

**Accelerated Simulation of Power-Law Traffic  
in Packet Networks**

**By**

**Ho I Ma**

**SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**

**Supervised by Dr. John A. Schormans**

**Department of Electronic Engineering  
Queen Mary  
University of London**

**September 2003**

*To my family*

## Acknowledgements

I would like to thank my supervisor Dr. John Schormans who has given me all the support, assistance and encouragement that I needed throughout my Ph.D. study. John, you were the one who make me believe that I can do this, thanks ever so much. I would also like to express my thanks to Prof. Laurie Cuthbert and Dr. Raul Mondragón. Moreover, my acknowledgement of the financial funding provided by EPSRC.

Many thanks to Lynda Rolfe and Phil Wilson, as well as the computing geniuses: Andy Martin, Kok Ho Huen and Julian Rodaway, for revitalizing the life of “penguin” whenever I needed it. Also, many thanks to Rob Stewart for his assistance and advice during my early research days.

Finally, my salute to Yasir Alfadhil, Jimmy Leung, Matthew Woolf, Shi Zhou and my dear Wee Kian Toh for their useful suggestions on this research. Thanks all my friends in the department for making the research lab such a enjoyable place to be in, and I am certainly privileged to be part of the Queen Mary family.

# Abstract

Packet networks have become an important research topic in telecommunications. This research focuses on simulations of Power-law traffic in packet networks, and more importantly, a novel method by which such simulations can be accelerated.

Considerable evidence has shown that network traffic may be dominated by Power-laws. This is due to the fact that file sizes available on the networks are heavy-tailed distributed, and the synchronisation of Transmission Control Protocol (TCP) connections. Power-law traffic possesses properties that contradict traditional Markovian assumptions, and cause very significant challenges for the designers of accurate traffic simulators. Heavy-tailed distributions decay hyperbolically, and thus results in extremely high or infinite variance in traffic. This leads to a high time consumption as the simulation requires a long period to reach steady-state. In cases in which rare events, e.g. buffer overflows, are involved, some of these simulations could take up to months or years.

In this research a technique called Traffic Aggregation (TA) was developed to speed up simulations of Power-law traffic. One of the factors that influences the run-time of a simulation is the number of traffic sources. TA aims to minimise this by reducing  $N$  sources to a single source and thereby offer acceleration. The technique enables large-scale steady-state simulations in the sense that a system is able to reach stability much earlier. TA also plays an important role in abridging complexity. Results show that TA is highly accurate while providing excellent speedup in network simulations. In addition, it is important to note that TA, and many other techniques for accelerating simulation, are not mutually exclusive: TA can be operated in parallel with other acceleration and variance reduction methods.

TA has been validated in a number of network scenarios. Firstly, the technique was applied to a bottleneck situation where Power-law sources were multiplexed. Traffic was generated using non-truncated and truncated heavy-tailed distributions, and both homogeneous and heterogeneous settings were used. Secondly, an end-to-end link with foreground (FG) and background (BG) traffic was considered. TA was applied to replace background sources at

individual nodes. Furthermore, the technique was also examined for use with the Weighted Round Robin (WRR) scheduling mechanism. Lastly, the conjunction of TA and RESTART (Repetitive Simulation Trials After Reaching Thresholds) was examined.

In addition, attempts were made to aggregate an end-to-end path, omitting all nodes but one. The technique is named Node Aggregation (NA), early results show that the technique is highly accurate in the prediction of delay time probabilities for an end-to-end path.

Besides simulation techniques, analytical and hybrid methods were also developed in the area of Power-law traffic for the prediction of queueing behaviours. These methods were validated by simulation results and were found to be highly accurate.

# Table of Contents

<b>Acknowledgements</b> .....	<b>2</b>
<b>Abstract</b> .....	<b>3</b>
<b>Table of Contents</b> .....	<b>5</b>
<b>List of Figures</b> .....	<b>9</b>
<b>List of Tables</b> .....	<b>11</b>
<b>Glossary</b> .....	<b>12</b>
<b>List of Mathematical Symbols</b> .....	<b>14</b>
<b>Chapter 1 Introduction</b> .....	<b>15</b>
1.1 Objective of this research .....	16
1.2 Contribution of this thesis.....	17
1.3 Organization of this thesis .....	17
<b>Chapter 2 Self-Similarity in IP Traffic</b> .....	<b>20</b>
2.1 IP networks - overview .....	20
2.1.1 Impact of self-similarity .....	21
2.2 Evidence of network traffic self-similarity .....	23
2.3 Stochastic self-similarity and long range dependence .....	24
2.3.1 A mathematical approach .....	25
2.3.2 Heavy-tailed distributions .....	27
2.3.3 Power-law queueing .....	29
2.4 Modelling approaches.....	30
2.5 Choice of model – ON-OFF model .....	31
2.6 Generating Pareto distributed numbers.....	36
2.7 Summary.....	38
<b>Chapter 3 Simulation of Telecommunication Networks</b> .....	<b>39</b>
3.1 Importance of simulation modelling.....	39
3.2 Literature review of simulation techniques.....	40
3.2.1 Static vs. dynamic simulation model.....	40

3.2.2	Deterministic vs. stochastic simulation model .....	41
3.2.3	Continuous vs. discrete simulation model .....	41
3.2.4	Simulation model overview .....	42
3.2.5	Simulation clock .....	43
3.2.5.1	Next-event time advance mechanism .....	43
3.2.5.2	Fixed-increment time advance .....	44
3.2.6	Random number generators .....	44
3.3	Simulation with reference to output analysis .....	46
3.3.1	Terminating simulations .....	47
3.3.2	Non-terminating simulations .....	47
3.3.3	Choice of approach – steady-state parameter analysis .....	49
3.4	Literature review of accelerated simulation techniques .....	49
3.4.1	Distributed and parallel simulation .....	50
3.4.2	Hybrid techniques .....	50
3.4.3	Statistical variance reduction .....	51
3.5	Validation and verification .....	52
3.5.1	Confidence intervals .....	53
3.5.2	Power-law bestfit and exponentially wider bins .....	55
3.6	Summary .....	57
<b>Chapter 4</b>	<b>Introduction to Traffic Aggregation (TA) .....</b>	<b>58</b>
4.1	Traffic Aggregation (TA) .....	58
4.2	Parameterization of TA .....	61
4.2.1	Deriving $T_{ON}$ .....	62
4.2.1.1	Relationship between $D_{ON}$ and $T_{ON}$ .....	62
4.2.1.2	Relationship between $D_{OFF}$ and $T_{ON}$ .....	64
4.2.1.3	Relationship between $N$ and $T_{ON}$ .....	65
4.2.1.4	Relationship between $R$ and $T_{ON}$ .....	66
4.2.1.5	Relationship between $C$ and $T_{ON}$ .....	66
4.2.1.6	Formula for $T_{ON}$ .....	68
4.2.2	Deriving $R_{ON}$ .....	73
4.2.3	Deriving $T_{OFF}$ .....	73
4.3	Validation using queueing behaviour .....	74
4.4	Accuracy of TA under different network scenarios .....	75
4.4.1	Non-truncated Pareto ON-OFF traffic .....	75
4.4.1.1	Overlapping homogeneous Pareto ON-OFF sources .....	75
4.4.1.2	Overlapping heterogeneous Pareto ON-OFF sources .....	82
4.4.2	Truncated Pareto ON-OFF traffic .....	85
4.4.2.1	Overlapping homogeneous truncated Pareto ON-OFF sources .....	86

4.4.2.2	Overlapping heterogeneous truncated Pareto ON-OFF sources.....	89
4.5	Acceleration achieved by TA.....	90
4.5.1	Traffic generation.....	91
4.5.2	Buffer monitoring process.....	93
4.5.3	Rate of reaching steady-state.....	95
4.6	Summary.....	97
<b>Chapter 5</b>	<b>TA in End-to-end Connections.....</b>	<b>98</b>
5.1	End-to-end network studies.....	98
5.1.1	Deriving an equivalent TA model for BG traffic substitution.....	100
5.1.2	Non-truncated Pareto FG and BG traffic.....	101
5.1.3	Truncated Pareto FG and Pareto BG traffic.....	104
5.2	End-to-end connections with WRR.....	107
5.2.1	Round Robin (RR) and Weighted Round Robin (WRR).....	107
5.2.2	Weights of FG and BG traffic.....	108
5.2.3	Poisson FG and non-truncated Pareto BG.....	109
5.2.4	Poisson FG and truncated Pareto BG.....	111
5.3	Acceleration.....	112
5.4	Summary.....	113
<b>Chapter 6</b>	<b>TA in conjunction with RESTART.....</b>	<b>114</b>
6.1	Introduction.....	114
6.2	RESTART.....	115
6.3	RESTART: in conjunction with TA.....	117
6.3.1	Simulation approach and comparison with analysis.....	118
6.3.2	TA + RESTART with a single threshold.....	119
6.3.3	TA + RESTART with multiple thresholds.....	120
6.4	Acceleration.....	122
6.5	Summary.....	123
<b>Chapter 7</b>	<b>Studies on End-to-end Paths.....</b>	<b>124</b>
7.1	Introduction.....	124
7.2	Node Aggregation (NA).....	127
7.2.1	Parameterization of the transformed FG traffic.....	127
7.2.2	Mean of the total delay.....	128
7.2.3	Numerical example.....	128
7.3	A hybrid method for the prediction of end-to-end delays.....	131
7.4	Summary.....	134
<b>Chapter 8</b>	<b>Discussion, Future Work and Conclusions.....</b>	<b>135</b>

8.1	Discussion.....	135
8.2	Future Work.....	139
8.3	Conclusions.....	140
<b>Appendix 1</b>	<b>Design Rules and Equivalent Capacity for Buffering of a Pareto Source .....</b>	<b>141</b>
<b>Appendix 2</b>	<b>Hybrid Technique for Analysis of Multiplexed Power-Law Traffic...</b>	<b>145</b>
<b>Appendix 3</b>	<b>Aggregate Variance Method for Estimating H .....</b>	<b>149</b>
<b>Appendix 4</b>	<b>Sum of Squares of Differences.....</b>	<b>151</b>
<b>Author's Publications</b>	<b>.....</b>	<b>153</b>
<b>References</b>	<b>.....</b>	<b>154</b>

# List of Figures

Figure 2-1	Traffic traces at different time scales: An Ethernet traffic trace (first column), a synthetic trace from a traditional Markovian traffic model (second column), a synthetic trace from a self-similar traffic model (third column). Reproduced from [Willinger97].	22
Figure 2-2	A Cantor set to four levels of recursion.	25
Figure 2-3	Comparisons between Exponential and Pareto distributions. Reproduced from [Pitts01].	28
Figure 2-4	Comparisons on Pareto and Poisson distributed random numbers.	29
Figure 2-5	A single ON-OFF process.	32
Figure 3-1	Next-event time advance mechanism.	43
Figure 3-2	Fixed-increment time advance mechanism.	44
Figure 3-3	Concept of a random number generator.	45
Figure 3-4	Type of simulations with regard to output analysis.	47
Figure 3-5	Power-law best fit with respect to raw data.	56
Figure 3-6	Power-law bestfit with respect to binned data.	56
Figure 4-1	Concept of TA.	59
Figure 4-2	Modified TA.	60
Figure 4-3	Relationship between $D_{ON}$ and $T_{ON}$ .	63
Figure 4-4	Effect of $D_{OFF}$ on the overall queueing behaviour.	64
Figure 4-5	Relationship between $N$ and $T_{ON}$ .	66
Figure 4-6	Relationship between $C$ and $T_{ON}$ .	68
Figure 4-7	Relationship between $\kappa$ and $\kappa_1$ .	69
Figure 4-8	Relationship between error and $D_{ON}$ .	72
Figure 4-9	Relationship between error and $N$ .	72
Figure 4-10	Relationship between error and $C$ .	72
Figure 4-11	Approximation for packet loss probability.	75
Figure 4-12	Raw queueing results with the variation of $D_{ON}$ .	78
Figure 4-13	Bestfit applied to queueing results in Figure 4-12.	78
Figure 4-14	Raw queueing results with the variation of $N$ .	79
Figure 4-15	Bestfit applied to queueing results in Figure 4-14.	79
Figure 4-16	Raw queueing results with the variation of $C$ .	80
Figure 4-17	Bestfit applied to queueing results in Figure 4-16.	80

Figure 4-18	Raw queueing results with the variation of N and C .....	81
Figure 4-19	Bestfit applied to queueing results in Figure 4-18.....	81
Figure 4-20	Validation of the equivalent homogeneous model for multiplexed heterogeneous sources.....	84
Figure 4-21	Comparisons on the queueing behaviour between the overlapping truncated traffic case and its equivalent non-truncated model .....	88
Figure 4-22	Application of TA to an overlapping truncated Pareto traffic scenario.....	88
Figure 4-23	Validation of the equivalent truncated homogeneous model for multiplexed truncated heterogeneous sources .....	90
Figure 4-24	Traffic generation (wall-clock) time with increasing number of timeslots .....	92
Figure 4-25	Traffic generation (wall-clock) time with increasing number of sources.....	92
Figure 4-26	Buffer monitoring (wall-clock) time with increasing number of timeslots .....	94
Figure 4-27	Buffer monitoring (wall-clock) time with increasing number of packets.....	94
Figure 4-28	Comparisons on the queue tail extension .....	95
Figure 4-29	Acceleration achieved by TA in terms of rate of reaching steady-state .....	96
Figure 5-1	An end-to-end connection with FG and BG traffic .....	99
Figure 5-2	An end-to-end connection with TA BG traffic.....	99
Figure 5-3	TA for non-truncated Pareto BG traffic .....	103
Figure 5-4	Validations for the example given in Table 5-1 .....	103
Figure 5-5	TA for truncated Pareto BG traffic.....	106
Figure 5-6	Validations for the example given in Table 5-2 .....	106
Figure 5-7	An example of a mesh network topology .....	108
Figure 5-8	TA for the non-truncated Pareto BG traffic in the WRR example .....	110
Figure 5-9	Validations for the WRR example given in Table 5-3 .....	110
Figure 5-10	Delay time probabilities of the example in given in Table 5-4 .....	112
Figure 5-11	Acceleration achieved by TA in an end-to-end path .....	113
Figure 6-1	Concept of RESTART .....	115
Figure 6-2	Validation of TA for the RESTART example.....	117
Figure 6-3	TA + RESTART with a single threshold .....	120
Figure 6-4	TA + RESTART with multiple thresholds.....	121
Figure 7-1	Concept of NA .....	125
Figure 7-2	Comparisons on the delay time distribution with increasing the number of nodes ....	126
Figure 7-3	Validation of NA.....	130
Figure 7-4	Delay time distributions of FG packets a different nodes .....	133
Figure 7-5	Relationships between the queue offset, c, and the number of nodes.....	134

# List of Tables

Table 4-1	Parameters for monitoring the effect $D_{ON}$ .....	62
Table 4-2	Parameters for monitoring $N$ .....	65
Table 4-3	Parameters for monitoring $C$ .....	67
Table 4-4	Parameters for the validation of TA .....	77
Table 4-5	Parameters for the heterogeneous traffic multiplexing scenario.....	82
Table 4-6	Parameters for validating the equivalent homogeneous model method .....	85
Table 4-7	Parameters for the truncated homogeneous source scenario .....	87
Table 4-8	Parameters for heterogeneous truncated sources .....	89
Table 5-1	End-to-end studies with non-truncated Pareto FG and BG traffic.....	102
Table 5-2	End-to-end studies with truncated Pareto FG and BG traffic.....	104
Table 5-3	WRR studies with Poisson FG and Pareto BG traffic .....	109
Table 5-4	WRR studies with Poisson FG and truncated Pareto BG traffic .....	111
Table 6-1	Summary of the acceleration provided by TA and TA + RESTART with respect to state probability .....	122
Table 7-1	Parameters of the NA example and corresponding measurements prior to the entry of the last node (node 6).....	129
Table 7-2	Acceleration achieved by NA and TA+NA, with respect to wall-clock time.....	130
Table 7-3	Parameters used in the hybrid technique example.....	132
Table 7-4	DR's at individual nodes .....	133

## Glossary

ARIMA	Auto-Regressive Integrated Moving-Averaging
ATM	Asynchronous Transfer Mode
BG	Background
CPU	Central Processing Unit
DiffServ	Differentiated Services
DR	Decay Rate
FBM	Fractional Brownian Motion
FE	Fixed Effort
FG	Foreground
FGN	Fractional Gaussian Noise
FIFO	First-In-First-Out
FS	Fixed Splitting
FSN	Fractional Stable Noise
FTP	File Transfer Protocol
IID	Identically and Independently Distribution
IP	Internet Protocol
IS	Important Sampling
LAN	Local Area Network
LRD	Long Range Dependence
NA	Node Aggregation
PADS	Parallel And Distributed Simulation
PIRS	Parallel Independent Replicated Simulation
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RESTART	Repetitive Simulation Trials After Reaching Thresholds
SBM	Standard Brownian Motion
SIC	Sensitive dependence on Initial Conditions
SRD	Short Range Dependence
TA	Traffic Aggregation

TCP	Transmission Control Protocol
VBR	Variable Bit Rate
WAN	Wide Area Network
WRR	Weighted Round Robin
WWW	World Wide Web

## List of Mathematical Symbols

A	Rare event
b	Scaling factor of a Power-law bestfit (i.e. the gradient in a log-log scale)
c	Coefficient (offset) of a Power-law bestfit
C	Overall service rate in packets/timeslot
$D_{\text{OFF}}$	Mean OFF sojourn time of an original Pareto source
$D_{\text{ON}}$	Mean ON sojourn time of an original Pareto source
$E(x)$	Mean value of variable x
$G_i$	Proportions of different traffic indexed by i
MT	Number of thresholds (in RESTART)
N	Number of overlapping sources
$O[h]$	P(a FG packet experiences a delay of h timeslots across an end-to-end path)
$P[h]$	P(h packets in the queue at a randomly chosen packet arrival instant)
$P_i$	$P[C_i/C_{i-1}]$
q	The uppermost value of a Pareto number from a random generator
$Q[h]$	P(no. of packets in the queue at a packet arrival instant $\geq h$ )
R	Rate of arrival in packets/timeslot
$r_i$	Number of retrials
$R_{\text{OFF}}$	Rate of OFF arrival of TA
$R_{\text{ON}}$	Rate of ON arrival of TA
s	Number of sources are ON simultaneously
$T_{\text{FG/BG}}$	Amount of FG/BG traffic
$T_{\text{OFF}}$	Mean OFF sojourn time of TA
$T_{\text{ON}}$	Mean ON sojourn time of TA
v	Number of nodes in an end-to-end path
w	The smallest non-zero value of a uniform random generator
$\alpha$	Shape parameter of the Pareto distribution
$\varphi$	Location parameter of the Pareto distribution
$\lambda$	Rate of arrival of a Poisson traffic
$\rho$	Utilisation

# Chapter 1

## Introduction

In the past decade, there has been a growing interest worldwide in studying Internet Protocol (IP) networks in general, and the Internet in particular. However, exploring these networks leads to many difficulties due to various complex characteristics. Firstly, the Internet is huge; there are approximately 180 million hosts world wide [Netsizer], and despite of its size, the Internet is still expanding. Moreover, the many different kinds of applications result in a wide range of packet traffic heterogeneity.

As well as direct experimentation, packet level phenomena can be studied by analysis and simulation modelling, both having their advantages and disadvantages. The major problem with the former is that many complex scenarios cannot be represented by solvable expressions and even in those cases where the system can be expressed analytically, solving the related equations can be infeasible. This leaves simulation modelling as often a more preferable choice, because any complicated network can always be modelled to an arbitrary level of accuracy. However, the drawback is that time and computational demand can be unaffordable.

There are many factors that influence the run-time of a simulation: the number of sources, the size of a network etc. Time consumption for a simulation increases with the total number of sources. This can be quantified by reference to an ON-OFF source scenario: a single ON-OFF source consists of only two states, but  $N$  of them results in  $2^N$  states in total. This results in expanding the overall time of a simulation in two ways: firstly, the simulation needs to run much longer to encounter rare events; secondly, the traffic generation time increases proportionally with the total number of sources. Therefore, simulation run-times are significantly amplified when there are a large number of nodes in a network model.

In addition, high-resolution measurements show that IP traffic appears to be statistically self-similar [Leland94]. Researchers have shown that traditional Markovian models are

inadequate to capture this phenomenon, and new traffic models have been developed such that network performance can be accurately predicted, and, eventually, services delivered with the expected quality. Self-similar traffic is vastly different from traditional voice and data traffic models. Such traffic is highly variable, and extremely long packet-trains can be encountered. This type of traffic can be described by heavy-tailed distributions which possess properties that, critically, make stability of a simulation hard to achieve, and as a result, simulations require a long period to converge to steady-state, and, even at steady-state, high variability can still be observed [Crovella97].

Because of these considerations, researchers have put their attention in the area of efficient and effective modelling schemes. The idea is to reduce the overall network simulation time while still guaranteeing accurate results. Many accelerated simulation methods have been suggested, e.g. Important Sampling (IS) [Rubinstein81] and Repetitive Simulation Trials After reaching Thresholds (RESTART) [Altamirano91]. In the context of self-similar traffic, there is no lack of literature on fast simulation techniques [Gallardo98 & 99, Li98, Paxson95 & 97a]. IS has been applied in different self-similar traffic models to accelerate simulations. Gallardo et al [Gallardo98 & 99] proposed a fast simulation technique for Fractional Stable Noise (FSN) processes which correspond to the aggregation of a relatively large number of traffic streams mixed together into a single flow. In [Li98], a speedup scheme based on Fractional Auto-Regressive Integrated Moving-Average (ARIMA) process was presented. The scheme is said to provide an efficient way to monitor queueing performance in a system in which the traffic possesses both Short Range Dependent (SRD) and Long Range Dependent (LRD) properties. Paxson proposed a fast approximation technique for synthesizing Fractional Gaussian Noise (FGN) traffic via a Fourier transform [Paxson95 & 97a]. Significant speedup, as far as 50 times, was obtained, however, the drawback is that there is often some loss of accuracy.

## **1.1 Objective of this research**

This research aims to provide a simple and accurate way to accelerate simulations of Power-law traffic.

There are currently no existing acceleration technique specifically that targets the generation of Power-law traffic using heavy-tailed distributions, and those present in the literature are mostly generic methods, often require a thorough understanding of the techniques

themselves prior to application. In addition, accuracy is always affected by the choice of parameters. These factors are not desirable particularly in the area of Power-law traffic in the sense that modelling such traffic itself is already complicated due to its statistical properties. A simple and accurate approach is vital in the context of teletraffic engineering, so that network modelling involves Power-law traffic can be carried out efficiently and effectively.

## **1.2 Contribution of this thesis**

An acceleration technique, Traffic Aggregation (TA), was developed for modelling Power-law traffic. This technique is simple, accurate and provides significant speedup.

TA is a powerful simulation algorithm in the sense that it is applicable to many different scenarios, including end-to-end network paths. The technique is not mutually exclusive to other acceleration methods; it can be applied in conjunction with other acceleration methods to achieve further speedup.

Furthermore, while some speedup techniques, e.g. [Gallardo98, 99], aiming at the acceleration of LRD traffic, are based on the fact that a large number of sources will aggregate into a particular model (i.e. overlapping Pareto processes become Fractional Brownian Motion (FBM) if the number of sources is large enough), it is a particular advantage of the scheme developed here that it is not limited in this fashion, and can apply equally well to a small number of sources being multiplexed. Very recent research has noted the serious difficulties encountered when trying to get accurate models for a small aggregation of packet traffic [Fraleigh02].

## **1.3 Organization of this thesis**

In Chapter 2 a review of IP networks and their importance is provided. The chapter also focuses on the self-similarity in network traffic, discusses its discovery, evidence as well as its impacts on network performance. Moreover, self-similar traffic is also discussed in the simulation modelling sense, in which the necessity of new simulation approaches for such traffic is highlighted.

Chapter 3 concentrates on the simulation related issues. The importance of simulation in the context of telecommunication networks is reviewed, and techniques in this aspect are

discussed. The chapter also provides details on the choice of approach of this research. In addition, the role of fast simulation techniques and related literature are studied. Finally, validation and verification techniques exploited in this thesis are presented and explained.

Chapter 4 describes the principle of TA. Details of TA including the fundamental concept, derivation of parameters, as well as validations are provided. In the validation section, TA was tested under various traffic settings. Firstly, the technique was applied to a single buffer with homogeneous non-truncated Power-law traffic, and results were found to be highly accurate. Secondly, the application of TA was extended to truncated Power-law traffic as well as traffic sources with heterogeneous settings. Again, excellent results were obtained. In addition, acceleration achieved by TA is quantified.

Chapter 5 is an extension of the validation of TA. Unlike studies in Chapter 4 where validations were limited to a single network node, an end-to-end path was introduced in this chapter. Such a network path was fed with foreground (FG) and background (BG) traffic, and TA was exploited to substitute the many BG sources. Excellent agreement between the queueing behaviour of the original path and that of the TA substitution was obtained. Furthermore, TA was tested using this network path with the Weighted Round Robin (WRR) scheduling mechanism implemented, and results of this study are presented.

Chapter 6 outlines the conjunction of TA with another fast simulation technique – RESTART. The success of applying the two techniques concurrently indicates that TA is not mutually exclusively to other fast simulation techniques. The combination is efficient and effective, allowing buffer states with a low probability to be estimated within an affordable time, offering a powerful tool for tackling prohibitively time-consuming Power-law traffic.

Chapter 7 consists of studies carried out on end-to-end paths. An accelerated simulation technique for modelling end-to-end paths, namely Node Aggregation (NA), is presented. The idea is to use a single network node to mimic the operation of an end-to-end path which is fed by FG and BG traffic. Accurate results were obtained and the technique provides significant speedup.

The chapter also covers a hybrid technique for end-to-end performance evaluations. An end-to-end path involves a large number of nodes requires excessive effort to model and simulate. Simulation results show that delay time distributions shift along the

x-axis as the number of nodes increases. The hybrid technique uses delay statistics measured from the first few nodes to draw a relationship that describes the amount of shifting with respect to the number of nodes, and such a relationship is applied to predict the delay statistics of nodes further down the network path.

Chapter 8 consists of a discussion, further work related to this research and conclusions.

## Chapter 2

# Self-Similarity in IP Traffic

This chapter firstly presents a brief overview of IP networks and their importance, followed by the discovery of self-similar traffic, its impact and some evidence. It is then moving onto the theoretical background of network self-similarity and related details in the context of traffic modelling.

### **2.1 IP networks - overview**

The development of today's giant inter-networks can be traced back to 1970's when packet switching was first introduced. Traditional circuit switching was found to be inadequate and inefficient to deal with the increasing amount of data transmission at the time. IP networks provide interconnectivity between geographic regions. The best-known example is the Internet, which consists of more than 180 million hosts world wide, and despite of its size, it is still expanding. The many kinds of applications result in a wide range of traffic heterogeneity. IP networks have become indispensable in many aspects, and a thorough understanding of the topic is essential.

The combination of traditional Public Switched Telephone Networks (PSTNs) and circuit switching technology provides an excellent media for voice transmission. A connection or a call between two hosts is used as a basic unit. Resources are allocated to a specific call, and are maintained throughout until the call ends. Traffic is forwarded by the circuit switching technology in which switches are used to pass traffic from one link to the next. Since a connection is not shared in voice environments, all active connections are recorded, so that all the other idle links can be recognised when a new call needs to set up. Homogeneity and invariance of voice traffic fit very well with the circuit switching concept. Although voice calls tend to last for a long time respectively, the amount of bandwidth for the entire duration of the call is highly predictable.

The circuit switching approach, however, is far from ideal for data transmission. Data calls are bursty<sup>1</sup>, heterogeneous and unpredictable in the sense that a data connection ranges from extremely short to long durations and from extremely low to high rates of transmission. If a connection is reserved for a specific data call, a lot of resources are wasted in many ways. Firstly, much of the time of a data connection is idle as information appears in bursts. Secondly, the lifetime of a data connection is uncertain. Maintaining idle but open connections is impractical and uneconomical. The idea of segmenting information into packets, and forwarding them across networks independently, is efficient and effective as packets from different connections/calls can be multiplexed onto the same physical link during transmission. The basic concept of packet switching has remained essentially the same during the past three decades and continues to represent one of the few effective technologies for data communication in networks.

The success of IP networks has brought a lot of open questions, such as Quality of Service (QoS) issues, resource provisioning and dimensioning etc., necessitating new research in the field of telecommunications. Studying such huge and complex systems is a challenging task [Paxson97b, Floyd01] due to the networks' nontrivial heterogeneity and rapid changes. In addition, recent traffic measurements show that IP traffic often exhibits self-similarity or LRD, which intensifies the difficulties.

### **2.1.1 Impact of self-similarity**

Previous research has concluded that call arrivals at links in networks are well described by the Poisson distribution, and holding times by the exponential [Bear80]. There is no lack of well-defined Markovian models in the literature (for examples, see [Pitts01]), and these models have been widely used for traffic modelling in the past few decades.

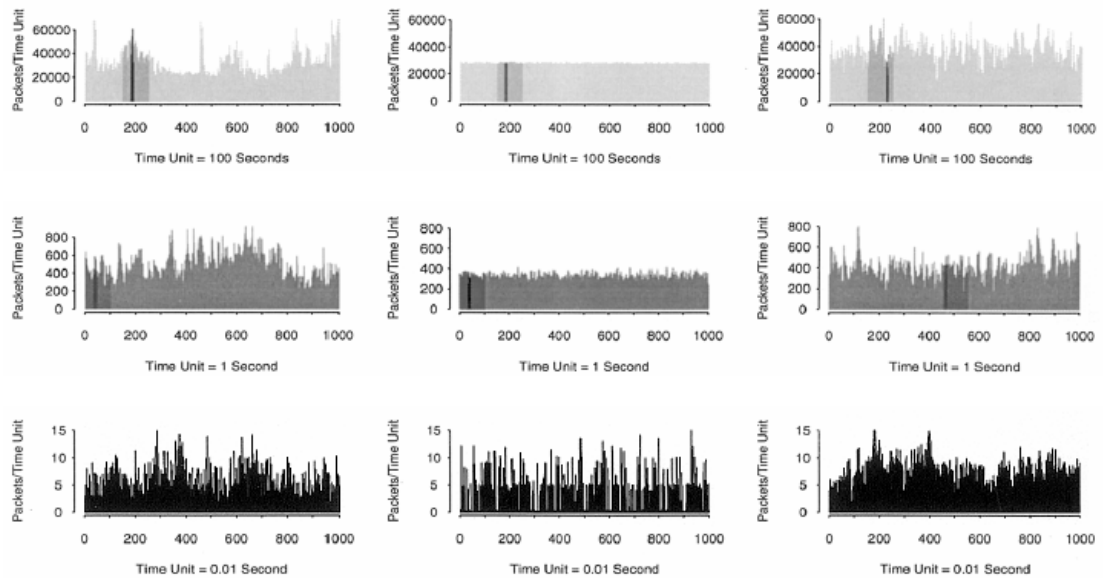
The impact of self-similar traffic in modern networks is nontrivial because this particular type of traffic disagrees with the long held paradigm of Markovian<sup>2</sup> traffic assumptions. The phenomenon is in stark contrast to the appearance of traditional traffic in the sense that bursts do not smooth out by simply aggregating over longer timescales. In other words, traffic burstiness remains the same regardless the degree of resolution [Leland94, Paxson94,

---

<sup>1</sup> Connections are very often brief – on the order of seconds, but intense.

<sup>2</sup> Refers to traditional Markovian cases only. There are new models for IP traffic developed based on Markovian approximations.

Crovella96], see Figure 2-1. In that figure, three sets of traffic traces are shown with respect to different time scales: 100s, 1s and 0.01s.



**Figure 2-1** Traffic traces at different time scales: An Ethernet traffic trace (first column), a synthetic trace from a traditional Markovian traffic model (second column), a synthetic trace from a self-similar traffic model (third column). Reproduced from [Willinger97].

The first column corresponds to a real Ethernet traffic trace in which traffic burstiness is persistent regardless of timescale. A Poisson synthetic trace is presented in the middle column, however, does not possess the same degree of invariance as the traffic smooths out as the time unit increases. This has certainly necessitated new traffic modelling approaches so that characteristics of self-similar traffic can be truly reflected. The last column consists of a synthetic trace which originates from a self-similar traffic model, and the invariant traffic burstiness is comparable with that of the real Ethernet traffic.

[Paxson94] provides evidence of how some crucial aspects of WAN traffic behaviour are inadequately described by Markovian models. Traditional packet traffic models give rise to exponential tail behaviour in queue size distributions, and such performance predictions appear to be too optimistic, resulting in insufficient resource allocations. Queue length

distributions of self-similar traffic models decay hyperbolically or slower-than-exponentially, and as a consequence, traffic packets encounter longer waiting times at network processing elements such as routers and switches. This deviation in the queueing behaviour is caused by two main reasons: the heavy-tailed nature of file sizes in networks [Crovella96] and the synchronisation of TCP connections [Peha97]. One of the most important characteristics of heavy-tailed distributions is the asymptotical shape of the tail of the distribution, which gives non-negligible probability to extremely large values. With respect to network traffic, this effectively means that transmissions can be in the form of extremely long bursts, contributing significantly towards the overall burstiness of the traffic.

## **2.2 Evidence of network traffic self-similarity**

Leland et al first introduced the notion of self-similarity to describe Local Area Network (LAN) traffic [Leland94]. Many studies also reported that traffic exhibits self-similarity in Wide Area Networks (WANs) [Addie95, Beran95, Crovella96]. In [Crovella96], World Wide Web (WWW) traffic was found to be self-similar. Data related to web file request times and transfer lengths were collected from the LAN in Boston University Computer Science Department. Several surveys were carried out on file size information at various WWW servers. File accessing patterns at both client and server sides were considered. Studies on the collected data showed that WWW traffic exhibited self-similarity on time scales of one second and above. Transmission and idle times were both heavy-tailed distributed. The former case was directly related to the Power-law distributed file sizes available in WWW. The latter case was determined by the thinking times of users.

Transmission Control Protocol (TCP) connections carrying File Transfer Protocol (FTP) and TELNET traffic have also been monitored [Paxson94]. Studies showed that the burstiness of TCP traffic was greatly underestimated. Firstly, TELNET traffic is well described by the Poisson approximation provided that the scale is fixed at hourly rates. However, the exponential inter-arrival times of the Poisson model result in a much less bursty traffic pattern. Secondly, FTP is responsible for carrying out large bulk transfer. Again, Poisson models do not capture sufficient burstiness to reflect the actual traffic appearance. While Poisson arrival process provides bursty traffic for small range of scales, burstiness over a wide range of scales is required to model FTP connections.

TCP is one of the reliable transport protocols in which an end-to-end congestion control mechanism is applied. One of the tasks of TCP is to adapt the transmission rate according to the actual network conditions in order to minimise lost packets. When transmitting self-similar traffic that is induced by Power-law file sizes, it was found that the adaptive nature of TCP preserves the self-similar characteristics of the traffic [Veres00]. Moreover, TCP propagates the effects to other parts of the network.

## 2.3 Stochastic self-similarity and long range dependence

The concept of self-similarity was pioneered by B. Mandelbrot [Mandelbrot63], and it has been applied to many aspects such as hydrology, biophysics, financial economics etc. Self-similarity is an important notion in describing the phenomenon where a certain property of an object – e.g. a nature image, the convergent sub domain of certain dynamical systems, a time series – is preserved with respect to scaling in space and/or time [Park00]. If an object is self-similar, the appearance of the object remains the same regardless of the degree of resolution. A Cantor set provides a graphical way to understand the concept, see Figure 2-2. The set is constructed using the following procedures [Stallings98]:

- 1) Begin with the closed interval  $[0,1]$ , represented by a line segment.
- 2) Remove the middle third of the line
- 3) For each succeeding step, remove the middle third of the lines created by the preceding step

These steps essentially form a recursive process that can be more precisely defined by assuming  $S_i$  is the Cantor set after  $i$  levels of recursion. Then, the following obtained:

$$S_0 = [0,1]$$

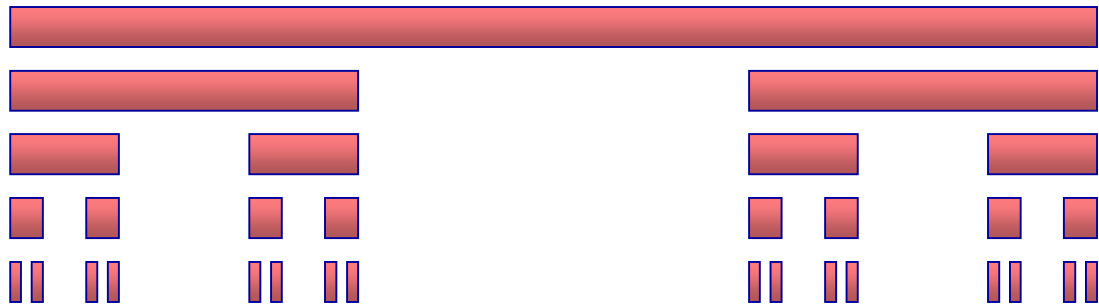
$$S_1 = [0, 1/3] \cup [2/3, 1]$$

$$S_2 = [0, 1/9] \cup [2/9, 1/3] \cup [2/3, 7/9] \cup [8/9,1]$$

and the pattern continues.

The Cantor line can be referred to as a time line, and each successive step magnifies the time scale by a factor of three. The Cantor set reveals two important properties observed in all self-similar processes: Firstly, the structure has arbitrarily small scales. If a part of the set is magnified repeatedly, the same complex pattern of points separated by gaps of various sizes is observed. However, the fine details of the structure are lost in the magnification process, a smooth and continuous line appears. Secondly, the appearance of the set remains the same at

each step. Nevertheless, these properties do not hold indefinitely for real phenomena. At some point under magnification, the structure and the similarity break down.



**Figure 2-2 A Cantor set to four levels of recursion**

The form of self-similarity demonstrated by the Cantor set is referred to as deterministic or exact self-similarity, in which the object possesses an exact resemblance of their parts with the whole at finer details. Surely, measured traffic traces cannot be exactly self-similar due to that the random nature of many network events. To study network traffic self-similarity, the degree of resemblance is no longer restricted to the exact appearance at different time scales. Instead, certain statistics of the rescaled time series are used as an index, and self-similarity of network traffic is determined by the preservation of its statistical properties. In particular, burstiness and/or variability are captured by second-order statistics, and the most important one is the autocorrelation function, in which its shape is monitored for the verification of self-similarity. Correlation, as a function of time lag, is assumed to decrease polynomially as opposed to exponentially. The existence of strong correlation “at a distance” is referred to as LRD.

### 2.3.1 A mathematical approach

Self-similarity can be expressed mathematically by considering a stationary discrete time stochastic process  $X = (X_1, X_2, X_3, \dots)$ .  $X$  can be used to describe the number of arriving packets in successive intervals of time units, where the process is second-order stationary. A new time series can be obtained by averaging  $X$  over non-overlapping blocks of size  $m$ ,  $X^{(m)}$ .

$$X^{(m)}(i) = \frac{1}{m} (X_{(i-1)m+1} + \dots + X_{im}) \quad i \geq 1 \quad \text{Eqn. 2-1}$$

$X^{(m)}$  is the aggregate process of the time series  $X$ . If the time scale is compressed in terms of  $X^{(m)}$  by assuming  $X^{(1)}$  is the highest magnification for the process, then increasing  $m$  results in moving towards smaller scales. For example, by definition,  $X^{(4)}$  is

$$X^{(4)}(i) = \frac{X_{4i-3} + X_{4i-2} + X_{4i-1} + X_{4i}}{4} \quad \text{Eqn. 2-2}$$

This is the same process but with a quarter the original resolution. However, the fine details of the highest magnification are expected to be lost through the averaging procedure. The way to determine whether  $X$  is self-similar is via a corresponding statistical measure such as variance and autocorrelation.

In general, the Hurst parameter,  $H$ , is used to describe the “strength” of the self-similarity, and this parameter is in the range of  $0.5 < H < 1$ . The higher the  $H$  value, the stronger is the self-similarity. Autocorrelation functions,  $r^{(m)}(k)$ , where  $k \geq 0$ , of the aggregated process  $X^{(m)}$  can be used to characterize second-order self-similarity. The process is said to be exactly second-order self-similar if

$$\begin{aligned} r^{(m)}(k) &= r(k) \\ &= \frac{1}{2} \left[ (k+1)^{2H} - 2k^{2H} + (k-1)^{2H} \right] \end{aligned} \quad \text{Eqn. 2-3}$$

It is called asymptotically second-order self-similar if

$$\begin{aligned} \lim_{m \rightarrow \infty} r^{(m)}(k) &= r(k) \\ &\sim k^{-(2-2H)} L(k) \end{aligned} \quad \text{Eqn. 2-4}$$

where  $L$  is a slowly decaying function at infinity.

A second-order self-similar process is said to exhibit LRD if the autocorrelation function has the following structure

$$r(k) \sim ak^{-\beta} \quad \text{Eqn. 2-5}$$

for  $0 < \beta < 1$ , where  $a$  is a positive constant. From this definition, it can be seen that the asymptotically second-order self-similar process  $X^{(m)}$  is LRD with  $\beta = 2-2H$ . LRD processes have nonsummable autocorrelation, i.e.  $\sum_k r(k) = \infty$ .

