to reconstruct a constant spacing generated by the high-bandwidth (faster) senders, mixes these reconstructed audio streams into a single stream, translates the audio encoding to a lower-bandwidth, and forwards the lower-bandwidth packet stream across the low-speed link. Thus, in the RTP header, there is a field call

CSRC (Contributing SouRCe) count to identify the number of CSRC identifiers in the RTP header.

The RTP header includes a sequence number and a time stamp. The timestamp reflects the sampling instant of the first octet in the RTP data packet. Thus, the receivers at the network edge can detect end-to-end loss events using those fields.

While the destination of a packet can be determined by reading its packet header, from the point of view of the provider network, it is the provider edge (PE) to provider edge (PE) QoS that is of interest. Thus, the end-to-end loss detection mechanism needs to know the egress PE node of the traffic flow. Fortunately, BGP is usually implemented at the edge nodes of a provider network. Since BGP maintains a table of known AS (Autonomous System) in the Internet, based on a given destination address, the edge node can almost surely determine which egress node will be used. In a case where MPLS is used (or another tunnelling protocol), the protocol that sets up the tunnel should know the egress point of the tunnel.

This thesis so far has been focused on the aggregated traffic flow allowed into the network where the destinations are not relevant. In this case, the end-to-end loss deals with only the aggregated traffic flow.

One can also perform traffic engineering on a more granular level, where individual source-destination flow pairs are considered. In this case, a likelihood ratio for the traffic from the sender has to be calculated and the receiver shall monitor the end-to-end loss events. EPIT will still work, but the computational load will be heavy if there are many source-destination flow pairs to be metered. The extra computational load is the price one

has to pay for a more granular analysis. It is important to note that the maximum number of end-to-end flows in a provider network equals the number of edge nodes squared.

To enable flow-based metering, NetFlow/IPFIX [62] can provide a detailed end-to-end view. Cisco's Flexible NetFlow can also be used. With Cisco's Netflow, all parts of the IP header and even payload sections can be tracked.

The next section shall cover the traffic measurement mechanism. Because Hurst parameter measurement is a complex topic, it is discussed in a separate section.

7.3 Traffic Measurement

One issue with traffic measurement is that large provider and enterprise networks generate enormous amounts of traffic monitoring data. For example, the network of a large provider was reported to generate 500 Gbytes of NetFlow data per day [62]. It is clearly inefficient for each edge node to report every sample, recorded in their traffic metering mechanism, to a central server so that the central server can then calculate the likelihood ratio needed by EPIT.

Our implementation is based on a semi-distributed approach to solve this problem. It is semi-distributed because some computational burden is distributed among the edge nodes while there is still a central server to implement the SA-EPIT algorithm. The traffic received on a particular node is measured/calculated for its mean, variance, and Hurst parameter. The likelihood ratio is then calculated on each edge node using that information. This likelihood ratio is retrieved by the central server at the end of a replication/window.

Because SA-EPIT will use different twist amounts to predict the network QoS as a function of the ingress traffic mean rates, an array of likelihood ratios are calculated in each edge node. Each likelihood ratio is calculated with respect to a specific twist amount value.

Table 7.1: Likelihood Ratio Table in Each Edge Node

Twisted Amount	Original traffic mean rate	New traffic mean rate	Likelihood ratio
0	v'	v'	$L(\vec{x}, v', v')$
Θ	v'	$v' - \Theta$	$L(\vec{x}, v' - \Theta, v')$
2Θ	v'	$v' - 2\Theta$	$L(\vec{x}, v' - 2\Theta, v')$
3Θ	v'	$v' - 3\Theta$	$L(\vec{x}, v' - 3\Theta, v')$
4Θ	v'	$v' - 4\Theta$	$L(\vec{x}, v' - 4\Theta, v')$
.....	.....	.....	.....

During the annealing processes, if SA-EPIT wishes to twist (adjust) the mean rate of specific traffic source, it can simply retrieve the likelihood ratio from that source corresponding to that twist amount value. Note that a quota Θ is defined as a step size in the heuristic search process in section 4.3. As a result, the edge node does not have to calculate the likelihood ratio for every possible value of twist amount, only those that are integer multiple of Θ .

The size of the table depends on the available memory resource on the edge node. The table is sent to the central server upon request at the end of a time window, and the table is cleared for the calculation in the new time window. Sending a table of calculated likelihood ratio at the end of a window will consume a lot less bandwidth than sending every sampled traffic trace data to the central server.

To calculate the mean, the variance, etc, of the incoming traffic, edge nodes need efficient algorithms to compute various quantities. These algorithms need to treat online traffic measurements as a data stream and process incoming data as it arrives. The likelihood ratio, according to equation (32), can be calculated recursively. In other words, likelihood ratio can be calculated as traffic arrives. The calculation of the mean and variance of the traffic can also be done recursively (this will be demonstrated shortly). The measurements restart at the beginning of every time-window.

