Accelerated Simulation for Packet Buffer with a Non FIFO Scheduler

By

Sharifah Hafizah Syed Ariffin

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Supervised by Dr John A. Schormans

Department of Electronic Engineering
Queen Mary
University of London

March 2006

To my beloved husband,
and our children Nur Dina, Dura Ain, Dalily and Nabil Ali

A b s t r a c t

Packet traffic in the IP networks has been found to possess self similar characteristics. Modelling IP network traffic with classical queuing system therefore can lead to inaccuracy. Furthermore, differential schedulers such the DiffServ, are very important in providing Quality of Service guarantees to the corresponding class of traffic. However, problems arise in modelling self similar traffic through non-FIFO schedulers where to obtain crucial performance results such as the rare event probabilities (i.e. packet loss , buffer overflow and packet delay) consumes extremely long simulation times.

Simulation acceleration methods allow these rare event probabilities to be found in less time while giving accurate results. This research focuses on providing acceleration for packet switched networks simulation with self similar traffic using non-FIFO scheduler. The method is a development of an existing method called Traffic Aggregation (TA), and it is called Enhanced TA (E_TA).

Acceleration is achieved by simulating only a single equivalent ON/OF source instead of a number of sources, and simulates only the packets that are responsible for building up queues in the buffer, which are called the Excess Rate (ER) packets. The novel technique used in this research is traffic generation of the ER batch, consisting of the ER packets and a non-FIFO scheduler to support two priority traffic levels. The Generalised Ballot Theorem is use to calculate the waiting time of the low priority traffic in the scheduler in order to take account of the high priority traffic that has been removed by the acceleration technique.

Validation of this approach shows that the E_TA method can mimic the behaviour of the conventional simulation. The queuing behaviour is measured by comparing E_TA to TA as well as conventional. Active probing is use where necessary to determine delay probabilities. The results show that a significant acceleration is achieved in E_TA over both conventional and TA.

A c k n o w l e d g e m e n t s

First of all, I would like to thank my supervisor, Dr John Schormans who had given me the guidance, support, critical advice, his time and patience throughout my study. Secondly to my second supervisor Dr Athen Ma, thank you.

Many thanks to the staffs of the Department of Electronic Engineering Queen Mary University of London, Head of Department Professor Laurie Cuthbert, Head of Group Professor Jonathan Pitts and also Lynda, Ho, Chris, Phil, George and many others. My special thanks to Maheen, Jib, Huifang, Vindya and Karen for their friendship.

I am also grateful to my children for their understanding and love. To my husband, M.Fuad, thank you for his endless support, encouragement and advice, who is always there in joy and tears from the beginning till the end of my study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AF	Assured Forwarding
BE	Best Effort
CBQ	Class-based Queuing
CBWFQ	Class-based Weighted Fair Queuing
CoS	Class of Service
DCBWFQ	Distributed Class-based Weighted Fair Queuing
DiffServ	Differentiated Services
DSCP	DiffServ packet Code Point
DWFQ	Distributed Weighted Fair Queuing
E_TA	Enhanced Traffic Aggregation
E_TA-FIFO	Enhanced Traffic Aggregation with FIFO scheduler
E_TA-PQ	Enhanced Traffic Aggregation with Priority Queuing scheduler
EF	Expenditure Forwarding
ER	Excess Rate
F-ARIMA	Fractional AutoRegressive Integrated Moving-Average Process
FE	Fixed Effort
FGN	Fractional Gaussian Noise
FIFO	First-In-First-Out
FS	Fixed Splitting
FTP	File Transfer Protocol
GBT	Generalize Ballot Theorem
HOL	Head of Line
HQ	Hybrid Queuing
IETF	Internet Engineering Task Force
IID	Independent Interval Distributions
IntServ	Integrated Services
IP	Internet Protocol
IS	Important Sampling
KP	Kernel Procedure
LAN	Local Area Network
LRD	Long Range Dependence
NS2	Networking Simulators version 2