### 2.3.2 Heavy-tailed distributions

Evidence has suggested that self-similarity originates from heavy-tailed distributed file sizes [Crovella96, Park96, Paxson94]. A distribution is heavy-tailed if

$$P[X > x] \sim x^{-\alpha} \quad x \rightarrow \infty \text{ and } 0 < \alpha < 2 \quad \text{Eqn. 2-6}$$

The asymptotic shape of a heavy-tailed distribution is hyperbolic, and follows a Power-law which results in a high or even infinite variance. The term “Noah effect” is used to describe the infinite variance syndrome [Taqqu97]. The Pareto distribution is the simplest form of heavy-tailed distributions and an example of its application is modelling the distribution of incomes of a population [Pareto1896]. The probability distribution and density function are given by:

$$F(x) = 1 - \left(\frac{\varphi}{x}\right)^\alpha \quad \text{Eqn. 2-7}$$

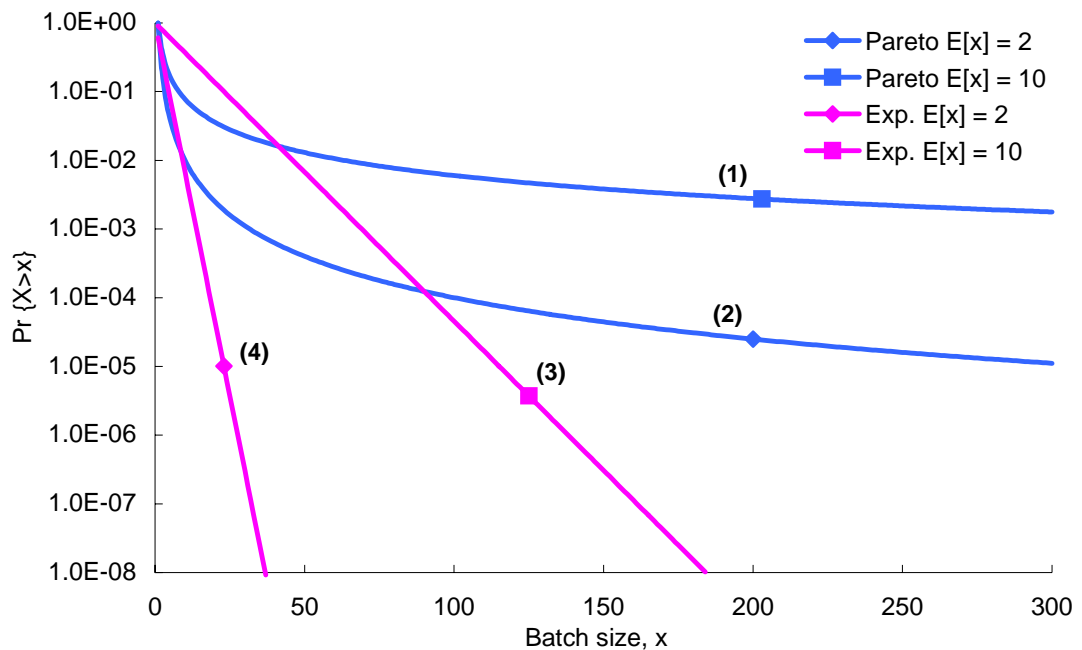
$$f(x) = \frac{\alpha}{\varphi} \cdot \left(\frac{\varphi}{x}\right)^{\alpha+1} \quad \text{Eqn. 2-8}$$

where  $\varphi$  is the location parameter which defines the minimum value that the distribution can accommodate, and  $\alpha$  is the shape parameter which determines the characteristic “decay” of the distribution. If  $\alpha \leq 2$ , the distribution has an infinite variance. If  $\alpha \leq 1$ , the distribution has an infinite mean. As  $\alpha$  increases, a larger portion of the random variables lies in the tail of the distribution. In general, the probability of obtaining exceptionally large values is non-negligible. The distribution has a mean equal to  $\alpha\varphi/(\alpha-1)$ , and  $\alpha$  is related to H by:

$$H = \frac{3 - \alpha}{2} \quad \text{Eqn. 2-9}$$

Figure 2-3 gives a good overview of the heavy-tailed concept. It is plotting the probability of getting more than x arrivals in any timeslot against x. For clarity, discrete distribution points are jointed together, forming solid lines. Curves (1) and (3) represent the Pareto and exponential distribution respectively. The two distributions have the same mean of 10, but there is a significant difference between the ways that their probability masses are located. The exponential distribution has the logarithm of the probability decaying linearly as x increases; the Pareto distribution falls very gradually with a large portion of random variables in the tail. For that reason, the possibility of getting extremely large values cannot

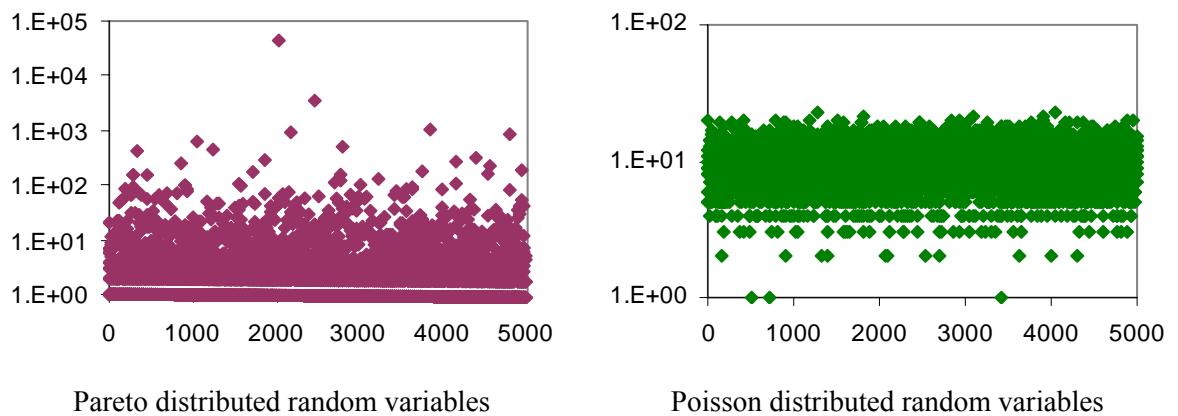
be ignored. The same result is also demonstrated by curves (2) and (4) which have a mean of 2.



**Figure 2-3 Comparisons between Exponential and Pareto distributions. Reproduced from [Pitts01]**

How do heavy-tailed distributions affect network studies in practical terms? A simple experiment was carried out to further investigate the heavy-tailed behaviour with respect to simulation modelling. A random number generator was constructed which gives Pareto distributed random variables with reference to the shape and location parameters. For this experiment,  $\phi$  was 1 and  $\alpha$  was  $10/9$  which provided a mean of 10. The generator was run for 5000 times and Figure 2-4 shows the obtained numbers. The majority of the variables from the generator were small, mostly below 100. There were a few variables which were much higher than the mean value, with values in the hundreds and thousands. The same experiment was performed for a traditional distribution. This time, Poisson distributed random variables with a mean of 10 were collected. The spread of the random variables was substantially different from the Pareto case. They were mainly around the mean value with a few of them at zero.

Comparing the two sets of results, the Pareto random variables appeared to be extremely variable. This is summarized by the infinite variance property of the distribution when  $\alpha \leq 2$ . Observations implied that heavy-tailed distributed sources can have long bursts, and sometime they are several orders higher than the expected value. Even though most of the variables are small, occasional large figures contribute the most to the mean and variance. In contrast, relatively minor variations are expected from Poisson traffic.



**Figure 2-4 Comparisons on Pareto and Poisson distributed random numbers**

### 2.3.3 Power-law queueing

Besides the actual distribution of transmission and idle times of the traffic, the research also concerns the actual decay of buffer queues. The term “Power-law” is used to describe the decay characteristics of queues, and they are expected to follow a Power-law, in other words, Eqn. 2-6 must be fulfilled.

A specified area has been selected in the research with respect to queues, the gradient of the queues to be precise. Power-law queues decay linearly in log-log scale, which can be described by

$$\log y = -b \cdot \log x + c \quad \text{Eqn. 2-10}$$

where  $x$  and  $y$  represent elements from the  $x$ -axis and  $y$ -axis respectively.  $b$  is the gradient of the linear decay and  $c$  is a constant. Specifications of  $x$  and  $y$  for this research are given in

4.3, here, the only focus is the condition set on the gradient,  $b$ , which is restricted in the range of  $1 < b < 2$ .

Considering Eqn. 2-10, if both sides carry out inverse log,

$$y = 10^{-b \cdot \log x + c}$$

$$y = 10^{-b \cdot \log x} \cdot 10^c$$

since  $c$  is a constant,  $10^c$  is also a constant

$$y = c \cdot 10^{-b \cdot \log x}$$

$$y = c \cdot (10^{\log x})^{-b}$$

and the below is obtained:

$$y = c \cdot x^{-b}$$

The gradient,  $b$ , therefore, corresponds to the characteristic parameter,  $\alpha$ , of a Power-law distribution.

## 2.4 Modelling approaches

The fact that Poisson assumptions are inadequate in describing LRD traffic has necessitated the development of new traffic models so that the long memory properties can be accurately reflected. Some of the common models in the literatures are ON-OFF models, FGN models, fractional ARIMA processes and chaotic deterministic maps.

The ON-OFF process was originally suggested by Mandelbrot for economic settings [Mandelbrot63] and Willinger et al applied the ON-OFF idea to stochastic self-similar traffic modelling [Willinger97]. This approach is attractive as it approximates network traffic by using the knowledge obtained from network measurements, providing an explanation of the observed data characteristics. An ON-OFF process alternates strictly between ON and OFF periods. The superposition of many ON-OFF sources with heavy-tailed sojourn times produces a self-similar process. Furthermore, ON-OFF models are very simple and rather good for IP traffic [Leland94, Paxson94].

Norros et al first suggested using FBN to generate traffic with respect to the mean input rate,  $H$  and the variance coefficient [Norros95]. It is aimed at producing a strictly self-similar process. For a second-order self-similar process, FGN, the incremental process of FBM, is used. These models were developed by referring to Standard Brownian Motion (SBM), which can be interpreted as the limit of random walk. SBM appeals for network modelling in the sense that the first order of approximation of the arrival behaviour in such a network can be viewed as independent events, and as a form of random walk.

ARIMA models were first introduced by Box and Jenkins in [Box70]. These models have been widely applied in many fields due to their simplicity and flexibility. Fractional ARIMA models are a natural extension of the ARIMA models. One of the important advantages of this modelling method is the ability to capture both SRD and LRD correlation structure in time series modelling, which allows modelling of complex structures like the case of Variable Bit Rate (VBR) video.

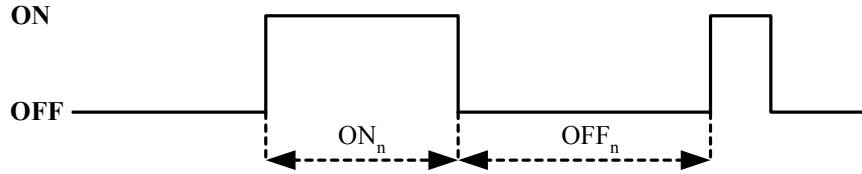
Fractal traffic modelling in terms of chaotic maps (also known as non-linear maps) was suggested by Erramilli et al [Erramilli94], and, in that paper, it was shown that low order chaotic systems can be applied to generate complex traffic behaviour. Such systems exhibit a chaotic behaviour that arises from a property known as Sensitive dependence on Initial Conditions (SIC). A number of models have been developed based on different classes of chaotic maps, e.g. intermittency maps [Pruthi95, Samuel99, Molnat97].

## **2.5 Choice of model – ON-OFF model**

The model adopted in this research is an ON-OFF model with Pareto distributed sojourn times. The choice of ON-OFF models is natural given its wide recognition in the context of teletraffic modelling [Crovella96, Gribble98, O'Mahony01, Schwefel99, Schwefel01, Willinger97, Zwart01].

This ON-OFF approach aims to reproduce the observed features of the traffic measurements, which represents traffic between a single source/destination pair. It alternates strictly between ON and OFF periods, the former case corresponds to an active period of a traffic source, such as a file transmission, Central Processing Unit (CPU) processing time for a specific job, whereas the latter represents the system idle time. Figure 2-5 displays a schematic view of an ON-OFF process.  $ON_1, ON_2, \dots$  are Identically and Independently

Distributed (IID) nonnegative random variables representing the ON sojourn times, and the mean ON duration time is  $D_{ON}$ .  $OFF_1, OFF_2, \dots$  are IID nonnegative random variables describing the duration of OFF states, and the mean OFF sojourn time is given by  $D_{OFF}$ .



**Figure 2-5** A single ON-OFF process

The relationship between the ON-OFF process and self-similarity is given in [Taqu97] which states that the superposition of many strictly alternating IID ON-OFF sources each of which exhibits the Noah effect<sup>3</sup> results in self-similar traffic. The aggregation of many such processes leads in the limit to FBM. This theory assumes the ON and OFF sojourn times are heavy-tailed, and this assumption is consistent with network measurements. OFF durations can be highly variable as some source model phenomena are triggered by humans which can have extremely long period of latency. And for ON periods, file sizes on networks are Power-law distributed, which implies that the transfer times for these files also have the same type of characteristics [Crovella96]. The ON-OFF model is extremely appealing due to its simplicity and the physical resemblance to network measurements.

If an individual traffic is represented as a time series  $\{W(t), t \geq 0\}$ , and viewing  $W(t)$  as the reward at time  $t$ .  $W(t) = 1$  when the source is ON and  $W(t) = 0$  when the source is OFF. The process follows the IID and the strict alternation assumptions. Now, consider the aggregation of  $M$  of these reward processes, each of them has its own reward sequence  $\{W^{(m)}(t), t \geq 0\}$ . The superposition of the cumulative cell/packet counts from the  $M$  sources in the given time  $t$  is

$$W_M(t) = \sum_{m=1}^M W^{(m)}(t) \tag{Eqn. 2-11}$$

<sup>3</sup> Refers to high variability or infinite variance

Then rescaling the aggregated process by a factor T gives:

$$W_M^*(Tt) = \int_0^{Tt} \left( \sum_{m=1}^M W^{(m)}(u) \right) du \quad \text{Eqn. 2-12}$$

which is the aggregated cumulative packet counts in the interval  $[0, Tt]$ . The main interest is to study the statistical behaviour of the stochastic behaviour of  $\{W_M^*(Tt), t \geq 0\}$  when T and M become large. Distributions for the ON and OFF periods need to be defined for one of the following two approaches: 1) FBM model for aggregate cumulative packet traffic, or 2) FGN (the increment process of FBM) for the aggregate traffic, which refers to the number of packets per unit time without cumulating. The choice of distribution needs to fulfil the condition that as  $M \rightarrow \infty$  and  $T \rightarrow \infty$ , the aggregate process  $\{W_M^*(Tt), t \geq 0\}$  adequately normalised becomes  $\{\sigma_{lim} B_H(t), t \geq 0\}$ , where  $B_H(t)$  is FBM and has a covariance structure dependent on H, and  $\sigma_{lim}$  is a finite normalising constant determined by the behaviour of the sojourn distributions in the ON and OFF states.  $\sigma_{lim}$  depends on the complementary distribution of sojourn times having the characteristic of a slow varying function at infinity.

Generally, the probability density of the ON or OFF distribution is denoted by  $f_j(x)$  and the cumulative distribution is given by

$$F_j(x) = \int_0^x f_j(u) du \quad \text{Eqn. 2-13}$$

where j is the subscript equals to 1 or 2, representing ON and OFF states respectively.

The complementary distribution is  $F_{jc}(x) = 1 - F_j(x)$ , which is the property required to be heavy-tailed and slowly varying at infinity, and it can be referred to as the probability of staying in one of the two states. For  $x \rightarrow \infty$ , an approximation expression is given by:

$$F_{jc}(x) \approx \frac{\ell_j}{x^{\alpha_j}} L_j(x) \quad \text{with } 1 < \alpha_j < 2 \text{ or } \sigma_j^2 < \infty \quad \text{Eqn. 2-14}$$

where  $\ell_j > 0$  is a constant and  $L_j > 0$  is a slowly varying function at infinity. For any  $t > 0$ ,  $\lim_{x \rightarrow \infty} L_j(tx)/L_j(x) = 1$  is valid.  $\alpha_j$  is the tail decay rate (DR). The mean length of distribution and the corresponding variance are given by

$$\mu_j = \int_0^{\infty} x f_j(x) dx \quad \text{Eqn. 2-15}$$

$$\sigma_j^2 = \int_0^{\infty} (x - \mu_j)^2 f_j(x) dx \quad \text{Eqn. 2-16}$$

respectively.

Defining notations with reference to Eqn. 2-14:

- 1) when  $1 < \alpha_j < 2$ , set  $a_j = \frac{\ell_j \Gamma(2 - \alpha_j)}{\alpha_j - 1}$ .
- 2) when  $\sigma_j^2 < \infty$ , set  $\alpha_j = 2$ ,  $L_j \equiv 1$  and  $a_j = \sigma_j^2$ .

The normalisation factors and the limiting constants in the theorem below depend on whether the following expression is finite, 0 or infinite.

$$b = \lim_{t \rightarrow \infty} t^{\alpha_2 - \alpha_1} \frac{L_1(t)}{L_2(t)} \quad \text{Eqn. 2-17}$$

If  $0 < b < \infty$ , in this case  $\alpha_1 = \alpha_2$  and  $b = \lim_{t \rightarrow \infty} \frac{L_1(t)}{L_2(t)}$ , set  $\alpha_{\min} = \alpha_1 = \alpha_2$ ,

$$\sigma_{\lim}^2 = \frac{2(\mu_2^2 a_1 b + \mu_1^2 a_2)}{(\mu_1 + \mu_2)^3 \Gamma(4 - \alpha_{\min})} \quad \text{where } L = L_2; \quad \text{Eqn. 2-18}$$

On the other hand, if  $b = 0$  or  $b = \infty$ , set

$$\sigma_{\lim}^2 = \frac{2\mu_{\max}^2 a_{\min}}{(\mu_1 + \mu_2)^3 \Gamma(4 - \alpha_{\min})} \quad \text{and } L = L_{\min} \quad \text{Eqn. 2-19}$$

where min is the index 1 if  $b = \infty$ , i.e. ON has the dominant sojourn, and is the index 2 if  $b = 0$ , i.e. OFF has the dominant sojourn.

**Homogeneous case:** for large M and T, the aggregate cumulative traffic process behaves statistically as follows:

$$W_M^*(Tt) = TM \frac{\mu_1}{\mu_1 + \mu_2} t + T^H \sqrt{L(T)M} \sigma_{\text{lim}} B_H(t) \quad \text{Eqn. 2-20}$$

where  $H = (3 - \alpha_{\text{lim}}) / 2$  and  $\sigma_{\text{lim}}$  is as above. The theorem depends on convergence in probability as the limits of T and M are taken in the right order, more precisely,

$$\mathfrak{S} \lim_{T \rightarrow \infty} \mathfrak{S} \lim_{M \rightarrow \infty} \left( \frac{W_M^*(Tt) - TM \frac{\mu_1}{\mu_1 + \mu_2} t}{T^H L(T)^{1/2} M^{1/2}} \right) = \sigma_{\text{lim}} B_H(t) \quad \text{Eqn. 2-21}$$

where  $\mathfrak{S} \lim$  represents convergence in the sense of the finite-dimensional distributions.

**Heterogeneous cases:** In the heterogeneous case, it is assumed that there are R distinctive types of r sources being aggregated, i.e.  $r = 1, \dots, R$  types of sources. The characteristics of each source of type r are denoted  $F_j^{(r)}$ ,  $\alpha^{(r)}$ ,  $\sigma^{(r)}$ ,  $L^{(r)}$ . If there is a total of M sources, there is a proportion  $M^{(r)}/M$  of sources that does not converge to 0 as  $M \rightarrow \infty$ . Assuming this proportion is not negligible, and the theorem for the homogeneous case can be modified for heterogeneous scenarios.

For large  $M^{(r)}$ ,  $r = 1, \dots, R$  and large T, the aggregated cumulative packet traffic behaves statistically like:

$$W_M^*(Tt) = T \left( \sum_{r=1}^R M^{(r)} \frac{\mu_1^{(r)}}{\mu_1^{(r)} + \mu_2^{(r)}} \right) t + \sum_{r=1}^R T^{H^{(r)}} \sqrt{L^{(r)}(T)M^{(r)}} \sigma_{\text{lim}}^{(r)} B_{H^{(r)}}(t) \quad \text{Eqn. 2-22}$$

where  $H^{(r)} = (3 - \alpha_{\text{min}}^{(r)})/2$  and the  $B_{H^{(r)}}$  are independent FBMs. The expression implies the limit is the superposition of independent FBM with different Hurst parameters,  $H^{(r)}$ . With respect to fluctuations, the term with the highest  $H^{(r)}$  (or equivalently smallest  $\alpha^{(r)}$ ) ultimately dominates as  $T \rightarrow \infty$ . The contribution of sources with finite variance is simple that of ordinary Brownian motion.

For an ON-OFF model with Power-law distributions of ON and OFF periods, these periods give two separate H values quantifying the strength of self-similarity of individual states. It is worth noting that the overall H is influenced by the stronger component. Typically, the mean OFF time is assigned to be longer than the mean ON time, which gives H of the ‘‘OFF’’

component a higher value. Therefore, OFF periods always have the dominant effect towards the final H value. However, research has shown that in fact the ON period is more important for teletraffic engineering [Mondragón01, Samuel99], particularly with respect to the overall queueing behaviour, and this is another limitation of relying only on H when modelling Power-law traffic. In this research, H is only referred to as a LRD index and is always maintained in the range of  $0.5 < H < 1$ . Its effects towards the overall queueing behaviour are not in the evaluations. The impact of the ON and OFF periods in the queue statistics is the focus.

## 2.6 Generating Pareto distributed numbers

Generating Pareto distributed numbers is complicated by the heavy-tailed nature of the Pareto distribution. In teletraffic modelling, pseudo-random number generators are deployed to provide  $U[0,1]$  random numbers, and these numbers are then converted to a specific distribution. This method works extremely well with light-tailed distributions, such as the exponential, Poisson etc., where the distribution mean and variance can be reproduced precisely. However, due to the limitation of the generators, difficulties arise when using these generators for heavy-tailed distributions. The related problems will be discussed later on in this section, but first it is important to understand the method exploited to create Pareto distributed numbers.

The probability distribution function of the Pareto distribution was modified in a fashion based on the inverse function method to produce an algorithm for generating Pareto distributed random numbers. The formula is given by:

$$X_{\text{Pareto}} = \frac{\Phi}{U^{1/\alpha}} \quad \text{Eqn. 2-23}$$

where U is an uniformly distributed value in the range of  $[0,1]$ . In theory, the Pareto distribution extends to infinity. However, in practice, generating Pareto distributed random numbers using a pseudo-random number generator is always restricted by the limitations of the generator, and specifically the maximum value of the Pareto distribution is limited by the minimum value provided by the random number generator. Therefore, extremely large values will not be generated. Any stream of random variables from a true Pareto distribution of sufficient length will have values that exceed the range generated by a computer. This has a negligible effect on light-tailed distributions, in the sense that the contribution of extremely

large values towards the distribution is minimal. However, when generating Pareto distributed numbers, the main problem associated with the missing tail is that the mean of the distribution will be too small. Consequently, much of the information from the system that related to the mean sojourn times is no longer valid (e.g. the system utilization). It is therefore necessary to be able to calculate the distribution mean of the generated value correctly.

Assuming  $w$  to be the smallest non-zero value that a uniform random generator may give, the uppermost Pareto value will be  $q$  which is derived from Eqn. 2-23

$$q = \frac{\phi}{w^{1/\alpha}} \quad \text{Eqn. 2-24}$$

so the actual mean value of the Pareto distribution that results from the generation process can be obtained from:

$$E(x) = \int_{\phi}^q x \cdot f(x) dx$$

$$E(x) = \frac{\alpha\phi}{\alpha-1} \left[ \frac{1 - \left(\frac{\phi}{q}\right)^{\alpha-1}}{1 - \left(\frac{\phi}{q}\right)^{\alpha}} \right] \quad \text{Eqn. 2-25}$$

substituting Eqn. 2-24 into Eqn. 2-25, the mean of the pseudo-generated Pareto distribution is given by:

$$E(x) = \frac{\alpha\phi}{\alpha-1} \left[ \frac{1 - w^{\frac{\alpha-1}{\alpha}}}{1 - w} \right] \quad \text{Eqn. 2-26}$$

Eqn. 2-26 is important in the teletraffic modelling sense because it allows the actual mean sojourn times of the traffic to be predicted. This is vital in calculating the effect that the simulated traffic sources will have on the simulated network, e.g. even to know the load they are providing is impossible without this knowledge. In this thesis, numerical figures are presented using the rectified value, i.e. the actual value.

## **2.7 Summary**

In this chapter, the concept of self-similarity in network traffic was reviewed. Some of the important properties of self-similar traffic were discussed. An overview on the related modelling methods was provided. The ON-OFF approach adopted in this thesis was described and explained in details with respect to both theoretical and practical sides.

The following chapter will cover materials related to simulation of networks. Simulation techniques and output analyses will be reviewed. Furthermore, validation and verification methods used in this research will also be discussed.

## Chapter 3

# Simulation of Telecommunication Networks

In this chapter, the importance of simulation in the context of network performance evaluations is reviewed. Various simulation techniques and analysis methods are addressed. Above all, the discrete-event simulation with steady-state output analysis, as the choice of this research, is discussed in detail. Furthermore, the difficulties encountered when carrying out this type of simulations, particularly in the area of LRD traffic, are discussed. The role of accelerated simulation techniques and related literature will be reviewed and summarised. The last section discusses the validation and verification techniques used in this research.

### **3.1 Importance of simulation modelling**

Studies are often interested in the operation of many real-life systems. For instance, a good understanding of a network is crucial in dimensioning for QoS. Studying any process scientifically normally requires a set of assumptions, which can be defined by mathematical or logical relationships.

Provided that the individual relationships that describe the model are simple enough, analytical solutions can be obtained to give exact information about the system. Unfortunately, most networks are far too complicated to solve analytically, necessitating the simulation approach. In a simulation, a model is constructed based on the mathematical or logical relationships, and data generated to give an estimate of the true characteristics of the model.

Simulation plays a nontrivial role in the study of communication systems due to network's complexity, large scale and rapid evolution. Any complicated network can always be modelled to an arbitrary level of accuracy, thus providing a good tool for evaluating the behaviour of the object. However, there are drawbacks: firstly, a model's complexity

intensifies as the scale of a system increases, and the related program construction can be gruelling. The good news is that there are now available well-developed software, e.g. OPNET<sup>4</sup>, NS-2<sup>5</sup> etc, and these provide an excellent platform for network modelling with built-in standard components. Secondly, many simulations can be prohibitively time consuming and require excessive computational effort. In particular, simulations requiring steady-state analysis need large amounts of time to run, and with large-scale network models, a great deal of CPU power is always mandatory.

## 3.2 Literature review of simulation techniques

The fundamental concept of a simulation model is a system and the associated system states. A system is defined as a collection of entities that act and interact together towards the accomplishment of some logical end. The state of a system refers to a collection of variables necessary to describe a system at a particular time, relative to the objectives of a study. In this research, a system is referred to as the buffer queue, and the state of the system is the number of packets accumulated in the queue at packet arrival instants. A simulation model can be classified according to three different types [Law00]: i) static vs. dynamic, ii) deterministic vs. stochastic and iii) continuous vs. discrete.

### 3.2.1 Static vs. dynamic simulation model

A static simulation model is a representation of a system at a particular time, or one that may be used to represent a system which time simply plays no role. Model elements do not change during the execution of the model and are therefore constant. These model elements may be hard-coded into the simulation model or read in during the initialization phase of the simulation, but remain the same during the entire execution.

Quite the opposite, a dynamic simulation model represents a system as it evolves over time. Model elements of this type of model have the potential to change their properties or attributes during the model execution. One form of simulation based on dynamic simulation elements is an interactive simulation where users can make modifications during runtime.

---

<sup>4</sup> OPNET<sup>®</sup> is a commercial simulation tool with established libraries developed by Mil3.

<sup>5</sup> NS-2 is the 2<sup>nd</sup> version of NS. It is a jointed project developed by UC Berkeley, USC/ISS, LBL and Xerox PARC for discrete event simulation modelling. The software is available at <http://www.isi.edu/nsnam/ns/>.

### **3.2.2 Deterministic vs. stochastic simulation model**

A simulation model is deterministic if it does not consist of any probabilistic element. The main characteristic of this type is that the inputs of the system determine the output as soon as they are entered, even though the process might take a lot of time to evaluate. A typical example is an analytical model of a network model in which appropriate mathematical or logical expressions are used to construct the model, and corresponding outputs are given once the input parameters are entered.

Models with one or more random input components belong to the stochastic class. As the output produced by a stochastic simulation is random, it is important to note that it can only be treated as an estimate of the true characteristics of the model. Hence, there is a need for statistical output analysis. Network models often belong to the stochastic type as traffic arrival patterns and other aspects of the workload are commonly obtained from a probability distribution.

### **3.2.3 Continuous vs. discrete simulation model**

Continuous simulation models focus on the modelling of a system over time by a representation in which the state variables change continuously with respect to time. Very often, differential equations provide the relationship for the rate of change of a particular state variable with time. For an example, if the variation of the load of a server over a 24-hour period needs to be predicted, then a continuous simulation will be a suitable choice.

On the other hand, discrete simulation models are used to monitor a system that evolves over time by a representation in which the state variables change instantaneously at separate points in time, and these specific points are defined as an event, which is an instantaneous occurrence that may change the state of the system.

### 3.2.4 Simulation model overview

Certain components of a network (buffer queues in particular) were studied in this research, and implementations were done using commercial software OPNET and MATLAB<sup>6</sup>. Simulations described in this thesis are classified as static, stochastic and discrete event-driven.

Static simulations were used mainly due to the focus of this research which was to accurately speed up simulations of network scenarios with Power-law traffic. To achieve this, the overall behaviour (queueing behaviour in this case) must be known thoroughly, so that the scenario can be mimicked in the fashion that the same behaviour can be obtained while providing a speedup. For a particular network scenario, its steady-state<sup>7</sup> behaviour is required which implies that none of the system's parameters should be changed during the execution, and time is only an index to specify the length of the simulation which has no direct impact to the output result (obviously the simulation needs to run sufficient long to achieve steady-state).

The stochastic nature of the simulations originates from the probabilistic property of the buffer behaviour. Unlike many well-established traffic models, such as M/D/1, in which a complete analysis is available for the model, there is not a defined analytical solution for the network scenarios in this research. This leaves the stochastic approach the best option so that a close approximation can be achieved. Traffic sources are defined from a probability distribution, the Pareto distribution in this case, the state of individual traffic switches randomly, and so as the state of the buffer which is determined by the incoming traffic and the transmission rate.

During the execution of a simulation, the state of a buffer is examined. However, the buffer monitoring process is only taken place at specific points, the continuous change of the buffer with respect to time is not of interest. As a result, the discrete event-driven simulation approach was chosen.

In discrete event-driven simulation, a computer model mimics the actual operation of the network, and events can occur any time during the execution of the program. In a network simulation model, an event can be referred to a wide variety of processes (e.g. packet loss),

---

<sup>6</sup> MATLAB® is a product of the MathWorks, Inc.

<sup>7</sup> Steady-state simulation is described in section 3.3.

and in this research, packet arrivals are the fundamental events, which allows traffic related statistics to be provided, e.g. packet delay time probability. Whenever a packet arrives at a buffer, the state of the buffer is studied. Recording and/or analysing output data are considered as subsidiary tasks commonly carried out by a simulation program.

### 3.2.5 Simulation clock

In dynamic discrete-event simulations, it is vital to keep track of the time of the process. The concept of a simulation clock is defined to describe the variable in a simulation model that gives the current value of simulated time. Very often, when general-purpose programming languages (e.g. C) are used to construct a simulation model, the unit of time for the simulation clock is not stated explicitly, and it is assumed to be in the same units as the input parameters. The actual wall-clock time of a simulation and the time of the simulation clock are not necessarily related in the sense that a short simulated time of a system can be extremely time consuming in reality (in wall-clock time).

In addition, a mechanism to advance simulated time from one instant to another is indispensable. Two of the commonly known time advance mechanisms are: next-event time advance and fixed-increment time advance.

#### 3.2.5.1 Next-event time advance mechanism

The next-event time advance mechanism is a widely used approach, and it is commonly found in simulation software. The simulation clock is initialised to zero and the times of occurrence of future events are determined. The simulation clock is advanced to the time of occurrence of the most imminent of these future events, and at this instant the system is updated to encounter an event that has occurred. Figure 3-1 illustrates the concept:  $e_n$  represents the instant where an event takes place and  $t_n$  refers to the time that the simulation clock is advanced as well as the update of the system states.

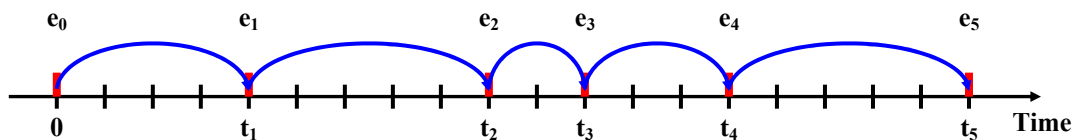


Figure 3-1 Next-event time advance mechanism

This time advancing process forwards the simulation clock from one event time to another continuously until eventually a pre-specified stopping condition is satisfied. Since all state changes occur only at event times for a discrete event simulation model, periods of inactivity are skipped over by jumping the clock from an event time to another event time. The scheme omits all idle periods in a simulation, providing an efficient approach for dynamic simulation modelling. This mechanism is found in OPNET models.

### 3.2.5.2 Fixed-increment time advance

In this time advancing approach, the simulation clock is forwarded in increments of exactly  $dt$  time units for some appropriate choice of  $dt$ . After each update of the clock, a check is made to determine if any events should have occurred during the previous interval of length  $dt$ . If one or more events were scheduled to have occurred during this interval, these events are considered to occur at the end of the interval and the system states are updated accordingly. The concept is demonstrated in Figure 3-2, where  $e_n$  is the actual time of occurrence of an event and the simulation clock is advanced regularly at a fixed interval of  $dt$ . In this research, all MATLAB models were built based on the fixed-increment time concept.

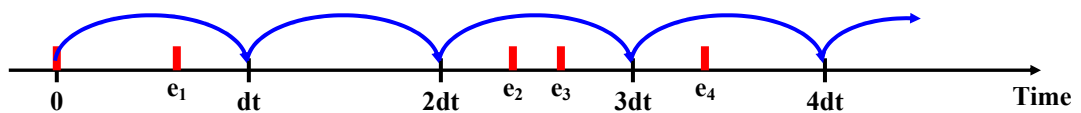


Figure 3-2 Fixed-increment time advance mechanism

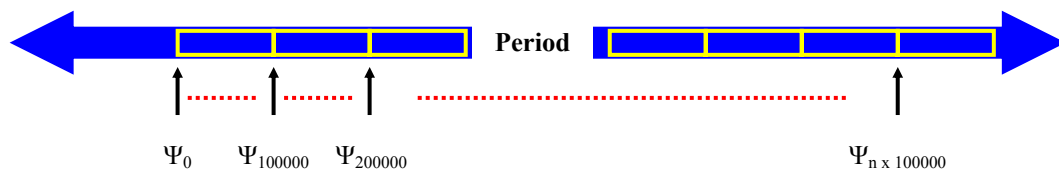
### 3.2.6 Random number generators

Models used in this research involve many stochastic elements, in which random number generators were extensively used to provide random numbers for, e.g. the inter-arrival times and burst lengths of individual traffic sources. These are derived from a specific distribution produced by appropriate mathematical algorithm acting on the  $U[0,1]$  random numbers. The performance of a random number generator has a direct link to the validity of any simulation results, and therefore, the choice of a random number generator is critical.

Random number generated should be IID on the interval [0,1]. These numbers should not exhibit any correlation with the other; otherwise, the randomness of the process no longer holds and the validity of the results are questionable.

Pseudo-random number generators were used to generate randomly distributed numbers. The term “pseudo” is used because these generators are constructed by different algorithms. Ideal generators do not exist as there are always limitations, e.g. the maximum and minimum values that can be generated computationally; nevertheless, there are many good examples available for simulation purposes, e.g. Mersenne Twister [Mastumoto92].

A random number generator can be seen as an extremely long sequence of numbers, and producing a random number is equivalent to retrieving a number from the sequence. Even though such a sequence is sufficiently lengthy in most generators, eventually, the generator will reach the end of the sequence. With large-scale stochastic simulation, in which millions of random events are required, it is critical that the end of the random number stream is recognised before the generator repeats itself within a single simulation, leading to correlated results. The length of the stream of a random number generator is known as the period, and the longer the period, generally the better the generator.



**Figure 3-3 Concept of a random number generator**

Besides the period, reproducibility is also an important property of a random number generator. Each number in the random number sequence can be pointed at by a reference, called the seed value and setting the seed of a generator corresponds to choosing a different starting point to retrieve numbers from. Remembering the initial seed used,  $\Psi_0$ , and initiate the generator with this value again allows the same sequence of random numbers to be produced again. In some simulations where independent streams of random numbers are required, the  $\Psi_i$ 's are generated at any point in the sequence by saving the final  $\Psi_i$  obtained previously and using it as the new seed; as a result, non-overlapping and “independent” sequences of random numbers can be obtained. The concept is illustrated in Figure 3-3. Streams are typically set up by specifying the initial seed. For example, if streams of a length

of 100000 each are required, then set  $\Psi_0$  for the first stream to some value and  $\Psi_{100000}$  for the second one and  $\Psi_{200000}$  for the third one and so on, until all the required sequences have been produced or the end of the period is reached. As a result, non-overlapping streams are adjacent sub-sequences of the single stream of random numbers.

In OPNET simulations, Sun Solaris<sup>TM</sup> is the fundamental operating system, which uses a non-linear additive feedback random number generator with a default state array of 31 long integers to return successive pseudo-numbers in the range from 0 to  $2^{31}-1$ . The period of this random number generator is approximately  $16 \cdot (2^{31}-1)$  [Sun].

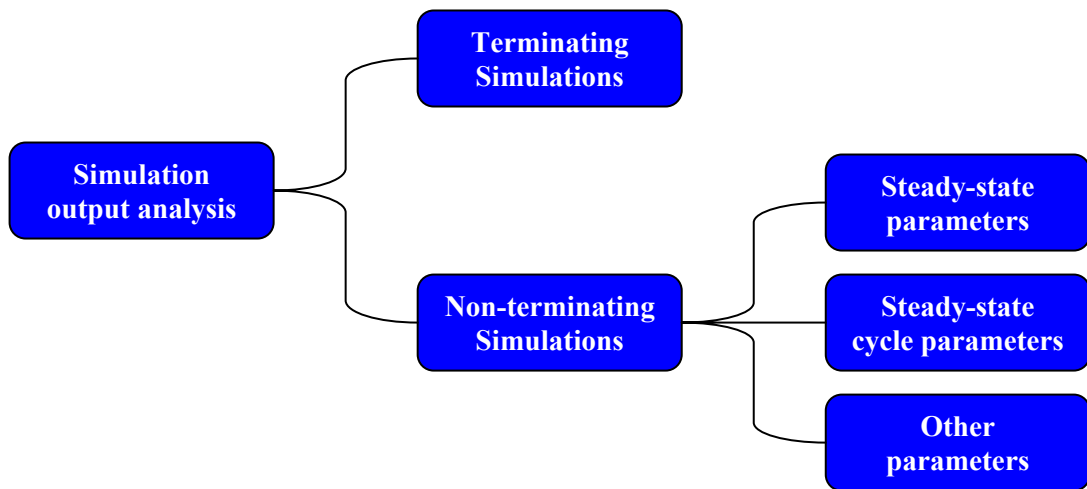
In MATLAB simulations, the built-in random number generator function was used. The generator can generate all the floating-point numbers in the closed interval  $[2^{-53}, 1-2^{-53}]$ . Theoretically, it can generate over  $2^{1492}$  values before repeating itself, which provides an excellent tool for traffic modelling.

Having random number generators with such a long period is extremely important and beneficial. This particular holds true for Power-law traffic modelling as simulations are always required to run for an extremely long period of time to obtain results with an acceptably small statistical error. Poor generators do have significant impact on the simulation results, and the creditability of studies can be questionable [Pawlikowski02].

### **3.3 Simulation with reference to output analysis**

Appropriate output analyses are indispensable in retrieving useful information from simulation results. Very often, a simulation may have been constructed by a piece of complicated code, and not much attention has been put into analysing the output data. To be more precise, a simulation is a computer-based statistical sampling experiment, and to attain useful information, suitable statistical techniques need to be designed for evaluating the experiment once it is completed.

There are two classifications of simulations with regard to output analysis [Law00]: terminating and non-terminating, see Figure 3-4. Depending on the objectives on the simulation study, the two approaches are applied accordingly.



**Figure 3-4** Type of simulations with regard to output analysis

### 3.3.1 Terminating simulations

A terminating simulation requires the presence of a “natural” event  $\zeta$  that specifies the length of the simulation. A number of replications of the simulation are carried out, referred to as different runs, using independent random variables and the identical initialization rule. This implies that comparable random variables output from the different runs are IID. The event  $\zeta$  is often defined as the point where the system is “cleaned out”, or a point in which no further useful information can be retrieved. The ending event is specified prior to the start of any runs, and the occurrence of the event can take place any time during the simulation. Initial conditions have significant influences on the desired measures in a terminating simulation, therefore it is important to have these conditions set with reference to those of a real system.

A terminating simulation can be applied to scenarios such as the performance evaluation of a new TCP protocol stack only during the office (peak) hours, e.g. from 9am to 5pm. In this case, traffic representing peak hour activities is simulated and the simulation is terminated exactly after 8 hours of simulated time.

### 3.3.2 Non-terminating simulations

When designing a new system or changing an existing system, very often behaviour of the system over a long period of time when it is under normal operating conditions is the subject of interest. Therefore, there is no natural event  $\zeta$  to specify the length of a simulation run. A

measure of performance for such a simulation is said to be a steady-state parameter if it is a characteristic of the steady-state distribution of some output stochastic process  $Y_1, Y_2, \dots$ . If the random variable  $Y$  has a steady-state distribution, then one might be interested in estimating the steady-state mean  $E(Y)$  or a probability  $P[Y \leq y]$  for some real numbers of  $y$ .

Interestingly, stochastic processes for most real systems do not have a steady-state distribution as the properties of the system change over time. For instance, characteristics of IP networks change rapidly and unpredictably along many dimensions, it is impossible to draw a conclusion on the overall networks' behaviour [Paxson97b, Floyd01]. However, simulation models, which can be described as an abstraction of reality, may have steady-state distributions. This is due to the fact that characteristics of a model are often treated as invariant. For instance, when modelling a network, it is possible to just focus on a particular time segment, such as the peak hours, by setting the appropriate parameters. As a result, the steady-state behaviour of such a period can be obtained. In the case where new information on the characteristics arrives or new time segment needs to be studied, the model can always be re-simulated for steady-state analysis. For example, if the long-term packet loss rate in a congested router needs to be estimated, this type of simulation approach is the most suitable.

In addition, steady-state cycle parameter analysis can be applied in cases where the stochastic process  $Y_1, Y_2, \dots$  for a non-terminating simulation does not have a steady-state distribution. To perform the evaluation, the time axis is divided into equal lengths, and contiguous time intervals are called cycles. Assume  $Y_i^C$  be the random variable defined on the  $i^{\text{th}}$  cycle, and suppose  $Y_1^C, Y_2^C, \dots$  are comparable. If the process  $Y_1^C, Y_2^C, \dots$  has a steady-state distribution  $F^C$  and that  $Y^C \sim F^C$ . A measure of performance can then be defined as a steady-state cycle parameter if it is a characteristic of  $Y^C$ . An example can be the mean  $E(Y^C)$ . As a result, a steady-state cycle parameter is just a steady-state parameter of the appropriate cycle process  $Y_1^C, Y_2^C, \dots$ .

An example of this will be the performance evaluation of a server of an organization over a 24-hour period. Within a day, there are times that the demand on the server is relatively stable, such as the high demand during office hours and idle during off peak. Therefore, individual steady periods can be analysed using the steady-state cycle parameter approach.

There are models that continue to change over time, and as a result, they do not have a steady-state distribution, and there is no appropriate cycle definition such that the corresponding process  $Y_1^C, Y_2^C, \dots$  has a steady-state distribution. However, there will typically be a fixed amount of data describing how input parameters change over time. This provides, in effect, a terminating event  $\zeta$  for the simulation, and analysis techniques for terminating simulations are applicable. This type of simulations is classified to the “other parameter” category.

Consider one is interested in evaluating the performance of a new scheme for downloading say 100 specific objects from a popular website. In this particular case, it can be assumed that users round the world can access the site at any time of the day, and a regular usage pattern is not available. Therefore, the steady-state and/or the steady-state cycle conditions cannot be fulfilled. The appropriate approach will be the “other parameter” (terminating) method, in which the simulation terminates as soon as the 100 “jobs” have been completed.

### **3.3.3 Choice of approach – steady-state parameter analysis.**

The decision on the steady-state method was based on the fact that the long-term behaviour of a buffer queue is required in this study, so that fair and accurate comparisons can be made between the non-accelerated simulation approach and TA. For a particular network simulation scenario, it is aimed to exploit TA for traffic substitution, and hence achieve speedup. To determine the applicability of TA, it is essential to obtain the steady-state queueing behaviour of the original non-accelerated case, so that the expected response of the system is known. Subsequently, the same scenario with traffic substituted by TA is simulated, and the corresponding steady output is collected. By comparing the two steady-state queueing behaviours (of the non-accelerated model and TA), it evidently provides a validation of TA, and its usability can be justified.

## **3.4 Literature review of accelerated simulation techniques**

Fast simulation techniques can be categorised into three major types: distributed and parallel simulation, hybrid techniques and statistical variance reduction.

### **3.4.1 Distributed and parallel simulation**

The most obvious way to speed up simulations is to increase the computational power so that a larger portion of the task is carried out within the same period of time. This method uses computational parallelism to reduce the experimental period by increasing the number of calculations per unit time. There are two different approaches: Parallel And Distributed Simulation (PADS) [Benveniste95, Heidelberger94, Kalantery94] and Parallel Independent Replicated Simulation (PIRS) [Kurose98, Lin94, Mishra99].