Samples of the ingress traffic have to be collected at the PE nodes at the edge of the core provider network, as shown in Figure 7.1. Traffic metering mechanisms should already exist in provider networks, since one of the top priorities in traditional provider networks is traffic transit and meeting service level agreements (SLAs) to other customer networks [62]. Thus, some kind of traffic metering mechanism must be present.

So far in this thesis, $x(i)$ refers to the traffic recorded in time slot i . The length of the time slot depends on the correlation of the traffic or the time scale chosen by the carrier. It also depends on how fine the network engineers want the measurement to be.

The sample mean and the sample variance of the traffic can be estimated as

$$v = \frac{\sum_{i=1}^T x(i)}{T} \quad (105)$$

$$\sigma^2 = \frac{\sum_{i=1}^N (x(i) - v)^2}{N - 1} = \frac{\sum_{i=1}^N x(i)^2 - 2v \sum_{i=1}^N x(i) + Nv^2}{N - 1} \quad (106)$$

If one defines $tempA = \sum_{i=1}^T x(i)$ and $tempB = \sum_{i=1}^T x(i)^2$, and when ever a new $x(T+1)$ is sampled, $tempA$ and $tempB$ can be updated as $tempA = tempA + x(T+1)$ and $tempB = tempB + x(T+1)^2$. The sampled mean and variance can then be calculated by $v = \frac{tempA}{T+1}$, $\sigma^2 = \frac{tempB - 2v(tempA) + Tv^2}{T}$. Thus, the mean and variance can be calculated recursively and quickly.

The above approach essentially has a measurement window of length equals to the total length of the sample traffic trace. There are other measurement window approaches that can be used instead:

Figure 7.2: Taking Measurements Using a Jumping Window

By setting the Windows size W to $k * T_{slot}$, where k is an integer and T_{slot} is the time slot duration, all the traffic arriving inside the time slot is aggregated into a single sample.

Another approach is the sliding window approach, which is similar to the jumping window except when it moves, instead of moving the entire window length, it only slides by only one time slot.

Figure 7.3: Taking Measurements Using Sliding Window

Another measurement window which can be used is the Double Recursive Double Moving Windows (DRDMW), which is two sliding windows with the smaller window (size E) inside the larger window (size J). The inner window can only move within the outer window. When a new time slot becomes available, the outer window will move by one time slot to include the new time slot.

Figure 7.4: Double Recursive Double Moving Window Measurement

The main difference between DRDMW and regular sliding window is that only the $x(i)$'s calculated inside the outer window are used while the outside ones are discarded.

The embedded software on the managed device can choose one of the window measurement implementations mentioned above to sample the ingress traffic in order to determine the statistical parameters such as mean, variance, etc.

As stated before, due to the large amount of traffic sampled in a provider network, the collection of the traffic data and the pre-processing steps (calculating mean, variance and Hurst parameter) are done in the edge nodes instead of the central EPIT server. This is because reporting every traffic sample to the server to calculate the mean, variance, etc., could consume a lot of network bandwidth.

As it will be discussed in section 7.6, the traffic measurement in the edge nodes will take place after the traffic shaping.

The next section shall discuss how the Hurst parameter can be measured from a traffic flow.

7.4 Hurst Parameter Measurement

As stated in section 4.1, the likelihood ratio requires the distribution of the input traffic process. The distribution of FGN traffic is described (partially) by the Hurst parameter. There are several ways to estimate the Hurst parameter (H):

- Variance and Time plot: The main objective of this approach is to analyze the long-time trend of the target traffic. Using the notation defined in section 2.4

where $x^{(m)}(k) = \frac{1}{m}(x(km - m + 1) + \dots + x(km))$, and create a function

$$f(m) = (\log(m), \log(\text{var}[X^{(m)}])) \quad (107)$$

Figure 7.5: Variance and Time Plot

By taking the derivative of $f(m)$ with respect to m to find the slope of the plot, the Hurst parameter is calculated as $H=1-(\text{slope}/2)$.