OPNET	Optimal Networking Software
PADS	Parallel and Distributed Simulations
PHB	Per Hop Behaviour
PQ	Priority Queuing
QoS	Quality of Service
RESTART	Repetitive Simulation Trials After Reaching Thresholds
RNG	Random Number Generator
RSVP	Resources Reservation Protocol
SLA	Service Level Agreement
SRD	Short Range Dependence
TA	Traffic Aggregation
TA-PQ	Traffic Aggregation with Priority Queuing scheduler
TCB	Traffic Conditioning Block
ToS	Type of Service
VBR	Variable Bit Rate
VoIP	Voice over IP
WAN	Wide Area Network
WFQ	Weighted Fair Queuing

LIST OF MATHEMATICAL SYMBOLS

$e_2(k)$	Batch waiting time distribution
$ST2$	Equivalent service time for low priority ER batch
ρ_1	Load of high priority traffic
ρ_2	Load of low priority traffic
$E[a]$	Mean arrival rate of both high and low priority packet.
$W_2(k)$	Probability a low priority packets has a total wait of k timeslots.
$s_2(k)$	Probability k low priority packets in the queue.
$a_1(k)$	Probability k packets of high priority arrive in any timeslot.
$a_2(k)$	Probability k packets of low priority arrive in any timeslot.
$gp(k)$	Probability that a Pareto batch contains k packets
$\overline{ST_B}$	The mean service time for an E_TA batch of packet arrivals
$\overline{ER}$	The mean size of the excess-rate (ER) batch
$u(k)$	Unfinished work
$V_2(k)$	Probability a low priority batch waits k unit due to the unfinished work of other high and low priority arrivals
$a_c(1,k)$	Probability k units of work of high and low priority arrive in any 1 timeslot
α	The probability of an individual traffic source being active
ρ	Utilization of the conventional model
B	The mean size of an ER batch in E_TA
C	Service rate (packets/ timeslot)
$Doff$	Mean OFF period for individual traffic
$Doff_{BG}$	Mean OFF period of the background traffic
$Doff_{FG}$	Mean OFF period of the foreground traffic
Don	Mean ON period for individual traffic
Don_{BG}	Mean ON period of the background traffic
Don_{FG}	Mean ON period of the foreground traffic
H	Hurst parameter
N	Number of sources aggregated by the acceleration method

N_{BG}	The number of aggregated background traffic sources
N_{FG}	The number of aggregated foreground traffic sources
q	Probability of there being a batch arrival in any random timeslot.
R	Arrival rate of individual packet (packet/ timeslot)
R_{BG}	The packet arrival rate of the background traffic per timeslot
R_{FG}	The packet arrival rate of the foreground traffic per timeslot
$Roff$	Arrival rate in the OFF period (packet/ timeslot)
Ron	Arrival rate in the ON period (packet/ timeslot)
T_{BG}	The mean total packet rate of the foreground traffic
T_{FG}	The mean total packet rate of the foreground traffic
$Toff$	Mean OFF period for aggregated traffic
$Toff_2$	Mean OFF period in E_TA
Ton	Mean ON period for aggregated traffic

Chapter 1

INTRODUCTION

1.1 Overview

The increasing usage of IP (packet-switched) networks has greatly expanded the importance of studying their behaviour, and the performance of the network's essential components, such as buffers. However, difficulties occur due to the complex characteristics of IP traffic, and the various applications using the IP networks.

Mathematical and simulation techniques are two general ways to evaluate buffer performance. Mathematical modelling is complementary to experiment results. Using mathematical techniques, results can be obtained faster if a mathematical algorithm can be used to describe the behaviour of the system under study. However, when it comes to modelling complex networking scenarios mathematical analysis may become intractable.

Simulation modelling uses a computer program that mimics the real system, or an abstraction of the real system. It is seen as an alternative technique that is able to predict performance prior to implementation, and can model very complex network scenarios. However, the drawback of simulation modelling is that it requires a large expenditure of time to obtain accurate results of interest.