In PADS, sub-models are defined from the original model, and they are then evaluated in parallel on different processors. In a network model, each queue or traffic source, can be treated as an individual parallel sub-models. However, the main difficulty is that it is not always straightforward to locate these sub-models for efficient parallel simulation. This is because maximum acceleration requires these sub-models to be independent. In cases where sub-models are related, at interaction points where synchronization is needed, the whole process is slowed down for the interaction among sub-models.

Stochastic simulations always require a number of independent runs to obtain an estimate within a specific confidence interval (further details on confidence interval is included in 3.5.1). PIRS is a way to obtain more observations in a shorter time by making replications of the simulation process and distributing them on different processors. Despite all simulation overheads, such as data collection from different machines, the speedup provided is  $N$ , which is equal to the number of applied processors.

### **3.4.2 Hybrid techniques**

When using hybrid techniques, the idea is to reduce the number of simulation events by combining the flexibility of simulations with the computational efficiency of analytical models. There are two common techniques: conditional sampling and decomposition [Frost88, Kurose98, Lewis88].

The former is a mathematical approach which aims to reduce the variance by using a known functional relationship between two random variables. The relationship can be conditional arguments, conditional expectation, and substructures of the random variables and system response. An analytical model is applied as a starting point of the evaluation and the remaining unknown elements that have no analytical solution are retrieved by simulations.

The second approach, decomposition, can be seen as an engineering method where the target model is divided into sub-models, either in time or space, e.g. [Sto191]. Some sub-models only require to be evaluated once throughout the entire simulation, via analysis or simulation. By correctly identifying these “one-time” sub-models, simulation acceleration is feasible by omitting unnecessary repeated simulations of sub-models. Furthermore, the application of an analytical solution contributes significantly towards the overall speedup.

A hybrid simulation technique was developed as part of the work done in this research, see Appendix 2. This technique was published in [Ma02c], for the analysis of multiplexed Power-law traffic in broadband networks. The technique allows the prediction for buffer overflow probabilities based on the simulation of the input traffic, measuring particular parameters, and then applying analysis to obtain queueing results that long before a purely simulation based approach could.

### **3.4.3 Statistical variance reduction**

This simulation acceleration technique uses some statistical properties of the model to reduce the variance in the output values. There are two major types under this category: Variance Minimization and Rare Event Provoking.

Variance Minimization involves rearranging the simulation set up without changing the underlying processes to achieve variance reduction. Commonly known methods include Antithetic Samples, Common Random Number and Control Variables [Lewis88, Bratley87]. The main difficulty in applying these techniques is to gain sufficient knowledge of the system behaviour so that appropriate correlated controls can be identified, and to predict their corresponding influences.

In the Rare Event Provoking approach, the occurrence of a particular rare event is increased by changing corresponding statistical properties, e.g. IS [Rubinstein81] and RESTART [Altamirano91].

IS was initially introduced for performing integration by Monte Carlo methods, i.e. by the generation of random numbers [Rubinstein81], and now has been applied to many other areas in recent years, e.g. [Nicola90]. In [Townsend98], the technique has been proposed to simulate rare events such as cell loss probabilities of the order of  $10^{-9}$  in ATM networks. The scheme involves modifying, or biasing, the underlying probabilities in such a way that rare

events occur much more frequently. To correct for this modification, the results are weighted in a way that yields a statistically unbiased estimator. There is a fundamental restriction on the original probability biasing in the sense that the biased probability mass must consist of an occurrence of the rare event of interest. The key to applying IS is to determine which parameter(s) of the system to bias and the amount of biasing.

M. and J. Villén-Altamirano [Altamirano91] presented the RESTART technique which conditions rare events on more common network states. The probability of rare events, such as packet loss probability, can be predicted based on a less rare event (e.g. a threshold level in the queue). System states leading to the target rare event are saved and re-entered a number of times as the starting point for sub-runs. This leads to a great increase in the occurrence of the rare event and hence speedup is achieved. The rare event estimate is obtained by combining results from sub-runs with measurements for reaching the threshold. The overhead required to restore the system for retrials does affect the efficiency, but despite this, it has been reported that RESTART can reduce simulation run times by several orders of magnitude.

In this thesis, RESTART is in fact exploited in conjunction with TA. A comprehensive explanation of RESTART is provided in Chapter 6, where the co-operation of the two fast simulation techniques will also be discussed.

### **3.5 Validation and verification**

In order to merit TA, appropriate statistical methods are essentially required to validate and verify simulation results. In simple terms, the queueing results of TA should be more or less same as the original non-accelerated scenario. However, saying “more or less” is inaccurate and unscientific, certainly cannot justify the functionality of TA. A suitable way is needed for the comparisons between the original non-accelerated model and the equivalent TA, so that the accuracy provided by TA can be clearly illustrated and any differences between the two models can be precisely quantified. One way to achieve this is by using confidence intervals [Law00]. This technique is also useful when comparing steady-state results from individual runs of the same simulation. The way to determine the “steadiness” is by comparing these results obtained from individual runs, and checks the differences among them.

Two other techniques were used in examining results are Power-law bestfit and exponentially binning [Adamic]. Further details are followed later on in this section.

### 3.5.1 Confidence intervals

In this thesis, confidence intervals were applied in the following two result analysis procedures. Firstly, results obtained from the original non-accelerated models and TA were compared and the confidence intervals were used as an indicator to reflect the closeness between the two sets of results. Secondly, the confidence intervals were applied in the verification of the steady-state distribution among different runs of individual simulations. In steady-state output analysis, individual sets of simulation, for both original and TA, were replicated and run with different seeds. If every set of results is obtained during the system's steady status, the obtained data from various runs are expected to follow the same distribution, although high variance is observed in some LRD cases [Crovella97]. To determine the precision of individual output distributions, the confidence intervals were applied.

A confidence interval is defined by a parameter  $\eta$ , which is the percentage of the desired confidence interval. To find a  $100(1 - \eta)\%$  confidence interval, assume  $X_1, X_2, \dots, X_n$  are IID random variables with finite mean  $\mu$  and finite variance  $\sigma^2$  ( $\sigma^2 > 0$ ). The sample mean and variance<sup>8</sup> are given by [Law02, Milton95]:

$$\bar{X}(n) = \frac{\sum_{i=1}^n X_i}{n} \quad \text{Eqn. 3-1}$$

$$S^2(n) = \frac{\sum_{i=1}^n [X_i - \bar{X}(n)]^2}{n} \quad \text{Eqn. 3-2}$$

Defining  $Z_n$  to be a random variable in which

$$Z_n = \frac{\bar{X}(n) - \mu}{\sqrt{\sigma^2 / n}} \quad \text{Eqn. 3-3}$$

---

<sup>8</sup> The sample mean and variance corresponds to the mean and variance obtained via different runs of same simulation, with the same input parameter settings.

The distribution function of  $Z_n$  is denoted as  $F_n(z)$  for a sample size of  $n$ , where  $F_n(z) = P[Z_n \leq z]$ . In accordance with the central limit theorem,  $F_n(z) \rightarrow \Phi(z)$  as  $n \rightarrow \infty$ , where  $\Phi(z)$  is the distribution function of a normal random variable,

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-y^2/2} dy \quad \text{for } -\infty < z < \infty \quad \text{Eqn. 3-4}$$

with  $\mu = 0$  and  $\sigma^2 = 1$ .

According to the theorem, if  $n$  is sufficiently large, the random variable  $Z_n$  will be approximately distributed as a standard normal random variable, regardless the underlying distribution of  $X_i$ 's. Furthermore, for a large number of  $n$ , the sample mean  $\bar{X}(n)$  is roughly distributed as a normal random variable with mean  $\mu$  and variance  $\sigma^2/2$ .

The difficult part is that  $\sigma^2$  is often unknown, however, as the sample variance  $S^2(n)$  converges to  $\sigma^2$  as  $n$  becomes large, it is feasible to replace  $\sigma^2$  by  $S^2(n)$ , giving  $Z_n = [\bar{X}(n) - \mu] / \sqrt{S^2(n)/n}$ , which is approximately distributed as a standard normal random variable. Hence, for large  $n$ ,  $Z_n$  follows

$$P\left(-z_{1-\eta/2} \leq \frac{\bar{X}(n) - \mu}{\sqrt{S^2(n)/n}} \leq z_{1-\eta/2}\right) = P\left[\bar{X}(n) - z_{1-\eta/2} \sqrt{\frac{S^2(n)}{n}} \leq \mu \leq \bar{X}(n) + z_{1-\eta/2} \sqrt{\frac{S^2(n)}{n}}\right]$$

$$\approx 1 - \eta$$

Therefore, for a given data  $X_1, X_2, \dots, X_n$ , provided that  $n$  is sufficiently large, an approximate  $100(1 - \eta)\%$  confidence interval for  $\mu$  is

$$\bar{X}(n) \pm z_{1-\eta/2} \sqrt{\frac{S^2(n)}{n}} \quad \text{Eqn. 3-5}$$

where  $\mu$  lies between the lower confidence-interval endpoint  $l(n, \eta) = \bar{X}(n) - z_{1-\eta/2} \sqrt{S^2(n)/n}$  and the upper confidence-interval endpoint  $u(n, \eta) = \bar{X}(n) + z_{1-\eta/2} \sqrt{S^2(n)/n}$ .

In this thesis, a 95% confidence interval was generally used for the comparison between the non-accelerated model and TA. In comparisons among different runs of the same simulation,

a steady-state distribution is expected from individual cases, and the confidence interval is expected to be more precise.

### **3.5.2 Power-law bestfit and exponentially wider bins**

Power-law bestfit was another procedure taken in this research for ensuring the accuracy of the simulation results. All simulations were run for several numbers of times and those which are within the 95% confidence interval were then averaged to achieve maximum accuracy. As results were obtained from stochastic processes, small variations are bounded to be found in the results. The bestfit concept was therefore introduced to describe the overall queueing behaviour. For Power-law traffic, unquestionably, a Power-law equation would be the best in characterising the corresponding queue decays. A Power-law bestfit is obtained by calculating the least squares fit through points by using  $y = c \cdot x^b$ , details refer to section 2.3.3.

Queueing behaviour obtained from traditional Markovian traffic models is not as bursty as Power-law traffic. This can be explained by the SRD property of the distribution used to produce the traffic, and so the probability of rare event (long traffic bursts) is negligible. Once a simulation is run for a sufficiently long period of time, meaningful results are available for collection. The problem with studying Power-law traffic lies in the heavy-tail property of their distributions. Those extremely long traffic bursts with nontrivial probability produced by the Pareto distribution contribute a Power-law pile-up in the buffer. When plotting the queueing results on a log-log graph, it is expected the queue to decay linearly. However, applying a Power-law bestfit directly to the collected data leads to a gradient that is too steep due to the enormous number of statistics at the end of the queue tail, which contribute unevenly towards the calculation of the bestfit, see Figure 3-5. To rectify this, output data are binned into exponentially wider bins [Adamic], so that the “noisy” tail of the queue will not contribute disproportionately towards the overall queueing trend. Results will appear evenly spaced on a log-log scale. Subsequently, a clear Power-law relationship can be obtained using the binned data, see Figure 3-6, and which the actual queueing behaviour is truly reflected. This technique was also applied when examining the distribution of both ON and OFF sojourn times.

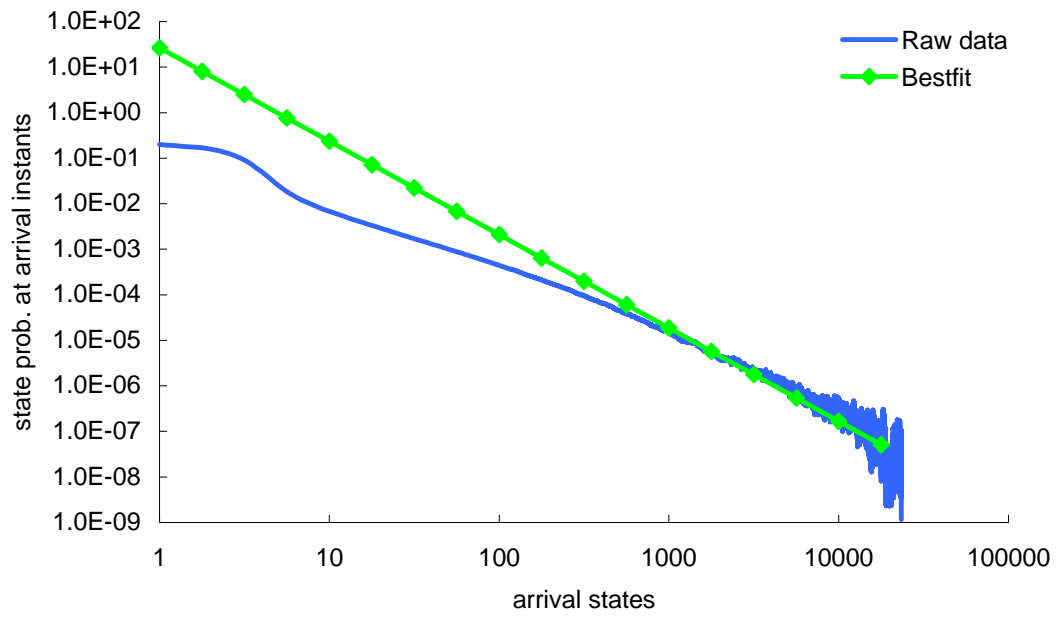


Figure 3-5 Power-law best fit with respect to raw data

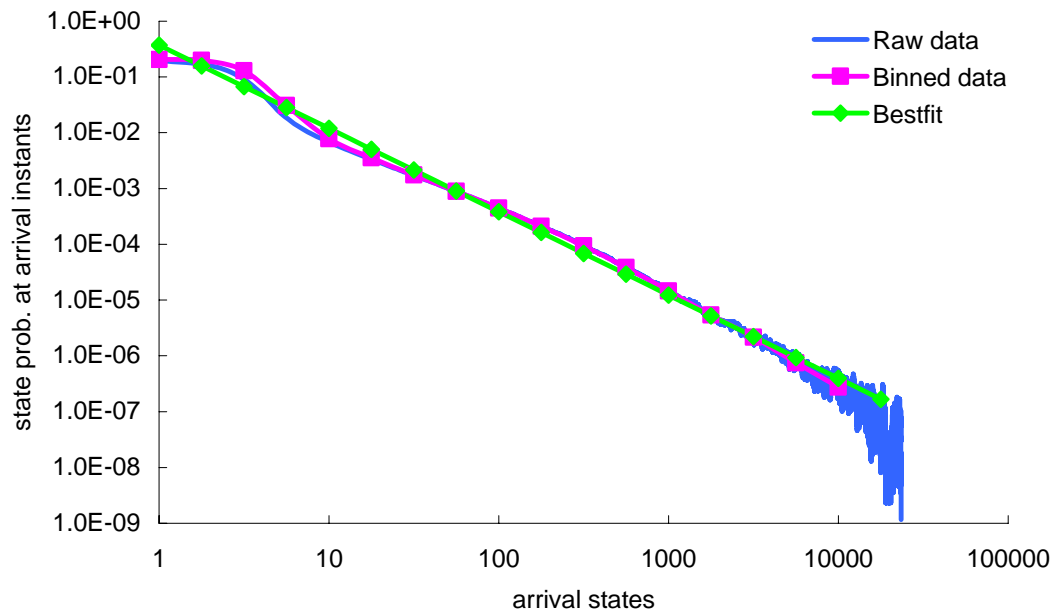


Figure 3-6 Power-law bestfit with respect to binned data

### **3.6 Summary**

The role of simulation modelling was discussed in this chapter. Modelling concepts and output analysis techniques were provided. The approach used in this research was classified and explanations were provided. A literature overview was presented on the topic of acceleration simulation techniques. Finally, the validation and verification methods used in this thesis were examined.

In the next chapter, TA, the core of this thesis will be described. The concept of TA will be presented, as well as the parameterization involved. Simulation results will be provided to validate the technique. The application of TA to different traffic settings will also be demonstrated.

## Chapter 4

# Introduction to Traffic Aggregation (TA)

The principle of TA is presented in this chapter. A similar traffic aggregation technique was first published in [Schormans00], and in that paper, Markovian traffic was the focus. The concept was to reduce an arbitrary number of independent sources to one single aggregate process, and effectively reduce the overall state-space from  $2^N$  states to just 2: an ON state and an OFF state. In general there are two advantages in using the aggregate process: 1) acceleration – only one single traffic source is needed instead of  $N$ , and the simulation time required by a two-state process is significantly shorter than that of a  $2^N$  process for the same stability, and 2) simplification - the overall state-space reduction in the sense that the total number of system states is decreased to only two, which may allow simple analytical solutions to be applied.

This research now targets Power-law traffic. The previous Markovian aggregation technique has been redeveloped and is applied to Power-law traffic. An algorithm has been developed for the parameterization of TA, and details are covered in this chapter. In addition, comparisons between the results of the original non-accelerated and TA models are also included and evaluated. The accuracy provided by TA is clearly shown, as well as its applicability to many scenarios. Models in this chapter were built in Matlab.

### 4.1 Traffic Aggregation (TA)

Consider a traffic model that comprises  $N$  homogeneous ON-OFF sources with Pareto distributed sojourn times. Individual traffic sources transmit packets at a rate of  $R$  packets/timeslot during ON periods, where a timeslot is the basic time unit. During OFF periods, these sources are inactive and no packets are sent. The mean ON and OFF sojourn durations are described by  $D_{ON}$  and  $D_{OFF}$  respectively, and the queueing system has an overall service rate of  $C$  packets/timeslot. These traffic sources switch between ON and OFF

states randomly, i.e. there is no synchronization on the state switching among these sources. A resultant traffic pattern can be obtained which is similar to the pattern shown in Figure 4-1(a). The First-In-First-Out (FIFO) queuing mechanism is used and the utilization of the system,  $\rho$ , is given by:

$$\rho = \frac{D_{ON}}{D_{ON} + D_{OFF}} \cdot N \cdot R \cdot \frac{1}{C} \quad \text{Eqn. 4-1}$$

This is referred to as the “original” scenario in this thesis, in which no acceleration technique has been applied.

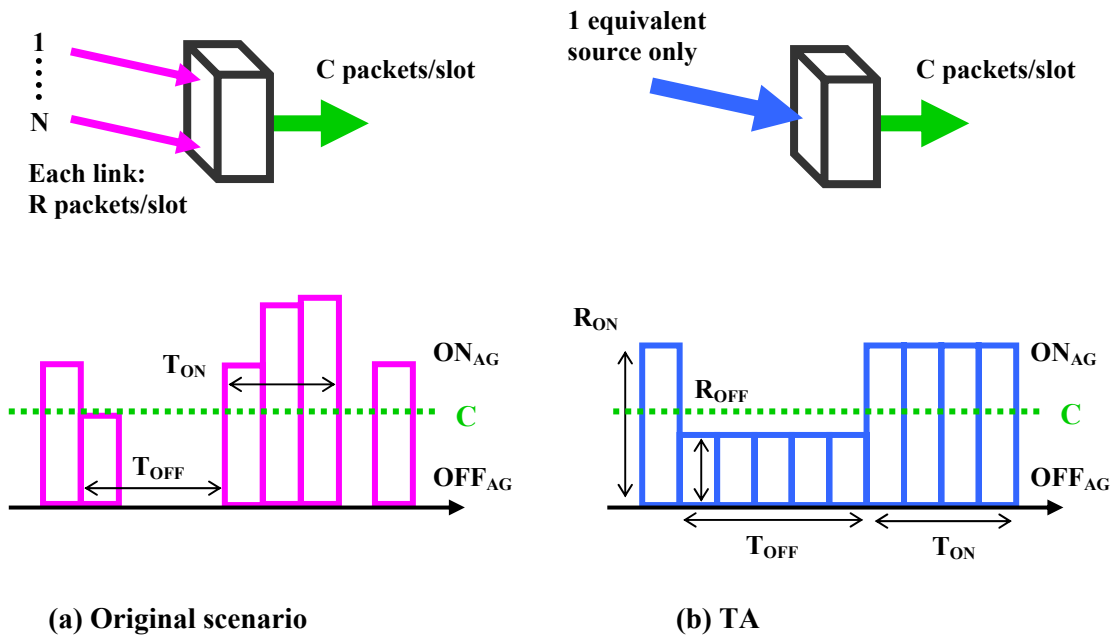
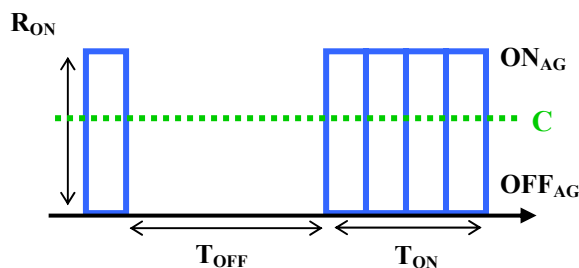


Figure 4-1 Concept of TA

In TA, acceleration and simplification are achieved by introducing a single equivalent ON-OFF process to replace the multiplexing scenario. The two-state process is defined by reference to the resultant traffic pattern, and a number of new definitions are essentially specified during the derivation.  $ON_{AG}$  and  $OFF_{AG}$  correspond to the ON and OFF states in the aggregated process, and they are derived with reference to the service rate,  $C$ . An  $ON_{AG}$  period occurs when the total number of incoming packets is greater than the service rate, i.e.  $NR > C$ , and the mean duration of this is denoted by  $T_{ON}$ . An  $OFF_{AG}$  period is obtained when

there is a contiguous number of timeslots in which the number of packet arrivals is less than or equal to the service rate, and the mean of this duration is referred to as  $T_{OFF}$ . As the rate of packet arrivals during an OFF period is always lower than the service rate, therefore, there is no piling up and effectively, there is spare capacity to release accumulated packets.  $NR=C$  is considered as OFF is due to the fact that there is no net effect on the buffer size, in other words there is no further accumulation of packets. The concept is demonstrated in Figure 4-1(b).

Previously, TA (the early version for Power-law traffic) involved evaluating both  $R_{ON}$  and  $R_{OFF}$  which denote the arrival rate during the ON and OFF states respectively. However, experimental results show that the OFF periods have negligible effects towards to overall queueing behaviour of the system for Power-law traffic (further detail is included in the parameterization section). Based on this observation,  $R_{OFF}$  has been set to zero in this research, and thus TA becomes a single Pareto ON-OFF traffic source with three parameters only:  $T_{ON}$ ,  $T_{OFF}$  and  $R_{ON}$  (i.e. an ordinary ON-OFF source), see Figure 4-2. Results in this thesis are all based on the modified scheme.



**Figure 4-2 Modified TA**

Setting  $R_{OFF} = 0$  has several major advantages: Firstly, further acceleration is offered in the sense that there are even fewer events which greatly decrease the overhead of a simulation. Secondly, the modelling process is further simplified as the number of parameters is reduced. Thirdly, with  $R_{OFF} > 0$ , the mean OFF time of TA,  $T_{OFF}$ , can be extremely large, which results in increasing the location parameter<sup>9</sup>,  $\phi$ , of the distribution so that the corresponding mean can be achieved. However, when high mean values are encountered, it is extremely

<sup>9</sup> Location parameter,  $\phi$ , denotes the minimum value of the Pareto distribution, see 2.3.2 for details.

difficult to predict the mean value accurately from a simulation as the variance between individual elements of the distribution can be extremely large. These considerations did not arise in the previous application to Markovian traffic. Furthermore, as applying TA to Power-law traffic requires the parameterization scheme to be re-established, with the absence of  $R_{OFF}$ , the process is certainly simplified.

## 4.2 Parameterization of TA

Applying TA to any network scenario requires knowing the value of  $T_{ON}$ ,  $T_{OFF}$  and  $R_{ON}$ . A parameterization scheme has been developed to derive precise values for these parameters provided that the settings of the original non-accelerated model are known. Individual parameters were formulated via a simulation approach. This approach was considered as a more appropriate choice than analysis because there are not many analytical studies on Power-law traffic available, to apply analysis to characterise a new scheme for the modelling of this new type of traffic has no guarantee on the success of a final algorithm. Moreover, with limited resources to refer to, the whole analytical process can take an unrealistic period of time, which can be unaffordable as far as this research is concerned.

Moreover, studies in the area of Power-law traffic mostly concentrate on the strength of self-similarity which is measured by the  $H$  parameter e.g. [Fonseca00, Willinger98]. Unfortunately, the value of  $H$  has no direct relationship with the actual queueing behaviour in an ON-OFF traffic model [Mondragón01]. This is due to the value of  $H$  is substantially affected by the stronger component of the two, and in most cases (if not all), the OFF periods have the dominant effect as they always have a higher mean. Contrarily, the queueing behaviour is always determined by the ON sojourn times. As a result, carrying out analysis is elusive. In addition, there are other issues that need to be taken into account, such as multiplexing gain when multiplexing traffic sources [Cao01]. Consequently, the simulation approach has become a more preferable option as a look-up table of some sort can often be produced by referring to the obtained results, if not the actual algorithm.

The parameterization scheme is divided into three parts: firstly, the most important part of the parameterization scheme is the derivation of  $T_{ON}$ . A formula has been defined for parameterizing  $T_{ON}$  based on simulation results. Data collected from these simulations were analysed, and a relationship was drawn between the original non-accelerated scenario and TA. Secondly, the evaluation of  $R_{ON}$  is to be carried out. This is done by using the binomial

distribution for calculating the rate of arrival during the system's ON state. Lastly,  $T_{OFF}$  is to be defined via the utilization formula, Eqn. 4-1.

#### 4.2.1 Deriving $T_{ON}$

The derivation of  $T_{ON}$  is the most crucial part of the parameterization process for two reasons: firstly,  $T_{ON}$  has the most influential effect towards the overall queueing behaviour, which directly affects the accuracy of TA. And secondly, the value of  $T_{OFF}$  is determined from  $T_{ON}$ . As  $R_{ON}$  is evaluated independently, once  $T_{ON}$  is obtained, the rest of the parameterization can be accomplished. Therefore,  $T_{ON}$  is considered as the prime parameter of TA.

Recalling the parameters of the original scenario,  $D_{ON}$ ,  $D_{OFF}$ ,  $N$ ,  $R$  and  $C$ . The way to derive  $T_{ON}$  was based on the fact that these individual parameters have separate influences on the value of  $T_{ON}$ , and simulation results have verified the existence of these relationships. For instance, increasing  $D_{ON}$  leads to a higher  $T_{ON}$ , providing the rest of the settings remain the same. Similarly, increasing  $N$  results in a raise in the value of  $T_{ON}$ . On the contrary, increasing  $D_{OFF}$  or  $C$  causes a decrease in  $T_{ON}$ . Logically,  $T_{ON}$  can be defined provided that the resultant effect of these parameters is established.

##### 4.2.1.1 Relationship between $D_{ON}$ and $T_{ON}$

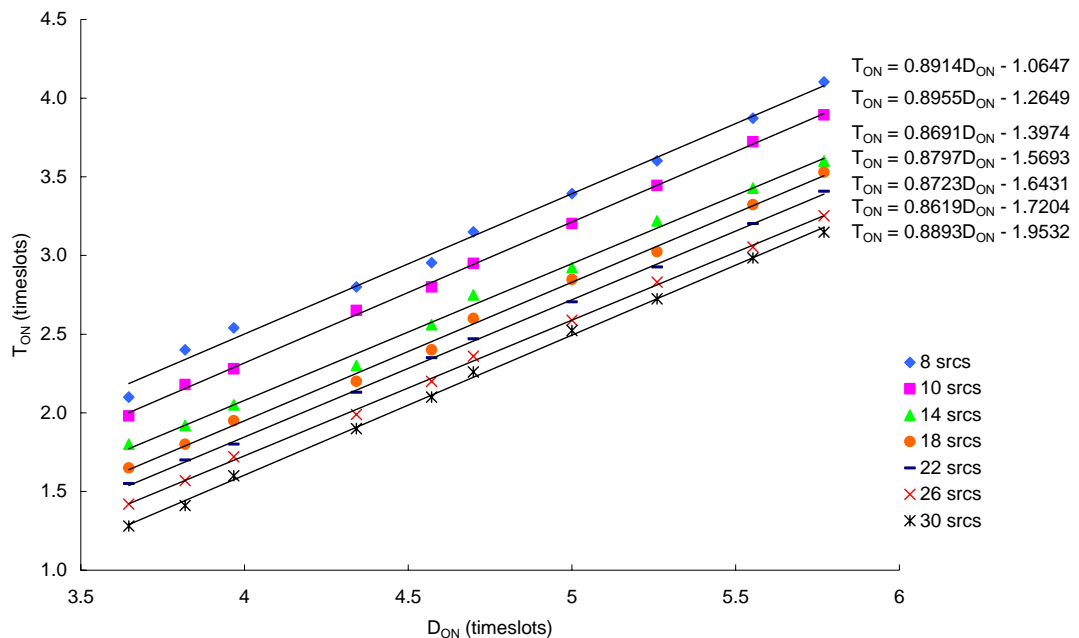
Studies were carried out to monitor the effect of  $D_{ON}$ , which involves varying the parameter over a range of values, while having a fixed setting for the rest of the configuration. A list of simulations was carried out, and parameters for these simulations are listed in Table 4-1.

<b>N</b>	8	10	14	18	22	26	30
<b>C</b>	4	5	7	9	11	13	15
<b>R</b>	1						
<b><math>D_{OFF}</math></b>	10						
<b><math>D_{ON}</math></b>	Varies from 3.6 to 5.8						

**Table 4-1** Parameters for monitoring the effect  $D_{ON}$

These  $D_{ON}$  values were chosen for several reasons, mainly as these values can be accurately generated. Values from the Pareto distribution are highly restricted by the pseudo-random number generator (see Chapter 3), and to be able to produce a specific distribution with a correct mean, it is important to keep the targeted mean value relatively small. Nevertheless, these values were chosen given the fact that the LRD requirements were fulfilled. Variance plot were carried out to ensure that the overall H parameters of these traffic scenarios were in the range of 1 and 2. See Appendix 3 for the details of variance plot method.

Appropriate values of  $T_{ON}$  were then obtained for each case, and the relationship between the two parameters investigated.  $D_{ON}$  and corresponding  $T_{ON}$  were plotted and presented in Figure 4-3. It can be seen that for a given combination of N, R and C,  $T_{ON}$  increases linearly with  $D_{ON}$ , giving a relationship in the form  $y = mx + c$  with x and y corresponds to  $D_{ON}$  and  $T_{ON}$  respectively. The gradient, m, is extremely important as it directly links the two parameters together. Referring to these linear regressions, m is technically insensitive to the changes of N and C, which essentially concludes that it describes the “sole” effect of  $D_{ON}$  on  $T_{ON}$ . The remaining c is the intercept of the linear equation, whose value is determined by N, R and C, seen as the offset of the queue.



**Figure 4-3 Relationship between  $D_{ON}$  and  $T_{ON}$**

The final value of  $m$  which is then used in part of the  $T_{ON}$  formula was derived from the linear equations obtained from the simulation results. There are seven equations and the gradient of these equations were averaged, and the following equation is defined to associate  $D_{ON}$  and  $T_{ON}$ .

$$T_{ON} = m \cdot D_{ON} + \kappa_1 \quad \text{Eqn. 4-2}$$

where  $m$  is the averaged gradient value which has a value of 0.87969 and standard deviation of 0.01157, and therefore, the confidence is 98% which implies the accuracy is extremely high.  $\kappa_1$  is a constant for a specific  $N$  and  $C$  combination, and its value varies according to the value of  $N$  and  $C$ .

#### 4.2.1.2 Relationship between $D_{OFF}$ and $T_{ON}$

TA was developed based on the assumption that  $D_{OFF}$  is significantly longer than  $D_{ON}$ , and this is a phenomenon observed in network traffic. In cases where  $D_{OFF}$  satisfies this condition, changing the value of  $D_{OFF}$  causes only a trivial effect in the overall queueing behaviour of the buffer. This benefits the parameterization process of TA in the sense that the effect brought by  $D_{OFF}$  can be neglected provided that this assumed condition is met, which effectively simplifies the process.

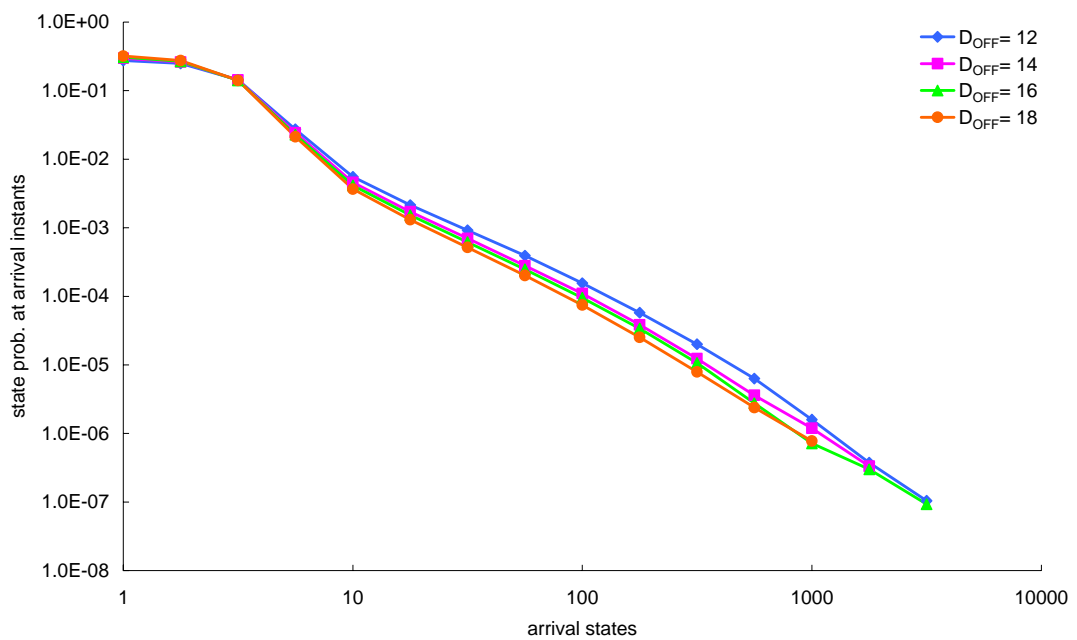


Figure 4-4 Effect of  $D_{OFF}$  on the overall queuing behaviour

Figure 4-4 illustrates the way that queueing behaviour is altered by the variation of  $D_{OFF}$ . The rest of the parameters are as follows:  $D_{ON} = 3.9$ ,  $N = 10$ ,  $R = 1$  and  $C = 5$ . This configuration fulfils the above assumption, and queueing results show minimal changes even when  $D_{OFF}$  was amplified considerably. Subsequently, no alternation is required for the value  $T_{ON}$  of the equivalent TA model for these cases. On the contrary, if  $D_{ON}$  were raised by the same amount, the queueing behaviour would have been drastically transformed.

Therefore,  $D_{OFF}$  has been omitted in the characterization of  $T_{ON}$  as it has trivial influence towards the overall queueing results, and hence the value of  $T_{ON}$ .

#### 4.2.1.3 Relationship between $N$ and $T_{ON}$

Increasing  $N$  obviously has the effect of increasing the burstiness of the traffic, and hence necessitating a higher  $T_{ON}$ . Figure 4-5 corresponds to the results of simulations of Table 4-2.  $N$  covers a range of values which reflects the number of sources from small to large.  $C$  was varied to show that the effect on  $T_{ON}$  by  $N$  is independent to the configuration of the rest of the parameters.

<b>N</b>	In the range of 7 to 28						
<b>C</b>	5	6	7	8	9	10	11
<b>R</b>	1						
<b>D<sub>OFF</sub></b>	10						
<b>D<sub>ON</sub></b>	3.9						

**Table 4-2 Parameters for monitoring  $N$**

$N$  was plotted against  $T_{ON}$ , and this demonstrates that  $T_{ON}$  increases exponentially with  $N$ . The relationship can then be generalised in the form of

$$T_{ON} = \kappa_2 \cdot e^{\varepsilon_1 N} \quad \text{Eqn. 4-3}$$

where  $\kappa_2$  is a constant for a particular setting of  $D_{ON}$  and  $C$ , and it varies according to the combination of the two parameters.  $\varepsilon_1$  is an index that describes the exponential relationship between  $N$  and  $T_{ON}$ . With reference to Figure 4-5, the scaling factor,  $\varepsilon_1$ , in the power term is highly close to each other regardless the variation in  $C$ . By averaging the value of  $\varepsilon_1$  from

individual cases presented in Figure 4-5, the final value of  $\epsilon_1$  concluded from those results is 0.13667, and the standard deviation is 0.00333. Again, the accuracy is highly respectable.

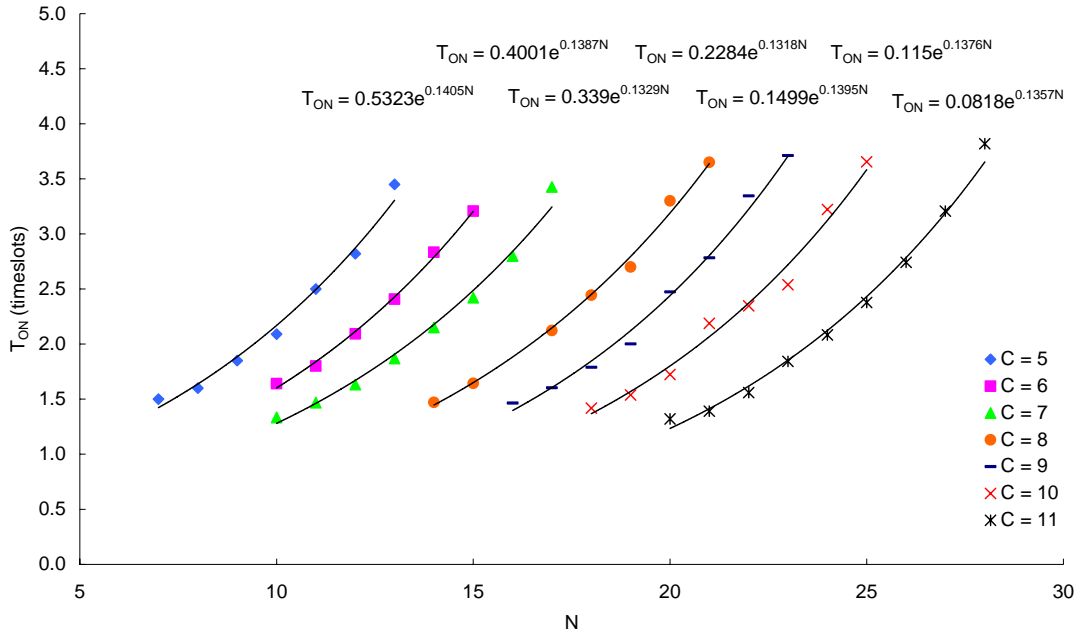


Figure 4-5 Relationship between N and T<sub>ON</sub>

#### 4.2.1.4 Relationship between R and T<sub>ON</sub>

The effect of R is not considered as part of the parameterization process for T<sub>ON</sub> due to the fact the arrival rate of TA, R<sub>ON</sub>, is calculated separately via the binomial distribution in terms of R. Therefore, any changes in R will be reflected in R<sub>ON</sub> during the calculation. See 4.2.2 for the calculation method of R<sub>ON</sub>.

#### 4.2.1.5 Relationship between C and T<sub>ON</sub>

Changing C directly affects the queueing behaviour in the buffer, and therefore, it is expected to have impact on T<sub>ON</sub>. If C is reduced solely when the other parameters remained unchanged, T<sub>ON</sub> is required to increase so that the extra delays in the queue can be reflected. In order to draw this relationship precisely, again, simulations were carried out.

Similar to previous investigations for  $D_{ON}$  and  $N$ , parameters were set in the sense that  $C$  was adjusted while the rest of the configuration kept unchanged. Table 4-3 consists of the settings of individual simulations, corresponding results are illustrated in Figure 4-6.

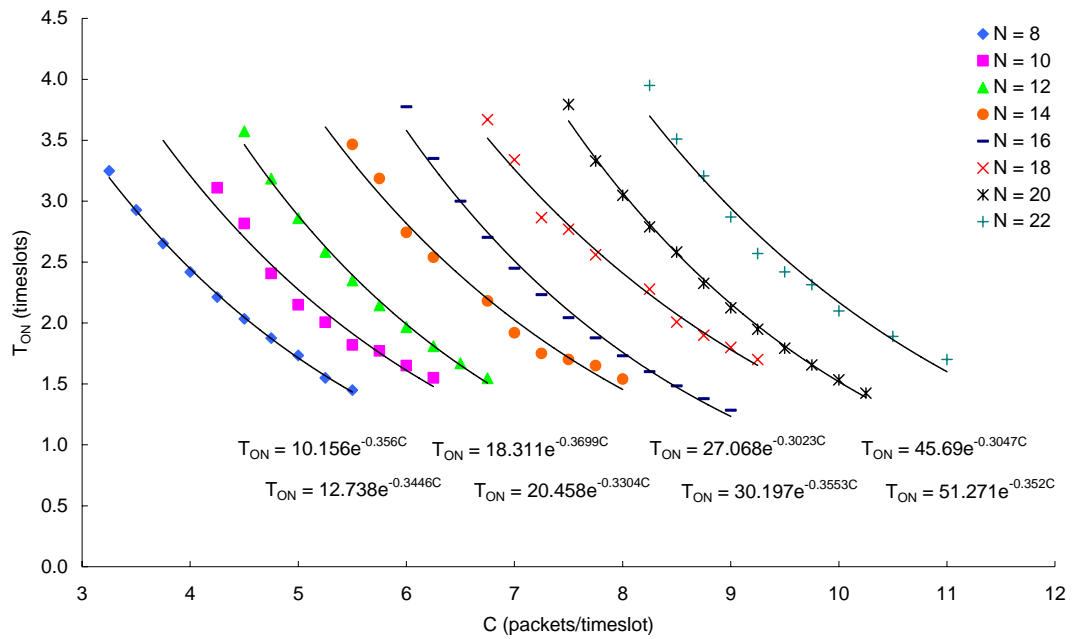
Observations show that  $C$  and  $T_{ON}$  are correlated exponentially, and this relationship holds true regardless the rest of the parameters. A general formula can be used to describe the way that  $T_{ON}$  varies as  $C$  is modified.

$$T_{ON} = \kappa_3 \cdot e^{-\varepsilon_2 C} \quad \text{Eqn. 4-4}$$

where  $\kappa_3$  is a constant for individual cases with the same  $D_{ON}$  and  $N$ , and as these two parameters are modified,  $\kappa_3$  changes accordingly.  $\varepsilon_2$  is the scaling factor in the power term that characterises the exponential distribution, and it is illustrated that the value of  $\varepsilon_2$  is technically unaltered by the change of other attributes. This concludes that  $\varepsilon_2$  can be referred to as an index that expresses the influence of  $C$  on  $T_{ON}$ . With respect to the results presented in Figure 4-6,  $\varepsilon_2$  was calculated by taking the mean of all  $\varepsilon_2$ 's, and has a value of 0.3394 and a standard deviation of 0.0248. The confidence is 95% indicating a tremendously high accuracy.

<b>N</b>	8	10	12	14	16	18	20	22
<b>C</b>	In the range of 3.25 to 11.5							
<b>R</b>	1							
<b>D<sub>OFF</sub></b>	10							
<b>D<sub>ON</sub></b>	3.9							

**Table 4-3**      **Parameters for monitoring C**



**Figure 4-6 Relationship between C and  $T_{ON}$**

#### 4.2.1.6 Formula for $T_{ON}$

Relationships between individual parameters and  $T_{ON}$  have been discovered (Eqn. 4-2, Eqn. 4-3, and Eqn. 4-4), the next step is to solve these equations so that it is feasible to obtain a straightforward formula in which  $T_{ON}$  can be evaluated by substituting for the associated parameters of the original scenario. This is equivalent to attaining the resultant effect of  $D_{ON}$ ,  $N$  and  $C$ .