- R/S Analysis: This approach creates two series $R(t_i, m)$ and $S(t_i, m)$ from a time series $x(t)$ of length T where it is divided into K non-overlapping blocks:

$$t_i = \lfloor T / K \rfloor (i-1) + 1.$$

$$R(t_i, m) = \max\{0, W(t_i, 1), \dots, W(t_i, m)\} - \min\{0, W(t_i, 1), \dots, W(t_i, m)\} \quad (108)$$

where

$$W(t_i, k) = \sum_{j=1}^k x(t_i + j - 1) - k \left(\frac{1}{m} \sum_{j=1}^m x(t_i + j - 1) \right) \quad (109)$$

$S(t_i, m)$ is the sample variance of $x(t_i), \dots, x(t_i + m - 1)$

First, R/S plot is used to plot $\log R(t_i, m)/S(t_i, m)$ vs. $\log m$. Then, a least squares line is fitted to the points of the R/S plot. The slope of the regression line for these R/S samples is estimated for the Hurst parameter H .

- Periodogram: A periodogram is an estimate of the spectral density of a sampled series. The main idea of the periodogram-based analysis is to fit the parameters of

a specific random process to a given empirical sample trace by using the goodness-of-fit approach. Since the spectral density of self-similar processes obeys a power law near the origin, the Hurst parameter H is determined simply by plotting the periodogram in a log-log grid and computing the slope of a regression line which is fitted to a number of low frequencies.

There are other approaches such as the Whittle's estimator and other wavelet-based methods to estimate the Hurst parameter. For more details, refer to [63] [17]. Estimators, like those in [65], essentially fit a straight line to a frequency spectrum derived using wavelets. A 95% confidence interval can be calculated; however, this should be interpreted *only* as a confidence interval on the fitted line and, not as a confidence interval on the fitted Hurst parameter.

The next section shall discuss how information such as traffic measurement and the end-to-end loss event notification can be delivered to the EPIT server, and how admission traffic rates calculated by EPIT, to satisfy the QoS requirement, are delivered back to edge nodes.

7.5 Information Delivery Mechanism

There is existing technology that delivers network management information within a network. Therefore, it is preferable to implement EPIT with existing system/network management technologies. Two most popular technologies are Simple Network Management Protocol (SNMP) and Web Based Enterprise Management (WBEM)/Common Information Model (CIM) technology.

SNMP, as the name implies, is designed to manage simple network elements. It is fast and light-weight in getting/setting a single “object” from a managed device. In a system management environment where the interaction between system elements are more complex and system management specification is more sophisticated, WBEM/CIM is used.

The best way to distinguish SNMP and WBEM/CIM is that SNMP provides managed “variables” to represent the managed element whereas CIM provides the class to describe the managed elements, since CIM employs object-oriented modeling. Thus, WBEM can incorporate inheritance, encapsulation and association into the model.

Figure 7.6: SNMP Model for the EPIT Implementation

The SNMP model is chosen to implement EPIT because, as it will be demonstrated next, the data required by EPIT are simple and mostly independent from each other. Thus, it does not require inheritance or aggregation capabilities. The simplicity of SNMP makes it ideal to realize the EPIT approach.

There is a SNMP manager running in the EPIT server while a SNMP agent is running on every edge node (edge routers) in the core provider network, as shown in Figure 7.6.

The managed edge node is positioned at the ingress edge of the core network where traffic flows enter the network and the flows are aggregated.

Periodically (at the end of each time-window), the central SNMP manager polls the SNMP agent on each edge node (PE) via `get()` SNMP operation and the edge nodes replies with `response()` SNMP operation to send the likelihood ratios calculated in the current time window. The information also includes whether an end-to-end loss event has been detected. The EPIT algorithm in the EPIT server uses this information for the calculation of the likelihood ratio and the end-to-end loss probability. If the probability exceeds a QoS target, EPIT finds the appropriate rate limit so QoS can be achieved. This information is sent back to the SNMP agent, on the edge nodes, via `set()` SNMP operation. The edge nodes use that value to control the rate of the traffic entering the core network.

The information collected by the edge nodes is stored in managed objects in the MIB of the node. A managed object is essentially variables that should be published to the SNMP manager:

Figure 7.7: Managed Object

The agent and manager communicate via SNMP Protocol Data Units (PDUs) that are typically encapsulated in UDP packets. The agent sends data (UDP ports 161) in response to a request from a manager (UDP ports 162).

The values of some managed object are updated by the SNMP manager: `traffic_limit_rate`, while others have to be updated and maintained by the router's (managed node's) firmware, driver, or application. For example, the value of the `current_time` needs to be updated by the driver according to the system clock on the router. The Hurst parameter, the mean, and the variance of the aggregated traffic have to be calculated by using the internal traffic accounting structure in the router-embedded software. The likelihood ratios are then calculated using the sampled traffic values, the mean, the variance and the Hurst parameter.

The embedded software of the router is charged with detecting the event of end-to-end loss using approaches described in section 7.2. If the event is detected, the router software will update the corresponding entry in the MIB. When the edge node is polled, the SNMP agent will send this information to the SNMP manager of the EPIT server.