Experiment evidence from ground breaking studies discovered self similar patterns in IP traffic [Lela94, Paxs95, Will97]. Self similar traffic is very different from e.g. the traditional voice traffic. Such traffic is highly variable and might have extremely long bursts of continuously arriving packets. This type of traffic is well described using heavy tailed distribution, but they have properties that make simulation hard to achieve because simulating usually requires a long period of time to converge to steady-state. Even at steady-state high variability can still be observed [Crov97].

Accelerated simulations use methods introduced to decrease simulation time without neglecting the performance accuracy when compared to the non-accelerated method. The existing simulation acceleration methods include Important Sampling (IS) [Heid95, Town98], Repetitive Simulation Trials After Reaching Thresholds (RESTART) [Vill94] and Traffic Aggregation (TA) [Scho00a, Ma03a].

In this research we are focusing on developing an equivalent model of the conventional packet buffer that can produce an output analysis (which in this case will be the steady state probability) in less time. This acceleration simulation method is a further development of the TA technique, which had previously been applied to FIFO buffers only.

1.2 Objectives

The first objective of this research is to develop a method to accelerate simulation of power-law traffic. Prior work on acceleration techniques tend to focus almost exclusively on FIFO buffering. The growing use of multimedia applications in IP networks now needs a form of priority scheduling to support different classes of services.

The second objective is to test our method against conventional (packet-by-packet) simulation. This should show that the new method does not significantly lose accuracy. It should also show significant speed up. In order to compare performance with the conventional method, we analyse the delay distribution using active measurement, and the queuing behaviour of the buffer at steady state.

1.3 Contribution of this Research

In this research work, we have developed an accelerated simulation method called Enhanced Traffic Aggregation or E_TA. This method reduces processing time especially in obtaining steady state output. Acceleration is achieved by:

- Replacing a number of traffic sources with a single equivalent ON/OFF source that generates Excess-Rate batches instead of individual packets.

- Using the Generalized Ballot Theorem (GBT) to calculate the waiting time of the low priority traffic in the priority buffer which allows the removal of the high priority traffic and thus provide a further speedup.

1.4 Layout of the Thesis

In Chapter 2 we review the scheduling that is implemented in IP networks, focusing on DiffServ. The groups of traffic defined in DiffServ are briefly presented as is the scheduler discipline that is used in DiffServ.

Chapter 3 presents the significance in considering self similar IP traffic and overviews the concept of self similarity in terms of its mathematics. It also gives a brief explanation of heavy-tailed distributions, power-law decaying queues and explains the ON/OFF model used in this thesis.

Chapter 4 details the simulation techniques and approaches used in this research. The types of simulation models in general are also briefly explained. The essential part of simulation, which is the random number generator is presented and the types of the output analysis that are needed. This chapter also covers some the existing accelerated simulation methods in the literature. The Traffic Aggregation (TA) accelerated simulation technique for power law traffic is briefly explained, and this leads to the introduction of Enhanced TA; E_TA with FIFO buffer (E_TA-FIFO). This chapter also presents the queuing behaviour at packet scale, burst scale and the queuing of the Excess-Rate (ER) packets. Active measurement is introduced as in this thesis it will be used as a means to compare to conventional simulation.

Chapter 5 explains the development of the proposed accelerated simulation method using FIFO buffer, E_TA-FIFO. The numerical results of the packet delay distribution of the conventional and E_TA-FIFO are compared using active measurement, also the state probabilities of the E_TA-FIFO with the existing accelerated simulation method, TA are compared.

Chapter 6 focuses on the priority buffer used with E_TA, E_TA-PQ. A hybrid method is introduced in the E_TA algorithm that involves the Generalized Ballot Theorem (GBT). The algorithm developed calculates a waiting time for the low priority traffic in the E_TA-PQ model; this is obtained prior to the simulations.

In Chapter 7 we apply the E_TA method to end-to-end network paths, by splitting the overall traffic into foreground (FG) and background (BG) traffic. This chapter is divided into two: end-to-end connections using E_TA-FIFO and end-to-end connections using of E_TA-PQ. In the first section, instead of N BG traffic sources, E_TA method is applied to act as the ambient traffic that passes through the buffers. In the second section we present a two priority network path in which the high priority traffic is the BG traffic. The idea of this is to apply the E_TA-PQ (of 1 node) in chapter 6 across a model of a network path.