First of all, the two exponential equations (Eqn. 4-3 and Eqn. 4-4) were combined into one by multiplying them, giving

$$T_{ON}^2 = \kappa_2 \cdot e^{\varepsilon_1 N} \cdot \kappa_3 \cdot e^{-\varepsilon_2 C}$$

$\kappa_2$  and  $\kappa_3$  are constant for a particular setting of  $N$  and  $C$ , it can then be assumed that the multiplication of two constants gives another constant. Therefore,

$$T_{ON} = \kappa \cdot e^{0.5(\varepsilon_1 N - \varepsilon_2 C)} \tag{Eqn. 4-5}$$

where  $\kappa$  is the product of  $\kappa_2$  and  $\kappa_3$  and yet another constant.

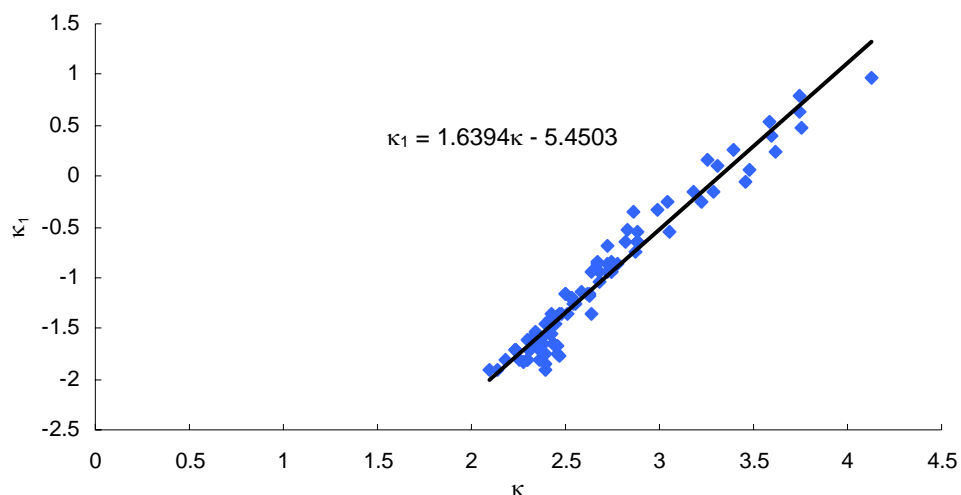
At this stage, there are two independent equations, with one common unknown  $T_{ON}$ . To solve these two equations, the simultaneous approach was used which requires a link between Eqn. 4-2 and Eqn. 4-5. This was done by relating  $\kappa_1$  with  $\kappa$ .

For a given combination of  $D_{ON}$ ,  $N$  and  $C$ , the same  $T_{ON}$  should be obtained regardless whether Eqn. 4-2 or Eqn. 4-5 is applied. With the aid of existing simulation results, it is possible to locate  $T_{ON}$  for a limited number of cases, and by substituting  $T_{ON}$  and corresponding parameters into the appropriate equations,  $\kappa_1$  and  $\kappa$  can subsequently be evaluated.

Values  $\kappa_1$  and  $\kappa$  were collected from all studies, and all  $\kappa_1$ 's were plotted against corresponding  $\kappa$ 's, illustrated in Figure 4-7. A linear relationship was observed and can be formulated into:

$$\kappa_1 = 1.6394\kappa - 5.4503 \quad \text{Eqn. 4-6}$$

Eqn. 4-6 plays a non-trivial role in the formulation of  $T_{ON}$  as it provides a connection between the two independent equations of  $T_{ON}$ , fulfilling the simultaneous equation solving requirement.



**Figure 4-7 Relationship between  $\kappa$  and  $\kappa_1$**

At this point, there are three simultaneous equations, and they can be solved by first of all substituting Eqn. 4-2 into Eqn. 4-5, giving,

$$\kappa \cdot e^{0.5(\epsilon_1 N - \epsilon_2 C)} = m \cdot D_{ON} + \kappa_1 \quad \text{Eqn. 4-7}$$

Then, substituting  $\kappa_1$  by putting Eqn. 4-6 into Eqn. 4-7,

$$\kappa \cdot e^{0.5(\epsilon_1 N - \epsilon_2 C)} = m \cdot D_{ON} + 1.6394\kappa - 5.4503 \quad \text{Eqn. 4-8}$$

Rearranging Eqn. 4-8 to express  $\kappa$  in terms of the rest of the parameters,

$$\kappa \left[ e^{0.5(\epsilon_1 N - \epsilon_2 C)} - 1.6394 \right] = m \cdot D_{ON} - 5.4503$$

$$\kappa = \frac{m \cdot D_{ON} - 5.4503}{e^{0.5(\epsilon_1 N - \epsilon_2 C)} - 1.6394} \quad \text{Eqn. 4-9}$$

Putting Eqn. 4-9 into Eqn. 4-5 to obtain an expression for  $T_{ON}$ ,

$$T_{ON} = \frac{m \cdot D_{ON} - 5.4503}{e^{0.5(\epsilon_1 N - \epsilon_2 C)} - 1.6394} \cdot e^{0.5(\epsilon_1 N - \epsilon_2 C)} \quad \text{Eqn. 4-10}$$

Eqn. 4-10 expresses  $T_{ON}$  in terms of  $D_{ON}$ ,  $N$  and  $C$ , which effectively provides a formula for characterising  $T_{ON}$ . However, as the formula was developed based on simulation results, a certain degree of error has been introduced. The term “error” here is used to describe the small disagreement between the  $T_{ON}$  from Eqn. 4-10 and actual  $T_{ON}$  value (with correction). These differences were found to be relatively small, and according to the results recorded the average error is  $\pm 0.26127$  timeslot. To minimise this error statistic, the amount of error of individual cases was analysed, and observations showed that as  $D_{ON}$  increases, the amount of the error also increases, see Figure 4-8. Errors in these studies, however, appear to be noise and have no direct relationship with  $N$  and  $C$ , see Figure 4-9 and Figure 4-10 respectively.

In order to evaluate  $T_{ON}$  accurately, the error term was introduced to make the necessary corrections. Let  $T_{ON(\epsilon)}$  be the value of  $T_{ON}$  with error correction. Then,

$$T_{ON(\epsilon)} = \xi(m \cdot D_{ON} - 5.4503) + \tau \quad \text{Eqn. 4-11}$$

where  $\xi = \frac{e^{0.5(\epsilon_1 N - \epsilon_2 C)}}{e^{0.5(\epsilon_1 N - \epsilon_2 C)} - 1.6394}$  and  $\tau$  is the error term.

The connection between  $D_{ON}$  and the amount of error can be expressed using a linear equation as shown in Figure 4-8.

$$\tau = 1.5717 \cdot D_{ON} - 5.9868 \quad \text{Eqn. 4-12}$$

Introducing the error term Eqn. 4-12 to Eqn. 4-11 to attain a final formula for  $T_{ON}$ , i.e.  $T_{ON(\epsilon)}$ ,

$$T_{ON(\epsilon)} = \xi(m \cdot D_{ON} - 5.4503) + 1.5717 \cdot D_{ON} - 5.9868$$

Substituting  $\xi$ ,

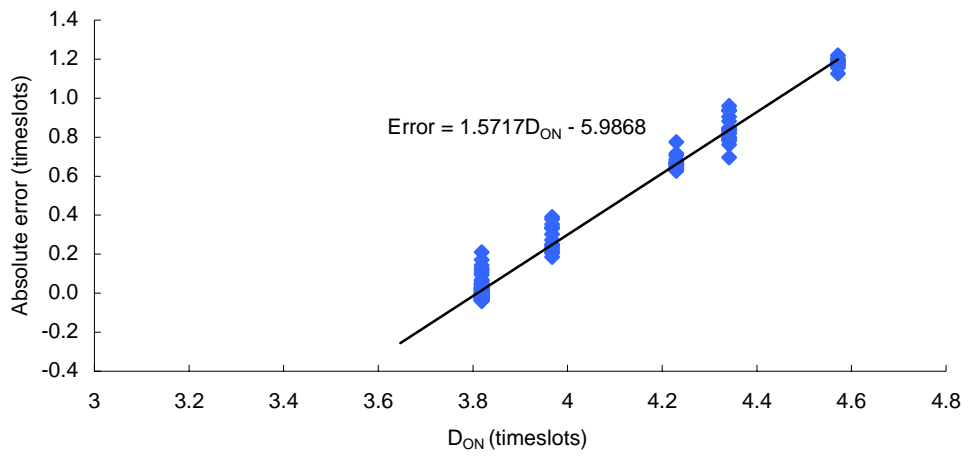
$$T_{ON(\epsilon)} = \frac{(m \cdot D_{ON} - 5.4503)}{e^{0.5(\epsilon_1 N - \epsilon_2 C)} - 1.6394} \cdot e^{0.5(\epsilon_1 N - \epsilon_2 C)} + 1.5717 \cdot D_{ON} - 5.9868$$

And finally, substituting the value of  $m$ ,  $\epsilon_1$  and  $\epsilon_2$ :

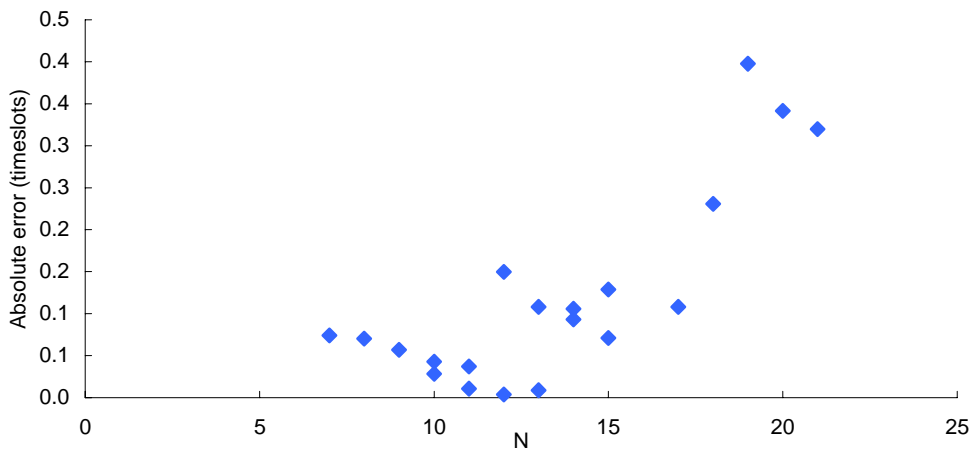
$$T_{ON(\epsilon)} = \frac{(0.87969 \cdot D_{ON} - 5.4503)}{e^{0.5(0.13667 \cdot N - 0.3394 \cdot C)} - 1.6394} \cdot e^{0.5(0.13667 \cdot N - 0.3394 \cdot C)} + 1.5717 \cdot D_{ON} - 5.9868$$

Eqn. 4-13

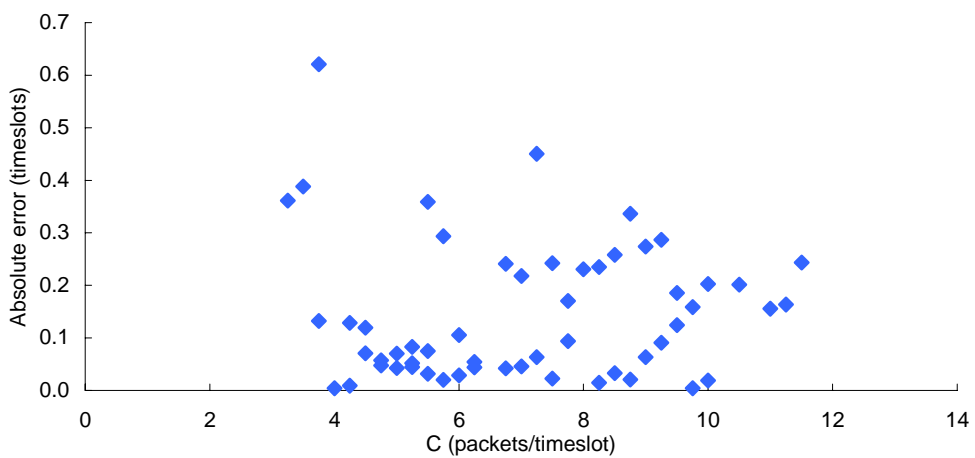
Eqn. 4-13 is in the final form for the evaluation of  $T_{ON}$  when defining a TA model. It is found to be very accurate in characterising an equivalent TA model for speeding up network simulations. When this equation was tested, the amount of error is reduced to an average of 0.0392 timeslot, which is negligible as differences in  $T_{ON}$  in that magnitude have non-noticeable effects on the overall queueing behaviour.



**Figure 4-8 Relationship between error and  $D_{ON}$**



**Figure 4-9 Relationship between error and N**



**Figure 4-10 Relationship between error and C**

### 4.2.2 Deriving $R_{ON}$

The overall distribution of the arrival rate of TA is given by the binomial distribution. Considering a single ON-OFF process, the probability of the source being ON is denoted by  $P_{ON}$  and is equal to  $D_{ON}/(D_{ON}+D_{OFF})$ . Let  $P(s)$  be the probability of  $s$  sources ON simultaneously at a randomly chosen time instant, then  $P(s)$  is given by:

$$P(s) = \binom{N}{s} P_{ON}^s \cdot (1 - P_{ON})^{N-s} \quad \text{Eqn. 4-14}$$

From the TA definitions, an ON state occurs when  $NR > C$  (in this example,  $R$  is assumed to be 1 for simplification). The probability of an  $ON_{AG}$  state is  $(P(s=C+1) \cup P(s=C+2) \cup \dots \cup P(s=N))$ , whereas the probability of an  $OFF_{AG}$  state becomes  $(P(s=0) \cup P(s=1) \cup \dots \cup P(s=C))$ . In this thesis, only the ON arrival rate,  $R_{ON}$ , is evaluated. During  $ON_{AG}$  periods, the number of active sources can vary from  $(C+1)$  to  $N$ , and the mean number of active sources during these periods is used to define the exact mean arrival rate. This involves first normalizing individual probabilities so that they all sum up to 1, i.e. find the conditional distribution given the aggregated model is in an  $ON_{AG}$  state. The expected (mean) number of sources being ON simultaneously during the active state,  $E_{ON}$ , is given by

$$E_{ON} = \sum_{i=C+1}^N P(i) \cdot i \quad \text{Eqn. 4-15}$$

where  $i$  specifies the number of sources;  $R_{ON} = E_{ON}$  for  $R=1$ ; otherwise, for scenarios in which individual  $R > 1$ , then the  $E_{ON}$  is multiplied by the original  $R$  of individual sources.

### 4.2.3 Deriving $T_{OFF}$

The characterization of  $T_{OFF}$  is based on the need to conserve the system's utilization between the original  $N$ -source model and TA. The load of the modified TA process (without  $R_{OFF}$ ) is given by:

$$\rho = \frac{T_{ON}}{T_{ON} + T_{OFF}} \cdot R_{ON} \cdot \frac{1}{C}$$

$$T_{OFF} = \frac{R_{ON} \cdot T_{ON} - \rho \cdot C \cdot T_{ON}}{\rho \cdot C} \quad \text{Eqn. 4-16}$$

where  $\rho$  is provided by Eqn. 4-1 in which the parameters of the original  $N$ -source model are applied.

### 4.3 Validation using queueing behaviour

Validation was done by comparing the queueing behaviour of the original non-accelerated buffer model and TA. The state probability at arrival instants was chosen for comparisons, which is defined as:

**$P[h]$  = P(h packets in the queue at a randomly chosen packet arrival instant)**

This statistic was obtained by monitoring the accumulated packets in the buffer specifically at packet arrival instants, which directly relates to the loss and delay of a given buffer. Queueing results can be interpreted as the estimation on packet waiting time probabilities and packet loss probabilities.

With regard to the packet waiting time probability, the deployed model is a timeslot based discrete system in which a single timeslot can be interpreted as any time units e.g. milliseconds, second. Say  $l$  packets already exist in the queue. A new arrival has to wait for all  $l$  packets being served before transmission (under the FIFO assumption anyway!). If the system serves at a rate of 1 packet/second, where one timeslot is equivalent to one second. Then, the new packet would have to wait for  $l$  seconds. Therefore, by registering the state probabilities at arrivals, the waiting time probabilities are also found providing the time unit is available.

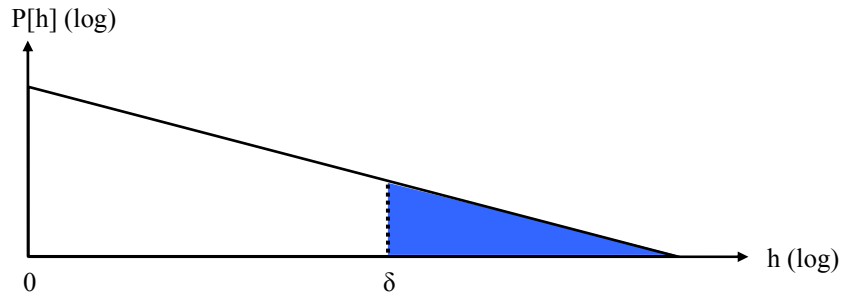
With regard to the packet loss probability, for a given buffer with a storage capacity of  $\delta$  packets, packet loss probabilities can be estimated directly using the arrival state probability. For instance, an incoming packet is lost when  $\delta$  is reached. Hence, probabilities  $P[h]$  for all  $h > \delta$  taken from an infinite buffer is equivalent to the portion that is lost for a buffer with a capacity of  $\delta$ , giving

$$\sum_{i=\delta+1}^{\infty} P[i] = \text{Overall packet loss probability}$$

where  $i > \delta$ , and therefore,

$$\text{the overall packet loss probability} = 1 - \sum_{h=0}^{\delta} P[h]$$

which is equivalent to the shaped part in Figure 4-11.



**Figure 4-11** Approximation for packet loss probability

## 4.4 Accuracy of TA under different network scenarios

Simulation results are presented in this section. Studies are categorised into two types according to the traffic applied: non-truncated Pareto traffic and truncated Pareto traffic, details of truncated Pareto traffic will be discussed later on in this chapter. In each case, homogeneous and heterogeneous traffic sources were applied, and a series of comparisons were carried out between the non-accelerated case and TA. These studies show that TA is highly accurate and simple to apply, providing an efficient and effective scheme for Power-law traffic modelling.

### 4.4.1 Non-truncated Pareto ON-OFF traffic

Non-truncated Pareto ON-OFF traffic refers as to the pseudo-generated traffic with no strict boundary applied to the process' sojourn periods. The only restriction acts upon on the process model originates from the pseudo-random number generator, as discussed in the Chapter 3.

#### 4.4.1.1 *Overlapping homogeneous Pareto ON-OFF sources*

The fundamental concept of TA has undergone validation tests using a single network node, which is the novel scenario that TA was originated from and in which it was designed to operate. To clearly demonstrate the functionality of TA, results featuring the variation of individual parameters, namely  $D_{ON}$ ,  $C$  and  $N$ , are included. Details of these simulations are listed in Table 4-4, along with the corresponding TA models derived from the algorithm.

Each set of comparisons is presented in two versions. The first version involves direct comparisons of the queueing behaviour between the original  $N$ -source model and TA. In the

second version, statistical fitting was applied to the queueing behaviour of the original N-source model, and Power-law bestfit trends were deployed to compare with TA. Power-law fitting was used to match these heavy-tail queueing. For example, Figure 4-13 is a reciprocal to Figure 4-12. These fittings were essential because as these comparisons were done, simulations for the both models were run for the same period of time (i.e. the same length), and non-ideal queueing behaviours were observed from the original N-source models, where the attained queues do not appear as a straight line on a graph with a log-log scale. This is due to the fact that these models reach steady-state much less swiftly. These bestfit trends were obtained when the original N sources were simulated for much longer periods of time, and the average taken of a number of studies, representing the “true” queue decay at steady-state. These assumptions provide an important reference for the validation of TA.

Also, in general, simulations using TA do cover the low probability regions better than the original simulation model. This is due to the fact the aggregation model has only two states, which enables steady-state to be reached far earlier. Hence, when running simulations for both original and TA with the same length, TA covers probabilities significantly further down the tail.

Figure 4-12 and Figure 4-13 demonstrate the queueing results where  $D_{ON}$  was varied. In the former figure, the solid lines represent results from the original N-source model and the dotted lines represent that of TA. The latter figure presents all the bestfit trends in terms of solid lines and results of TA are again in dotted lines. Individual  $D_{ON}$  studies are indicated by different markers in both figures. This presentation is used for the three remaining sets of comparisons.

Figure 4-12 (and Figure 4-13) shows that as  $D_{ON}$  increases, the gradient of the queue decay increases, which indicates that long decays occur with higher probabilities. With the aid of TA, these queueing behaviours were imitated correctly, and the steady status is reached much earlier.

Moving on to the variation of N, buffers with a higher number of multiplexed sources are expected to have a worse queueing behaviour (longer decays experienced), provided that the service rate of the buffer is the same. TA is able to mimic this phenomenon accurately, and Figure 4-14 and Figure 4-15 illustrate that excellent agreement is attained between the results of the original models and TA equivalents. It is worth noting that from these results that the

stable status of the system is harder to achieve when the number of sources is high prior to TA. This is due to the increase in the magnitude of state-space. Referring to Figure 4-14, the scenario with  $N = 20$ , with the same simulation duration, the stability of the queueing results of the original case is far from what is provided by TA. And, the accuracy of TA can only be justified by comparing with the bestfit line.

	Original N-source model					TA			
	$D_{OFF}$	$D_{ON}$	N	R	C	$T_{OFF}$	$T_{ON}$	$R_{ON}$	$C_{TA}$
<b>Changing <math>D_{ON}</math></b> Figure 4-12 & Figure 4-13	10	3.1	10	1	5	2.3258	1.8059	6.2258	5
	10	3.3	10	1	5	2.2492	1.9315	6.2459	5
	10	3.9	10	1	5	2.2292	2.3086	6.2871	5
	10	4.5	10	1	5	2.1926	2.6856	6.3285	5
	10	5.0	10	1	5	2.2022	2.9998	6.3569	5
<b>Changing N</b> Figure 4-14 & Figure 4-15	10	3.9	14	1	8	1.6442	1.5401	9.2586	8
	10	3.9	16	1	8	1.2857	1.9254	9.3898	8
	10	3.9	18	1	8	1.0120	2.4899	9.5407	8
	10	3.9	20	1	8	0.7985	3.3858	9.7140	8
<b>Changing C</b> Figure 4-16 & Figure 4-17	10	3.9	8	1	3.5	1.3125	2.9559	4.7396	3.5
	10	3.9	8	1	4.0	1.6135	2.4643	5.2396	4.0
	10	3.9	8	1	4.5	1.8494	2.0933	5.6315	4.5
	10	3.9	8	1	5.0	2.1503	1.8042	6.1315	5.0
<b>Changing N &amp; C</b> Figure 4-18 & Figure 4-19	10	3.9	8	1	4	1.6135	2.4643	5.2396	4
	10	3.9	10	1	5	1.4871	2.3086	6.2871	5
	10	3.9	14	1	7	1.3353	2.0410	8.3605	7
	10	3.9	18	1	9	1.2461	1.8198	10.4156	9
	10	3.9	26	1	13	1.1441	1.4765	14.4942	13

**Table 4-4 Parameters for the validation of TA**

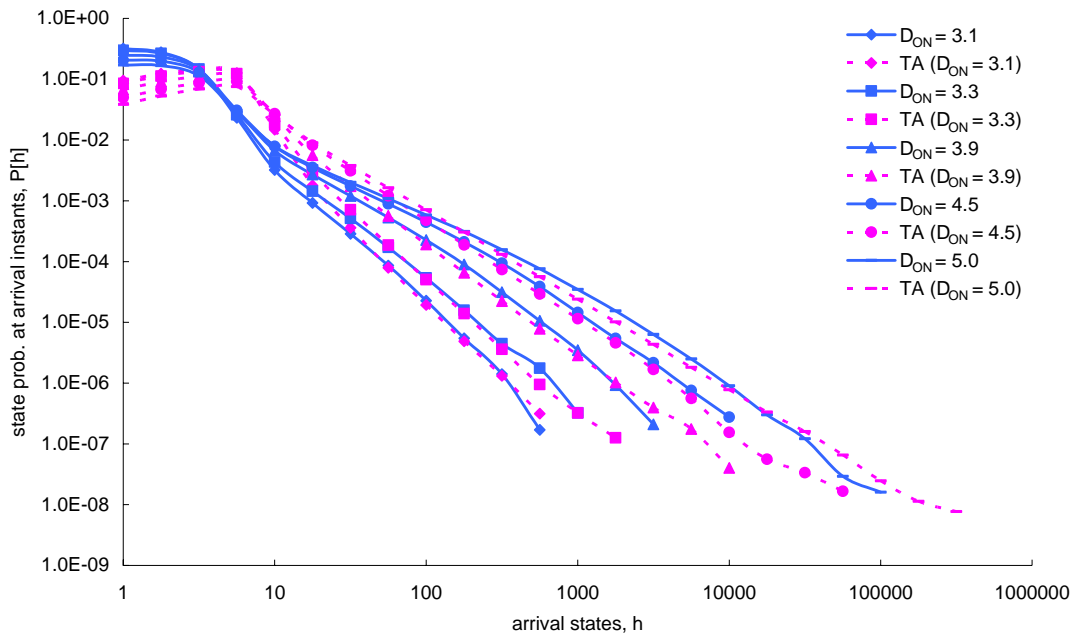


Figure 4-12 Raw queuing results with the variation of  $D_{ON}$

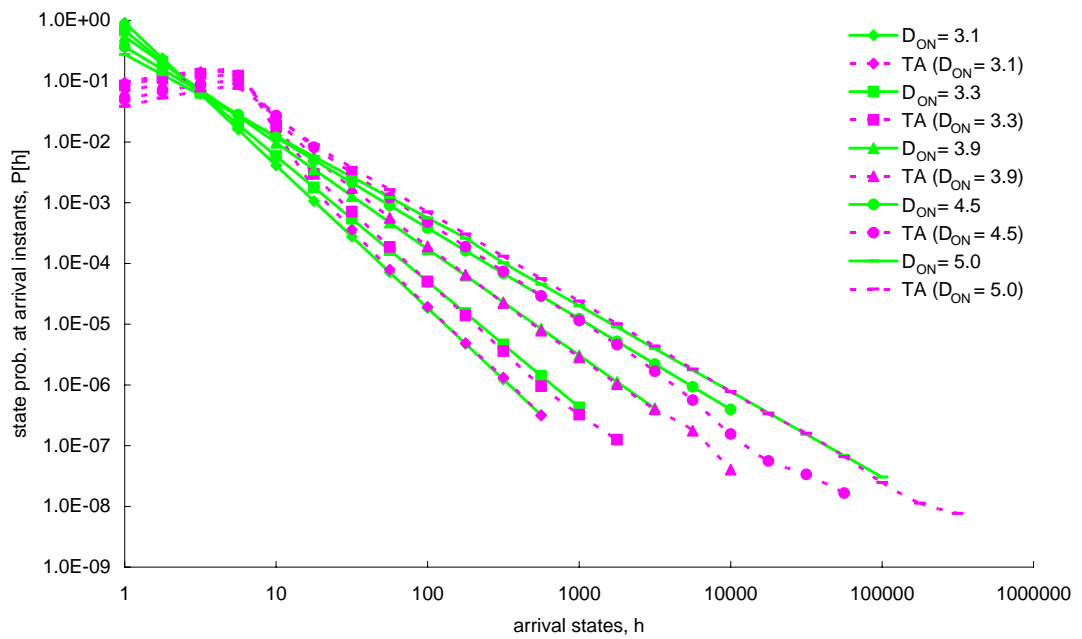


Figure 4-13 Bestfit applied to queuing results in Figure 4-12

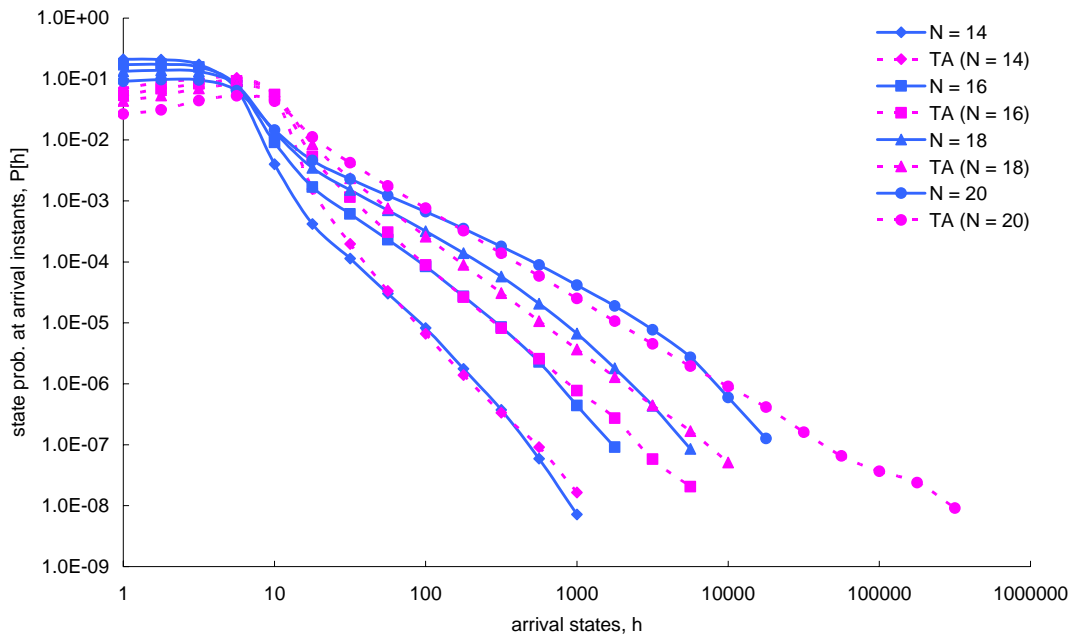


Figure 4-14 Raw queuing results with the variation of  $N$

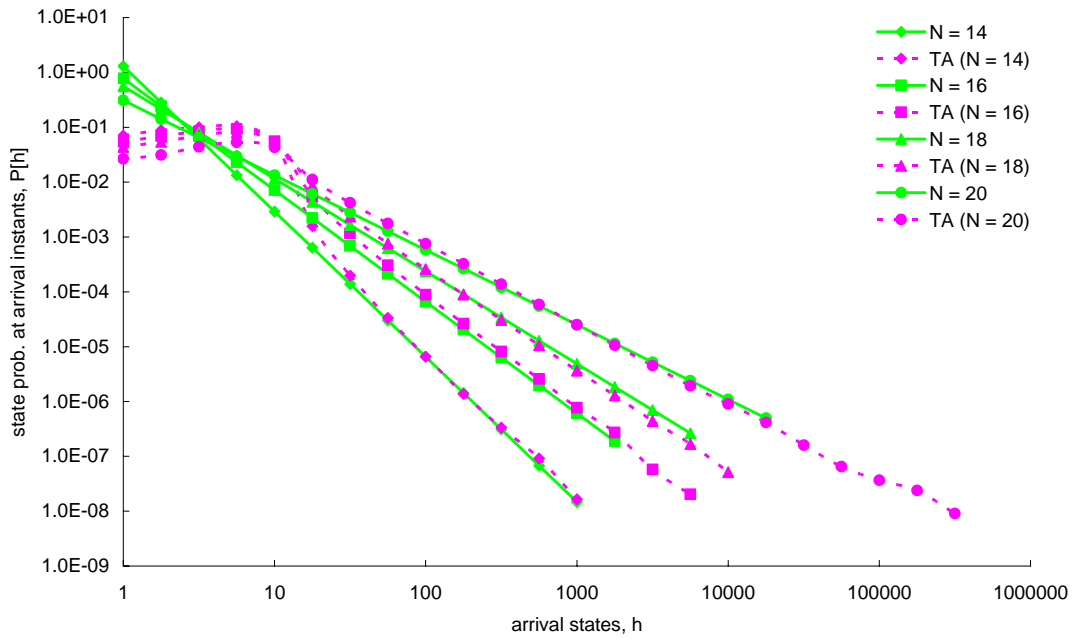


Figure 4-15 Bestfit applied to queuing results in Figure 4-14

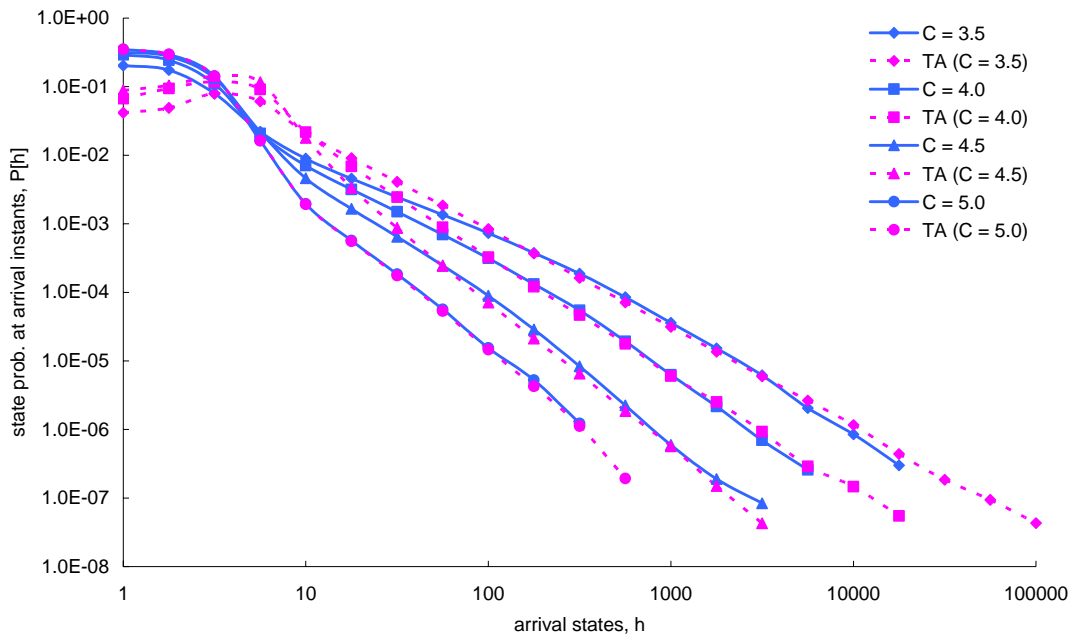


Figure 4-16 Raw queuing results with the variation of  $C$

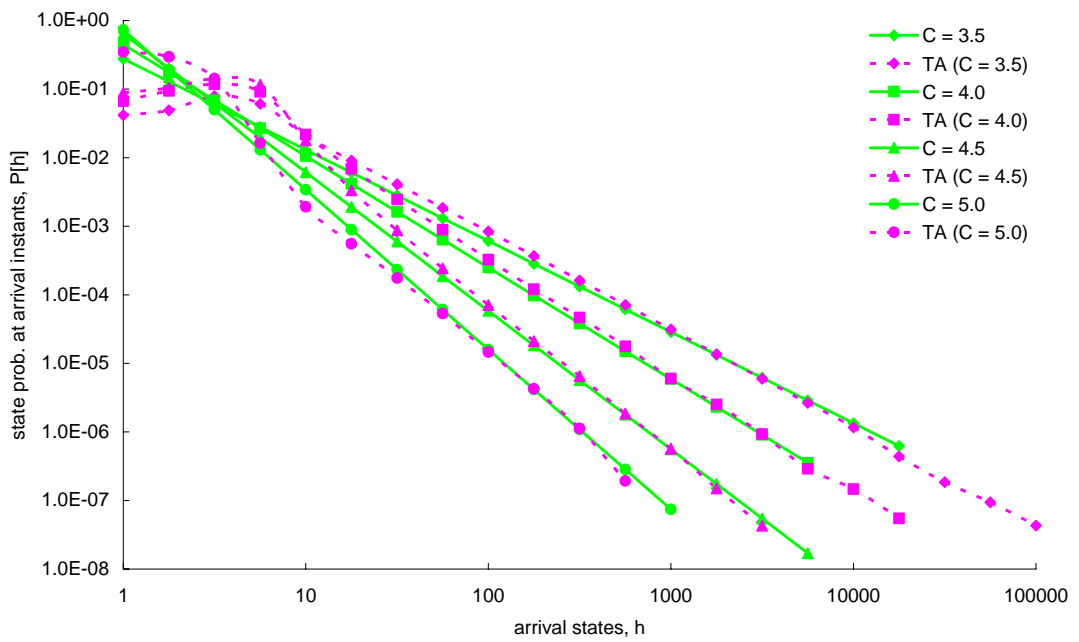


Figure 4-17 Bestfit applied to queuing results in Figure 4-16

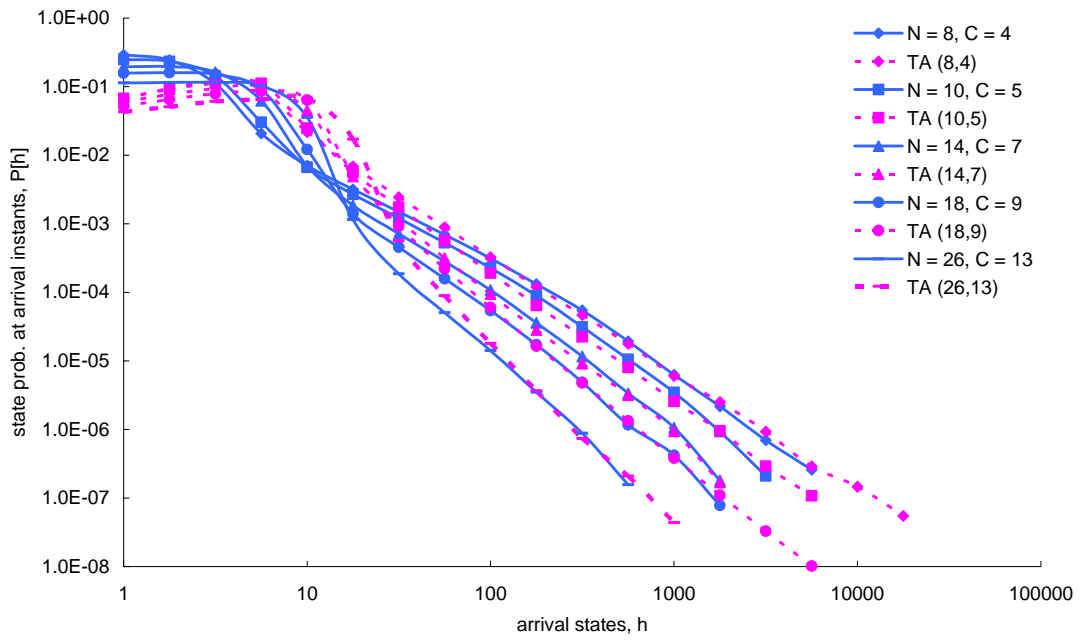


Figure 4-18 Raw queuing results with the variation of  $N$  and  $C$

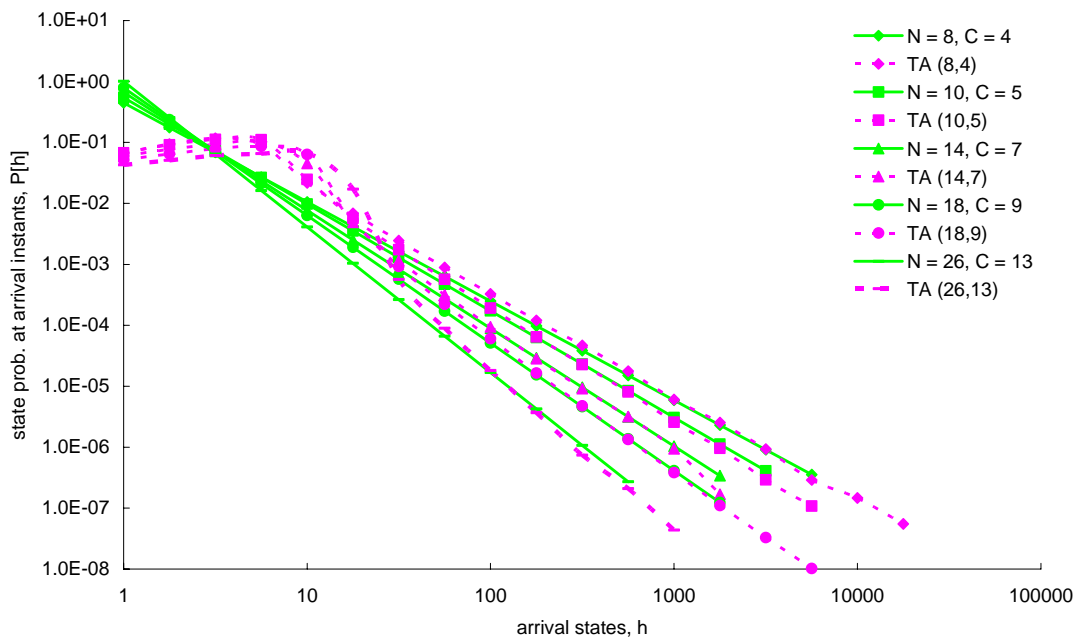


Figure 4-19 Bestfit applied to queuing results in Figure 4-18

Results on the variation of C are illustrated in Figure 4-16 and Figure 4-17, and results on the variation of both N and C are presented in Figure 4-18 and Figure 4-19. These examples are included to further show that TA is insensitive to the change of model attributes. Accuracy is maintained throughout these results, which implies that TA is applicable to many network scenarios.

#### 4.4.1.2 *Overlapping heterogeneous Pareto ON-OFF sources*

A heterogeneous environment is formed by multiplexing Pareto ON-OFF sources with different  $D_{ON}$ 's and  $D_{OFF}$ 's periods and/or R's. In order to apply TA, the combined traffic characteristics of these sources need to be known so that an equivalent homogeneous traffic model can be obtained to substitute for the heterogeneous case. Once the corresponding homogeneous process is obtained, the aggregation of heterogeneous traffic sources follows the original TA for Power-law traffic.

The method used to derive an equivalent homogeneous process is straightforward, provided that all parameters of the original heterogeneous process are known. To decide the mean ON and OFF time ( $D_{ON}$  and  $D_{OFF}$ ) of the equivalent homogeneous process, the proportion of individual heterogeneous traffic is exploited. Consider a scenario in which there are four different types of traffic, and Table 4-5 lists the individual traffic types.

Traffic	$D_{OFF}$	$D_{ON}$	R	N
1	10	3	1	5
2	12	4	1	5
3	8	3	1	5
4	14	5	1	5

**Table 4-5 Parameters for the heterogeneous traffic multiplexing scenario**

The overall system output capacity is 10 packets/timeslot and the total number of sources is

$$N = N_1 + N_2 + N_3 + N_4 = 20$$

Assume  $G_1$ ,  $G_2$ ,  $G_3$  and  $G_4$  are the proportion of traffic1, traffic2, traffic3 and traffic4 with respect to the overall number of sources respectively, and obtaining

$$G_1 = G_2 = G_3 = G_4 = 5/20 = 0.25$$

The mean ON and OFF times of the equivalent homogeneous process can then be calculated by:

$$D_{ON\ Equ} = \sum_{i=1}^{\max.} D_{ONi} \cdot G_i \quad \text{Eqn. 4-17}$$

$$D_{OFF\ Equ} = \sum_{i=1}^{\max.} D_{OFFi} \cdot G_i \quad \text{Eqn. 4-18}$$

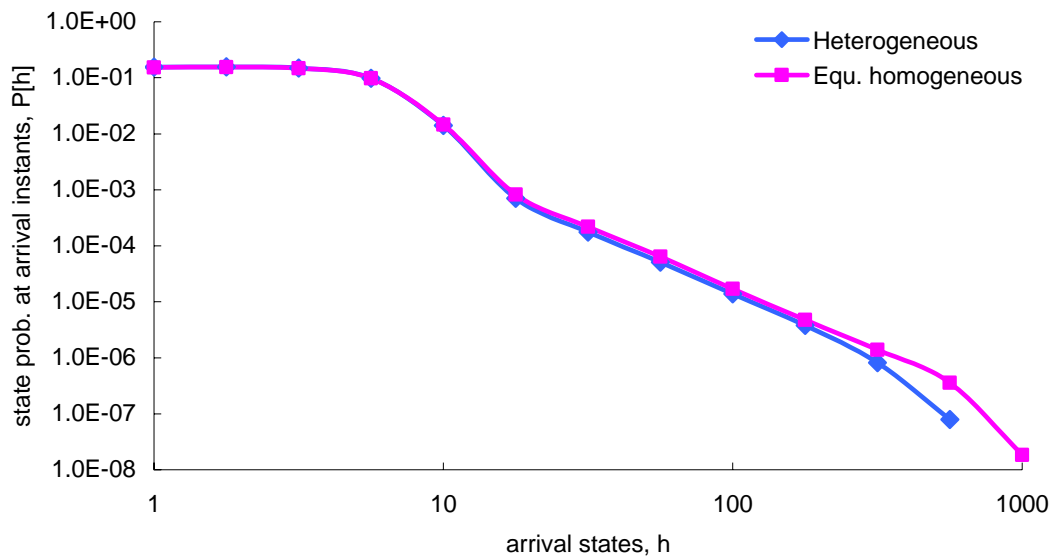
where i indicates the different types of traffic. Therefore,  $D_{OFF\ Equ}$  and  $D_{ON\ Equ}$  are:

$$\begin{aligned} D_{ON\ Equ} &= 10(0.25) + 12(0.25) + 8(0.25) + 14(0.25) \\ &= 11 \end{aligned}$$

$$\begin{aligned} D_{OFF\ Equ} &= 3(0.25) + 4(0.25) + 3(0.25) + 5(0.25) \\ &= 3.75 \end{aligned}$$

The equivalent homogeneous model has the same N, C and R as that of the heterogeneous case. Figure 4-20 is the comparison of the queueing results between the heterogeneous and the corresponding homogeneous model. It can be seen that excellent accuracy is achieved. Consequently, heterogeneous traffic sources can be aggregated by compressing the equivalent homogeneous model.