The router-embedded software can retrieve or set managed object value in the MIB directly, or via local SNMP get() and set() operations (the choice is left to the network engineer). Lastly, it is also the job of the firmware/driver to enforce the traffic rate assigned EPIT so that the aggregated ingress traffic allowed into the network does not exceed it. Some traffic shaping algorithms shall be discussed next.

7.6 Traffic Shaping Algorithms

Both ATM and IP perform traffic shaping function. Traffic shaping is to control the rate of sending data streams into the network to ensure conformance to a traffic requirement (in the EPIT case, the requirement is the traffic rate allowed so the QoS target in the network can be achieved). Traffic shaping is different from traffic policing in the sense that a shaper delays the traffic that violates the predefined condition while traffic policing simply drops the traffic. A shaper delays the traffic when the traffic rate exceeds a threshold with a queue. Some traffic loss would occur if the queue becomes full, and this scenario can be included in the end-to-end QoS provision in EPIT by including the event of buffer overflow at the network edge in the end-to-end event in which EPIT is interested.

It is important to note that EPIT does not impose any burstiness restrictions on the traffic. Thus, the burstiness of the ingress traffic is not shaped, only the mean rate of the traffic is shaped. Thus, after shaping takes place, the variance and the Hurst parameter can still be measured/calculated. As for measuring the mean of the traffic, one can just use the limiting rate of the traffic shaper.

7.6.1 Leaky Bucket

ATM networks use a continuous state version of the leaky bucket algorithm called the generic cell rate algorithm (GCRA) to control the traffic at the entrance of the ATM network. The leaky bucket analogy refers to a bucket with a hole in the bottom that causes it to 'leak' at a certain rate corresponding to a traffic cell-rate parameter. The 'depth' of the bucket corresponds to a tolerance parameter (cell delay variation tolerance). Each cell arrival creates a 'cup' of fluid flow 'poured' into one or more buckets to check if they meet the predefined traffic condition.

Figure 7.8: Leaky Bucket Implementation

In the algorithm, a cell counter represents a bucket. This counter is incremented by one for each incoming cell. The ‘leak rate’ in the algorithm is the decrement rate that reduces the counter value by one at *certain intervals*.

The GCRA algorithm is characterized by an “increment” and a “limit”. The “increment” denotes the average inter-arrival time between cells. This “limit” places a bound on the burstiness of the traffic. Since EPIT does not pose any burstiness restriction on the traffic, the value for the “limit” will be extremely large.

Figure 7.8 shows the implementation of leaky bucket in the edge nodes of the network. When different traffic flows arrive at the edge of the core network, the PE nodes aggregates them and performs measurement on the aggregated flow (detail discussed in section 7.3 and section 7.4). This information is used by the embedded software in the PE node to calculate the mean, variance and the Hurst parameter. The embedded software in turn controls the time-slot used in the measurement. It also resets the measurement for each new time window.

The embedded software also controls the Increment parameter of the leaky bucket to make sure it conforms to the `traffic_limit_rate` value in the MIB.

7.6.2 Token Bucket

IP uses a Token Bucket approach, where a token bucket is used instead of a leaky bucket to control the average transmission rate. A token bucket of a traffic regulator has three components:

- 1) Mean rate ν : It is measured in bytes of IP datagram per unit time on average permitted by the token bucket.
- 2) Bucket depth: It specifies in bits per burst what can be sent within a given unit of time without creating scheduling concerns.
- 3) Time Interval, also called the measurement interval. It specifies the time quantum in seconds per burst.

In the token bucket analogy, tokens are put into the bucket at a certain rate. The bucket itself has a specified capacity. If the bucket is filled to capacity, newly arriving tokens are discarded. Each token is the permission for the source to send a certain number of bits downstream into the network. To transmit a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.

If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens or the packet is discarded (if the bucket is full). The algorithm adds tokens to the bucket at a rate of ν bytes per second. By determining the maximum amount of data that can arrive over an interval T , it has been shown, analytically, that as T approaches infinity, the actual rate of traffic allowed into the network approaches the desired rate ν .

Since the burstiness of the traffic is not shaped, the bucket depth will be very “deep” and the interval T will be very long, so the burstiness of the traffic is not inhibited. The implementation of the token bucket is similar to that of the leaky bucket except the embedded software of the edge node will be generating the stream of token for the token bucket.

Figure 7.9: Token Bucket Implementation

The embedded software on the managed devices (routers/switches) can implement one of the two traffic-shaping algorithms to ensure that traffic entering the network complies with the value specified in the `traffic_limit_rate` managed object.

In summary, this chapter has demonstrated that EPIT can be implemented with existing technologies while remain consistent with the future autonomous management architecture trend.