Chapter 8 gives conclusions and direction for further work.

Chapter 2

PACKET SCHEDULING SCHEMES

2.1 Introduction

This chapter presents a survey of the packet scheduling schemes relevant to this research. The use of IntServ and DiffServ in IP networks is discussed. The three types of Class-of-Service (CoS) in DiffServ (i.e. Expedited Forwarding, Assured Forwarding and Best Effort) are explained. Some of the scheduler disciplines that are supported in DiffServ to provide Quality of Service (QoS) are also detailed.

2.2 Previous Approaches to Solve the QoS Problem

The internet, which was originally designed only to support best-effort packet forwarding, is now expanding rapidly. Priority queues are becoming very important in IP networks due to the heterogeneity of the network customers. The Internet Protocol (IP), which is the underlying 'transport mechanism' of the internet, is currently one of the enabling technologies for multi-service applications. Different applications have varying needs for delay, jitter or loss, and these parameters form the basis of their required QoS. For example, real-time applications such as audio and video streaming requires very low loss, jitter and delay. Whereas a file transfers application can tolerate both delay and jitter (see Appendix 3 for more detail). So to guarantee service in an IP network, such a network has to provide CoS beyond the best-effort service to multiple types of traffic.

Originally the Internet Engineering Task Force (IETF) defined two models to facilitate end-to-end QoS in IP networks: Integrated Services (IntServ) and Differentiated Services (DiffServ). The IntServ model requires a module in every IP router along the path to reserve resources for each transmission, and then ensure that each data packet in transit is checked to see what resources it is entitled to receive. The reservations are requested using a specially designed protocol called Resources Reservation Protocol (RSVP). Admission control (which is a process that controls the traffic allowed into the network) is important in IntServ. However if an RSVP request fails, the transmission will not start, or will only provide degraded service. There are two disadvantages in IntServ [Carp02, Cisc49], which are:

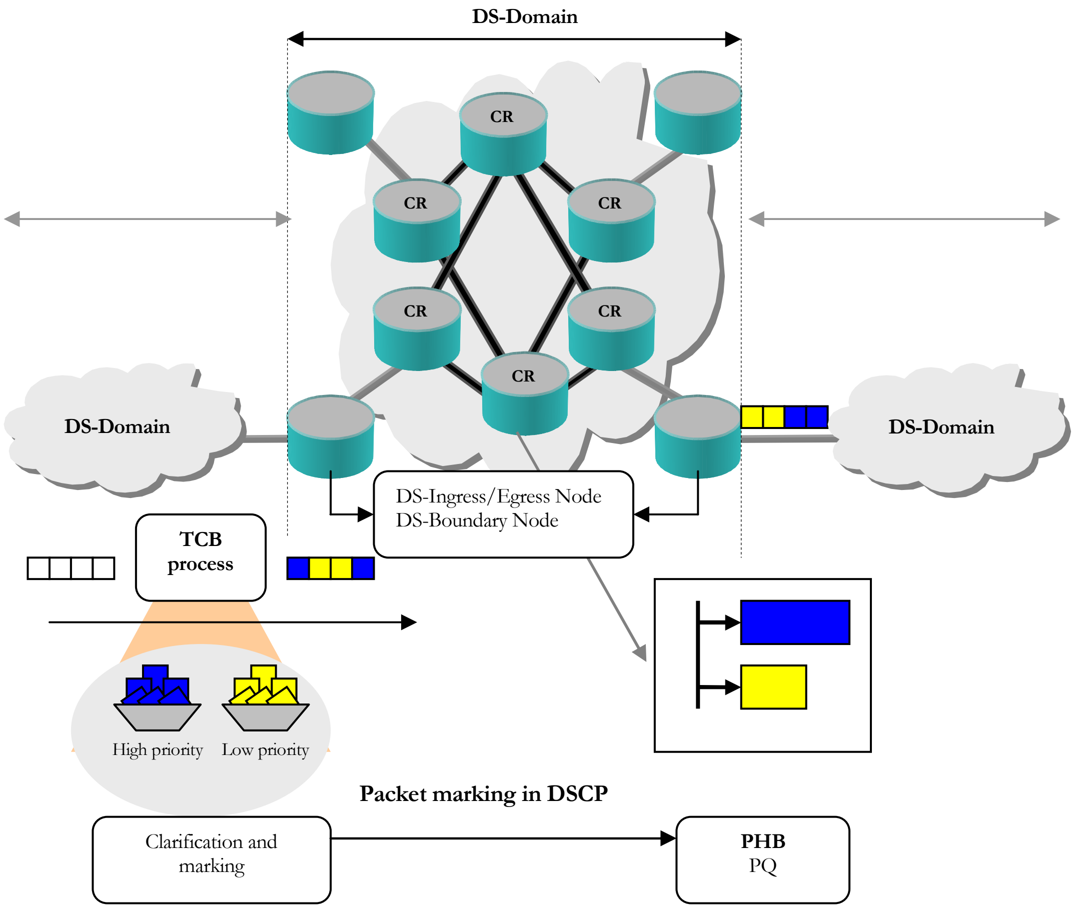
- Nonscalable - Every device along the packet's path, including the end systems such as servers and PCs, need to be fully aware of RSVP and capable of signalling the required QoS.
- Nonworkable - State information for each reservation needs to be maintained at every router along the path. If IntServ were to be used by a major internet service provider (ISP) on a trunk connection, carrying millions of packets per second, the overhead per packet of implementing the necessary checks and resource management is widely believed to be unacceptable.