Table 4-6 presents further validations on heterogeneous traffic multiplexing, which aimed to demonstrate that the derivation of an equivalent homogeneous model is irrespective to the heterogeneous settings.



**Figure 4-20 Validation of the equivalent homogeneous model for multiplexed heterogeneous sources**

A selection of heterogeneous combinations was exploited based on the fact that they have the same  $D_{ONEqu}$  and  $D_{OFFEqu}$ , and their queue DR's (decay rates) were then compared. As these models are supposed to have the same equivalent homogeneous model, their DR's should be equal. Four sets of comparisons are presented here and results show that given that the final  $D_{ONEqu}$  and  $D_{OFFEqu}$  of individual heterogeneous cases are identical, DR's are found to be highly close to the others, and the deviations among them are so small that the DR's of these cases can be considered as the same. These DR's were then compared against that of the corresponding equivalent homogeneous model, and they were found to be matched. Therefore, the equivalent homogeneous method can be applied to scenarios with different heterogeneous parameters.

	Heterogeneous					Equ. Homogeneous
	$D_{OFFi}$	$D_{ONi}$	$N_i$	C	DR	
1	12, 12, 12, 12	2.5, 3.4, 2.7, 3.8	5, 5, 5, 5	20	-4.2535	$D_{OFFEqu} = 12$ $D_{ONEqu} = 3.1$
	12, 14, 10.75	2.9, 3.6, 2.9625	7, 5, 8		-4.1160	
	12, 14, 8.67	2.9, 3.0, 3.467	4, 10, 6		-4.1928	
	Average				-4.1874	DR = -4.2056
	S.T.D.				0.056	
2	12, 12, 12, 12	3.5, 3.0, 2.4, 4.3	5, 5, 5, 5	20	-3.7780	$D_{OFFEqu} = 12$ $D_{ONEqu} = 3.3$
	10, 14, 12.5	3.3, 3.8, 2.9875	7, 5, 8		-3.6641	
	11, 12, 12.67	2.8, 3.0, 4.13	4, 10, 6		-3.7601	
	Average				-3.7341	DR = -3.7206
	S.T.D.				0.050	
3	12, 12, 12, 12	3.7, 4.2, 4.0, 3.7	5, 5, 5, 5	20	-2.5871	$D_{OFFEqu} = 12$ $D_{ONEqu} = 3.9$
	10, 14, 12.5	3.8, 4.0, 3.925	7, 5, 8		-2.5399	
	11, 12, 12.67	3.6, 4.1, 3.76	4, 10, 6		-2.5604	
	Average				-2.5625	DR = -2.5549
	S.T.D.				0.019	
4	12, 12, 12, 12	4, 4.6, 3.8, 4.4	5, 5, 5, 5	20	-2.1441	$D_{OFFEqu} = 12$ $D_{ONEqu} = 4.2$
	10, 14, 12.5	4, 4.2, 4.375	7, 5, 8		-2.0587	
	11, 12, 12.67	3.6, 4.32, 4.4	4, 10, 6		-2.1329	
	Average				-2.1119	DR = -2.0941
	S.T.D.				0.038	

**Table 4-6 Parameters for validating the equivalent homogeneous model method**

#### 4.4.2 Truncated Pareto ON-OFF traffic

This section specifically focuses on situations in which the Pareto distribution is truncated. Taking into consideration that file sizes do not grow indefinitely in real networks, an upper limit has been introduced to ON periods, truncating Power-law distributions in a fashion consistent with reality, as widely reported [Nuzman01, O’Mahony01, Queseth01, Schwefel99]. OFF periods are left unbound as often idle times of networks are always unrestricted.

Truncating Power-law distributions seems to be an irrational concept in a strict mathematical sense, as the slow-decay of the distribution is the unique characteristic which gives non-negligible probabilities to extremely large values. These large values contribute the most to both the mean and the variance, and omitting the tail results in damaging the purity of the Power-law nature. However, as mentioned in Chapter 2, the generation of Pareto distributed numbers in computational simulations is always restricted by the random number generator, so the distribution is always truncated anyhow, but the credibility of simulation based on Power-law traffic modelling is still highly respected. From an engineering point of view, the reason for the truncation is straightforward: a boundary does exist in file sizes! Provided that not the entire “tail” is missed out, the Power-law nature can still be preserved reasonably, which sojourn times can be retained in a fashion consistent with real network situations.

As part of the “tail” of the distribution is removed, it is essential to recalculate the mean and normalize the probability of individual elements within the range. Let  $L$  be the upper limit of a Pareto distribution, then having  $q = L$  which can be substituted into Eqn. 2-24, and  $w$  in this case becomes the minimum value accepted for the traffic generation. By applying Eqn. 2.26, it is then able to locate the mean of the truncated Pareto distribution.

There is not an established limit for truncating Power-law traffic, however, some studies do have their truncation limits specified, e.g. [Nuzman01, O’Mahony01, Pitts01, Queseth01], and file transmission times are restricted in the range of 300s to 22000s. Truncation limits in this thesis are selected within this range. It is worth noting again that examples presented in the following sections are expressed in terms of the actual values after truncation.

#### ***4.4.2.1 Overlapping homogeneous truncated Pareto ON-OFF sources***

A single node scenario was again used in this section. The aggregation of truncated Pareto sources can be categorized into two steps: firstly obtaining an equivalent original Pareto ON-OFF model which provides an upper bound for the truncated case (hence, the aggregation of the original Pareto model can be done via TA), and secondly, locating the upper boundary of the TA model to ensure the consistency of the decay of the states probabilities in the queue.

An example is presented in Table 4-7. The scenario with overlapping truncated Pareto traffic is given under the name of “Truncated”, and parameters listed are the actual values applied to the model. The mean ON time of the truncated model is used as  $D_{ON}$  of the equivalent

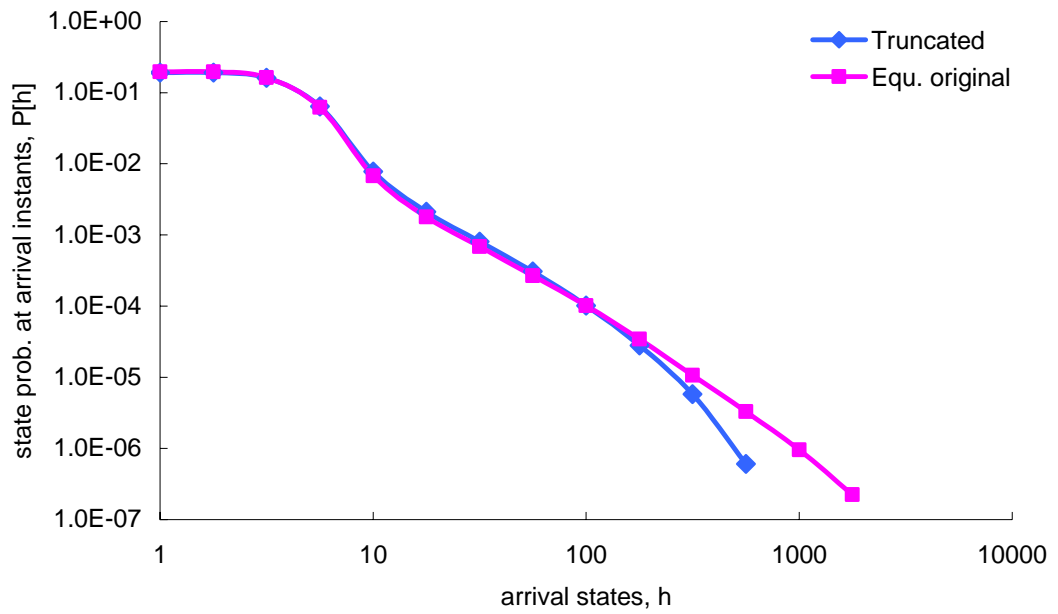
model with original Pareto sources, see the row “Eqn. Original”, which is the corresponding original Pareto model for TA to take place. Figure 4-20 shows that comparisons on the queueing results of the two cases, and apart from the tail part (as expected), the two sets of results are indistinguishable.

Once an equivalent original Pareto model is obtained, TA can effectively be applied to aggregate the sources in the model, see Table 4-7 for the consequent TA model. The next step is to take the truncated tail into consideration. Unfortunately, applying  $L_{ON}$  of the truncated case to the defined TA model is not an accurate way, and therefore, a new limit needs to be assigned for the TA model so that the truncated tail can be reflected.

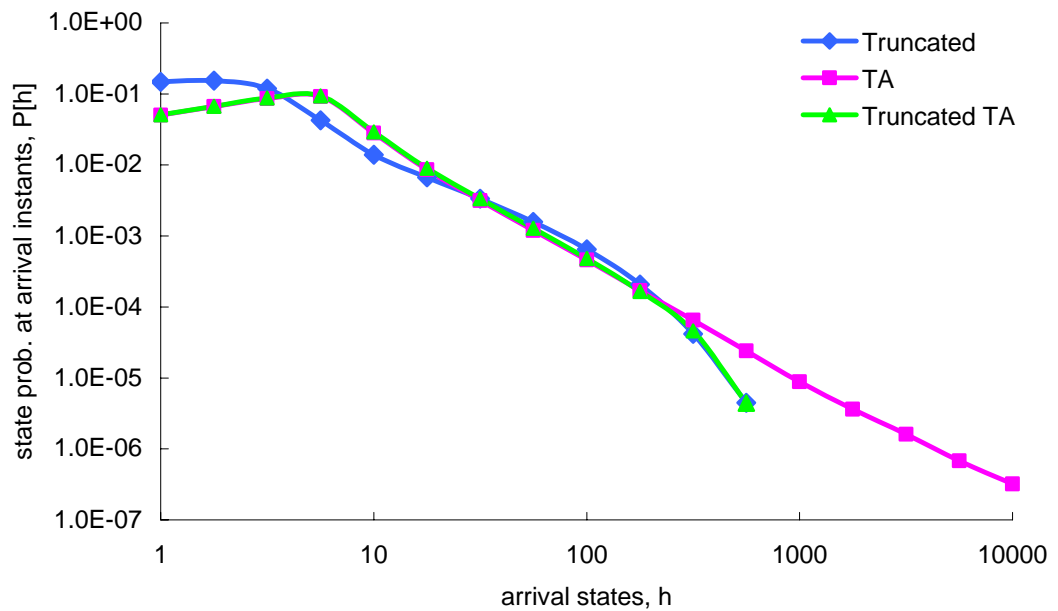
	$D_{OFF}$	$D_{ON}$	$L_{ON}$	$N$	$R$	$C$	$T_{OFF}$	$T_{ON}$	$L_{ON(TA)}$	$R_{ON}$	$C_{TA}$
<b>Truncated</b>	10	3.824	1000	14	1	7	–	–	–	–	–
<b>Eqn. Original</b>	10	3.824	–	14	1	7	–	–	–	–	–
<b>TA</b>	–	–	–	–	–	–	2.258	1.981	–	8.286	7
<b>TA with limit</b>	–	–	–	–	–	–	2.258	1.981	900	8.286	7

**Table 4-7 Parameters for the truncated homogeneous source scenario**

For this particular example, a  $L_{ON(TA)}$  of 900 timeslots was introduced, and the related queueing results are compared with the truncated model and non-truncated TA model, illustrated in Figure 4-22. It can be seen that an accurate allocation of  $L_{ON(TA)}$  enables the aggregation of truncated Pareto sources provided that an equivalent original Pareto model is defined.



**Figure 4-21 Comparisons on the queuing behaviour between the overlapping truncated traffic case and its equivalent non-truncated model**



**Figure 4-22 Application of TA to an overlapping truncated Pareto traffic scenario**

#### 4.4.2.2 *Overlapping heterogeneous truncated Pareto ON-OFF sources*

The procedure of aggregating truncated heterogeneous traffic sources is similar to the non-truncated heterogeneous case, in which an equivalent homogeneous model is required. The method used to derive the homogeneous model is the same as the previous multinomial example. An example is presented in Table 4-8 where parameters set for the various traffic sources are listed.

Traffic	D <sub>OFF</sub>	D <sub>ON</sub>	L <sub>ON</sub>	R	N
1	10	5.55	1800	1	7
2	14	3.6	1800	1	11
3	13	4.34	1800	1	12

**Table 4-8** Parameters for heterogeneous truncated sources

The overall system output capacity = 15 packets/time unit.

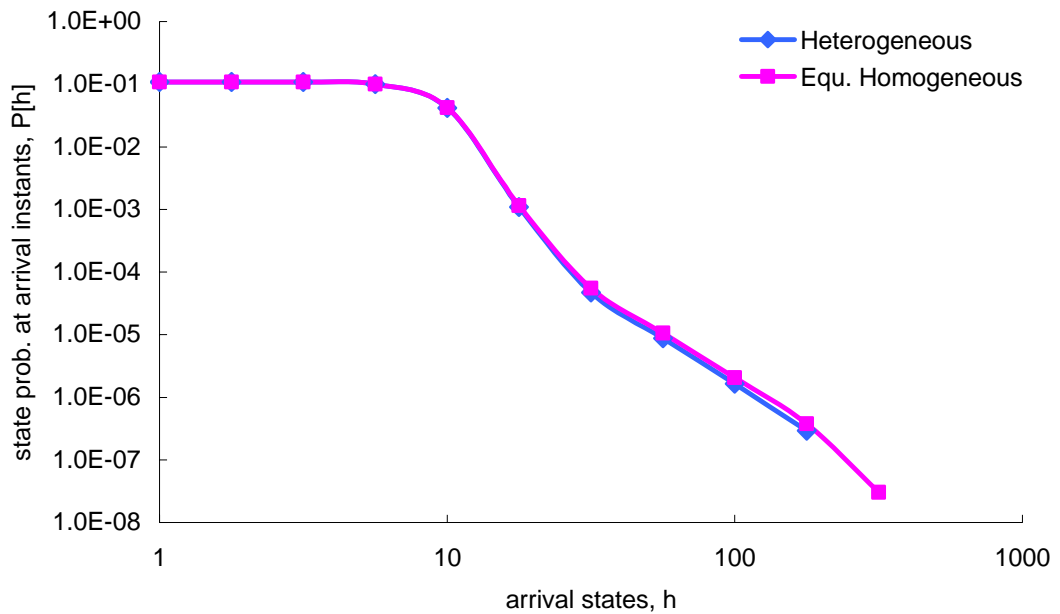
Referring to the procedures in 4.4.1.2, N is 30 in this example and  $G_1 = 0.2333$ ,  $G_2 = 0.3667$ , and  $G_3 = 0.4$ . Then,  $D_{OFFequ}$  and  $D_{ONEqu}$  are given by:

$$\begin{aligned} D_{OFFequ} &= 10(0.2333) + 14(0.3667) + 13(0.4) \\ &= 12.67 \end{aligned}$$

$$\begin{aligned} D_{ONEqu} &= 5.55(0.2333) + 3.6(0.3667) + 4.34(0.4) \\ &= 4.35 \end{aligned}$$

and, ON periods of the equivalent homogeneous model have the same truncation limit.

A comparison of the state probabilities at arrival instants for the two models is illustrated in Figure 4-23. It is again shown that this method of finding an equivalent model is highly accurate, and excellent agreement is attained between the two sets of results.



**Figure 4-23 Validation of the equivalent truncated homogeneous model for multiplexed truncated heterogeneous sources**

## 4.5 Acceleration achieved by TA

Besides accuracy, another important feature of TA is the acceleration provided by the technique. Speedup is available with TA is mainly due to the following two reasons: firstly, the overhead in running the model is much less than original traffic overlapping cases due to the fact that only a single traffic source is required. Secondly, and more importantly, the reduction in the number of states in the TA model allows simulation to reach steady-state with a smaller number of events, and hence, acceleration is achieved.

In this section, the amount of acceleration offered by TA is quantified with reference to three different aspects: traffic generation, buffer monitoring process, and rate of reaching steady-state.

### 4.5.1 Traffic generation

Traffic generation concerns the actual generation time of the traffic sources, and in these evaluations, the actual wall-clock time of individual traffic generation was recorded and compared.

An original N-source scenario was set up with  $D_{\text{OFF}} = 10$ ,  $D_{\text{ON}} = 5$ ,  $N = 10$ ,  $R = 1$  and  $C = 5$ . In addition, an equivalent TA model was derived for this particular case with  $T_{\text{OFF}} = 2.2022$ ,  $T_{\text{ON}} = 2.9998$ ,  $R_{\text{ON}} = 6.3569$  and  $C = 5$ . The former case is named “Original 1” and the latter is referred to as “TA 1”, and the two models will be recalled in the next two sub-sections.

In Figure 4-24, the times taken by the two models were monitored when they were run for the same duration, i.e. the same number of timeslots. Seed values are kept the same in both simulations to provide a fair comparison. It is shown that the run-time of a simulation increases linearly with the increase of simulation length, and this holds true for both models. However, TA is advantageous in the sense that its run-time is always shorter than that of the original model, and the rate at which its run-time increases with the simulation length increment is much lower. This effectively indicates that the overhead involved in TA is significantly lesser.

Figure 4-25 presents an overview of the magnitude of speedup in simulations in which there are a high number of sources involved. The settings of the two models are unchanged, and traffic sources with a length of  $56 \times 10^6$  timeslots were produced in each case. Again, the actual wall-clock time taken by individual simulations was recorded. It is clearly shown that while the run-time of the original N-source model rises linearly with the number of sources, TA introduces no extra time with the variation of the number of traffic. This is due to the fact that only a single traffic model is needed regardless of the number of traffic sources involved in the original scenario.

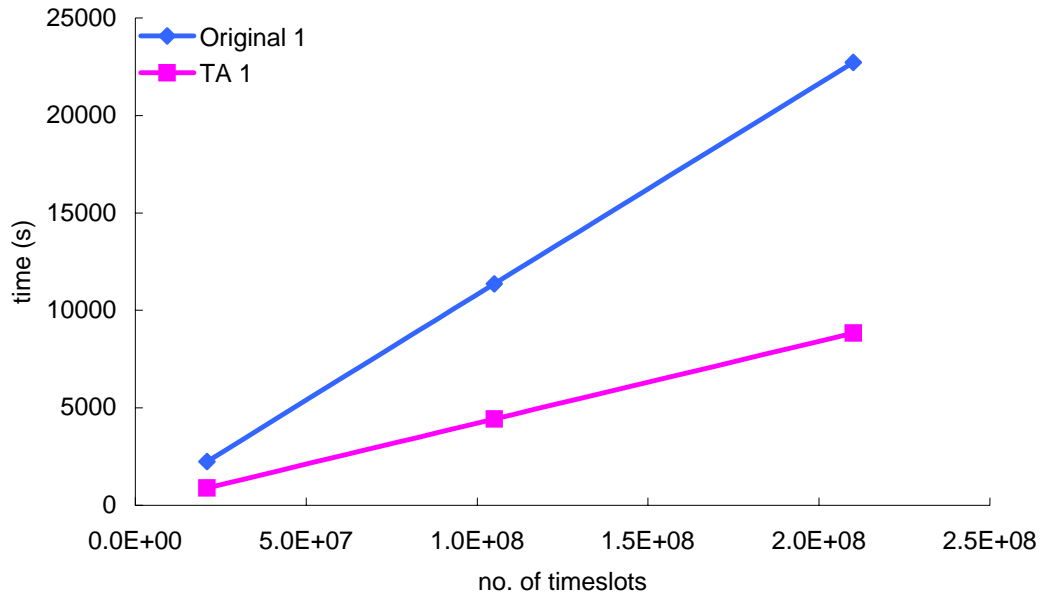


Figure 4-24 Traffic generation (wall-clock) time with increasing number of timeslots

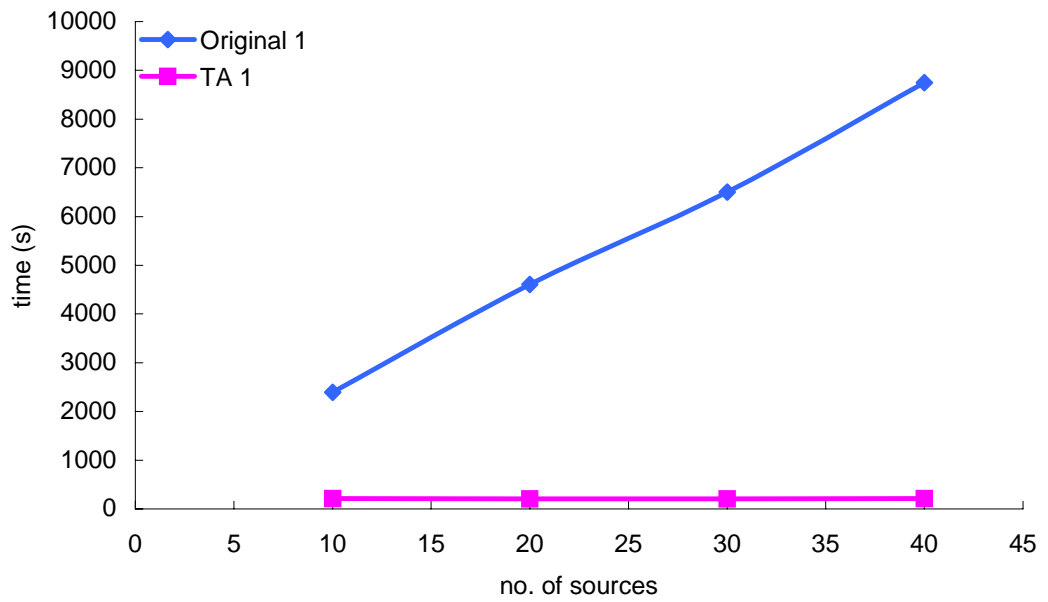


Figure 4-25 Traffic generation (wall-clock) time with increasing number of sources

#### **4.5.2 Buffer monitoring process**

This part of evaluation takes into account the time required to generate traffic sources and complete the buffering process for all the packets. “Original 1” and “TA 1” were applied in this study with the configuration unaltered.

The first set of comparisons is presented in Figure 4-26 in which the two models were run for the same length and the corresponding wall-clock times taken were registered. Results show that the run-time of the two models increases as the simulation length extends, however, the rate of run-time expansion of the two models differs from each other in the sense that the time increment of TA is smaller. This can be explained by the fact that for the same simulation length, TA generates fewer packets, and hence speedup is achieved. Furthermore, less overhead involved in TA simulations also contributes towards the acceleration, which is demonstrated in Figure 4-27.

Figure 4-27 compares the time taken for the two models to generate the same number of packets. Similar linear growth can be observed, and again, it is demonstrated that TA consumes less wall-clock time to carry out the assigned task. This is again supported by the intuition that there is less overhead in the TA model.

The acceleration shown in Figure 4-26 and Figure 4-27 may seem insignificant as the difference between the run-time of the two models is marginal. However, the most important characteristic of TA and the most significant contribution towards its acceleration is its rate of achieving steady-state, which is not demonstrated in these results. The next section discusses this particular property in details and related acceleration is illustrated.

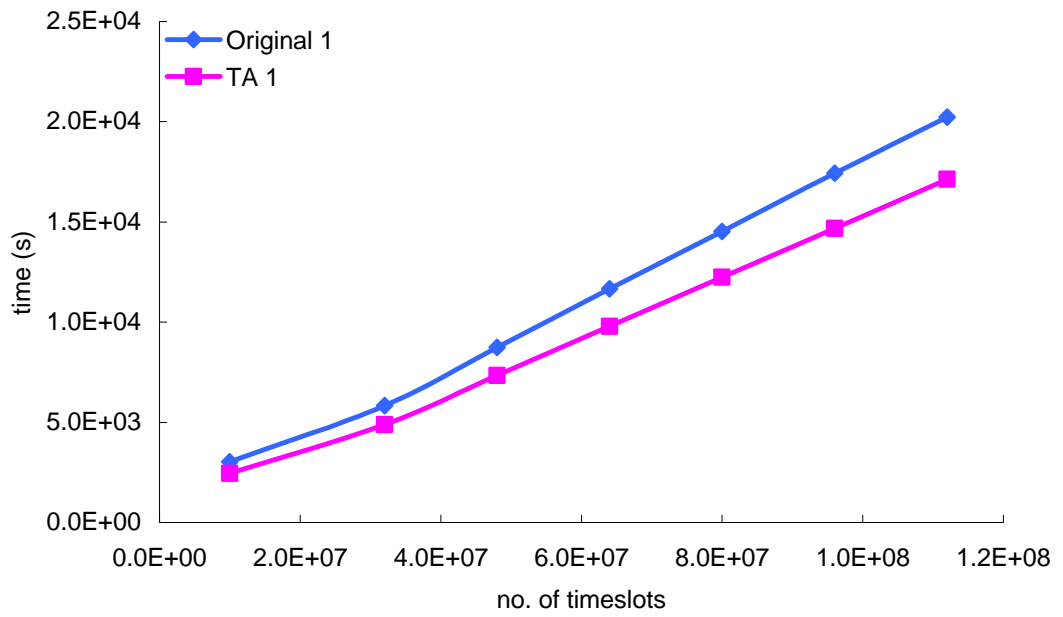


Figure 4-26 Buffer monitoring (wall-clock) time with increasing number of timeslots

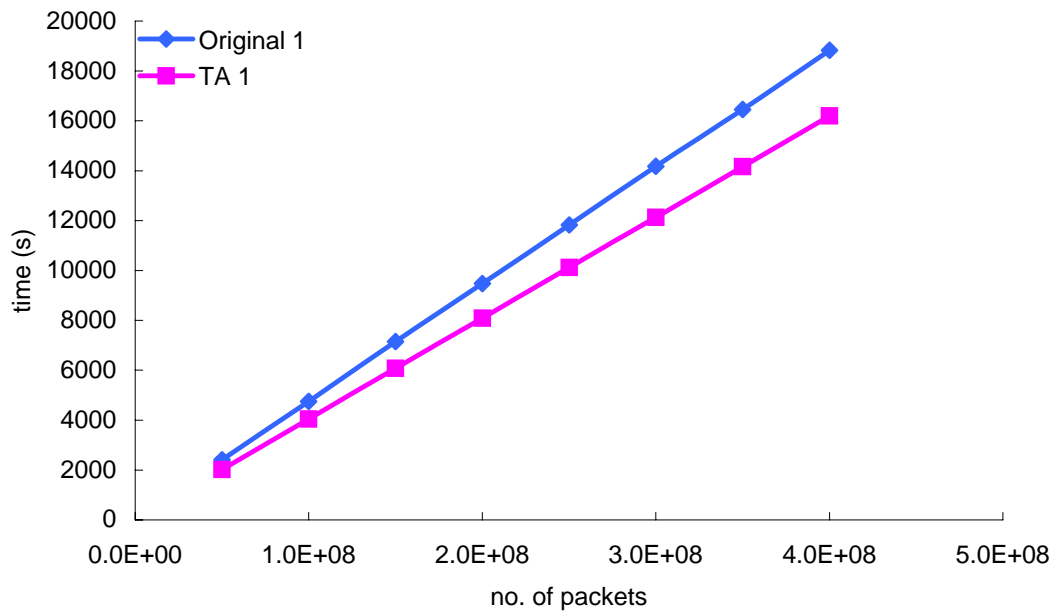


Figure 4-27 Buffer monitoring (wall-clock) time with increasing number of packets

### 4.5.3 Rate of reaching steady-state

The rate of reaching steady-state is the best way to express the acceleration of TA. Due to the traffic input process in TA consisting of only two states, it is able to reach stability within a shorter period of time. To demonstrate this property, the two previous models were run for the same period of time and their queueing results were plotted in Figure 4-28. An important point to note is that the queueing results of TA extend much further down the queue tail in comparison to that of the original model, indicating that the ability of TA to reach steady-state with shorter simulation lengths. This effectively means that simulation lengths of TA can be assigned to a shorter period, directly reducing the run-time.

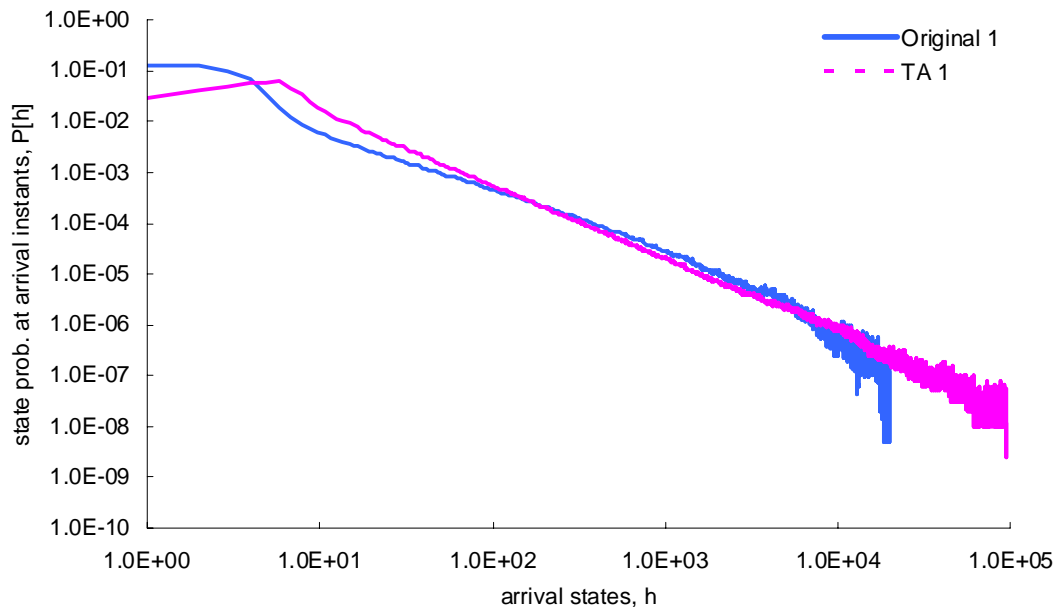


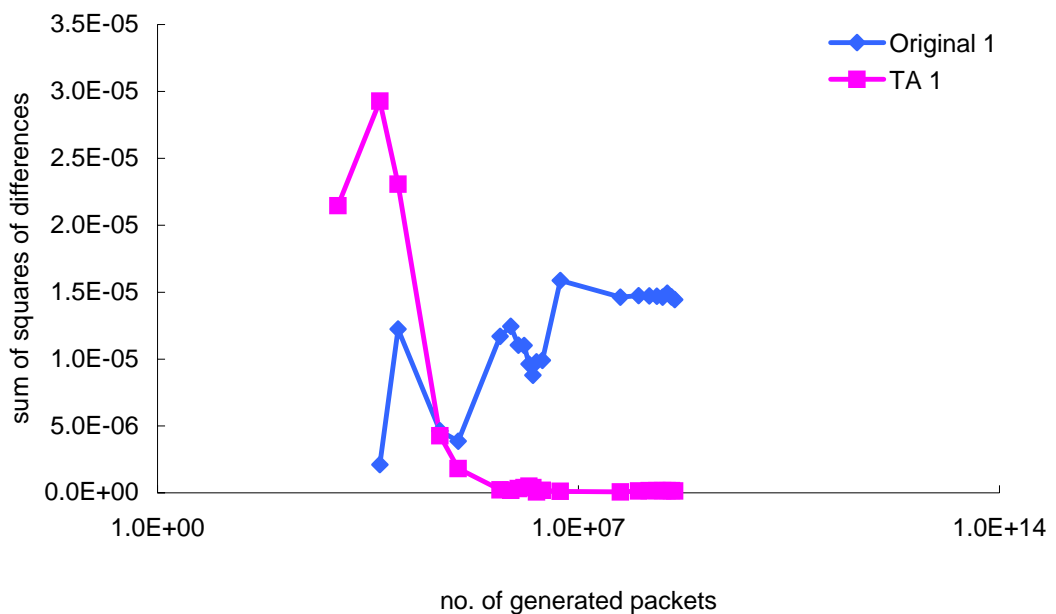
Figure 4-28 Comparisons on the queue tail extension

In addition, the sum of squares of differences method was introduced to visualise the difference in the rate of reaching stability of the two models (see Appendix 4). First of all, “Original 1” (described in 4.5.1) was simulated for ten times with different seed values over a long period of  $10 \times 10^8$  timeslots. The queueing results obtained from these simulations are then averaged, and the corresponding queue distribution is treated as the steady-state response of this particular model. The method of least squares was then used to define a Power-law bestfit to characterise the trend of the queue decay (the Power-law bestfit appears

as a linear trend line on a log-log scale). The first ten queue states have been omitted when deriving the Power-law bestfit as these states always deflect from the trend of the queue. The obtained Power-law bestfit is also considered as the steady-state response of TA, as “Original 1” and “TA 1” are expected to give the same overall queuing behaviour.

The next step was to set up an independent simulation for both “Original 1” and “TA 1”. Their corresponding queuing behaviours are monitored at different stages of the simulation, and the focus is to observe the rate of the two models reaching stability. At every queue evaluation, state probabilities of the two independent models were compared with the steady-state results, and their sum of squares of differences were calculated. These values were then plotted against the number of packets generated at the time the related statistics were collected, illustrated in Figure 4-29. By comparing this index at difference stages, the rate that a model reaches its stability can be estimated. The smaller the value of this index, the closer the individual data to the steady-state results, and the more stable is the model.

Referring to Figure 4-29, it is clearly shown that “Original 1” is far from stability even after  $4 \times 10^8$  packets have been generated. On the contrary, “TA 1” swiftly reaches steady-state after only  $5 \times 10^5$  packets. As a result, TA is 800 times faster in reach stability in this particular case. Similar magnitudes of acceleration are observed in many other comparisons.



**Figure 4-29 Acceleration achieved by TA in terms of rate of reaching steady-state**

## 4.6 Summary

An acceleration simulation technique, namely TA, has been presented in the chapter. The technique was developed based on the notion of state-space reduction, where the number of states of a traffic model is reduced, and effectively offering acceleration.

TA targets network scenarios with overlapping Pareto ON-OFF traffic. The parameterization process of the technique involves monitoring the effect of individual parameters towards the equivalent TA model, basing on the following assumptions:

- The mean OFF time of the original model (i.e.  $D_{\text{OFF}}$ ) has a mean value at least 10 timeslots, in other words, alpha value no greater than 1.1111.
- The gradient of the queue decay,  $b$ , has to be in the range of  $-2 < b < -1$ , which effectively is the range of that the data were collected for the algorithm derivation. And more importantly, it is the range where Power-law queueing lies as the gradient corresponds to the value of  $\alpha$ , the shape parameter, see Chapter 2.

Simulation results show that TA is highly accurate. Significant speedup, in the magnitude of 800, is obtained, offering a powerful accelerated simulation technique for Power-law traffic modelling. In addition, the application of TA to many network scenarios was demonstrated.

The following chapter will examine how TA can be applied to replace BG sources of an end-to-end connection. In addition, the chapter will also demonstrate that TA remains highly accurate even in the presence of the weighted round robin scheduler.

## Chapter 5

# TA in End-to-end Connections

This chapter covers the application of TA and can be seen as an extension of the validation of TA. Studies are no longer restricted to a single buffer with overlapping Pareto traffic, and have been extended to an end-to-end connection through multiple buffers. The importance of end-to-end studies will be discussed in depth in Chapter 7. Here, the focus is to look at how TA can be effectively applied to different end-to-end network connections with different configurations. End-to-end models in this chapter were implemented in Opnet.

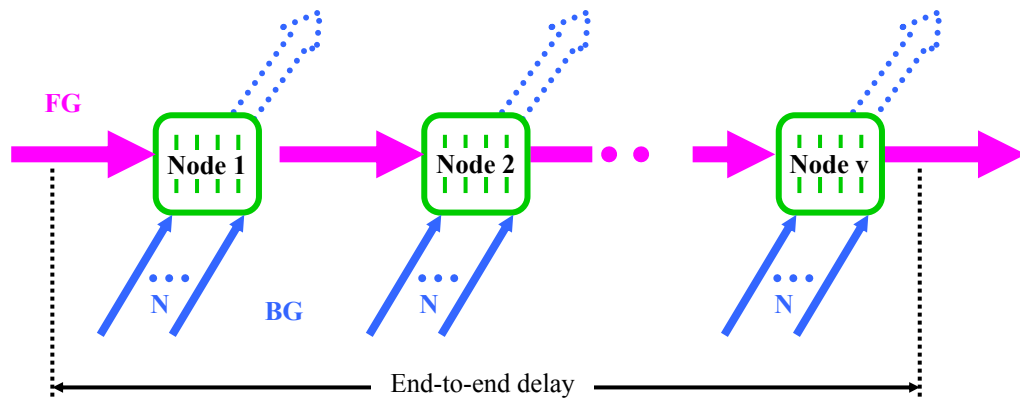
The chapter is divided into two major parts: in the first part, the focus is a simple end-to-end connection with independent and homogeneous nodes. Traffic fed into the connection is divided into two types: FG and BG. TA was applied to substitute for the BG traffic, effectively reducing many BG sources to only one. Both non-truncated and truncated Pareto traffic were simulated, and results show excellent agreement between the queueing behaviour of the original model and that of the TA substitution.

In the second part, WRR scheduling was introduced to the end-to-end connection. WRR schedules individual servers along the connection so that they transmit packets according to the weights of the traffic. Simulation results show that the presence of WRR introduces no loss of accuracy in TA, and yet the scheme works perfectly well as a BG traffic substitution, providing accurate results with considerable acceleration.

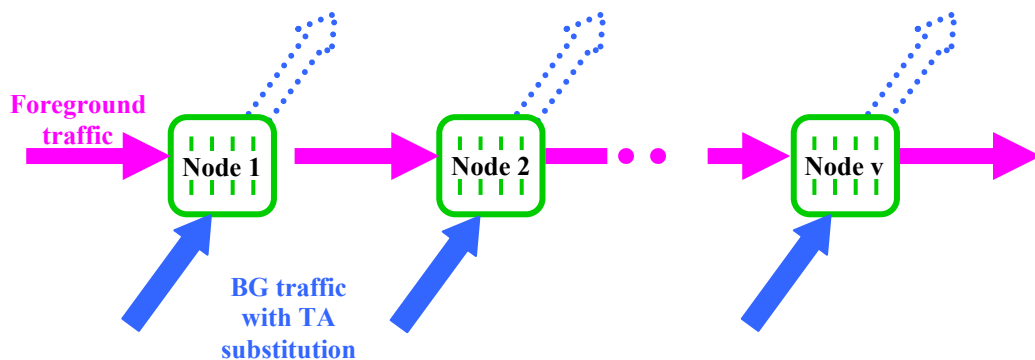
### 5.1 End-to-end network studies

An end-to-end connection with a number of independent and homogeneous nodes was deployed, each node acts as a buffer, in such a way as to try to reflect the way packet networks operate. The traffic is divided into two types: FG and BG [Schormans01]. The FG and BG assumption has been previously used for Markovian traffic, and has proven to be a

reliable modelling method. FG is the traffic flow of interest which passes every node along the path, whereas BG is inserted at individual nodes to reflect the ambient traffic. As soon as the BG traffic passes through the designated node, it is then removed from the system without interfering in any other nodes along the connection. Figure 5-1 illustrates the concept of an end-to-end link.



**Figure 5-1** An end-to-end connection with FG and BG traffic



**Figure 5-2** An end-to-end connection with TA BG traffic

TA was used to model the BG traffic so that the process of generating individual sources is abolished, and only a single traffic is used as a substitute. Figure 5-2 shows the concept of BG traffic replacement. With the aid of TA, only a single equivalent BG source is fed into all the network nodes. The replacement method first of all greatly simplifies the modelling

procedure, and secondly offers significant speedup, as fewer traffic sources are required. Thirdly, simulations of this kind are able to reach steady-state much earlier due to the fact that TA reduces the number of states of the traffic involved (see also Chapter 4). Lastly, the application of TA to end-to-end network scenarios is demonstrated.

Evaluations on the end-to-end connection consist of the following two parts, unless specified otherwise. Firstly, the validation of an equivalent TA for the BG traffic will be shown via a single node model. This is same as the examinations done in Chapter 4 in which the state probabilities,  $P[h]$ , were monitored. The idea is to provide verification for TA for a particular scenario. Secondly, the actual end-to-end connection is studied by referring to the delay time probability of FG packets, which is defined as:

**$O[h] = P(\text{a FG packet experiences a delay of } h \text{ timeslots across the end-to-end path})$**

This statistic directly reflects the effect of the BG component. It is expected that the burstier the BG traffic, the longer the delay experienced by the FG packets. Note that both FG and BG traffic are bursty traffic, Power-law traffic to be precise. Comparisons on  $O[h]$  of the end-to-end connection before and after the BG traffic replacement will be presented.

### 5.1.1 Deriving an equivalent TA model for BG traffic substitution

To derive an appropriate TA model for substituting for the BG traffic, a suitable  $C_{BG}$ , must be defined. This is because in the operation of FIFO buffers in the absence of scheduling mechanisms the service rate is shared between the FG and BG traffic. The amount of capacity used by BG traffic must be known to provide the parameters defining TA.

Given the parameters of the original non-accelerated end-to-end connection settings:

$D_{OFF(FG)}$	Mean OFF sojourn time of the foreground traffic
$D_{ON(FG)}$	Mean ON sojourn time of the foreground traffic
$N_{FG}$	Number of foreground traffic sources
$R_{FG}$	Rate of packet arrival of the foreground source
$D_{OFF(BG)}$	Mean OFF sojourn time of the background traffic
$D_{ON(BG)}$	Mean ON sojourn time of the background traffic
$N_{BG}$	Number of background traffic sources
$R_{BG}$	Rate of packet arrival of the background traffic
$C$	Overall service rate of the buffer in packets/timeslot

The distribution of C between FG and BG traffic is proportional to the amount of time that the server spends transmitting each type of traffic, which is determined by the total amount of traffic in each case. For instance, the amount of FG traffic,  $T_{FG}$ , is calculated by,

$$T_{FG} = \frac{D_{ON(FG)}}{D_{ON(FG)} + D_{OFF(FG)}} \cdot N_{FG} \cdot R_{FG} \quad \text{Eqn. 5-1}$$

and similarly, the amount of BG traffic,  $T_{BG}$ , is given by,

$$T_{BG} = \frac{D_{ON(BG)}}{D_{ON(BG)} + D_{OFF(BG)}} \cdot N_{BG} \cdot R_{BG} \quad \text{Eqn. 5-2}$$

Once the amount of FG and BG is known, it is then possible to calculate the proportion of the service rate for the BG case, with regard to the ratio of the amount of traffic. The corresponding service rate for the BG traffic,  $C_{BG}$  is

$$C_{BG} = \frac{T_{BG}}{T_{FG} + T_{BG}} \cdot C \quad \text{Eqn. 5-3}$$

After finding  $C_{BG}$ , there are then sufficient parameters to characterise a TA model to replace the BG traffic.