Chapter 8 Conclusion

This thesis has described the EPIT approach, which is a combination of simulation analysis and closed-form analysis. It can predict the QoS on an end-to-end network path, with respect to the mean rates of the ingress traffic flows. It can work with any type of QoS metric (congestion, loss, or delay), as long as the quantity can be described by stochastic events. The thesis also provides a framework to integrate EPIT and Simulated Annealing in order to provide a traffic engineering solution with a minimal reduction cost (maximum utilization) that satisfies multiple end-to-end QoS requirements.

What distinguishes this approach from other IS approaches is that: 1) this approach is the first to use an IS-based approach in a multi-queue system under FGN input, and; 2) this approach is a novel way to use IS in traffic engineering applications in which EPIT can treat the network as a black box. The proposed approach has been shown to be accurate, flexible and scalable.

This thesis also performed efficiency analysis on EPIT to demonstrate that EPIT can project a wide range of overflow probabilities, even if the probability is extremely small, without having to increase the computation cost. This analysis has been separated into two parts: a closed-form analysis in a single node with single LRD input, and a P2P and VPN networks based simulation analysis in which multi-queue with multiple LDR input

scenario is used. In both analyses, it has been show that EPIT can be efficient under LRD input.

Since EPIT is based on IS, and no efficiency analysis has been performed on IS under LRD input, this thesis fills this gap. First, an analytical equation characterizing the normalized variance of the IS simulation is derived. Then, the optimal twist value for the IS biasing is derived. This also enables one to observe the behaviour of the IS as a function of the simulation environment parameters. The derivation of the normalized variance equations provided the basis for an efficiency analysis on IS simulations. Based on existing efficiency criteria, it was shown that the IS simulation can (with the right biasing) be efficient, under a LRD input and in a single node system.

In summary, the analysis and simulation results of this thesis have demonstrated that the proposed EPIT and SA-EPIT technique can predict a wide range of overflow probabilities, in complex networks with traffic rate control at the network edge, under a LRD traffic model. There is also a significant gain in terms of simulation time using EPIT in comparison with the classic Monte Carlo approach. Its ability in fulfilling end-to-end QoS requirements while maximizing the network utilization makes EPIT an attractive solution to applications such as VPN and Peer-to-Peer communications.

Chapter 9 Limitations of EPIT and Future Works

There has been little work done to study the limitation of Importance Sampling based approaches (in this case, EPIT), except the domination condition for IS that is stated in [8].

Based on the workings of IS, the event of interest must be stochastic, and the probability of event occurrences must be well defined. In other words, the system in which the event occurs must be in steady state. For example, if a queuing network is unstable where the queue backlog is growing and never stops, the probability of buffer overflow will always be one; thus IS based approach cannot be used in this case.

This thesis used FIFO queue as an example in the simulation. For networks that implement the Differserv feature, some form of priority-based scheduling is used in its nodes. As long as the scheduler(s) keep the network in a steady and stable state, EPIT should remain effective. If there is a feedback mechanism in a network (measure the performance of the network and automatically take actions to ensure a certain QoS), we can view the network as a control system problem. In the case that the system is unstable and oscillating, EPIT cannot be used since the system is constantly changing, (i.e. each replication used in EPIT will be as if each replication was run on a different system).

Wireless network have garnered a lot of research attention in recent years. If the motion of the mobile users of a wireless network is random and a steady state is achieved in the network eventually, EPIT can be used. On the other hands, if the motion not random, for example, all users move in one direction, EPIT is not applicable.

A single TCP flow follows the predefined phases of slow-start, congestion avoidance, fast retransmit, and fast recovery. Since its behavior is non-stationary, EPIT is not a good candidate in this case. However, if one is analyzing a large number of aggregated TCP flows as a whole which results in an equilibrium state, then EPIT can be employed.

A future research topic can be dedicated to the study of the limitation of EPIT in various scenarios and applications.

Another research topic is to integrate other heuristic search algorithms with EPIT. Search algorithms such as the genetic search, tabu search, can be combined with EPIT to see if they can yield a solution faster. Future work can also check if those solutions will have a smaller reduction cost than the EPIT-SA solutions while still meeting the QoS requirements.

Another area to investigate is the use of discrete traffic model instead of the fluid traffic model. By actually quantizing the traffic into packets, and assigning packet size and other packet-related characteristics to the traffic, the simulation model will be more complex. However, it is anticipated that the simulation run time of a discrete traffic model will be much longer than the fluid based traffic model.

A mixture of long-range dependent traffic (Hurst parameter close to 1.0) and short-range dependent traffic (Hurst parameter close to 0.5) is used in the simulation of this

thesis. It would be interesting to apply a mixture of different stochastic distribution (for example, FGN mixed with Poisson) for the simulated traffic. Clearly, the likelihood ratio has to be recalculated. Based on the theoretical properties of EPIT (IS), the author believes that the overall EPIT approach will most probably remain effective.