Hence, IntServ has failed to be adopted widely because it is non-scalable and non-workable administratively.

2.3 Differentiated Services (DiffServ)

More recently the IETF has taken another approach, which is DiffServ. Currently DiffServ is being finalized by the IETF and implemented by various vendors [Carp02]. In DiffServ, the individual flows are aggregated together then treated on a per-class basis. The incoming packets are classified by the router/gateway at the edge of the DiffServ network into the defined service classes by examining the packet header.

Figure 2.1 gives a pictorial overview of the DiffServ (DS) architecture. A DS-Domain is made up of DS Ingress nodes, DS Interior nodes (core router) and DS Egress nodes. Further, an Ingress or Egress node might be a DS Boundary node, connecting two DS- Domain together.



CR – core router
TCB – traffic conditioning block
PHB – per-hob behaviour
DSCP – Diffserv packet code point

Figure 2.1: DiffServ architecture

Typically, the DS Boundary node performs traffic conditioning[1] at the Traffic Conditioning Block (TCB) where it classifies the incoming packets into a pre-defined aggregate, meters them to determine compliance to traffic parameters (whether, the traffic are in profile or out of profile), marking each packet appropriately (e.g. in Figure 2.1, blue for high priority and yellow for low priority), and finally shapes (buffers to achieve a target flow rate) or drops the packets in case of congestion. The packet marking at DS boundary nodes uses the Type of Service (ToS) field of the packet header for IPV4[2]. As for IPV6,

---

[1] Traffic conditioning has four essential components: *markers*, which place or change the DSCP mark in packet header; *meters*, (or policing) which measure traffic and check it against traffic profiles; *shapers*, which delay traffic to make it conform to a certain traffic profile; and *droppers*, which simply drop packets. These four components work together.

[2] The header of IPV6 is larger than IPV4, to cope with the ever growing user of the internet

the packet marking is in the DS field. The DS field has 8 bits, however only 6 bits are used and the DS field is called a DiffServ Packet Code Point (DSCP). This field enables each router to determine the traffic class.

A DS core router enforces the appropriate per-Hop behaviour (PHB) by employing policing or shaping techniques. Policing or shaping behaviour of a node is configured by a Service Level Agreement (SLA)[3] or policy[4]

To date, three standard PHBs are available to construct a DiffServ-enabled network and implement end-to-end CoS; Expedited Forwarding, Assured Forwarding and Best Effort.