### 5.1.2 Non-truncated Pareto FG and BG traffic

In this example, both FG and BG are non-truncated Pareto sources, and the settings are given in Table 5-1. The overall capacity, C, at individual nodes is 9 packets/timeslot and there are four nodes along the connection. To replace the original Pareto BG sources, it is essential to calculate the portion of the capacity that is devoted to the BG traffic, so that an equivalent TA model can be defined.

Referring to Eqn. 5-1,  $T_{FG}$  is given by,

$$T_{FG} = \frac{2.5}{7.5 + 2.5} \cdot 1 \cdot 6 = 1.5$$

And  $T_{BG}$  is calculated via Eqn. 5-2,

$$T_{BG} = \frac{5}{10 + 5} \cdot 9 \cdot 1 = 3$$

Therefore,  $C_{BG}$  can be evaluated using Eqn. 5-3:

$$C_{BG} = \frac{3}{1.5 + 3} \cdot 9 = 6$$

Once  $C_{BG}$  is defined, a TA model can be derived to substitute for all the BG sources via the TA algorithm; see Table 5-1 for the details of the TA model obtained in this example.

		Original BG	TA BG traffic
<b>Foreground</b>	<b>D<sub>OFF(FG)</sub></b>	7.5	7.5
	<b>D<sub>ON(FG)</sub></b>	2.5	2.5
	<b>N<sub>FG</sub></b>	1	1
	<b>R<sub>FG</sub></b>	6	6
<b>Background</b>	<b>D<sub>OFF(BG)</sub></b>	10	3.1096
	<b>D<sub>ON(BG)</sub></b>	5	2.6862
	<b>N<sub>BG</sub></b>	9	1
	<b>R<sub>BG</sub></b>	1	7.1921
	<b>C<sub>BG</sub></b>	–	6

**Table 5-1 End-to-end studies with non-truncated Pareto FG and BG traffic**

Prior to the actual BG traffic replacement, the derived TA model was tested. The validation is the same as the method used in Chapter 4 in which a single node was tested, and the corresponding queueing statistic,  $P[h]$ , was monitored and compared. To achieve this, first of all a single node was set up according to the values of  $D_{OFF(BG)}$ ,  $D_{ON(BG)}$ ,  $N_{BG}$ ,  $R_{BG}$  and  $C_{BG}$ , and the associated state probabilities were collected. Then, the same network node was modified to the TA settings, and the same statistics were recorded. Comparisons were done on the two sets of queueing behaviour, illustrated in Figure 5-3. Excellent agreement is shown between the results of the model with original BG traffic settings and TA, which can be interpreted as the validation of TA for the BG sources.

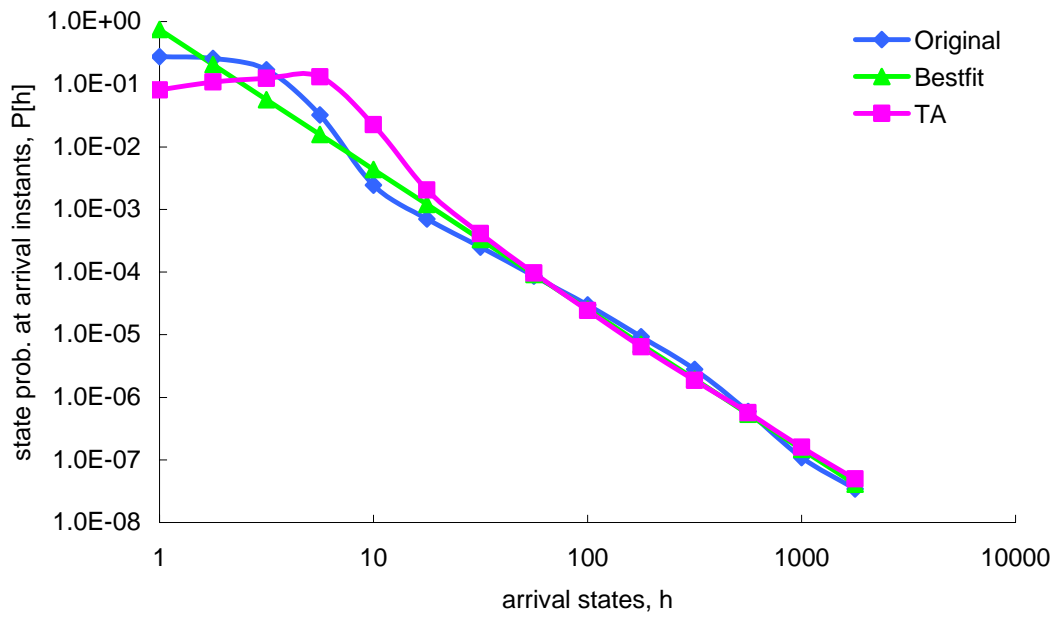


Figure 5-3 TA for non-truncated Pareto BG traffic

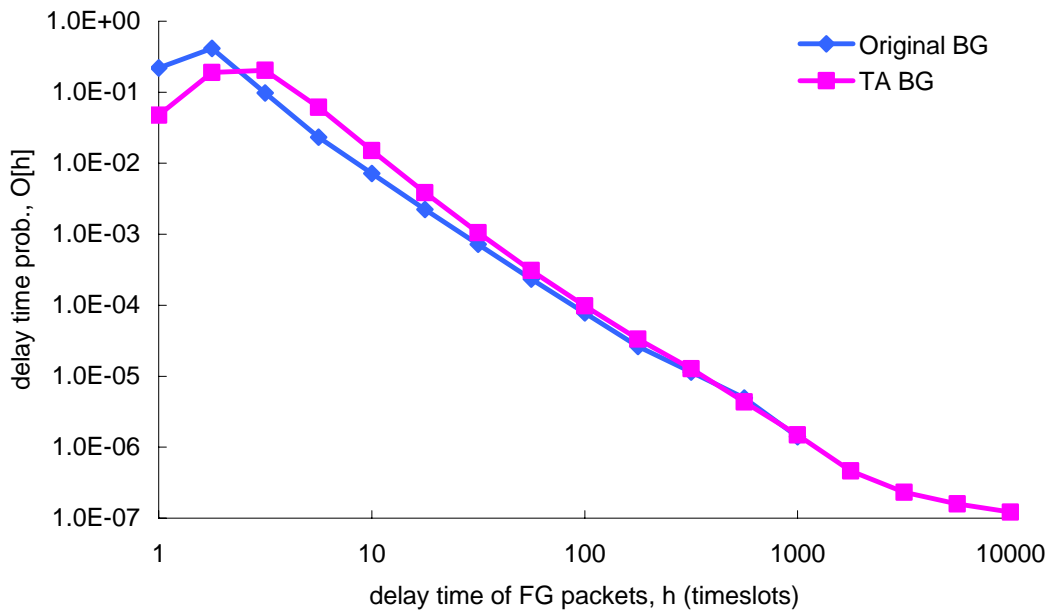


Figure 5-4 Validations for the example given in Table 5-1

The next step was to re-configure the end-to-end network settings for the replacement to take place. The associated TA model substitutes all BG sources at all nodes. Simulations were carried out for the end-to-end connection before and after the BG traffic replacement, and the delay time probability,  $O[h]$ , of the FG packets was monitored. Figure 5-4 presents the comparisons on the two sets of results, and excellent agreement is attained between the two scenarios.

### 5.1.3 Truncated Pareto FG and Pareto BG traffic

The end-to-end connection was also studied in the presence of truncated Pareto traffic. The configuration of the connection is similar to that of 5.1.2, however, both FG and BG sources are replaced by truncated Pareto sources. An example is presented in Table 5-2, and in this example, the overall capacity,  $C$ , at individual nodes is 5 packets/timeslot and the total number of nodes is 8. It is worth noting again that the values presented here are their actual values after truncation.

		Original BG	TA BG traffic
<b>Foreground</b>	<b>D<sub>OFF(FG)</sub></b>	10	10
	<b>D<sub>ON(FG)</sub></b>	4.1	4.1
	<b>L<sub>ON(FG)</sub></b>	1200	1200
	<b>N<sub>FG</sub></b>	1	1
	<b>R<sub>FG</sub></b>	2	2
<b>Background</b>	<b>D<sub>OFF(BG)</sub></b>	10	3.1895
	<b>D<sub>ON(BG)</sub></b>	4.1	2.5741
	<b>L<sub>ON(BG)</sub></b>	1200	1200
	<b>N<sub>BG</sub></b>	8	1
	<b>R<sub>BG</sub></b>	1	5.2087
	<b>C<sub>BG</sub></b>	–	4

Table 5-2 End-to-end studies with truncated Pareto FG and BG traffic

To attain an equivalent TA model for the substitution of truncated BG traffic, it is again required to define a capacity which corresponds to the BG traffic prior to the derivation of TA. With respect to Eqn. 5-1, Eqn. 5-2 and Eqn. 5-3, the following were obtained:

$$T_{FG} = \frac{4.1}{10 + 4.1} \cdot 1 \cdot 2 = 0.58156 \quad \text{and} \quad T_{BG} = \frac{4.1}{10 + 4.1} \cdot 8 \cdot 1 = 2.32624$$

therefore,

$$C_{BG} = \frac{2.32624}{2.32624 + 0.58156} \cdot 5 = 4$$

Figure 5-5 illustrates three sets of queueing results which were used to validate the truncated TA model for the BG traffic replacement in this example. The procedure is identical to that of 4.4.2.1, where a non-truncated TA model is required before a final truncated TA is defined. The first set of results originates from a single node scenario in which the truncated BG traffic was fed into, and the state probabilities were recorded. These queueing results were then used for further comparisons.

The second and third sets of results are associated with the non-truncated TA and the truncated TA for the BG traffic. In the former case, the non-truncated TA was tested using the single node setting. In the latter case, a limit of 1200 was introduced to create a truncated TA model for the BG traffic. Queueing behaviours of the two cases were compared with that of the truncated BG traffic, and excellent agreement was obtained.

Comparisons on the delay time probabilities,  $O[h]$ , of the end-to-end model with original and TA BG traffic can be found in Figure 5-6. The two sets of results match closely to each other, indicating a successful application of TA.

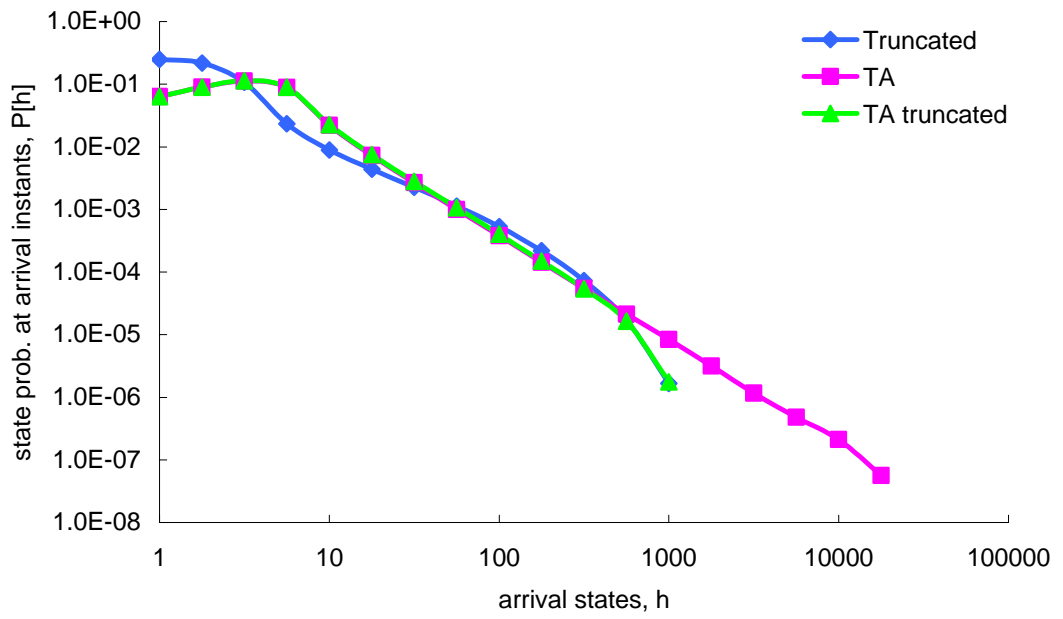


Figure 5-5 TA for truncated Pareto BG traffic

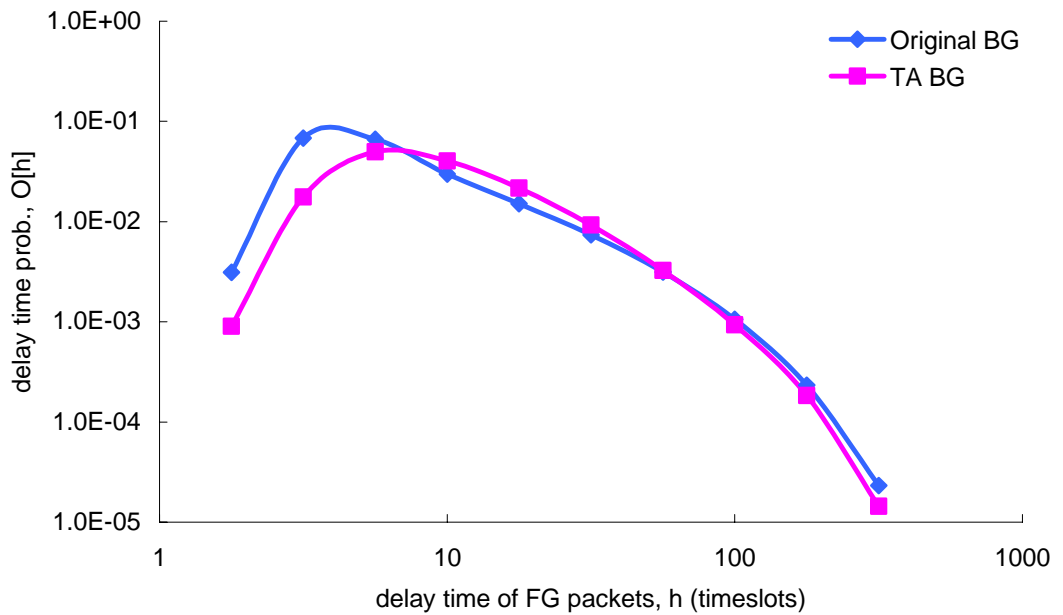


Figure 5-6 Validations for the example given in Table 5-2

## 5.2 End-to-end connections with WRR

The WRR scheduling mechanism was introduced to the end-to-end connection to further explore the applicability of TA. WRR is a queueing policy commonly adopted in many network scenarios e.g. [Ha98, Takahashi02]. Previous research on accelerated simulation has also used WRR as a testing platform [Stewart02]. It is important because future IP will employ Differentiated Services (DiffServ) which is WRR based.

### 5.2.1 Round Robin (RR) and Weighted Round Robin (WRR)

Unlike FIFO, RR serves packet queues in cyclical fashion. Packets are selected from each input stream and transmitted at fixed intervals. All packets are treated equally regardless of the number incoming connections or response time each server is experiencing.

WRR works in a similar fashion as RR, except the fact that the service time allocated to each queue is according to a weight value. Each flow or class of a connection has one queue, and individual queues are assigned a weight value which is determined by different properties of the traffic, e.g. the average traffic volume. Details of WRR can be found in [Keshav97].

In this research, the FG and BG traffic act as two different classes and WRR serves packets from these flows with reference to the ratio assigned. FG is no longer Pareto traffic, Poisson traffic [Pitts01] is used instead to intimate real-time traffic. For instance, in a real network, a single node is fed by different kinds of traffic, and these traffic can then be classified into two types: delay sensitive and loss sensitive. The former refers to real-time traffic, e.g. voice, in which Poisson traffic has been an excellent replica [Bear80]. The latter associates with data traffic in which any packet losses can result in meaningless data. Traffic originates from WWW, e-mails etc belongs to this class, which is often found to be Power-law e.g. [Crovella96]. As a result, by constructing an end-to-end path with Poisson traffic and Pareto traffic as FG and BG respectively, a general model that deals with different classes of traffic is provided.

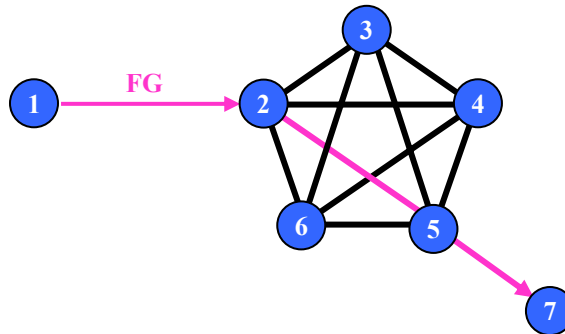
Poisson traffic is characterised by the rate of arrivals,  $\lambda$ , and packet arrivals follow:

$$\Pr\{z \text{ arrivals in time } T\} = \frac{(\lambda \cdot T)^z}{z!} \cdot e^{-\lambda \cdot T} \quad \text{Eqn. 5-4}$$

where  $z$  is the number of arrivals and  $T$  is the time of interest. With regard to the BG traffic, two versions were exploited: non-truncated Pareto and truncated Pareto.

### 5.2.2 Weights of FG and BG traffic

The weights of FG and BG traffic were determined by taking network topologies into account. An example of a network with a mesh connection is given in Figure 5-7. In that example, the mesh network is connected to two other nodes, 1 and 7, at nodes 2 and 5 respectively. Consider the FG traffic of interest propagates along the highlighted path, i.e.  $1 \rightarrow 2 \rightarrow 5 \rightarrow 7$ , and node 2 and 5 are the intermediate nodes along this FG path. The rest of the connections linked to these nodes can then be referred to as BG traffic. Therefore, in this specific example, the FG:BG ratio is 1:3.



**Figure 5-7** An example of a mesh network topology

The ratios used in this thesis for WRR studies lie between 1:4 to 1:5 (FG:BG), and these ratios were chosen by considering in most networks provided that network nodes are sparsely connected, the FG:BG ratio at individual nodes along a given network path would be approximately these ratios. In [Korkmaz00], examples on network topology are given in which FG and BG flows were assigned in terms of these ratios.

### 5.2.3 Poisson FG and non-truncated Pareto BG

An end-to-end connection where each node's buffer has the WRR scheduler was used in this part. Table 5-3 presents an example with Poisson FG traffic and non-truncated Pareto BG traffic. The overall capacity of individual nodes were assigned to be 6 packets/timeslot, and the allocation of this capacity to FG and BG traffic is 1:5 respectively. The connection consists of four network nodes.

The derivation of  $C_{BG}$  in this case is straightforward, and no longer needs to refer to the actual amount of traffic.  $C_{BG}$  can be simply calculated by referring to the assigned weights. For a particular FG:BG ratio, the portion of the capacity devoted to the BG traffic is given by:

$$C_{BG} = \frac{W_{BG}}{W_{FG} + W_{BG}} \cdot C \quad \text{Eqn. 5-5}$$

where  $W_{FG}$  and  $W_{BG}$  are the ratio of weight values for the FG and BG traffic respectively. And therefore,  $C_{BG}$  for this particular example is calculated to be 5 packets/timeslot.

Figure 5-8 illustrates the queueing results of the BG traffic and the equivalent TA model using a single node source overlapping scenario (as previous validations in Chapter 4). Yet again, TA demonstrates high accuracy. The actual results for end-to-end packet delays are shown in Figure 5-9. Evidently, the substitution of BG traffic with a corresponding TA model is highly accurate.

		Original BG	TA BG traffic
<b>Foreground</b>	<b>Poisson</b>	$\lambda = 0.97$	$\lambda = 0.97$
<b>Background</b>	<b>D<sub>OFF(BG)</sub></b>	8.1195	2.1769
	<b>D<sub>ON(BG)</sub></b>	3.8184	2.2543
	<b>N<sub>BG</sub></b>	10	1
	<b>R<sub>BG</sub></b>	1	6.2871
	<b>C<sub>BG</sub></b>	–	5

**Table 5-3 WRR studies with Poisson FG and Pareto BG traffic**

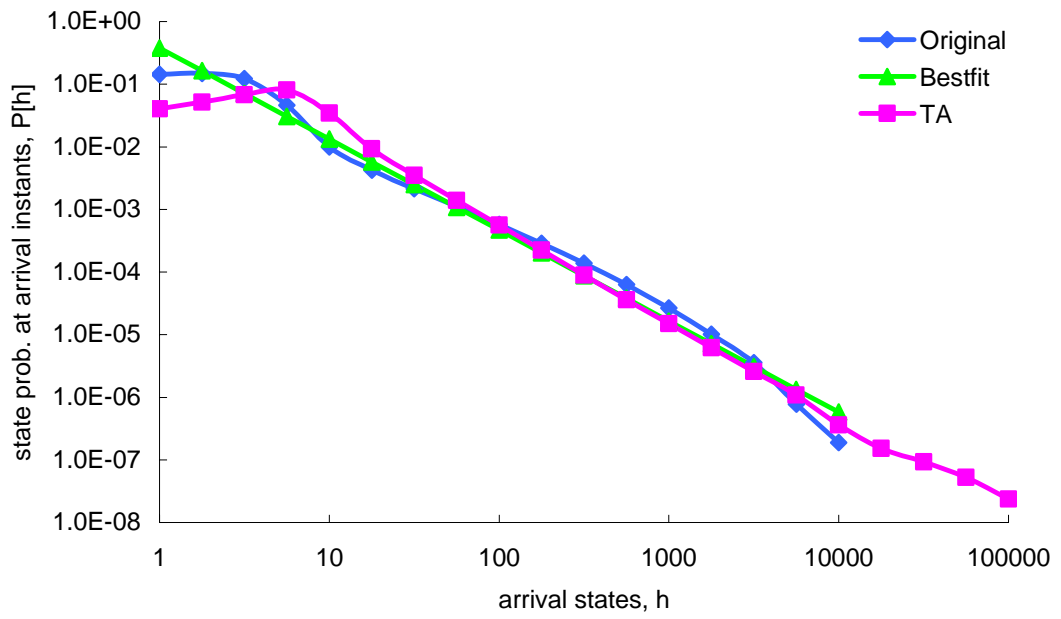


Figure 5-8 TA for the non-truncated Pareto BG traffic in the WRR example

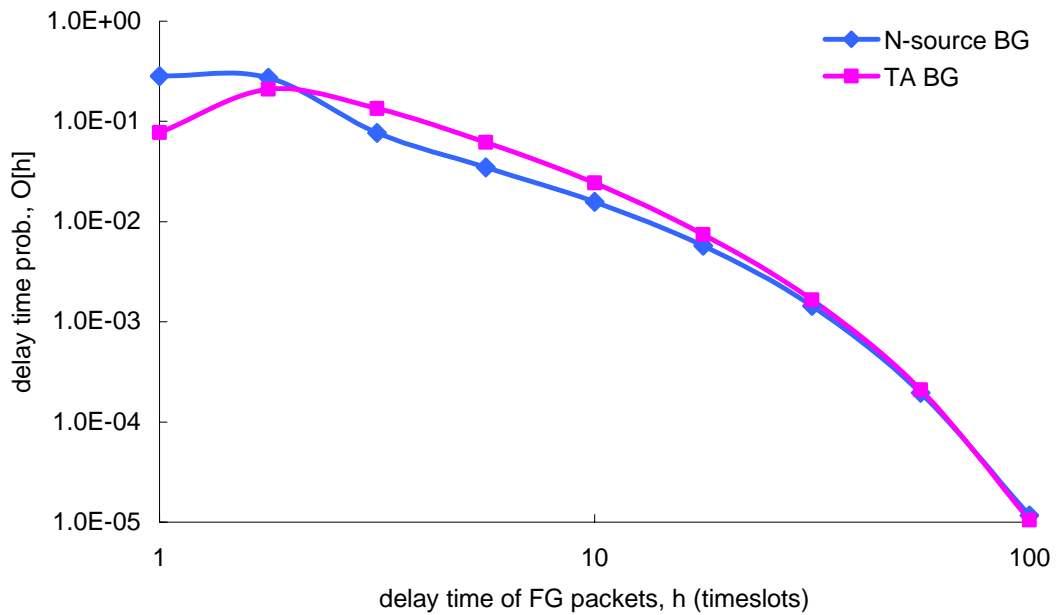


Figure 5-9 Validations for the WRR example given in Table 5-3

### 5.2.4 Poisson FG and truncated Pareto BG

This part is similar to 5.2.3, the only difference is that the Pareto BG traffic is in the truncated form. The BG traffic setting used in 5.1.3 was re-applied here in the presence of the WRR scheduler. The related TA validation can refer to the relevant section, and it is not included here to avoid repetition of the content. Table 5-4 lists the parameters exploited in this particular end-to-end study.

The overall link capacity is 5 packets/timeslot for each link and it is distributed to the FG and BG traffic in the ratio of 1:4 respectively. The total number of nodes along the connection is 6.

Simulations were performed for the end-to-end connection with the original BG traffic and TA BG traffic. Delay time probabilities of FG packets of the two scenarios were monitored and expressed in Figure 5-10. The accuracy of TA is once again clearly demonstrated.

		Original BG	TA BG traffic
<b>Foreground</b>	<b>Poisson</b>	$\lambda = 0.95$	$\lambda = 0.95$
<b>Background</b>	<b>D<sub>OFF(BG)</sub></b>	10	3.1895
	<b>D<sub>ON(BG)</sub></b>	4.1	2.5741
	<b>L<sub>ON(BG)</sub></b>	1200	1200
	<b>N<sub>BG</sub></b>	8	1
	<b>R<sub>BG</sub></b>	1	5.2087
	<b>C<sub>BG</sub></b>	–	4

**Table 5-4 WRR studies with Poisson FG and truncated Pareto BG traffic**

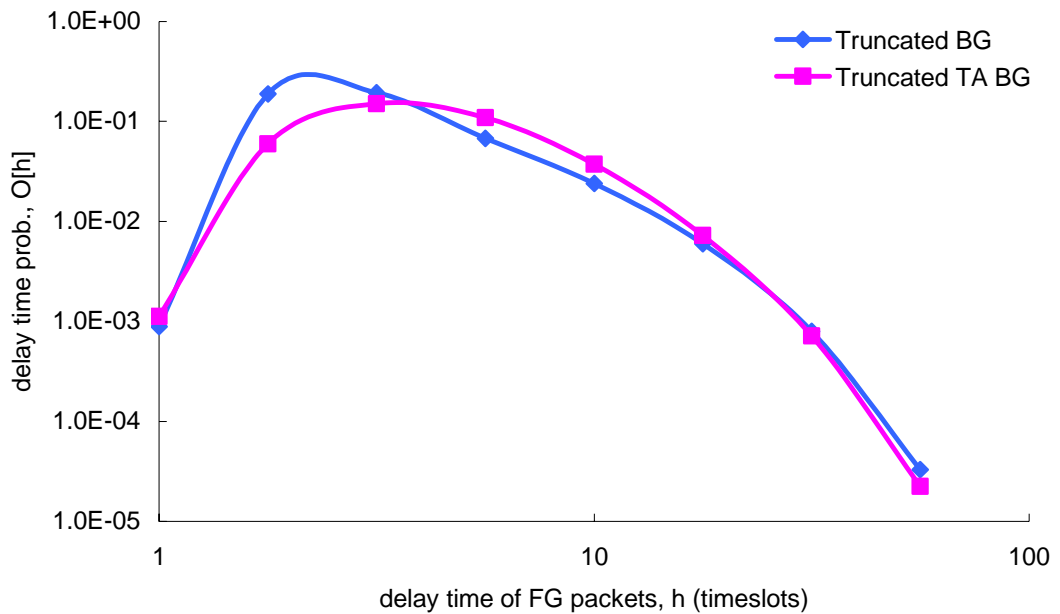


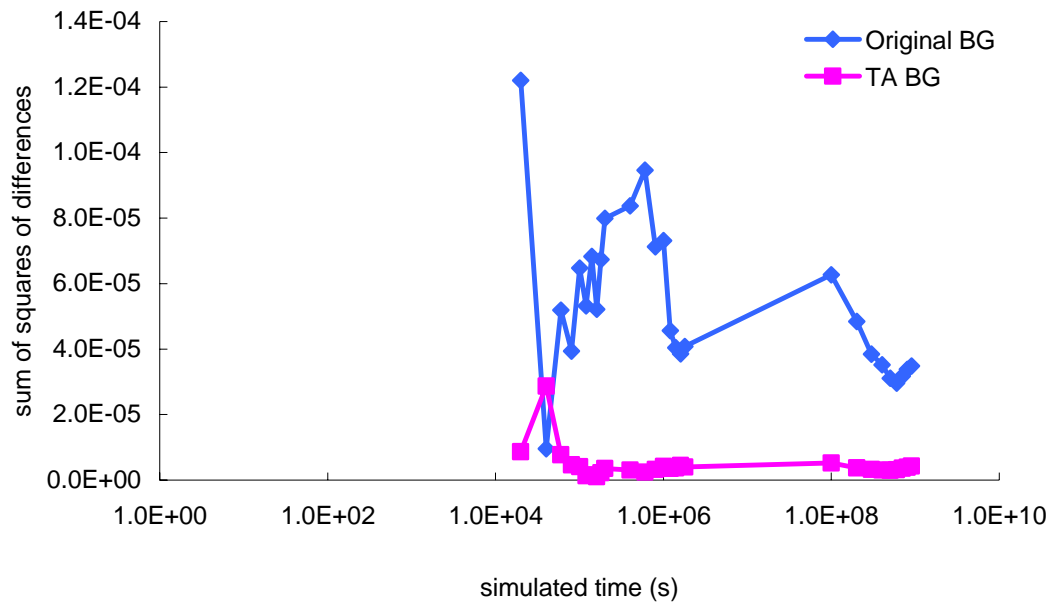
Figure 5-10 Delay time probabilities of the example in given in Table 5-4

### 5.3 Acceleration

Acceleration achieved by TA in the substitution of BG traffic is measured using the rate of reaching steady-state. Again, the sum of squares of differences method was applied.

The example introduced in 5.1.2 is used in this section to investigate the acceleration achieved by TA. A simulation was set up using the original BG traffic setting. Delay time probabilities were collected and analysed at different simulated times. These results were then compared against the corresponding steady-state results which were collected separately. The sum of squares of differences of these different stages were then plotted against their collection times. The same procedures were carried out for a simulation with TA BG traffic substitution.

Figure 5-11 illustrates the comparisons between the sum of squares of differences at various stages of the original and TA models. Once again, TA has shown its ability to reach steady-state much earlier than the original model. While the original model shows unstable behaviour even after  $1 \times 10^9$  seconds, TA reaches its stability at  $1 \times 10^6$  seconds, indicating a speedup by a factor of a 1000!



**Figure 5-11 Acceleration achieved by TA in an end-to-end path**

## 5.4 Summary

This chapter has reviewed the application of TA to end-to-end connections. The network path was fed by both FG and BG traffic, and TA was successfully substituted to replace the many BG sources. Evaluations on delay time probabilities of FG packets reveal the technique of BG traffic replacement is highly accurate, while significantly simplifying the modelling process.

In addition, the WRR scheduling mechanism was introduced to the connection. FG traffic was replaced by a Poisson source which mimics delay-sensitive traffic. The WRR scheduler distributes the available capacity according to the assigned ratio. Similarly, TA was applied as a replacement for the BG traffic. High accuracy was obtained with respect to queuing results.

In Chapter 6, the combination of TA and RESTART will be illustrated. First of all, the concept of RESTART will be discussed. Simulation results together with the acceleration provided by the combination will also be presented.

## Chapter 6

# TA in conjunction with RESTART

In this chapter, the conjunction of TA and the well-known rare event simulation technique – RESTART - is demonstrated. This signifies 1) TA is, unlike many other acceleration techniques, not mutually exclusive with other acceleration methods, and 2) the applicability of RESTART to Power-law traffic, which has not been previously done. The combination presented here is robust, efficient and effective, allowing buffer states with a low probability to be estimated within an affordable time. This provides a powerful tool for tackling prohibitively time-consuming LRD traffic simulations. RESTART studies in this thesis were done in Matlab.

### 6.1 Introduction

As many studies on networks rely on evaluating very low probabilities, it is necessary to further develop TA so that it is enhanced for rare events. TA allows a system to reach stability earlier and therefore makes a larger portion of the low probability statistics available. Nevertheless, it is beneficial to further extend the probability coverage. RESTART [Altamirano91, 94 & 02] was introduced in addition to TA to explore those buffer states with an extremely low probability, e.g. packet loss probability in the order of  $10^{-9}$ . Combining the two methods has several advantages in the evaluation of rare events. Firstly, “dual” speedup is obtained in the sense that simulations benefit from time reduction through the two acceleration schemes. Secondly, as network scenarios are greatly simplified by TA, RESTART can be exploited easily to achieve supplementary speedup without having to deal with complicated issues such as the state of individual traffic sources during retrials.

Two versions of RESTART were applied: with one threshold [Altamirano91] and with multiple thresholds [Altamirano94]. Results from various network scenarios have shown that

the combination of TA with RESTART is robust and accurate, and it provides great acceleration in the estimation of rare event probabilities.

## 6.2 RESTART

The fundamental concept of RESTART was introduced by A. J. Bayes in [Bayes70]. Manuel and José Villén-Altamirano invented a new version of the technique and named it RESTART [Altamirano91]. A full description of the method can be found in [Altamirano91, 94 & 02]. This section reviews only the fundamental concepts so that its use in conjunction with TA can be better understood.

RESTART allows the estimation of extremely low probabilities in a much shorter period of time than with a conventional simulation method. For a particular rare event, such an event tends to occur more frequently in a simulation if the starting point of the simulation is closer to the event of interest. In RESTART, appropriate starting points, referred to as thresholds, are defined, and at these points the system state is saved and re-entered a number of times to speed up the occurrence of the rare event. As a result, the system spends much more time in the region that actually contributes to the rare event probability. The time spent in the “high” probability states (with probability close to 1) is greatly reduced; hence acceleration is achieved.

The early version of RESTART uses a single threshold [Altamirano91]. Subsequently, the method was enhanced by the introduction of multiple thresholds, providing further acceleration in the estimation of rare events.

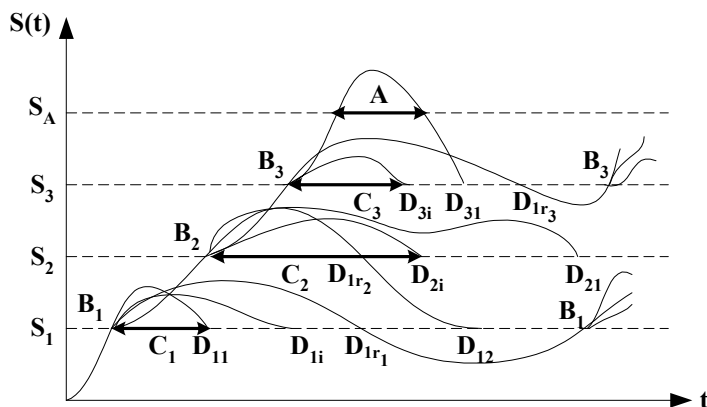


Figure 6-1 Concept of RESTART

The concept of a multi-threshold RESTART simulation is illustrated in Figure 6-1. Consider the probability of a rare event A is to be estimated, and it is associated with the system parameter S, which is required to be equal to or greater than  $S_A$ . Events  $C_i$  are defined and these events occur when the value of S is equal to or higher than the corresponding threshold value  $S_i$ . When a threshold is upcrossed (an upcrossing represents the point at which an event  $C_i$  occurs having  $\bar{C}_i$  occurred previously), an event  $B_i$  occurs and the system state is saved. When a threshold is downcrossed (transition from  $C_i$  to  $\bar{C}_i$ ), an event  $D_i$  takes place, and the system state of the last event  $B_i$  is restored and the interval between the two events is simulated again for  $r_i$  times (number of retrials). During one trial of level i, the system may upcross level i+1. Under this circumstance, the system completes all the retrials of level i+1 before returning to the trial of level i. In the case of an event  $D_{ir}$ , the simulation continues as normal without any retrials.

The probability of occurrences of rare event A is calculated using:

$$\hat{P}(A) = J_A \left( J \prod_{i=1}^{MT} r_i \right)^{-1} \quad \text{Eqn. 6-1}$$

where  $J_A$  is the number of event A in all retrials, J is the number of all events simulated in the first trial of each set of retrials and MT is the number of thresholds.

The way to define thresholds is based on certain statistics of the system (explained explicitly in [Altamirano91, 94 & 02]). In the case of a single threshold, define  $P = P[A]$ ,  $P_1 = P[C]$  and  $r = r_1$ . Using the assumptions in [Altamirano91], the formulas for the optimal value of  $P_1$  and r are approximately:

$$P_1 = \sqrt{P}; \quad r = \frac{1}{\sqrt{P}} \quad \text{Eqn. 6-2}$$

With regard to scenarios with multiple thresholds, some new notations are required:  $P_1 = P[C_1]$  and  $P_i = P[C_i/C_{i-1}]$  where  $(1 < i \leq MT)$ . The optimal values are obtained via [Altamirano94]:

$$P_i = e^{-2}; \quad r_i = e^2 \quad \text{MT such that } C_{MT} \supset A \supset C_{MT-1} \quad \text{Eqn. 6-3}$$

There are various implementations of RESTART, such as the fixed splitting (FS) and the fixed effort (FE) methods [Garvels98]. In this paper the former case, which is in the original

RESTART implementation was chosen. The analytical side of the technique is not considered here, as the focus is the collaboration of TA and RESTART. The following two areas were given special attention: 1) the accuracy provided by the combination and 2) the acceleration accomplished in the context of rare event simulations.

### 6.3 RESTART: in conjunction with TA

In both versions of RESTART, events A and  $C_i$  were defined with respect to the queue level inside the buffer. Packet arrival instants are defined as the reference points of the simulation, which can be referred to as the instants of interest in the context of RESTART. Previous studies of systems with Power-law inputs notably show that the asymptotic characteristic of buffer queues causes great difficulties in evaluating events with a low probability. With the aid of RESTART, it is able to further examine the buffer queue state towards the tail, covering even more “rare events” in the lower probability zone than using TA alone.

An overlapping Pareto ON-OFF traffic scenario was used in this particular study. The model has the following settings:  $D_{ON} = 5$ ,  $D_{OFF} = 10$ ,  $N = 10$ ,  $C = 5$  and  $R = 1$ . The corresponding TA model has  $T_{ON} = 2.9983$ ,  $T_{OFF} = 2.6740$ ,  $C_{TA} = 5$  and  $R_{ON} = 6.3061$ , derived from the TA algorithm. The comparison of the two models is illustrated in Figure 6-2.

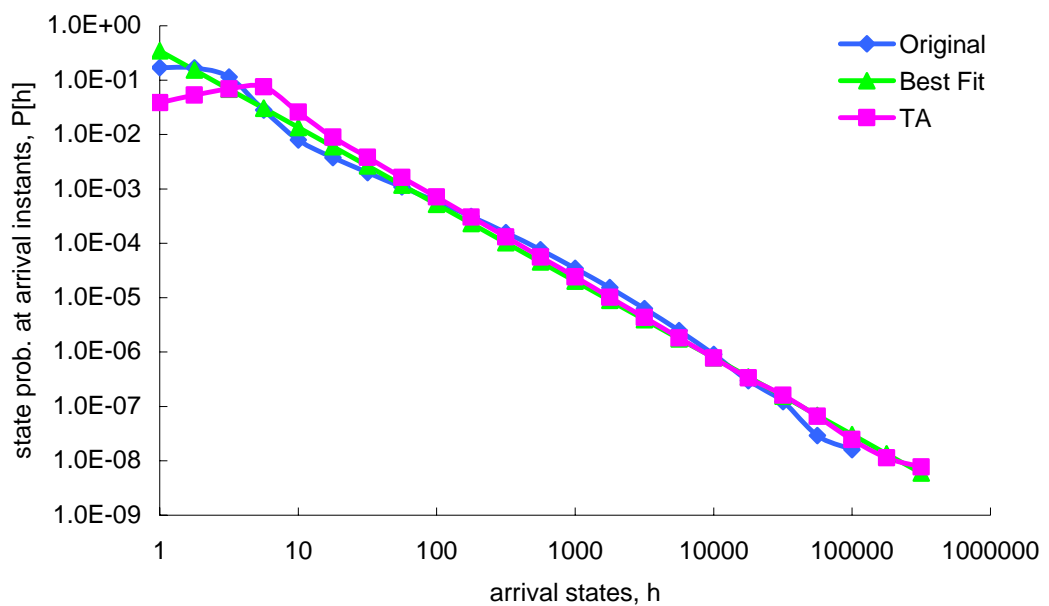


Figure 6-2 Validation of TA for the RESTART example

The two versions of RESTART, with a single-threshold and multiple thresholds, were applied in addition to TA for rare event enhancement. The next two subsections describe the way RESTART was applied to the above example. A new statistic,  $Q[h]$ , was introduced for comparing the queue behaviour of individual models.  $Q[h]$  is defined as:

$$Q[h] = \text{Pr}(\text{no. of packets in the queue at a packet arrival instant} \geq h)$$

One of the main reasons that this specific statistic was chosen was because the slow-decaying property of a Power-law queue is very well demonstrated. Referring to Figure 6-3 and Figure 6-4 (actual RESTART results), even with the aid of TA and RESTART, the lowest value of  $Q[h]$  of the model is only in the order of  $10^{-2}$ .

In addition, the results presented in this chapter also demonstrate the degree of stability of individual models given the same number of generated events. With the same number of events, the fluctuation in queueing behaviour varies in cases with and without acceleration techniques. Observations show that a model simulated using TA and RESTART reaches the steady status much earlier. Further details on the system's stability are included in the discussion section.

In order to quantify the amount of acceleration provided by individual techniques,  $500 \times 10^6$  arrival packets (can be interpreted as the number of events) were generated in all original, original + RESTART, TA and TA + RESTART simulations.

### **6.3.1 Simulation approach and comparison with analysis**

In this part of the research, acceleration is measured in terms of extension further down the queue tail, which the degree of stability of the results can be investigated. The terminating simulation approach was used in the sense that RESTART related simulations were only carried out up to a specified number of events. Simulations were assigned to generate a fixed number of events (the number of packet arrivals in this case), and queueing results were collected and compared. These terminating simulations were aimed to demonstrate the rate that individual schemes allow a system to reach stability. The closeness of these results to steady-state is determined by comparing against an analysis [Ma02c], and in that paper, a hybrid technique was presented for the evaluation of queue statistics when buffering multiple Pareto ON-OFF sources. By taking direct measurements of the resultant traffic and combining with large deviation analytical techniques, it is feasible to predict the queueing

behaviour at a buffer. To obtain an analytical solution for the example used here, measurements were taken from a number of different runs of the original N-source simulation, and the mean value of individual collected statistics were used to calculate the required analytical results. To avoid interrupting the flow of this chapter, full details on the derivation of the analytical solution are included in Appendix 2.

The next two sections cover the actual research carried out on the topic of RESTART. Both RESTART with a single threshold and multiple thresholds are discussed. However, comparisons on the two approaches are not taken into account as the concentration is on the collaboration between RESTART and TA. In individual RESTART studies, the main aim is to locate the best combination in terms of accuracy and speedup for TA and the corresponding version of RESTART.