References

- [1] MADRAS, N. 2002. *Lectures on Monte Carlo Methods*. American Mathematical Society, Providence, RI.
- [2] ROSEN, E., VISWANATHAN, A., AND CALLON, R. 2001. Multiprotocol Label Switching Architecture, RFC-3031, Available online at www.ietf.org/rfc/rfc3031.txt
- [3] RAVINDRAN, R. S., HUANG, C., AND THULASIRAMAN, K. 2007. Managed Dynamic VPN Service: Core Capacity Sharing Schemes For Improved VPN Performance. In *Proceedings of the 2007IEEE International Conference on Communications*, Glasgow, Scotland, June 2007, 211 – 216.
- [4] KHALIL, I., AND BRAUN, T. 2000. Edge provisioning and fairness in VPN-Diffserv networks. In *Proceedings of the 9th International Conference on Computer Communications and Networks*, Las Vegas, USA, October, 2000, 424 – 431.
- [5] BASET, S.A., AND SCHULZRINNE, H.G. 2006. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol, In *Proceedings of the 25th IEEE International Conference on Computer Communications*, Barcelona, Spain, April 2006, 1 – 11.
- [6] PURANDARE, D., AND GUHA, R. 2007. An Alliance Based Peering Scheme for P2P Live Media Streaming, *IEEE Transactions on Multimedia*, 9(8), 1633 – 1644
- [7] ZHANG, Y.H., AND FANG, Y. 2007. A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks, *IEEE Transactions on Parallel and Distributed Systems*, 18(8), 1134 – 1145.
- [8] RUBINSTEIN, R.Y., AND MELAMED, B. 1998. *Modern Simulation and Modeling*. John Wiley & Sons. New York, NY.
- [9] SADOWSKY, J.S., AND BUCKLEW, J.A. 1990, On Large Deviations Theory and Asymptotically Efficient Monte Carlo Estimation, *IEEE Transactions on Information Theory*, 36(3), 579-588
- [10] DUFFIELD, N.G., AND O'CONNELL, N. 1995. Large deviations and overflow probabilities for the general single server queue, with application. In *Proceedings of Cambridge Philosophical Society*, July 1995, 118, 363-374.
- [11] LELAND, W., TAQQU, M., WILLINGER, W., AND WILSON, D.V. 1994. On the Self-similar Nature of Ethernet Traffic (extended version), *IEEE/ACM Trans. on Networking*, 2(1), 1-15.
- [12] PAXSON, V., AND FLOYD, S. 1995. Wide-area Traffic: the Failure of Poisson Modeling, *IEEE/ACM Transactions on Networking*, 3(3), 226-244.

- [13] LEDESMA, S., AND LIU, D. 2000. A fast method for generating self-similar network traffic. In *Proceedings of 2000 International Conference on Communication Technology*, Beijing, China, August, 2000, 54 – 61.
- [14] Ma, S., AND JI, C. 2001. Modeling heterogeneous network traffic in wavelet domain. *IEEE/ACM Transactions on Networking*, 9(5), 634 – 649.
- [15] GEIST, R., AND WESTALL, J. 2000. Co-relational and distributional effects in network traffic models. In *Proceedings of the 2000 Computer Performance and Dependability Symposium*, Chicago, USA, March, 2000, 113 – 122.
- [16] Roughan, M., AND Veitch, D. 1999. Measuring long-range dependence under changing traffic conditions. In *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies*. New York, USA, March, 1999, 1513 - 1521.
- [17] HAGIWARA, T., DOI, H., TODE, H., AND IKEDA H. 2000. High-Speed Calculation Method of the Hurst Parameter Based on Real Traffic. In *Proceedings Of the 25th Annual IEEE Conference on Local Computer Networks*. Tampa, USA, November, 2000, 662 – 669.
- [18] BOX, G.E.P., AND JENKINS, G. M. 1976. *Time Series Analysis: Forecasting and Control 2nd edition*. Holden Day, San Francisco, CA.
- [19] HOSKING, J.R.M. 1984. Modeling Persistence in Hydrological Time Series Using Fractional Differencing. *Water Resources Research*, 20(12), 1898-1908.
- [20] HUANG, C., DEVETSIKIOTIS, M., LAMBADARIS, I., AND KAYE, A.R. 1995. Self-Similar Modeling of Variable Bit Rate Compressed Video: A Unified Approach. In *Proceedings of ACM SIGCOMM'95*, Cambridge, USA, August, 1995, 114-125.
- [21] RIBEIRO, V.J., RIEDI, R.H., AND BARANIUK, R.G. 2006. Multi-scale Queueing Analysis. *IEEE/ACM Transactions on Networking*, 14(5), 1005 – 1018.
- [22] BREGNI, S., AND JMODA, L. 2007. Accurate Estimation of the Hurst Parameter of Long-Range Dependent Traffic Using Modified Allan and Hadamard Variance. *IEEE Transactions on Communications*, 55(11), 2224 – 2224.
- [23] MASUGI, M. 2006. Recurrence Plot-Based Approach to the Analysis of IP-Network Traffic in Terms of Assessing Non-stationary Transitions Over Time. *IEEE Transactions on Circuits and Systems*, 53(10), 2318–2326.
- [24] NEUTS, M.F. 1981. *Matrix-Geometric Solutions in Stochastic Modeling: An Algorithmic Approach*. Johns Hopkins University Press, Baltimore, MD.
- [25] WALRAND, J. 1988. *An Introduction to Queueing Networks*. Prentice Hall, New York, NY.
- [26] KESIDIS, G., WALRAND, J., AND CHANG, C.S. 1993. Effective Bandwidth for Multi-class Markov Fluids and Other ATM Sources. *IEEE/ACM Trans. Networking*, 1(4), 424-428.