- Expedited Forwarding (EF) [Jaco99]
  This is similar to the guaranteed service class associated with IntServ. EF PHB is the key means by which DiffServ provides a low loss, low latency, low jitter and assured bandwidth service. EF can be implemented using priority queuing (PQ) (refer to Appendix 1). EF PHB is especially suitable for applications such as voice over IP (VOIP), video and online trading applications.
- Assured Forwarding (AF) [Hein99]
  AF PHB provides services with a minimum rate guarantee and a low loss rate. Four AF classes namely AF1, AF2, AF3, and AF4 are defined, and each class has three levels of drop precedence[5]. AF PHBs are suitable for network management protocols such as Telnets, SMTP, FTP and HTTP.
- Best Effort (BE)[Nich98]
  All IP packets belonging to the BE class are not policed and get the 'traditional' best effort service from a DS compliant node[6].

[3] SLA is a means of a formal service definition between two organizations, usually between a supplier of services and its customer.

[4] An example of policy might be that the high priority traffic gets 20 percent and low priority traffic gets 80 percent of the bandwidth.

[5] Drop precedence is a concept of priority within the AF classes

[6] Compliant node: A network node that complies with the entire core DiffServ requirements

The implementation of PHB relies on the scheduling and queuing schemes used in DiffServ compliant switches and routers. In order to provide premium service to EF traffic, packets belonging to the EF class should be served prior to packets belonging to any other class (AF and BE). On the other hand, to avoid starvation of the other classes, a minimum service rate should be guaranteed. In practice, networks need scheduling and queuing schemes which are efficient in providing differentiated services for different traffic classes with high throughput, and simple in implementation.

DiffServ supporting scheduling schemes including Class-based Weighted Fair Queuing (CBWFQ) [Buch01], Class-Based Queuing (CBQ) [Jaco99] and Priority Queuing (PQ) [Bona01, Quan03].

2.4 The DiffServ Scheduler

A scheduling discipline selects a packet out of the queue whenever the server becomes available. As most experts agree, future applications will carry at least two types of traffic [Kesh97]. In packet switched networks, scheduler disciplines are most important because they are the key to fairly sharing network resources and provide performance-critical traffic, such as telephony and interactive multi-participant applications (i.e. net meeting), with performance guarantees. A scheduler discipline has two components: it decides the order in which packets are serviced, and manages the queues of packets awaiting services. Scheduling disciplines create different delays for different users by their choice of service order. The following are some of the scheduler disciplines that may find use in current and future packet networks.

2.4.1 Priority Queuing (PQ)

PQ ensures that important traffic gets the fastest possible handling at each node where it is used. It was designed to give strict priority to important traffic. However PQ can flexibly prioritize according to network protocols (e.g. IP, IPX or AppleTalk), incoming interface, packet size, source or destination addresses and so on. In PQ each incoming packet is placed in the appropriate queue based on an assigned priority. The scheduler gives higher priority queues absolutely preference over low priority queues. There are two types of PQ; non pre-emptive and pre-emptive (see Appendix 1).

2.4.1.1 Non Pre-emptive PQ

Non pre-emptive PQ, also known as the Head-Of-Line (HOL), is simple and the most fundamental in priority queuing. The HOL priority queuing system was first studied by [Cobh54]. HOL is also known by the name of strict priority queuing or fixed priority queuing. The HOL structure is shown in Figure 2.2, in which packets are separated according to priority groups: higher priority (blue) and low priority (yellow). Once a packet has commenced service, the packet will not be interrupted by a higher priority packet that arrives after the service has started. The higher priority packet will join the waiting line ahead of any lower priority packets present and within the same priority class packets are served first-in-first-out (FIFO). Average waiting time in the system is obtained by the basic relationship that time-in-system is service time plus waiting time. This applies to each class.