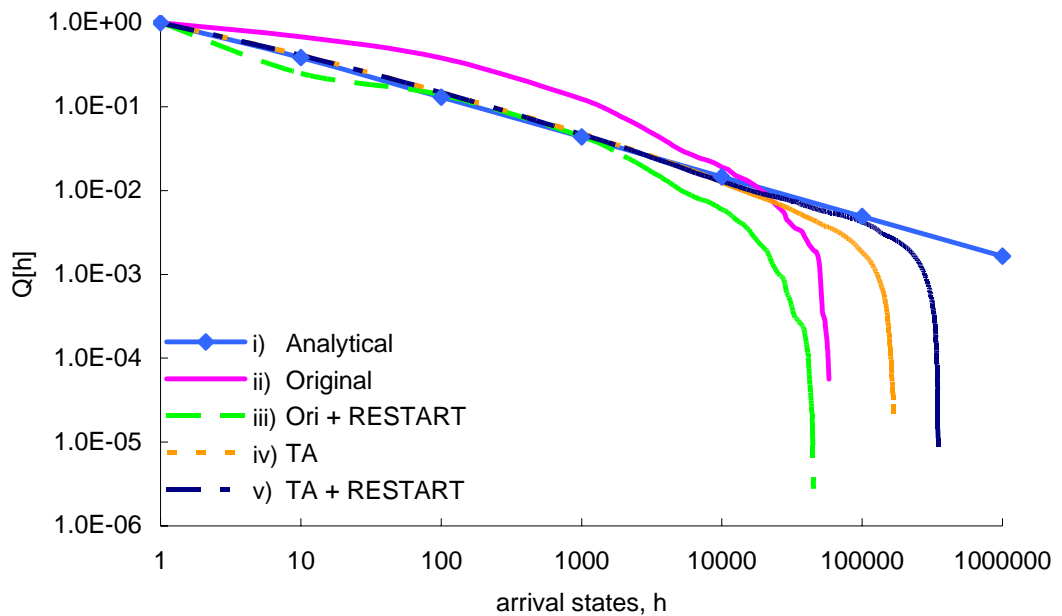
### **6.3.2 TA + RESTART with a single threshold**

Defining a threshold in the case of Power-law traffic requires more than just applying the equation (Eqn. 6-2). This is due to the asymptotical nature of Power-law queues. For instance, if a rare event in the probability region of  $10^{-2}$  (i.e.  $h = 10^5$ ) is to be estimated, and according to the equation,  $P_1$  should be assigned as  $10^{-1}$ , which gives a threshold of  $10^3$ . However, when referring to the actual queue there is in fact a huge “distance” between the threshold and the rare event itself. Even with the aid of RESTART, the rare event is hardly ever reached.

Several attempts were done on RESTART with a selection of threshold values. The best experimental results obtained was with a threshold of 35000, which has a probability of  $1.34 \times 10^{-7}$ , illustrated in Figure 6-3. There are five sets of results presented in the figure: i) analytical solution for the given example, ii)  $Q[h]$  obtained from the original model of the example, iii)  $Q[h]$  from the original model with the application of RESTART, iv)  $Q[h]$  of the TA model and v)  $Q[h]$  from the TA model with the application of RESTART.

First of all, by comparing results from ii) with the analysis i), it is clearly shown that the stability of the original model is far from ideal, indicating that a much longer simulation duration is mandatory. However, promising results were obtained from TA and the TA + RESTART combination, which both agree with the analytical solution. TA results are highly stable and the tail of the queue extends much further. In addition, with the aid of RESTART, further extension is achieved while maintaining the accuracy.

To study the sole effect of RESTART on the example (the original model without TA), RESTART has been applied to the model with the same threshold, i.e. 35000. The associated simulation results are also plotted in Figure 6-3. Although there is no sign of tail extension in the presence of RESTART, the queueing results do show a better agreement with the analytical solution, which reflects a high stability.



**Figure 6-3 TA + RESTART with a single threshold**

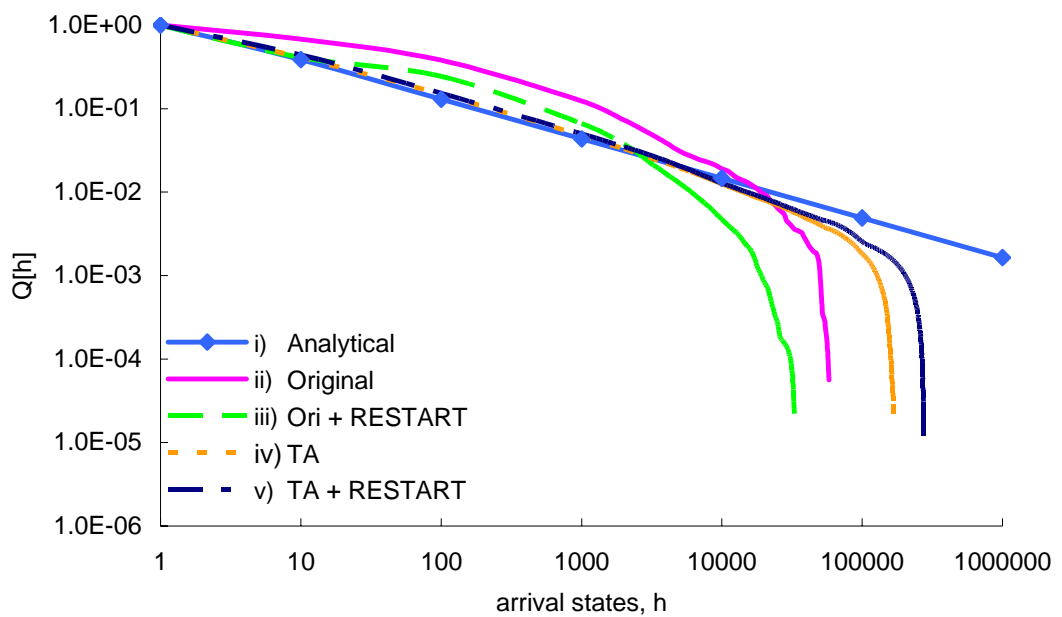
### 6.3.3 TA + RESTART with multiple thresholds

Similar to the single threshold scenario, difficulties are experienced when defining thresholds on the multiplexed queue length. Only a few thresholds could be specified on the queue length due to the very small difference between individual state probabilities of a Power-law queue. When Eqn. 6-3 was applied, as least as two thresholds were assigned with the last threshold equals to 4741. Again, due to the great difference between these thresholds and the targeted rare event, the obtained queueing output was far from satisfactory.

Many different values of  $P_i$  were tested and experimental results show that the operation of RESTART is negligibly affected by the choice of thresholds, but some choices of thresholds do give better results than the others in the case of Pareto traffic.

Figure 6-4 presents one set of results for RESTART with multiple thresholds.  $P_i = e^{-1.2}$  and four thresholds were defined with the last threshold of a queue length of 21528 with a probability of  $3.34 \times 10^{-7}$ . The number of retries is calculated by  $r_i = 1/P_i$ .

Similarly, five sets of results are obtained and illustrated in Figure 6-4, with i), ii) and iv) identical to those in 6.3.2. Referring to the results from v), i.e. TA + RESTART, once again, tail extension is achieved with excellent agreement with the analysis. This version of RESTART was applied directly to the original model with the absence of TA, see iii). Limited improvement is shown with reference to the queuing result, which could be due to numerous reasons, such as the choice of the thresholds. Nevertheless, it is more important in this research to show that with the aid of TA, the application of RESTART has been greatly simplified with significant acceleration provided.



**Figure 6-4 TA + RESTART with multiple thresholds**

## 6.4 Acceleration

As the study of rare events is the major interest in this chapter, the acceleration considered here is quantified by the additional amount of the statistics further down the queue tail provided by the combination of TA and RESTART.

Wall-clock times of each simulation is not useful for examining speedup because the same number of packets were generated in all simulations, which implies that the same number of events were produced in each case. Therefore, the difference in the actual wall-clock time taken for different simulations is insignificant.

A summary is presented in Table 6-1. Three categories were monitored for specifying the amount of acceleration: state coverage, P[h] and Q[h]. The state coverage concerns the number of states that is covered in the output queueing results. The range of the coverage ranges from minimum buffer state, 0, to the maximum queue size. The maximum queue size is a mean value which is obtained by averaging the maximum queue state obtained from a large number of runs of the same simulation with different seed values. P[h] and Q[h] are previously defined and these statistics were used to investigate the extension of individual probabilities into the lower occurrence zone.

It is clearly shown that state probability is significantly extended with the application of TA. Furthermore, the combination of TA and RESTART is robust, providing statistics of queue states with a very low probability.

	<b>Original simulation</b>	<b>TA</b>	<b>TA + RESTART single threshold</b>	<b>TA + RESTART multiple thresholds</b>
<b>No. of packets</b>	$500 \times 10^6$	$500 \times 10^6$	$500 \times 10^6$	$500 \times 10^6$
<b>State coverage</b>	$\sim 10^4$	$> 10^5$	$> 3 \times 10^5$	$> 3 \times 10^5$
<b>P[h]</b>	$10^{-6}$	$10^{-8}$	$10^{-9}$	$10^{-9}$
<b>Q[h]</b>	Not stable	$10^{-1.5}$	$10^{-2}$	$10^{-2}$

**Table 6-1 Summary of the acceleration provided by TA and TA + RESTART with respect to state probability**

## 6.5 Summary

The feasibility of RESTART in conjunction with TA was examined in this chapter. The simulation results evidently demonstrate that the application of RESTART enables more rare events to be estimated while maintaining the high accuracy.

Furthermore, these results can be used as an index of the rate at which individual simulation schemes achieve stability. While different studies were simulated for the same number of events, those with the application of TA and RESTART provide much more stable queueing outputs when comparing to that of the original non-accelerated simulations. Unlike the validation between the original and TA model, in which steady-state simulation were performed (Figure 6-2), RESTART related simulations were only carried out up to a specified number of events. In other words, the terminating approach was used in this part. By terminating individual simulations at the same specific point, it is able to observe the difference in stability in the overall queueing behaviour when different schemes were applied. Referring back to Figure 6-3 and Figure 6-4, it is obvious that the original non-accelerated case requires a much longer simulation length before the system reaches steady-state.

This combination of TA and RESTART signifies two important aspects: 1) the feasibility of combining TA with other existing speedup methods, which is a great advantageous and beneficial in the sense that “double” acceleration can be obtained in particular in the exploration of “rare-event” statistics, and 2) the recognition of RESTART in modelling Power-law traffic scenarios with asymptotical queueing behaviour.

In conclusions, the combination of TA and RESTART that presented in this thesis is robust, accurate and has the ability in estimating “low” probabilities, providing a very useful tool in the context of Power-law traffic modelling.

In the next chapter, the aggregation concept will be extended to end-to-end links where many network nodes are aggregated into a single node, and hence offers acceleration. A hybrid technique for predicting end-to-end delay probabilities will also be presented.

## Chapter 7

# Studies on End-to-end Paths

This chapter covers two areas with regard to end-to-end paths: firstly, an overview of Node Aggregation (NA) of an end-to-end path is provided. The method for this involves reducing many nodes of a network path down to a single node, with “transformed” FG traffic. Acceleration is achieved by omitting all but one of the nodes, significantly cutting down on the simulation overheads. Secondly, a hybrid technique for predicting end-to-end delay probabilities is developed. This technique analyses the delay statistics of the first few nodes of the path, and uses them to predict the delay probabilities of the nodes further down the path. Opnet was used to develop the techniques in this chapter.

### 7.1 Introduction

The interest in end-to-end performance evaluations lies on the deployment of real-time applications in future multi-service networks, e.g. VoIP [Moffaert01]. Real time applications are delay sensitive and are susceptible to jitter, resulting in the failure to fulfil QoS requirements. IP networks are based on statistical multiplexing, and jitter is introduced mainly due to queueing in routers. Jitter will multiply along the path from the sender to the receiver, explaining the necessity of some kind of a de-jitter buffer at the receiver site to compensate for these variations. Statistics with regard to end-to-end delays are therefore an important parameter for several reasons [Bovy02, Osterbo03]: firstly, these statistics enable the dimensioning of de-jitter buffers, helping to improve the overall QoS of a network. Secondly, they provide upper bounds on the total network delay for particular services, and it is among the most important QoS parameters in networks deploying statistical multiplexing, as the “worst” case can be anticipated. Lastly, the underlying properties of the current Internet can be examined, in particular the topology and traffic patterns.

Fast and accurate end-to-end delay predictions are desirable as simulations of a network path with a large number of nodes lead to excessive time consumption. The concept of NA is presented in this chapter, which involves aggregating many nodes along a network connection. Figure 7-1 demonstrates the concept of NA. The original network path consists of  $v$  nodes and both FG and BG traffic are Pareto distributed ON-OFF traffic with predefined parameters. NA aims to reduce the number of nodes to one only by using a transformed FG traffic. The focus is the estimation of the end-to-end delay of FG packets.

Figure 7-2 demonstrates the queueing statistics at various points along the end-to-end path. It is clearly shown that as the number of nodes increases, the delay experienced by the FG packets also increases. Provided the characteristics of the FG traffic that enters the node of interest, and the overall delay information are established, the corresponding “shape” of the queue can be regenerated. Therefore, the entire end-to-end path can be modelled by a single-buffer configuration.

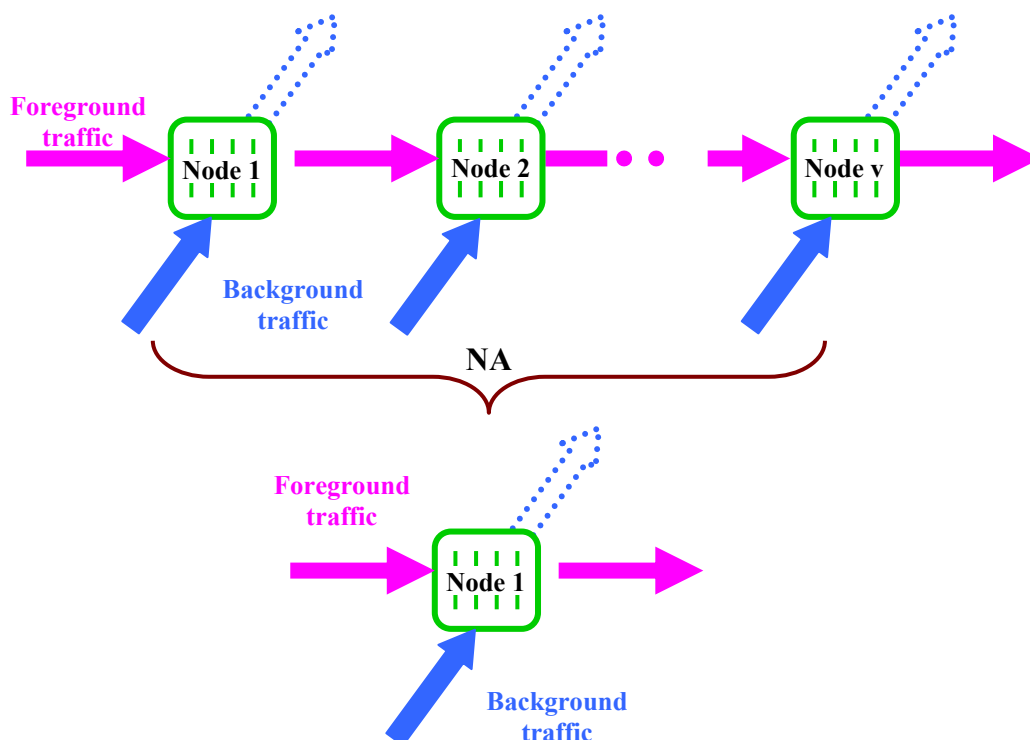
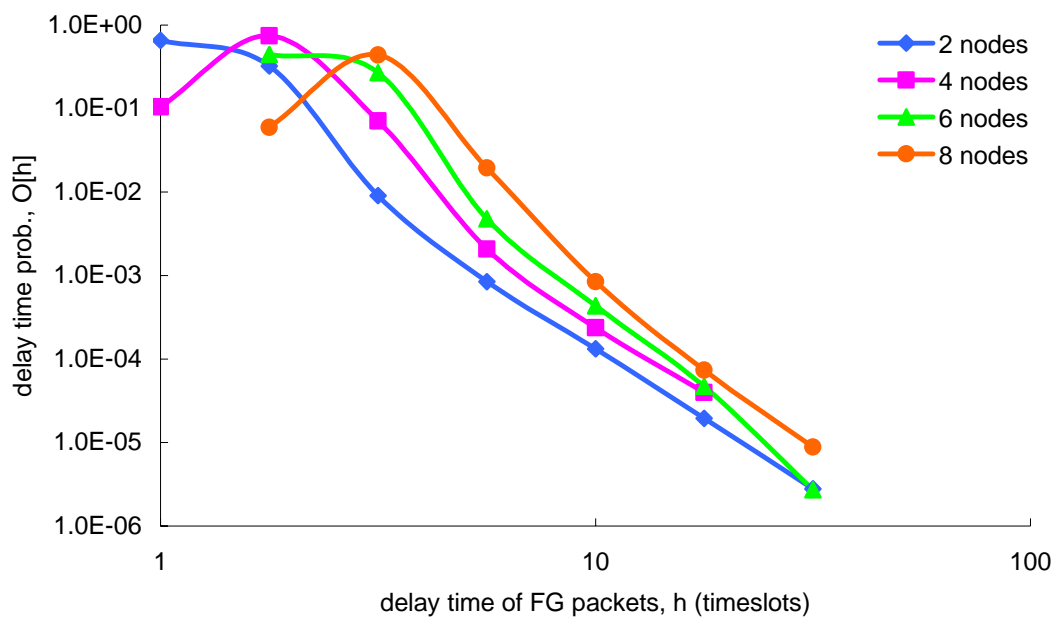


Figure 7-1 Concept of NA

In addition, this chapter also covers a hybrid technique which is used to evaluate the performance of an end-to-end path. The entire path is fed with a uniform load; in other words, the same BG traffic is inserted to all nodes. This is an interesting scenario because there is not an influential component, i.e. all BG sources have the same H parameter. Generally, for a given end-to-end path with non-uniform Power-law loads at different nodes, queueing statistics are always dominated by the one with the highest H value. Observations show that queue state probability distributions decay at individual nodes along the connection are in a similar fashion, and the difference among them is that as the number of nodes increases, the whole delay distribution shifts along the x-axis while the DR remains the same. Therefore, by knowing the decay for the first few nodes of the path, the queueing behaviour for the rest of the network buffer can be predicted by reference to the amount of shifting involved.



**Figure 7-2 Comparisons on the delay time distribution with increasing the number of nodes**

## 7.2 Node Aggregation (NA)

Given an end-to-end connection with homogeneous nodes with FG traffic fed at the first node, i.e. the sender. Such traffic enters the first node, and after buffering with BG traffic, FG packets exit the first node and enter the second node, and so on. Due to the queueing process FG traffic is transformed every time it propagates through a network node, in the sense that the original mean sojourn times are altered by the time the FG traffic enters the consecutive node. By the end of the path, the FG traffic would be statistically quite different from its original “shape”. Furthermore, each FG packet would have experienced a certain amount of delay prior to its entry to the last node. In order to carry out NA, the transformed sojourn times and the mean delay experienced need to be found.

### 7.2.1 Parameterization of the transformed FG traffic

The mean sojourn times of the FG traffic are taken from the entry point of the last node of the network path. For a path with  $v$  nodes, the sojourn times are taken between the exit of node  $(v-1)$  and the entry of node  $v$ . These two parameters are crucial as any changes in the sojourn times correspond to changes in traffic characteristics. Only precise values of these parameters can lead to accurate characterization of the transformed FG traffic.

Despite these measurements, the mean arrival rate of the transformed FG traffic at the last node needs to be calculated to complete the parameterization of the transformed FG traffic. This is due to the fact that the variation in the sojourn times, and the mean arrival rate needs to be adjusted accordingly to maintain the overall utilization. Studies here all consist of buffers with infinite capacity, and therefore, it can be assumed the absence of packet loss. This means that the number of FG packets is conserved at every node entry point of the path, leading to the following:

$$T_{FG} = \frac{D_{ON(FG)i}}{D_{ON(FG)i} + D_{OFF(FG)i}} \cdot R_{(FG)i} \quad \text{Eqn. 7-1}$$

where  $T_{FG}$  is the amount FG traffic (also used in Chapter 5) and  $i$  is an index which defines the corresponding entry node. As  $T_{FG}$  remains as a constant throughout,  $T_{FG}$  is the same for the first node and the last node, giving:

$$\frac{D_{ON(FG)1}}{D_{ON(FG)1} + D_{OFF(FG)1}} \cdot R_{(FG)1} = \frac{D_{ON(FG)v}}{D_{ON(FG)v} + D_{OFF(FG)v}} \cdot R_{(FG)v}$$

$$R_{(FG)v} = \frac{D_{ON(FG)1} \cdot R_{(FG)1} \cdot (D_{ON(FG)v} + D_{OFF(FG)v})}{D_{ON(FG)v} \cdot (D_{ON(FG)1} + D_{OFF(FG)1})} \quad \text{Eqn. 7-2}$$

The value of  $R_{(FG)v}$  is needed for NA to be carried out, which value is given by:  $D_{ON(FG)1}$ ,  $D_{OFF(FG)1}$  and  $R_{(FG)1}$  represent the initial FG traffic parameters, which “1” can be omitted and the expression can be simplified as:

$$R_{(FG)v} = \frac{D_{ON(FG)} \cdot R_{(FG)} \cdot (D_{ON(FG)v} + D_{OFF(FG)v})}{D_{ON(FG)v} \cdot (D_{ON(FG)} + D_{OFF(FG)})} \quad \text{Eqn. 7-3}$$

With the measurements of  $D_{(FG)v}$  and  $D_{(BG)v}$  together with Eqn. 7-3, the value of  $R_{(FG)v}$  can be calculated, and therefore the transformed FG traffic can be characterised.

### 7.2.2 Mean of the total delay

The mean of the total delay experienced prior to the entry of the last node is required. If there are  $v$  nodes, the mean of the total delay equals the average of the total delay experienced by FG packets from node 1 to  $(v-1)$ , denoted by  $\theta_{(v-1)}$ . This delay information is then included in the evaluation of the delay of the FG packets over the entire end-to-end path.

Consider an end-to-end path with  $\theta_{(v-1)}$  evaluated. To accurately mimic the overall end-to-end delay distribution, delays occurring in the previous nodes need to be captured. These delays can be described by the Pareto distribution, as delay distributions follow Power-law decays. This is achieved by taking  $\theta_{(v-1)}$  as the mean of the distribution. Whenever there is a “transformed” FG packet entering the NA node, an additional delay is drawn from this distribution and introduced to the packet. The overall delay of a FG packet is equal to the sum of the additional delay and the actual queueing delay occurs inside the last node. And it is the overall delay of a FG packet used in the evaluation of the delay time probabilities,  $O[h]$ , refers to Chapter 5 for definition.

### 7.2.3 Numerical example

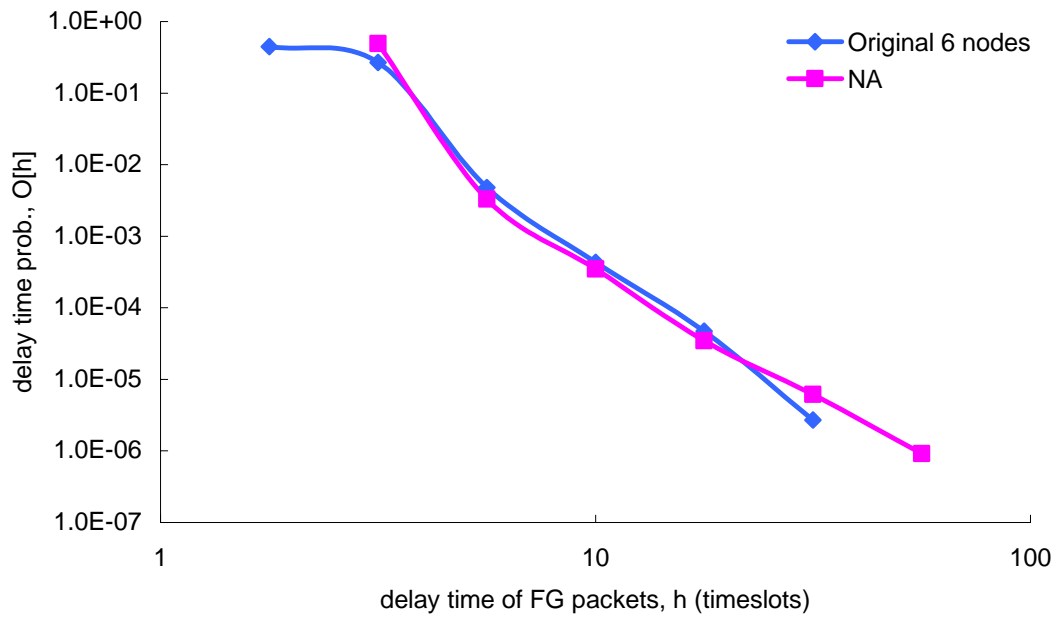
Table 7-1 presents a numerical example of an end-to-end path which consists of six nodes, and the overall  $C$  on each link is 6 packets/timeslot. Two simulation studies were carried out: an original model and a NA model. The former is the original simulation of the end-to-end path in which all six nodes are included. The given FG traffic is fed into the first node and it

propagates along the entire path, and the BG traffic is inserted at all nodes. The latter is the NA model in which only a single node is simulated. The required measurements are listed in the table and they were used as the parameters of the transformed FG traffic. The same BG traffic, i.e. the original BG traffic of the end-to-end path, was fed into the NA node. The measured  $\theta_{(v-1)}$ , i.e.  $\theta_5$ , was 1.4188 timeslots.

At the entry of node i		i = 1	i = 6
<b>Foreground</b>	<b>D<sub>OFF(FG)i</sub></b>	10	13.1474
	<b>D<sub>ON(FG)i</sub></b>	3	8.9153
	<b>N<sub>(FG)i</sub></b>	1	1
	<b>R<sub>(FG)i</sub></b>	2	1.1422
<b>Background</b>	<b>D<sub>OFF(BG)</sub></b>	10	10
	<b>D<sub>ON(BG)</sub></b>	2.5	2.5
	<b>N<sub>(BG)</sub></b>	10	10
	<b>R<sub>(BG)</sub></b>	1	1

**Table 7-1 Parameters of the NA example and corresponding measurements prior to the entry of the last node (node 6)**

Comparisons of the two models were in terms of the delay time probability of FG packets,  $O[h]$ . Results are illustrated in Figure 7-3, and it is clearly shown that excellent agreement is obtained between  $O[h]$  of the two models, indicating that the NA technique is highly accurate.



**Figure 7-3 Validation of NA**

Acceleration achieved by NA is substantial, and the higher the number of network nodes in an end-to-end path, the greater the acceleration. This is because the run-time of a simulation increases linearly as the number of nodes increases due to the multiplication of overheads. On the other hand, the run-time of NA changes trivially with the increment on the number of nodes. For the given numerical example, acceleration provided by NA is expressed in Table 7-2. All examples were simulated for the same length, a period of  $50 \times 10^6$  timeslots. While a substantial amount of time is consumed with the original simulation, NA offers a huge reduction in the simulation run-time. In addition, the combination of NA and TA provides further speedup, allowing simulations to be carried out even faster.

	<b>Original Path</b>	<b>NA</b>	<b>TA + NA</b>
<b>No. of nodes</b>	6	1	1
<b>No. of BG sources</b>	10	10	1
<b>Duration (timeslots)</b>	$50 \times 10^6$	$50 \times 10^6$	$50 \times 10^6$
<b>Wall-clock time (hours)</b>	150	15	7

**Table 7-2 Acceleration achieved by NA and TA+NA, with respect to wall-clock time**

### 7.3 A hybrid method for the prediction of end-to-end delays

The concept of NA is based on the fact that the distribution of queue delay probabilities shifts along the x-axis as the number of nodes increases (see Figure 7-2). A hybrid technique was developed by reference to the amount of shifting involved every time an additional node is included, and a relationship is then derived to describe this shifting phenomenon. Results from simulations show that the way that the delay queue shifts follow a trend, and if knowledge of this trend is obtained, queues at any node of a given end-to-end path can be estimated. This is achieved by collecting delay statistics from the first few nodes of the path, and relates the amount of shifting to the number of nodes.

Consider an end-to-end path that consists of  $v$  nodes. FG and BG traffic are fed into the path, see Table 7-3 for their corresponding parameters. The overall link capacity is 5 packets/timeslot for each link. The end-to-end path was simulated with the same traffic settings for different numbers of network nodes, and in this particular example only the first four nodes were modelled. Delay time probabilities of FG packets,  $O[h]$ , of these nodes were recorded and plotted in Figure 7-4. These results were analysed in two ways: the DR of individual queues and the offset of these queues.

Evaluations on DR's associated with the "linear" decay of Power-law queues with respect to a log-log scale. Power-law bestfit was again applied to collected queueing statistics. The bestfits are in the form of  $y = c \cdot x^b$ , where  $c$  corresponds to the offset of a given queue delay distribution and  $b$  is the parameter of the Power-law bestfit which is the decay gradient here as logarithm is applied. These decay gradients collected from different nodes are listed in Table 7-4. A mean of -2.2397 and a standard deviation of 0.0426, which can be concluded that the DR's at different nodes along the same end-to-end path are equal.

<b>Foreground</b>	<b>D<sub>OFF(FG)</sub></b>	10
	<b>D<sub>ON(FG)</sub></b>	3.1
	<b>N<sub>(FG)</sub></b>	1
	<b>R<sub>(FG)</sub></b>	2
<b>Background</b>	<b>D<sub>OFF(BG)</sub></b>	10
	<b>D<sub>ON(BG)</sub></b>	3.1
	<b>N<sub>(BG)</sub></b>	8
	<b>R<sub>(BG)</sub></b>	1

**Table 7-3 Parameters used in the hybrid technique example**

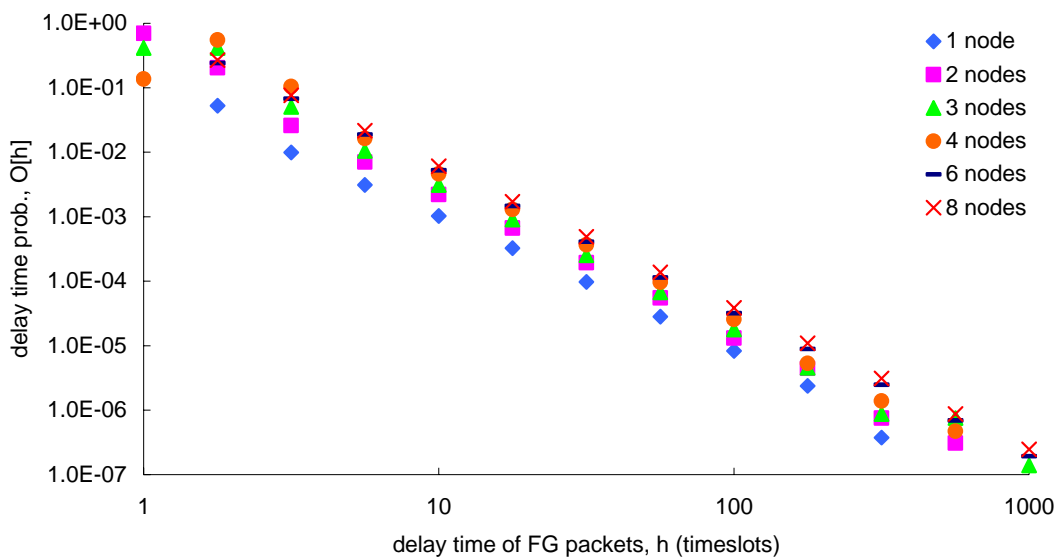
As the DR remains constant regardless of the number of nodes in the network path, the offset of the queue can be seen as the amount of shifting. The relationship between the offset of a queue distribution,  $c$ , and the number of nodes in the network path was studied. The offset values of the buffer queues in this example are plotted against the number of nodes in Figure 7-5. A relationship is derived between the offset and the number of nodes, which can be described by a logarithm equation. By referring to this equation and the universal DR of a given end-to-end path, it is then feasible to predict the queueing delay for a particular node further down the path.

To validate the derived relationship, the delay time probabilities at node 6 and 8 were also collected independently. The corresponding DR's are also included in Table 7-4, and they are within the same confidence interval region as the rest of the cases. In addition, offsets of the DR's of node 6 and 8 were plotted in Figure 7-5 to check the validity of the derived relationship. It is clearly shown that these offsets follow the relationship, which indicates the accuracy of the hybrid technique.

Nodes	DR
1	-2.1756
2	-2.2952
3	-2.2480
4	-2.2399
6	-2.2163
8	-2.1955

**Table 7-4 DR's at individual nodes**

So far at most four nodes are required to accurately predict delay time probabilities up to ten nodes. However, when the number of nodes is higher, e.g. >10, the changes in offset is negligible as the distributions of the delay time probabilities appear to be extremely close together, see Figure 7-4. It therefore can be concluded that the hybrid technique presented here is robust and highly accurate for the predication of delay time probabilities of an end-to-end path.



**Figure 7-4 Delay time distributions of FG packets a different nodes**

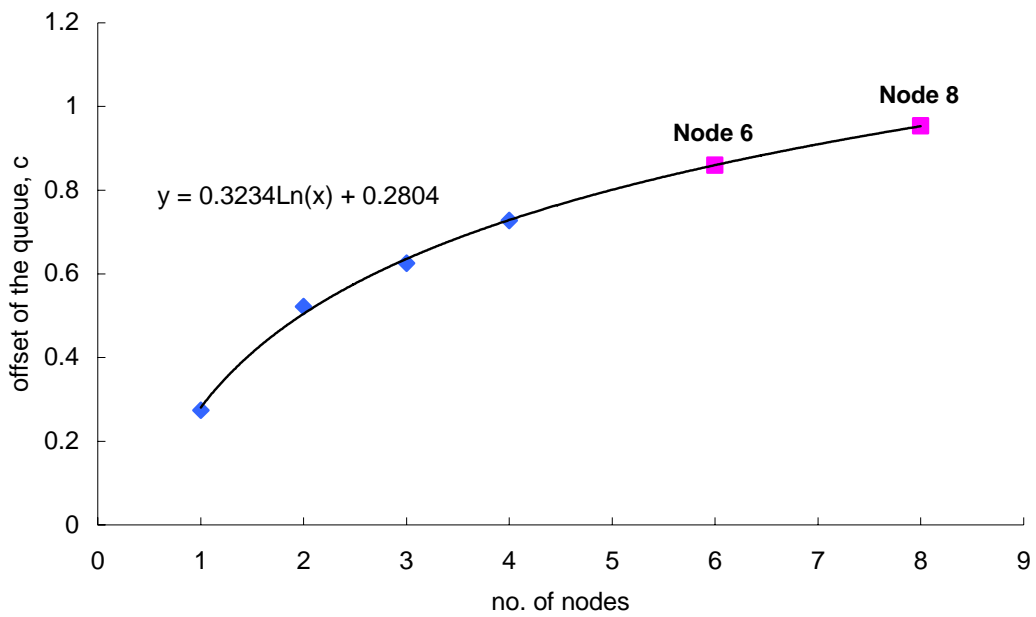


Figure 7-5 Relationships between the queue offset, c, and the number of nodes

## 7.4 Summary

This chapter has presented an aggregation technique, NA, which reduces the number of nodes of an end-to-end connection to a single node. The technique requires appropriate measurements to define parameters. Excellent results are provided and acceleration is offered by the significant cut down in simulation overheads. Moreover, a hybrid technique for the prediction of end-to-end delay probabilities has been presented. Queueing statistics of nodes at the far end of an end-to-end connection can be estimated by analysing the corresponding statistics obtained from the first few nodes.

The technical part of this thesis ends in the chapter. Chapter 8 will provide a discussion of the research. The chapter will also provide an insight of possible future work. Finally, conclusions will be provided.

## Chapter 8

### Discussion, Future Work and Conclusions

#### 8.1 Discussion

Studying Power-law traffic has become one of the major challenges in traffic modelling. A good understanding of such traffic is vital as such traffic possesses characteristics that are vastly different from traditional Markovian assumptions, necessitating new definitions for traffic models. Besides, some of these properties contribute to excessive run-times and computational effort, causing simulations to be extremely difficult, or even infeasible, to achieve. Therefore, there is a need for efficient and effective simulation techniques for Power-law traffic modelling.

The objective of this research was to develop an acceleration scheme that is accurate, simple and more importantly, fast for Power-law traffic modelling. This is achieved by Traffic Aggregation – TA.

TA is a novel approach for simulating Power-law traffic. It was developed based upon the concept of state-space reduction, previously applied to Markovian traffic [Schormans00]. Reducing the number of state-space of a system shortens the time required by the system to reach steady-state, and hence, speedup is achieved as simulations no longer need to run as long as previously.

A new algorithm was derived for TA which based immensely upon simulation results over a range of different parameters. Those parameters were chosen due to a number of reasons. For instance, with reference to  $D_{ON}$  and  $D_{OFF}$ , as the generation of random numbers is limited by the pseudo-random number generator, and such influence becomes apparent when generating Pareto numbers. Therefore, only Pareto distributions with a mean within a certain

range can be produced. Selections of  $N$  and  $C$  were based upon the resultant queueing behaviour. Theoretically, the number of overlapping sources is unbounded, provided that  $D_{ON}$ ,  $D_{OFF}$ ,  $R$  and  $C$  are properly configured so that the system will not be overloaded. However, simulation results illustrate that as the number of sources increases, the overall queueing behaviour improves. Eventually, the queue can hardly be categorised as Power-law, but appear to be SRD. This multiplexing gain phenomenon is also supported by [Cao01]. As a result, combinations of  $N$  and  $C$  were chosen within the Power-law queueing range.

TA was validated via a series of statistical procedures. Techniques such as confidence intervals, Power-law bestfit and exponentially wider bins were exploited. The confidence intervals were used in two aspects. First of all, to ensure the stability of simulation results. In this thesis, simulations were carried out for a number of runs with different seed values to eliminate unreliable results. Secondly, the confidence intervals were employed as a standard for the comparison between the original model and TA; differences between the two were restricted to  $\pm 5\%$ .

Power-law bestfits were applied in many areas in this thesis. Its primary use was to define an overall trend to describe the obtained queueing results. These bestfits play a non-trivial role in this research in the sense that they are referred to as the steady-state response of a given original model, in which TA is compared against and acceleration with respect to stability is justified.

High numbers of elements located at the end of the queue tail contribute disproportionately when deriving a Power-law bestfit. By putting data into exponentially wider bins, results then appear evenly spaced on a log-log scale. And, only bestfits obtained from the binned data can represent the true overall queueing behaviour of the system.

The application of TA was tested under a wide range of scenarios. Firstly, TA was validated using its target scenario in which a single buffer with overlapping Pareto sources was used. Models with different combinations of  $D_{ON}$ ,  $D_{OFF}$ ,  $N$ ,  $R$ , and  $C$  were simulated as well as their corresponding TA's. Results in Chapter 4 demonstrate that TA is highly accurate and insensitive to the setting of the parameters.

In addition, the feasibility of TA in aggregating truncated Pareto traffic was examined. The term “truncated” refers to traffic in which an upper bound has been applied to limit the ON sojourn times of the traffic. This is logical in the engineering sense as file sizes do not grow indefinitely in networks, and therefore, ON periods (transmission times) are bound to have an upper limit. The process of applying TA to truncated traffic requires deriving an equivalent original Pareto model, and using this equivalent model to define an appropriate TA. An upper limit is then applied to the obtained TA in order to reflect the truncated queue. It was found that TA equally works equally well with truncated sources.

In the case of a single buffer, TA was extended to heterogeneous sources to encounter the fact that a mixture of traffic sources is often found at multiplexing points of a network. Again, an equivalent original Pareto model is required prior to the application of TA, and this equivalent model is characterised by referring to the proportion of individual sources, details are again in Chapter 4. Power-law DR's were obtained from heterogeneous cases with different traffic proportions that lead to the same equivalent model, and they were found to be the same, independent to the heterogeneous settings.

Acceleration achieved by TA is substantial, and various measures were applied to quantify the amount of speedup. As TA generates far fewer events as compared to the original model, run-times of TA are generally much less. In particular the most significant speedup is demonstrated with regard to the rate of reaching steady-state. The sum of squares of differences method was introduced to analyse queueing results at different stages of the original and TA simulations, so that the rate of stability can be visualised. Studies have shown that TA enables a simulation to reach stability at least 800 times faster than the original simulation. This is an enormous speedup!

In addition to a single buffer, an end-to-end path was introduced in Chapter 5 as a testing ground for TA. Such path was fed by FG and BG, with the latter replaced by TA. The substitution is robust and accurate, which greatly simplifies and accelerates end-to-end simulations that consist of Power-law traffic. Furthermore, WRR was brought into the end-to-end path, and studies were carried out to investigate if the presence of WRR has any effects on the application of TA. Results show that TA is not affected by the scheduling mechanism, and yet, accurate queueing results were obtained. Tremendous acceleration in the magnitude of 1000 times was achieved.

One of the valuable properties of TA is that it is not mutually exclusive to other fast simulation techniques; and therefore, it can be applied in conjunction with other acceleration techniques to achieve dual speedup. In this research, a joint venture was formed between TA and RESTART in Chapter 6. The original formula for defining thresholds for RESTART was found to be inappropriate when applying to Power-law traffic due to the slow-decaying nature of the queues. However, by increasing the probability of individual thresholds, i.e.  $P_i$ , proper thresholds can be defined on the Power-law queues, and excellent RESTART results were obtained. The combination of TA and RESTART allows buffer states with a very low probability to be estimated, i.e. statistics further down the tail are made available.

To accelerate a simulation, there are many dimensions that can be focused on. For example, taking an end-to-end path as an example, if the number of multiplexing sources is considered as the path's "width"; under the same token, the number of nodes in the path can then be seen the "length". Attempts had been made to aggregate the many nodes of an end-to-end path, so that the associated simulation time can be reduced.

Chapter 7 covers the concept of NA, which was developed with reference to the fact that FG traffic transforms as it propagates along the path. Provided that the transformation is accurately captured, a single node can be used to substitute the entire path. At present, NA relies on measurements obtained from the simulation of the original path. Results show that the concept of NA works well, and provides significant speedup due to the enormous overhead reduction.

Both TA and NA are extremely useful in accelerating Power-law traffic simulation, and the combination of the two techniques forms a powerful tool. A network, regardless of the number of the multiplexed sources and the number of nodes along the path, can be represented in terms of a single buffer. Results in Chapter 7 have already shown that an incredible amount of speedup is achieved by the combination.

Two hybrid techniques related to this research were developed. The first technique is to analyse a multiplexed Power-law traffic scenario and details can be found in Appendix 2. By measuring certain characteristics of the multiplexed traffic, queueing statistics of the traffic can be predicted using the formula adapted from the analytical solution developed for a single Pareto source in Appendix 1 [Ma00].

The second hybrid technique concerns the evaluation of an end-to-end path and is included in Chapter 7. Based on the observation that the delay time distribution of individual nodes appears the same, and only difference is the offset along the x-axis. By knowing such distribution for the first few nodes, it is possible to predict the statistics for nodes further down the path. At most, only four nodes are required for predicting the queueing distribution for a path with ten nodes or more.

## 8.2 Future Work

The development of TA can be considered as the first successful step in the whole development of fast Power-law traffic modelling. There are many ways to further extend TA, and some of these possibilities are reviewed in this section.

TA operates in the packet level, in which packets are generated one by one in a simulation. Acceleration is achieved mainly by the reduction in state-space and hence the system is allowed to reach steady-state within a much shorter time. One way to bring TA to the next level of acceleration is to apply burst level simulation. In the burst level, traffic is no longer formed by individual packets; bursts are used to represent traffic. Further speedup is achieved by the significantly reduced overheads. [Ariffin02] is an example in the direction of burst level modelling for acceleration simulation.

Furthermore, the concept of TA can be further developed in the application of other network scenarios. For instance, a mobile network can be used as a testing platform. Due to the rising multi-media applications, e.g. picture messaging, in mobile networks, traffic is bound to have the Power-law nature, and the related modelling is inevitable.

Another valuable aspect to research on would be to fully develop NA. The novel concept has been examined in this thesis and proved to be a success. The next immediate step of the research would be deriving an algorithm for the parameterization of the scheme, which can be achieved either by analysis or simulation, so that the transformed characteristics of FG traffic can be predicted, and hence applying NA.