- [27] De VECIANA. G., AND KESIDIS. G. 1996. Bandwidth Allocation for Multiple Qualities of Service Using Generalized Process Sharing. *Transaction on Information Theory*, 42(1), 268-272.
- [28] MICHAEL, M., RUDIGER, M., AND JOACHIM, C. 2006. Capacity Over-provisioning for Networks with Resilience, In *Proceedings of SIGCOMM'06*, Pisa, Italy, September, 2006, 87-98.
- [29] JIANG, Y. 2006. A Basic Stochastic Network Calculus. In *Proceedings of SIGCOMM'06*, Pisa, Italy, September, 2006, 123-134.
- [30] DUFFIELD, N.G., AND O'CONNELL, N. 1993. Large Deviations and Overflow Probabilities for the General Single-Server Queue, with Applications. *Technical Report DIAS-STP-93-30*, Dublin Institute for Advanced Studies.
- [31] NORROS, I. 1993. Studies on a Model for Connectionless Traffic, Based on Fractional Brownian Motion. In *Proceeding of Conference on Applied Probability in Engineering, Computer and Communication Sciences* Paris, France, June, 1993.
- [32] NORROS, I., 1994. A storage model with self-similar input. *Queueing Systems*, 16, 387—396.
- [33] MASSOULIE, L., AND SIMONIAN, A. 1999. Large buffer asymptotics for the queue with FBM input. *Journal of Applied Probability*, 36(3), 894-906.
- [34] EUN, D.Y., AND SHROFF, N.B. 2005. Network Decomposition: Theory and Practice. *IEEE/ACM Transaction on Networking*, 13(3), 526- 539.
- [35] COURCOUBETIS, C., AND WEBER, R. 1996. Buffer Overflow Asymptotics for a Buffer Handling Many Traffic Sources. *Journal of Applied Probability*. 33(3), 886-903.
- [36] CRUZ, R.A. 1991. A calculus for network delay, Part II: Network analysis. *IEEE Transactions on Information Theory*, 37, 132-141.
- [37] ZHANG, H., AND KNIGHTLY, E. 1994. Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models. In *Proceeding of ACM SIGMETRICS '94*, Nashville, TN, May, 1994, 211-220.
- [38] WISCHIK, D. 1999. The output of a switch, or, effective bandwidths for networks. *Queueing Systems*, 32, 383-396.
- [39] ZHAO, Y., CHEN, Y., AND BINDEL, D. 2006. Towards Unbiased End-to-End Network Diagnosis. In *Proceedings of SIGCOMM'06*, Pisa, Italy. September, 2006, 219-230.
- [40] JIN, Y., BALI, S., DUNCAN, T.E., AND FROST, V.S. 2007. Predicting Properties of congestion Events for a Queueing System With fBm Traffic. *IEEE/ACM Transactions on Networking*, 15(5), 1098 – 1108.
- [41] HUANG, C., DEVETSIKIOTIS, M., LAMBADARIS, I., AND KAYE, A.R. 1999. Fast Simulation of Queues with Self-Similar Traffic. *Communication in Statistics-Stochastic Models*, 15(3), 429-460.
- [42] CRANE, M.C., AND IGLEHART, D.L. 1975. Simulating Stable Stochastic Systems: III. Regenerative Processes and Discrete-Event Simulations. *Operations Research*, 23(1), 33-45.