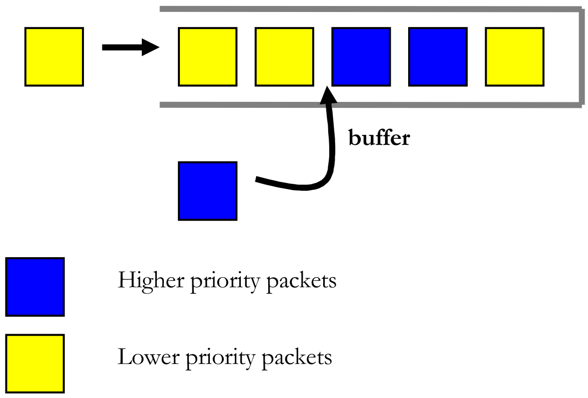


Figure 2.2: Head-of-line priority queue.

2.4.1.2 Pre-emptive PQ

With pre-emptive PQ, a higher priority packet will pre-empt the server if the server is dealing with a lower priority packet. There are two types of pre-emptive PQ; pre-emptive resume and pre-emptive restart (refer Appendix 1). The pre-emptive resume is illustrated in Figure 2.3.

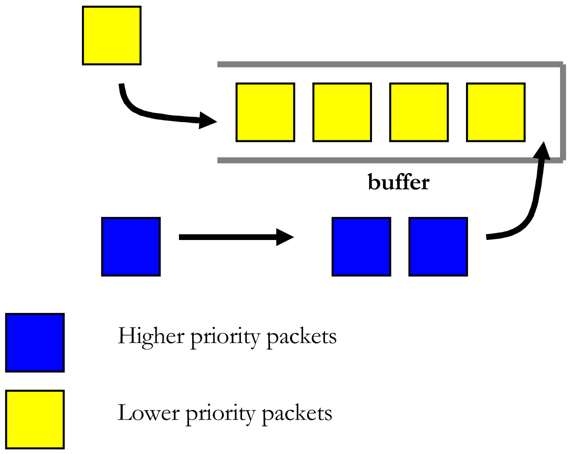


Figure 2.3: Pre-emptive priority queue.

It has been suggested that delay sensitive traffic such as VoIP needs to be given strict (static, HOL) priority, all along the packet path [Quan03]. To make this happen in this thesis, the PQ-HOL scheduler is used as the scheduler of the priority buffer.

2.4.2 Class Based Queuing (CBQ)

CBQ divides traffic into a hierarchy of classes, based on a combination of e.g. IP addresses, protocol or application types. CBQ can be used to meet business priorities, e.g. a company's accounting department may not need the same Internet access privileges as the engineering department.

CBQ is sometimes known as hierarchical link-sharing in resource management research [Floy95], as it distributes bandwidth on local links in response to local needs. CBQ can be either static (permanently assigned by network administrator) or dynamic (varying in response to current conditions on the network, according to some predetermined algorithm).

Figure 2.4 shows CBQ specifying link-sharing by allocating bandwidth to each class (expressed as a percentage of the overall link bandwidth). CBQ provides an algorithm that enables bandwidth to be allocated to each class over some interval of time, in periods of congestion. As a consequence, in times of congestion, some classes will be restricted to their minimum allocated bandwidth.

The zero percent for mail in Figure 2.4 means that the bandwidth received by this class is determined by other scheduling mechanisms at the gateway; however during congestion the class is not guaranteed any bandwidth.