Finally, further research can be done on the application of TA. For instance, the combination of TA and NA can be used as a testing platform for the study of a protocol stack. Protocol related studies often require a network of some sort, which requires a lot of effort to even

construct the network itself. By applying the combination in network modelling, the whole study is greatly simplified and sped up, and the interaction of protocols can be investigated efficiently and effectively.

### **8.3 Conclusions**

Simulations of Power-law traffic are prohibitively time and computational demanding, and there is a need for efficient and effective simulation techniques for modelling Power-law traffic.

The objective of this research was to develop an acceleration scheme for Power-law traffic modelling so that accurate statistics can be collected within an affordable time; and this objective has been fulfilled by the development of TA.

TA, as the main contribution of this research, was developed based on the concept of state-space reduction which was originally used to aggregate Markovian traffic. The concept was then adapted to Power-law traffic, and a new algorithm has been derived for defining parameters for TA. Validation results show that TA is highly accurate while an tremendous speedup in the magnitude of 800 times is achieved.

Moreover, the application of TA was examined under a wide range of scenarios. Studies provided in this thesis evidently show that TA is highly applicable, providing a powerful simulation tool for Power-law traffic modelling. Furthermore, TA was used in conjunction with RESTART to demonstrate that the technique is not mutually exclusive to other acceleration schemes.

Finally, the acceleration of an end-to-end path has been explored. The concept is to aggregate the many nodes of a network path (NA), and to use a single node for substitution. Results show high accuracy, while excellent speedup is achieved. In addition, the combination of TA and NA is robust, providing a powerful acceleration scheme for modelling Power-law traffic in packet networks.

## Appendix I

# Design Rules and Equivalent Capacity for Buffering of a Pareto Source

Formulae were developed that act as design rules when buffering an ON-OFF source with LRD characteristics [Ma00]. These can be used to predict the buffer length for a particular required overflow probability, or the equivalent capacity required to ensure a specified buffer overflow probability given a particular buffer length. The traffic source has Pareto distributed ON and OFF periods. For generality, “data units” refer to packets, cells or bytes as appropriate to the networking scenario.

$D_{OFF}$  and  $D_{ON}$  are the mean OFF and ON sojourn times of the single ON-OFF source.  $R$  is the rate of arrival from a single source (when ON) to the buffer, and  $C$  is the service rate at which data units are transmitted out of the buffer, where  $R > C$ . Given  $h$  is the buffer length in data units, defining  $Q[h]$ ,

**$Q[h] = P(\text{infinite queue contains } > h \text{ data units})$**

As the sojourn times of the source follow the Pareto distributions, the increase and decrease of the queue length are therefore approximated by the Pareto distributions.

Let  $P(\text{queue length increases by } > j \text{ data units during an ON period}) = \varphi_{ON} \cdot j^{-\alpha_{ON}}$ ,  
and  $P(\text{queue length decreases by } > j \text{ data units in an OFF period}) = \varphi_{OFF} \cdot j^{-\alpha_{OFF}}$

where  $\alpha_{ON}$  and  $\alpha_{OFF}$  are the shape parameters of the Pareto distributions describing the effect on the buffer of the ON and OFF periods respectively. The scaling factor (location parameter) are such that:  $\varphi_{ON} = \varphi_{OFF} = 1$  data unit, and therefore:

$$\alpha_{ON} = \frac{V}{V-1} \text{ or } V = \frac{\alpha_{ON}}{\alpha_{ON}-1}$$

where  $V$  = the mean number of data units increasing the buffer level in an ON period, and

$$\alpha_{OFF} = \frac{W}{W-1} \text{ or } W = \frac{\alpha_{OFF}}{\alpha_{OFF}-1}$$

and  $W$  = the mean number of data units transmitted out of the buffer in an OFF period. As the Pareto distribution is a single parameter distribution (since  $\varphi$  has been assumed to be 1), this is sufficient to fully parameterize the “up” and “down” periods in the growth of the queue level.

To exploit the Power-law shape of the Pareto distribution, a system of balance equations are required to link the probability that the buffer content is of an order of magnitude  $g$ . The developed method is completely different to the usual method of designing a balance equation for a queueing system, which would usually create a Markov chain linking the probabilities of the buffer containing exactly  $g$  units with the probability of it containing “ $g+1$ ” units, “ $g-1$ ” units, “ $g-2$ ” units, etc. Define  $P[g]$ ,

**$P[g]$  = P(number in buffer is of an order of magnitude “ $g$ ” at the end of an ON period)**

Consequently,  $P[g] = Q[g-1] - Q[g]$ . Using the fact that the probability of moving down from any state “ $g-1$ ” to a state “ $g$ ” (for instance, from a state of having 1000-10000 in the buffer to having 100-1000 in the buffer), requires that the source stay OFF long enough to drain approximately 10000 data units from the buffer. Once this has happened, the buffer is effectively empty anyway. Equally, when moving from a state “ $g$ ” to a state “ $g+1$ ” the source must remain active long enough to increase the buffer level by an amount equal to “ $g+1$ ”, in other words, it must introduce 10000 data units in the case of moving between a state of having 100-1000 in the buffer to having 1000-10000 in the buffer. This means that the probability increasing the buffer level by one state, i.e. an order of magnitude, virtually the same as if the buffer were being filled from empty. This is a natural consequence of having Power-law (Pareto) ON and OFF periods.

As a result, the following relationship is obtained:

$$\frac{P[g+1]}{P[g]} = \frac{a^g \cdot (1-s^g)}{s^g \cdot (1-a^g)} \approx \frac{a^g}{s^g}$$

where  $a = 10^{-\alpha_{ON}}$  and  $s = 10^{-\alpha_{OFF}}$ , giving

$$\frac{P[g+1]}{P[g]} = \frac{10^{-\alpha_{ON}}}{10^{-\alpha_{OFF}}} = 10^{[-\alpha_{ON} + \alpha_{OFF}]}$$

and therefore:

$$Q[h] = \left(10^{[-\alpha_{ON} + \alpha_{OFF}]}\right)^{\log h} \quad \text{Eqn. A. 1}$$

where  $h$  is the actual buffer length and not an order of magnitude. It is important to note that the actual buffer sizes do not have to be restricted to 10, 100, 1000 etc., and that, in general, such values will simply give non-integer  $\log(h)$  values in the analysis, which does not affect the accuracy of the results.

The following relationships are given in [Schormans94],

$$\alpha_{ON} = D_{ON}(R-C)/(D_{ON}(R-C)-1)$$

$$\alpha_{OFF} = D_{OFF} \cdot C/(D_{OFF} \cdot C - 1)$$

Substituting into Eqn. A. 1 gives:

$$Q[h] = \left(10^{[D_{OFF}C/(D_{OFF}C-1) - D_{ON}(R-C)/(D_{ON}(R-C)-1)]}\right)^{\log h} \quad \text{Eqn. A. 2}$$

The advantage of this approach is that Eqn. A. 2 can be further developed to yield an equivalent capacity formula analogous to that in [Schormans94], but, importantly in this case, applying to LRD sources of the type considered in this thesis. Specifically, let

$$\Gamma = \frac{\log_{10} Q[h]}{\log_{10} h}$$

Substituting into Eqn. A. 2, obtaining

$$\Gamma = \frac{D_{OFF}C}{D_{OFF}C-1} - \frac{D_{ON}(R-C)}{D_{ON}(R-C)-1}$$

Rearranging the equation, forming a quadratic equation with an unknown C:

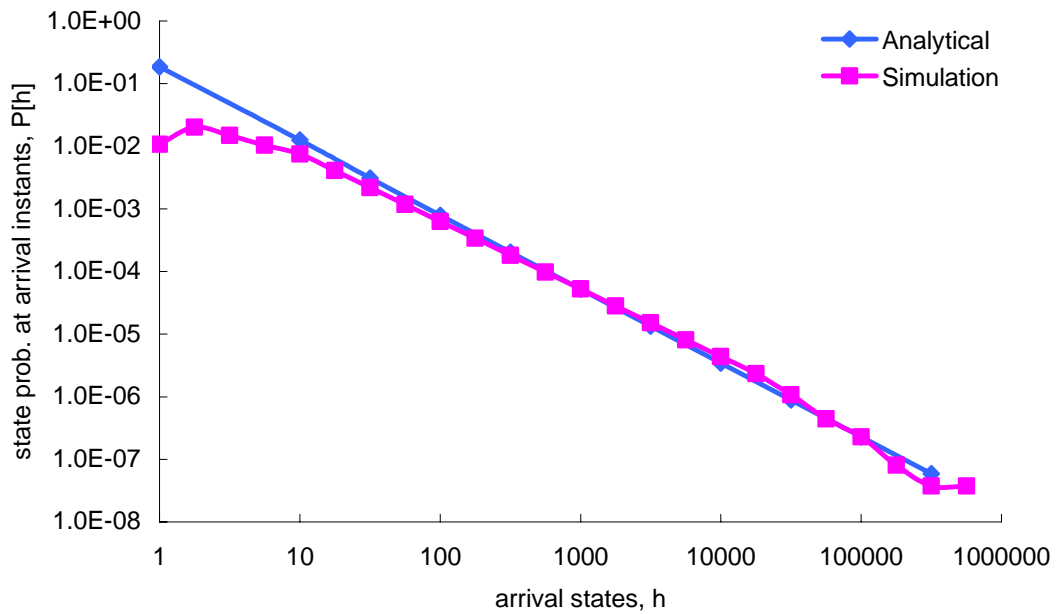
$$-\Gamma D_{ON} D_{OFF} C^2 + (\Gamma D_{ON} D_{OFF} R - \Gamma D_{OFF} + \Gamma D_{ON} + D_{ON} + D_{OFF})C + [\Gamma(1 - D_{ON} R) - D_{ON} R] = 0$$

Solving the quadratic equation to give the equivalent C, therefore:

$$C = \frac{-\Psi \pm \sqrt{\Psi^2 - 4(\Gamma D_{ON} D_{OFF})[\Gamma(1 - D_{ON} R) - D_{ON} R]}}{-2\Gamma D_{ON} D_{OFF}} \quad \text{Eqn. A. 3}$$

where  $\Psi = \Gamma D_{ON} D_{OFF} R - \Gamma D_{OFF} + \Gamma D_{ON} + D_{ON} + D_{OFF}$  which is the coefficient in the middle term.

A numerical example of a single Pareto source is presented here with  $D_{OFF} = 15$ ,  $D_{ON} = 5$ ,  $R = 2$  and  $C = 1$ .  $P[h]$  was monitored and compared with Eqn. A. 2 using the relationship  $P[h] = Q[h-1] - Q[h]$ . Results are illustrated in Figure A. 1, and excellent agreement is obtained.



**Figure A. 1 Comparison of simulated arrival state probabilities with analysis**

## Appendix 2

# Hybrid Technique for Analysis of Multiplexed Power-Law Traffic

To predict the buffer overflow probability in cases where there are  $N$  overlapping Pareto ON-OFF sources, the resultant aggregate traffic of all  $N$  sources is converted into a two-state ON-OFF process, and appropriate measurements are taken from the defined process for the purpose of analysis. The single ON-OFF process becomes analysable through an extension (presented here) of the work in [Ma00].

$N$  ON-OFF sources are buffered to a bottleneck.  $D_{OFF}$  and  $D_{ON}$  are the mean OFF and ON sojourn times of individual sources respectively.  $R$  is the rate of arrival from a single source (when ON) to the buffer.  $C$  is the service rate at which data units are transmitted out of the buffer. Measurements are taken with respect to the definition of TA, more precisely, the mean sojourn times and the rate of arrival of the two states of the single source process. However, these measurements do not associate with the actual parameters of TA. To avoid any confusion,  $M_{ON}$ ,  $M_{OFF}$ ,  $r_{ON}$  and  $r_{OFF}$  are introduced to denote individual parameters in the aspect of hybrid measurements.

Let  $N_{ON}$  specify the total number of sources that are “ON” simultaneously, so  $N_{ON} \cdot R$  is the total aggregate input rate. Under the TA definitions, an  $ON_{AG}$  state occurs when  $N_{ON} \cdot R > C$  and the average duration of which is denoted as  $M_{ON}$ . On the other hand, an  $OFF_{AG}$  state is obtained when  $N_{ON} \cdot R \leq C$ , and  $M_{OFF}$  denotes the average duration. The mean packet arrival rate of the two states are  $r_{ON}$  and  $r_{OFF}$  where  $r_{ON} > C$  and  $r_{OFF} < C$ .

For generality, “data units” can refer to packets, cells or bytes as appropriate to the networking scenario.

To compute the buffer overflow probability, an enhancement of the approach of [Ma00] was developed. As the sojourn times follow Pareto distributions, it is then possible to apply the same assumptions as in Appendix 1:

$$P(\text{queue length increases by } >j \text{ data units during an ON period}) = \varphi_{\text{ON}} \cdot j^{-\alpha_{\text{ON}}}$$

$$P(\text{queue length decreases by } >j \text{ data units in an OFF period}) = \varphi_{\text{OFF}} \cdot j^{-\alpha_{\text{OFF}}}$$

where  $\alpha_{\text{ON}}$  and  $\alpha_{\text{OFF}}$  are the shape parameters of the Pareto distributions describing the effect on the buffer of the ON and OFF periods respectively. Again, assuming the scaling factors are such that:  $\varphi_{\text{ON}} = \varphi_{\text{OFF}} = 1$  data unit, and therefore:

$$\alpha_{\text{ON}} = \frac{V}{V-1} \text{ or } V = \frac{\alpha_{\text{ON}}}{\alpha_{\text{ON}} - 1}$$

where  $V$  = the mean number of data units increasing the buffer level in an ON period, and

$$\alpha_{\text{OFF}} = \frac{W}{W-1} \text{ or } W = \frac{\alpha_{\text{OFF}}}{\alpha_{\text{OFF}} - 1}$$

and  $W$  = the mean number of data units transmitted out of the buffer in an OFF period.

Once more, the system of balance equations is designed such as to link the probability that the buffer content is of an order of magnitude “g” to the probability that it is of an order of magnitude “g+1”. Define:

**P[g] = P(number in buffer is of an order of magnitude “g” at the end of an ON period)**

**Q[g] = P(number in buffer is > an order of magnitude “g” at the end of an ON period)**

$$\text{and } P[g] = Q[g-1] - Q[g].$$

So as before, the following relationship is obtained:

$$\frac{P[g+1]}{P[g]} = \frac{a^g \cdot (1-s^g)}{s^g \cdot (1-a^g)} \approx \frac{a^g}{s^g}$$

where  $a = 10^{-\alpha_{\text{ON}}}$  and  $s = 10^{-\alpha_{\text{OFF}}}$ , giving

$$\frac{P[g+1]}{P[g]} = \frac{10^{-\alpha_{\text{ON}}}}{10^{-\alpha_{\text{OFF}}}} = 10^{[-\alpha_{\text{ON}} + \alpha_{\text{OFF}}]}$$

and therefore:

$$Q[h] = \left(10^{[-\alpha_{ON} + \alpha_{OFF}]}\right)^{\log h} \quad \text{Eqn. A. 4}$$

where  $h$  is the actual buffer length.

Deriving an equation for  $\alpha_{ON}$ :

$$\alpha_{ON} = M_{ON}(r_{ON} - C) / (M_{ON}(r_{ON} - C) - 1)$$

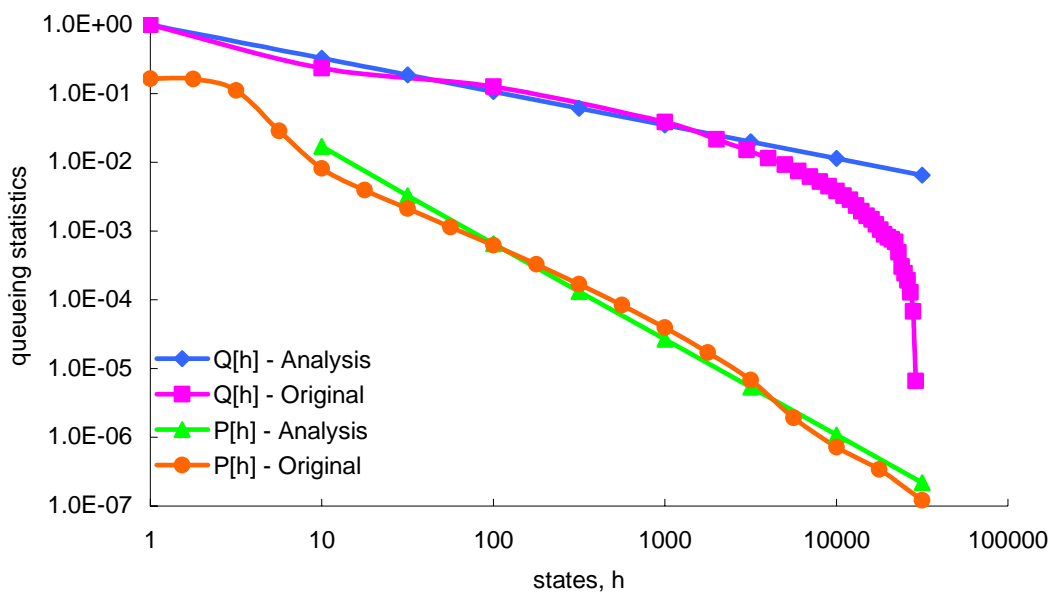
However, so far,  $r_{OFF}$  has not yet been taken into consideration, as [Ma00] did not allow more than a single source (i.e.  $N=1$  always). This ‘‘OFF’’ arrival rate,  $r_{OFF}$ , is a crucial factor that determines the rate of emptying buffer content, and plays an important role in the calculation of  $Q[h]$  as the probability a source remains in the OFF state is a function of the value of  $r_{OFF}$ . The buffer empties accumulated packets at a rate of  $C - r_{OFF}$ , and therefore, for  $\alpha_{OFF}$ :

$$\alpha_{OFF} = M_{OFF}(C - r_{OFF}) / (M_{OFF}(C - r_{OFF}) - 1)$$

Substituting into Eqn. A. 4 gives:

$$Q[h] = \left(10^{[M_{OFF}(C - r_{OFF}) / (M_{OFF}(C - r_{OFF}) - 1) - M_{ON}(r_{ON} - C) / (M_{ON}(r_{ON} - C) - 1)]}\right)^{\log h} \quad \text{Eqn. A. 5}$$

Recalling the example in Chapter 6 in which  $D_{ON} = 5$ ,  $D_{OFF} = 10$ ,  $N = 10$ ,  $C = 5$  and  $R = 1$ , giving a load,  $\rho = 0.6667$ . Measurements obtained using the above method are:  $M_{OFF} = 27.1704$ ,  $M_{ON} = 2.2527$ ,  $r_{OFF} = 3.0869$  and  $r_{ON} = 6.3061$ . These values were then substituted into Eqn. A. 5, and the results were used to compare against simulation results, illustrated in Figure A. 2, which consists of the analytical solutions and the associated steady-state simulation results for both  $P[h]$  and  $Q[h]$ . The analysis presented here was used throughout this research as guidance for the accuracy of the results, and it is particularly useful in the RESTART studies in Chapter 6, in which terminating simulations were used.



**Figure A.2** Analytical solutions for multiple Pareto traffic sources

## Appendix 3

### Aggregate Variance Method for Estimating H

Divide the original time series, e.g. packet arrival pattern,  $X = (X_1, X_2, X_3, \dots)$  into blocks of size  $m$  and average within each block, that is considered as the aggregated series

$$X^{(m)}(i) = \frac{1}{m} (X_{(i-1)m+1} + \dots + X_{im})$$

$$X^{(m)}(i) = \frac{1}{m} \sum_{j=(i-1)m+1}^{im} X(j) \quad i \geq 1 \quad \text{Eqn. A. 6}$$

for successive values of  $m$ . The index  $i$  labels the block. Then take the sample variance of  $X^{(m)}(i)$ ,  $i = 1, 2, \dots$  within each block. This sample variance is an estimator of  $\text{Var}X^{(m)}$ . Since, for FGN and fractional ARIMA,  $\text{Var}X^{(m)} \sim \sigma_0^2 m^\beta$  as  $m \rightarrow \infty$  where  $\beta = 2H - 2 < 0$ , an estimate for  $\beta$ , or  $H$ , can be obtained by proceeding as follows.

For a given  $m$ , divide the data,  $X_1, \dots, X_Z$ , into  $Z/m$  blocks of size  $m$ , calculate  $X^{(m)}(i)$ , for  $i = 1, 2, \dots, Z/m$ , and its sample variance

$$\hat{\text{Var}}X^{(m)} = \frac{1}{Z/m} \sum_{j=1}^{Z/m} (X^{(m)}(j))^2 - \left( \frac{1}{Z/m} \sum_{j=1}^{Z/m} X^{(m)}(j) \right)^2 \quad \text{Eqn. A. 7}$$

Repeat this procedure for different values of  $m$  and plot the logarithm of the sample variance versus  $\log m$ . Choose values of  $m$ ,  $\{m_j, j > 1\}$ , that are equidistant on a log scale, so that  $m_{i+1}/m_i = C$ , where  $C$  is a constant which depends on the length of the series and the desired number of points.

Since  $\hat{\text{Var}}X^{(m)}$  is an estimate of  $\text{Var}X^{(m)}$ , the resulting points should form a straight line with a slope  $\beta = 2H - 2$ ,  $-1 < \beta < 0$ . In practice, the slope is estimated by fitting a line to the points obtained from the plot. It is assumed here that both  $m$  and  $Z$  are large, and that  $m \ll Z$ ,

so that the length of each block and the number of blocks are large. If  $X$  has (short-range or) no dependence, the slope obtained should equal  $-1$ . Further details can refer to [Beran94].

## Appendix 4

### Sum of Squares of Differences

For a given set of data, the least squares method is commonly used to derive a bestfit so that the sum of squares of the residuals is minimal [Barford85, Devore99, Milton95]. In this thesis, this was applied to evaluate the rate that models reach steady-state.

For a given case study, the steady-state response is obtained by averaging several sets of results and taking the Power-law bestfit. This steady-state response is then used as a reference to which responses from the original model and/or TA are compared. For instance, say data set “a” is the steady-state response for a given study, which can be an analytical solution or a set of known steady results. Data “b” and “c” are the results obtained from the simulation of the original non-accelerated settings and TA respectively at a specific time instant (e.g. after 100000 packet arrivals). How much data b and c differ from data a at that specific time can be quantified by using the sum of squares of differences.

Consider  $a = a_1, a_2, a_3 \dots a_i$ ,  $b = b_1, b_2, b_3 \dots b_i$  and  $c = c_1, c_2, c_3 \dots c_i$ . The sum of squares of differences between a and b is given by:

$$\omega_{ab} = \sum_{j=1}^i (b_j - a_j)^2$$

and similarly, the sum of square of difference between a and c is

$$\omega_{ac} = \sum_{j=1}^i (c_j - a_j)^2$$

In theory, the lower the  $\omega$ , the closer the related data set is to the compared data.

To compare the original model and TA, two independent simulations were set up, and the queueing responses of the two models were recorded at precisely the same instants, see Figure A. 3. By applying the sum of squares of differences method, the variation of  $\omega$  for a

given model over time can be observed. In other words, the convergence to steady-state for the given model can be obtained. Once the value of  $\omega$  no longer changes significantly with time, it can be concluded that the corresponding system has reached steady-state. This method has been used in this thesis to evaluate the speedup achieved by TA, and TA was found to be 800 times faster!

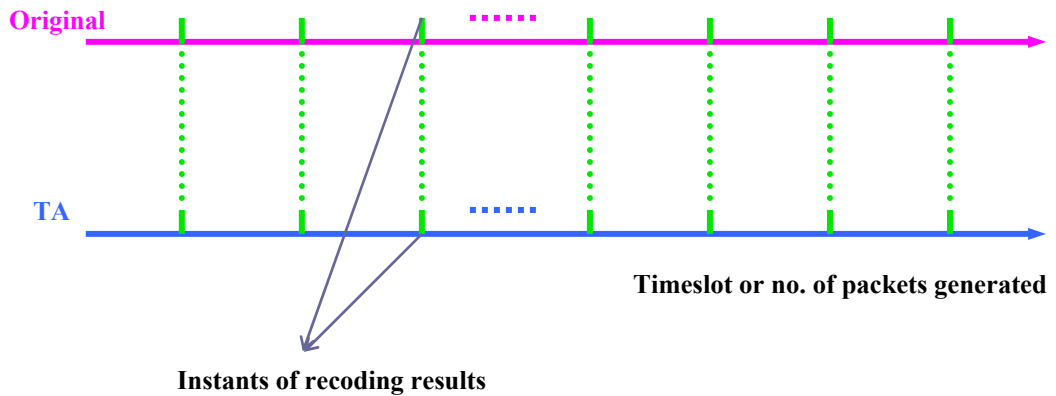


Figure A. 3 Simulation for steady-state evaluations

## Author's Publications

- [Ma03] MA, A., SCHORMANS, J. and CUTHBERT, L.: 'Aggregation technique for networks with power-law traffic and application to accelerated simulation', IEE Proc. Commun., 2003, **150** (3) pp. 177-183
- [Leung02] LEUNG, C.M., SCHORMANS, J.A. and MA, A.H.I: 'Measurement-based queue length distribution estimation for Power-law traffic', Elec. Lett., 2002, **38** (24) pp. 1608 - 1610
- [Ma02a] MA, A.H.I and SCHORMANS, J.A.: 'Accelerated simulation modelling of Power-law traffic via aggregation'. ICT'02, June 2002, Beijing, China
- [Ma02b] MA, A.H.I and SCHORMANS, J.A.: 'Efficient simulation modelling for long range dependent traffic'. RESIM'02, April 2002, Madrid, Spain
- [Ma02c] MA, A.H.I and SCHORMANS, J.A.: 'Hybrid technique for the analysis of multiplexed power-law traffic in broadband networks', Elec. Lett., 2002, **38** (6) pp. 295-297
- [Ma01] MA, A.H.I and SCHORMANS, J.A.: 'A fast simulation method for modelling IP networks'. 17th UKTS, May 2001, Dublin, Ireland
- [Ma00] MA, A.H.I, SCHORMANS, J.A., PITTS, J.M., SCHARF, E.M., PEARMAN, A.J. and PHILLIPS, C.I.: 'Design rules and equivalent capacity for buffering of Pareto source', Elec. Lett., 2000, **36** (15) pp. 1274-1275
- [Schormans00] SCHORMANS, J.A., PITTS, J.M., MA, A.H.I and MONDRAGÓN, R.J.: 'Accurate results for traffic engineering in the presence of LRD'. 16th UKTS, May 2000, Nortel Networks, Harlow, UK

## References

- [Adamic] ADAMIC, L.A.: 'Zipf, Power-laws, and Pareto – a ranking tutorial'. Xerox Palo Alto Research Centre, Palo Alto, CA, USA.
- [Addie95] ADDIE, R., ZUKERMAN, M. and NEAME, T.: 'Fractal traffic: Measurements, Modelling and Performance Evaluation'. IEEE INFOCOM'95, Apr. 1995, Boston, USA, pp. 977 - 984
- [Altamirano91] VILLÉN-ALTAMIRANO, M. and VILLÉN-ALTAMIRANO, J.: 'RESTART: A method for accelerating rare event simulations'. ITC13, June 1991, Copenhagen, Denmark, pp. 71-76.
- [Altamirano94] VILLÉN-ALTAMIRANO, M. and VILLÉN-ALTAMIRANO, J.: 'Enhancement of the accelerated simulation method RESTART by considering multiple thresholds'. ITC14, June 1994, Antibes Juan-les-Pins, France, pp. 797-810
- [Altamirano02] VILLÉN-ALTAMIRANO, M. and VILLÉN-ALTAMIRANO, J.: 'Analysis of RESTART simulation: Theoretical basis and sensitivity study', ETT, 2002, **13** (4) pp. 373-385
- [Ariffin02] Ariffin, S.H.S. and Schormans, J., 'ON/OFF Pareto Distributed Source At Burst Level: A Study of Fast Simulation Methods for Packet Networks'. PGNET'02, Apr. 2002, Liverpool, UK
- [Barford85] BARFORD, N.C.: 'Experimental measurements: precision, error and truth' (John Wiley & Sons Ltd, 1985 2nd edn.)
- [Bayes70] BAYES, A.J.: 'Statistical techniques for simulation models', Australian computer journal, 1970, **2** pp. 180-184
- [Bear80] BEAR, D.: 'Principles of Telecommunication-traffic engineering' (IEE telecommun. series 2, Peter Peregrinus Ltd., 1980)
- [Benveniste95] BENVENISTE, C. and HEIDELBERGER, P.: 'Parallel simulation of the IBM SP2 interconnection network'. WSC'95, Dec. 1995, Arlington, USA, pp. 584-589
- [Beran94] BERAN, J.: 'Statistics for long-memory processes' (Chapman & Hall, New York, 1994)

- [Beran95] BERAN, J., SHERMAN, R., TAQQU, M.S. and WILLINGER, W.: 'Long-range dependence in variable-bit rate video traffic', IEEE trans. on commun., Feb./Mar./Apr. 1995, **43** (2/3/4) pp. 1566 - 1579
- [Bovy02] BOVY, C., MERTODIMEDJO, H., HOOGHIEMSTRA, G., UIJTERWAAL, H. and VAN MIEGHEM, P.: 'Analysis of end-to-end delay measurements in Internet'. PAM'02, Mar. 2002, Fort Collins, USA, pp. 26-33
- [Box70] BOX, G., JENKINS, G. and REINSEL, G.: 'Time series analysis: forecasting and control' (Holden day, San Francisco, 1970)
- [Bratley87] BRATLEY, P., FOX, B.L. and SCHARGE, L.E.: 'A guide to simulation' (Springer-Verlag, 1987)
- [Cao01] CAO, J., CLEVELAND, W.S., LIN, D. and SUN, D.X.: 'The effect of statistical multiplexing on the long-range dependence of Internet packet traffic'. Bell Labs Technical Report, 2001
- [Crovella96] CROVELLA, M.E. and BESTAVROS, A.: 'Self-similarity in world wide web traffic: evidence and possible causes'. ACM SIGMETRICS'96, May 1996, Philadelphia, USA, pp. 160-169
- [Crovella97] CROVELLA, M.E. and LIPSKY, L.: 'Long-lasting transient conditions in simulations with heavy-tailed workloads'. WSC'97, Dec. 1997, Atlanta, USA, pp. 1005-1012
- [Demers89] DEMERS, A., KESHAV, S. and SHENKER, S.: 'Analysis and simulation of a fair queueing algorithm'. ACM SIGCOMM'89, Sept., 1989, pp. 1-12
- [Devore99] DEVORE, J. L.: 'Probability and statistics for engineering and the sciences' (Brooks/Cole, USA, 1999 5th edn.)
- [Erramilli94] ERRAMILI, A., SINGH, R.P. and PRUTHI, P.: 'Chaotic maps as models of packet traffic'. ITC14, June 1994, Antibes Juan-les-Pins, France, pp. 329-338
- [Floyd01] FLOYD, S. and PAXSON, V.: 'Difficulties in simulating the Internet', IEEE/ACM trans. on networking, 2001, **9** (4) pp. 392-403
- [Fonseca00] FONSECA, N.L.S., MAYOR, G.S. and NETO, C.A.V.: 'On the equivalent bandwidth of self-similar sources', ACM TOMACS, 2000, **10** (2) pp. 102-124

- [Fraleigh02] FRALEIGH C.J.: 'Provisioning Internet Backbone Networks to Support Latency Sensitive Applications'. PhD Thesis, Stanford University, California, USA, May 2002
- [Frost88] FROST, V.S., LARUE, W.W. and SHANMUGAN, K.S.: 'Efficient techniques for the simulation of computer communications networks', IEEE journal on selected areas in commun., Jan. 1988, **6** (1) pp. 146-157.
- [Gallardo98] GALLARDO, J.R., MAKRAKIS, D. and OROZCO-BARBOSA, L.: 'Use of Alpha-Stable self-similar stochastic processes for modelling traffic in broadband networks'. SPIE Conference on Performance and Control of Network Systems II, Boston, USA, Nov. 1998
- [Gallardo99] GALLARDO, J.R., MAKRAKIS, D. and OROZCO-BARBOSA, L.: 'An approximation to Alpha-Stable long range dependent stochastic processes suitable for simulation of communication systems'. 5<sup>th</sup> Bayona Workshop on Emerging Technologies in Telecoms, Sept., 1999, Bayona, Spain
- [Garvels98] GARVELS, M.J.J. and KROESE, D.P.: 'A Comparison of RESTART implementations'. WSC' 98, Dec. 1998, Washington, USA, pp. 601-608
- [Gribble98] GRIBBLE, S., MANKU, G., ROSELLI, D., BREWER, E. GIBBON, T. and MILLER, E.: 'Self-similarity in file systems'. ACM SIGMETRICS'98, June 1998, Madison, USA, pp. 141-150
- [Ha98] HA, S., LEE, K.W. and BHARGHAVAN V.: 'Performance evaluation of scheduling and resource reservation algorithms in an integrated packet services network environment'. IEEE ISCC'98, June 1998, Athens, Greece, pp. 94-99
- [Heidelberger94] HEIDELBERGER, P. and NICOL, D.: 'Parallel simulation of Markovian queueing networks'. MASCOT '94, Jan. 1994, pp. 35-36.
- [Jefferson85] JEFFERSON, D.R.: 'Virtual time', ACM TOPLAS'85, July 1985, **7** (3) pp. 404-425
- [Kalantery94] KALANTERY, N. and WINTER, S.: 'High performance simulation of telecommunication networks'. IEE colloquium on high performance computing and advance control, Dec. 1994, London, UK
- [Keshav97] KESHAV, S.: 'An engineering approach to computer networking: ATM networks, the internet, and the telephone network' (Addison-Wesley, 1997)

- [Korkmaz00] KORKMAZ, T. and KRUNZ, M.: 'Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks', ACM TOMACS, 2000, **10** (4) pp. 295-325
- [Kurose98] KUROSE, J.F. and MOUFTAH, H.T.: 'Computer-aided modelling, analysis and design of communication networks', IEEE journal on selected areas in commun., Jan. 1998, **6** (1) pp. 130-145
- [Law00] LAW, A.M. and KELTON, A.D.: 'Simulation modelling and analysis' (McGraw-Hill series in industrial engineering and management science, 2000, 3rd edn.)
- [Leland94] LELAND, W.E., TAQQU, M.S., WILLINGER, W. and WILSON, D.V.: 'On the self-similar nature of Ethernet traffic (Extended version)'. IEEE/ACM trans. on networking, 1994, **2** (1) pp. 1-15
- [Lewis88] LEWIS, P.A.W. and ORAV, E.J.: 'Simulation methodology for statisticians, operations analysts and engineers' (Wadsworth & Brooks / Cole Advance Books & software, 1988, vol. 1)
- [Li98] LI, J.S., WOLISZ, A. and POPESCU-ZELETIN, R.: 'Fast simulation of self-similar traffic'. IEEE ICC'98, June 1998, Atlanta, USA, pp. 1829-1833
- [Lin94] LIN, J.Y.B.: 'Parallel independent replicated simulation on networks of workstations'. PADS '94, July 1994, Edinburgh, Scotland, pp. 73-80
- [Mandelbrot63] MANDELBROT, B., 'New methods in statistical economics'. Political economy, 1963, **71** (5), pp. 421-440
- [Mastumoto92] MASTUMOTO, M. and NISHIMURA, T.: 'Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator', ACM TOMACS, 1998, **8** (1) pp. 3-30
- [Milton95] MILTON, J.S. and ARNOLD, J.C.: 'INTRODUCTION TO PROBABILITY and STATISTICS Principles and applications for engineering and the computing sciences' (McGraw-Hill Book Co., Singapore, 1995, 3rd edn.)
- [Mishra99] MISHRA, A. and CHRISTENSEN, K.: 'Improving the Simulation Process in Model-Based Codesign with Remote Execution with Load Observation and Distribution (RELOAD)'. Int'l Conf. on Parallel and Distributed Processing Techniques and Applications, June 1999, pp. 3009-3015

- [Moffaert01] VAN MOFFAERT, A., DE VLEESCHAUWER, D., BÜCHLI, M. JANSSEN, J., and PETIT, G.: ‘Tuning the VoIP gateway to transport international voice calls over a best-effort IP backbone’. IFIP01, June 2001, Budapest, Hungary, pp. 193-205
- [Molnat97] MOLNAT, S. and VIDACS, A.: ‘On modelling and shaping self-similar ATM traffic’. ITC15, July 1997, Washington, USA, pp. 1409-1420.
- [Mondragón01] MONDRAGÓN, R.J., ARROWSMITH, D.K. and PITTS, J.M.: ‘Chaotic maps for traffic modelling and queueing performance analysis’, *Performance Evaluation*, 2001, **43** pp. 223-240
- [NAGLE87] NAGLE, J.B.: ‘On packet switches with infinite storage’, *IEEE trans. on commun.*, 1987, **35** (4) pp. 435-438
- [Netsizer] Telcordia Netsizer, Internet growth forecasting tool, 2003. Available: <http://www.netsizer.com>
- [Nicola90] NICOLA, V.F., NAKAYAMA, M.K., HEIDELBERGER, P. and GOYAL, A.: ‘Fast simulation of dependability models with general failure, repair and maintenance processes’. 20<sup>th</sup> int’l symposium on fault-tolerant computing, June 1990, Newcastle Upon Tyne, England, pp. 491-498
- [Norros95] NORROS, I.: ‘On the use of fractional Brownian motion in the theory of connectionless networks’, *IEEE journal on selected areas in commun.*, 1995, **13** (6) pp. 953-962
- [Nuzman01] NUZMAN, C.J., SANIEE, I., SWELDENS, W. and WEISS, A.: ‘A compound model for TCP connection arrivals’. Specialist sem. on IP traffic modelling measurement and management, Monterey, USA, Sept. 2000
- [O’Mahony01] O’MAHONY, M.J., SIMEONIDOU, D., HUNTER, D.K. and TZANAKAKI, A.: ‘The application of optical packet switching in future communication networks’, *IEEE Commun. Mag.*, 2001, **39** (3) pp. 128-135
- [Osterbo03] OSTERBO, O.: ‘Models for end-to-end delay in packet networks’. Scientific report, telenor R&D, Fornebu, Norway, 2003
- [Pareto1896] PARETO, V.: ‘Cours d’Economie Politique’ (Droz, Geneva Switzerland, 1896)
- [Park00] PARK, K. and WILLINGER, W.: ‘Self-similar network traffic and performance evaluation’ (John Wiley & Sons Inc., USA, 2000)

- [Park96] PARK, K., KIM, G. and CROVELLA, M.: 'On the relationship between files sizes, transport protocols and self-similar network traffic'. ICNP'96, Oct. 1996, Columbus, USA, pp. 171-180
- [Pawlikowski02] PAWLIKOWSKI, K., JEONG, H. and LEE, J.: 'On credibility of simulation studies of telecommunication networks', IEEE commun. mag., 2002, **40** (1) pp. 132 – 139
- [Paxson94] PAXSON, V. and FLOYD, S.: 'Wide-area traffic: The failure of Poisson modelling'. ACM SIGCOMM'94, Feb. 1994, London, England, pp. 257-268
- [Paxson95] PAXSON V., 'Fast approximation of self-similar network traffic'. Lawrence Berkeley Lab. and EECS Division, University of California, Berkeley, CA, Tech. Rep. LBL-36750, Apr. 1995
- [Paxson97a] PAXSON V., 'Fast, approximate synthesis of fractional Gaussian Noise for generating self-similar network traffic', Computer Commun. Review, 1997, **27** (5) pp. 5-18
- [Paxson97b] PAXSON, V. and FLOYD, S.: 'Why We Don't Know How To Simulate The Internet'. WSC'97., Dec. 1997, Atlanta, USA, pp. 1037 –1044
- [Peha97] PEHA, J.M.: 'Protocols can make traffic appear self-similar'. IEEE/ACM/SCS on commun. Networks and distributed systems modelling and simulation, 1997, San Diego, CA, USA, pp. 47-52
- [Pitts01] PITTS, J.M. and SCHORMANS, J.A.: 'Introduction to IP and ATM design and performance' (John Wiley & Sons Ltd., 2001, 2nd edn.)
- [Pruthi95] PRUTHI, P. and ERRAMILI, A.: 'Heavy-tailed ON-OFF source behaviour and self-similar traffic', ICC '95, June 1995
- [Queseth01] QUESETH, O.: 'On the performance of unlicensed data access systems'. Nordic radio symposium, Apr. 2001
- [Rubinstein81] RUBINSTEIN, R.Y.: 'Simulation and the Monte-Carlo method' (Wiley, New York, 1981)
- [Samuel99] SAMUEL, L. G.: 'The application of non-linear dynamics to teletraffic modelling'. PhD Dissertation, Department of Electronic Engineering, Queen Mary and Westfield College, University of London, U.K., 1997
- [Schormans00] SCHORMANS, J.A., PITTS, J.M., SCHARF, E.M., PEARMAIN, A.J. and PHILLIPS, C.I.: 'Buffer overflow probability for multiplexed on-off VoIP sources', Elect. Lett., 2000, **36** (6) pp. 523-524

- [Schormans01] SCHORMANS, J.A., LIU, E., CUTHBERT, L.G. and PITTS, J.M.: 'A hybrid technique for the accelerated simulation of ATM networks and network elements', IEEE/ACM TOMACS, Apr. 2001, **11** (2) pp. 182-205
- [Schwefel01] SCHWEFEL, H.P.: 'Behaviour of TCP-like elastic traffic at a buffered bottleneck router'. IEEE INFOCOM, Apr. 2001, Anchorage, Alaska, USA.
- [Schwefel99] SCHWEFEL, H.P. and LIPSKY, L.: 'Buffer size issues in the presence of self-similar traffic'. 3rd IFIP workshop on traffic management and design of ATM networks, Apr. 1999, London, UK, pp.11/1-11/12
- [Stallings98] STALLINGS, W.: 'High-speed networks: TCP/IP and ATM design principles' (Prentice-Hall Inc., 1998)
- [Stewart02] STEWART, R.: 'End-to-end delay analysis for small/medium scale IP networks'. PhD Dissertation, Department of Electronic Engineering, Queen Mary, University of London, 2002
- [Stol91] STOL, N. and HELVIK, B.E.: 'End-to-end traffic in ATM networks: network chain simulator'. Technical report STF40 A91035, ELAB-RUNIT, February 1991
- [Sun] [www.doc.sun.com](http://www.doc.sun.com)
- [Takahashi02] TAKAHASHI, J., TODE, H. and MURAKAMI, K.: 'QoS enhancement methods for MPEG Video transmission on the Internet'. IEEE ISCC'02, July 2002, Taormina-Giardini Naxos, Italy, pp. 106-113
- [Taqqu97] TAQQU, M.S., WILLINGER, M. and SHERMAN, R.: 'Proof of a fundamental result in self-similar traffic modelling', ACM CCR, 1997, **26** (2) pp. 5-23
- [Townsend98] TOWNSEND, J.K., HARASZTI, Z., FREEBERSYSER, J.A. and DEVETSIKIOTIS, M.: 'Simulation of rare events in communication networks', IEEE Commun. Mag., 1998, **36** (8) pp. 36-41
- [Veres00] VERES, A., KENESI, Z.S., MOLNAR, S. and VATTAY, G.: 'On the propagation of long-range dependence in the Internet'. ACM SIGCOMMS'00, Aug. 2000, Stockholm, Sweden, pp. 243-254
- [Willinger97] WILLINGER, W., TAQQU, M.S., SHERMAN, R. and WILSON, D.V.: 'Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level', IEEE/ACM trans. on networking, 1997, **5** (1) pp. 71-86

- [Willinger98] WILLINGER, W. and PAXSON, V.: 'Where mathematics meets the Internet', Notices of the AMS, 1998, **45** (8) pp. 961-970
- [Zwart01] ZWART, B., BORST, S. and MANDJES, M.: 'Exact queueing asymptotics for multiple heavy-tailed On-Off flows'. IEEE INFOCOM'01, 2001, Anchorage, USA, **1** pp. 279-288