- [43] BUCKLEW, J.A. 2004. *Introduction to Rare Event Simulation*, Springer, New York, NY.
- [44] FALKNER, M., DEVETSIKIOTIS, M., AND LAMBADARIS, I. 1999, Issues on Fast Network Simulation by Use of Effective and Decoupling Bandwidths. *ACM Transactions on Modeling and Computer Simulation*, 9(1), 45-58.
- [45] L'ECUYER, P., AND CHAMPOUX, Y. 2001. Estimating Small Cell-loss Ratios in ATM Switches via Importance Sampling. *ACM Transactions on Modeling and Computer Simulation*, 11(1), 76—105.
- [46] SCHLEBUSCH, H.J. 1993. On the asymptotic efficiency of importance sampling techniques. *IEEE Transactions on Information Theory*, 39(2), 710 – 715.
- [47] SANDMANN, W. 2007. Efficiency of Importance Sampling Estimators. *Journal of Simulation*, 1(2), 137-145.
- [48] LI, C., BURCHARD, A., AND LIEBEHERR, J.A. 2007. Network Calculus With Effective Bandwidth. *IEEE/ACM Transactions on Networking*, 15(6), 1442 – 1453.
- [49] KREHER, D.L., AND STINSON, D.R. 1999. *Combinatorial Algorithms*. CRC Press. New York, NY.
- [50] NORROS, I. 1995. On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks. *IEEE JSAC*, 13(6), 953-962.
- [51] MANDELBROT, B.B. 1983. *The Fractal Geometry of Nature*. W. H. Freeman, New York, NY.
- [52] BANKS, J., CARSON, J.S., NELSON, B.L., AND NICOL, D.M. *Discrete-Event System simulation 4th edition*, Prentice Hall, New York, NY.
- [53] LI, J.S., WOLISZ, A., AND POPESCU-ZELETIN, R. 1998. Modeling and Simulation of Fractional ARIMA Processes based on Importance Sampling. In *proceedings of the 1998 ACM symposium on Applied Computing*, Atlanta, USA, February, 1998, 453-455.
- [54] CHENG, Y., AND ZHUANG, W. 2003. Effective Bandwidth of Multi-class Markovian Traffic Sources and Admission Control with Dynamic Buffer Partitioning. *IEEE Transactions on Communications*, 51(9), 1524 – 1535.
- [55] BERTSIMAS, D., PASCHALIDIS, I.C., AND TSITSIKLIS, J.N. 1998. Asymptotic Buffer Overflow Probabilities in Multiclass Multiplexers: an Optimal Control Approach. *IEEE Transactions on Automatic Control*, 43(3), 315 – 335.
- [56] PAREKH, A. K., AND GALLAGER, R.G. 1993. A Generalized Processor Shainrg Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE Transactions on Networking*, 1(3), 344-357.
- [57] ROMANOW, A., AND FLOYD, S. 1995. Dynamics of TCP Traffic over ATM Networks. *IEEE Journal on Selected Area in Communication*, 13(4), 633-641.
- [58] PAXSON, V. 1999. End-to-End Internet Packet Dynamics. *IEEE/ACM Transaction on Networking*, 7(3), 277-292.

- [59] BORELLA, M.S., SWIDER, D., ULUDAG, S., AND BREWSTER, G.B. 1998. Internet packet loss: measurement and implications for end-to-end QoS. In *proceedings of the 1998 ICPP Workshops on Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing*, Minneapolis, USA, August, 1998, 3 – 12.
- [60] KUNG, H.T., AND WANG, S.Y. 1999. In *proceedings of the 7th Annual International Conference on Network Protocols*, Toronto, Canada, November, 1999, 222-235
- [61] SCHULZRINNE H., CASNER, S., FREDERICK, R., AND JACOBSON, V. RTP: A Transport Protocol for Real-Time Applications, RFC-3550, Available online at www.ietf.org/rfc/rfc3550.txt
- [62] KIND, A., DIMITROPOULOS, X., DENAZIS, S., CLAISE, B. 2008. Advanced Network Monitoring Brings Life to the Awareness Plane, *IEEE Communications Magazine*, October, 2008.
- [63] ROSE, O. 1996. Estimation of the Hurst Parameter of Long-Range Dependent Time Series. *University of Wurzburg Institute of Computer Science Report*, 137.
- [64] YE, T., KAUR, H., KALYANARAMAN, S., and YUKSEL, M. 2008 Large-Scale Network Parameter Configuration Using an On-Line Simulation Framework. *IEEE/ACM Transaction on Networking*. 16(4), 777-790.
- [65] CLEGG, R. 2006. A Practical Guide to Measuring the Hurst Parameter. *International Journal of Simulation: Systems, Science & Technology*, 7(1), 3-14.