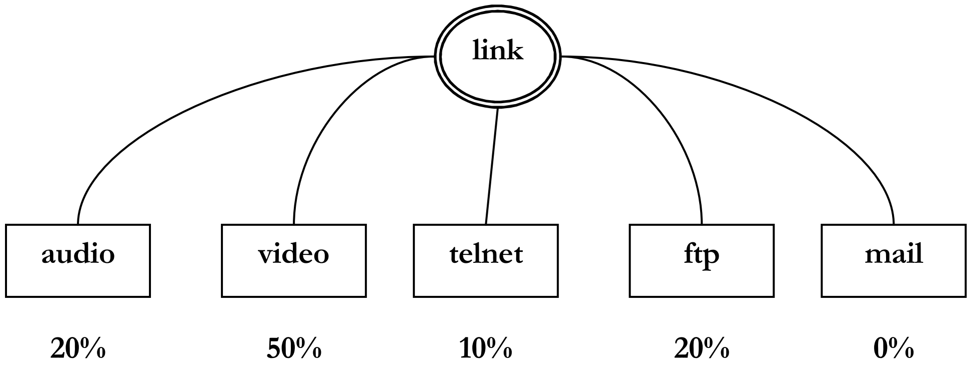


Figure2.4: A link-sharing structure in CBQ

2.4.3 Class-Based Weighted Fair Queuing (CBWFQ)

CBWFQ is one of the four types in the WFQ family (WFQ, DWFQ, CBWFQ and DCBWFQ). CBWFQ supports used-defined traffic classes in which traffic classes are defined based on customer and commercial requirements [Cisc49]. All incoming packets that satisfy the matching criteria according to the defined class will be put into the appropriate queue. Each queue of the same class will be served as First-In-First-Out (FIFO). After the classes are defined, a user can assign the characteristics to the bandwidth: weight and maximum packet limit (the number of packets allowed to accumulate in the queue for the class). The bandwidth assigned to each class will be guaranteed during congestion.

For CBWFQ, the weight assigned for each class becomes the weight of each packet in the class. The weight for a packet belonging to a specified class is derived from the bandwidth assigned to the class (when it was configured).

For example Figure 2.5 shows a system using CBWFQ. The classifier reads the DSCP in the IP header to determine packet class. Each queue i (i=1, 2, 3) is assigned a weight ϕ_i. For example, the capacity for class 1 can be found using

$$SR_{CBWFQ}1 = \phi_1 \Big/ \sum_i \phi_i \cdot C \qquad (2.1)$$

where C is the total output capacity.

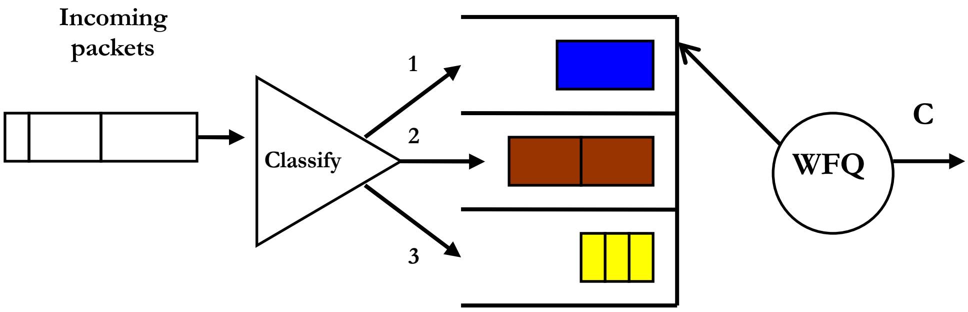


Figure 2.5: Class-based WFQ

2.5 Summary

This chapter has reviewed the importance of differential services in IP networks. Also it briefly overviews some of the scheduler discipline use in DiffServ. The non preemptive PQ scheduler will be use in the priority buffering of this thesis.

Taking consideration of the experimental evidence in [Will97, Lela94], in this thesis, the IP traffic is divided into two major classes: a high priority class (i.e. EF PHB) and a low priority class (i.e. AF PHBs and BE PHB). The phrases 'high priority' and 'low priority' will be use in this thesis, in the sense that packet with a high priority will be given preference over low priority. When it comes to analysis, the classes of 1 and 2 will be label, meaning that class 1 has high priority and class 2 has low priority. PQ-HOL is used as simpler method to achieve this priority buffer (Note: Section 8.2 of further work will point to the extension of this issue).

The following chapter covers material related to the concept of self similarity and power law traffic.

Chapter 3

SELF SIMILARITY IN PACKET TRAFFIC

3.1 Introduction

Traditionally the Poisson process has been used to model a lot of network traffic because of its inherent simplicity (M/M/1 or M/D/1) and that it has been formed to provide a satisfactory level of accuracy in traditional packet networks like X.25. Studies of LAN traffic [Lela94] and WAN traffic [Paxs95] however argues that IP network traffic is better modelled using *self similar processes*. This chapter overviews the significance in considering self similar characteristics in modern packet traffic.

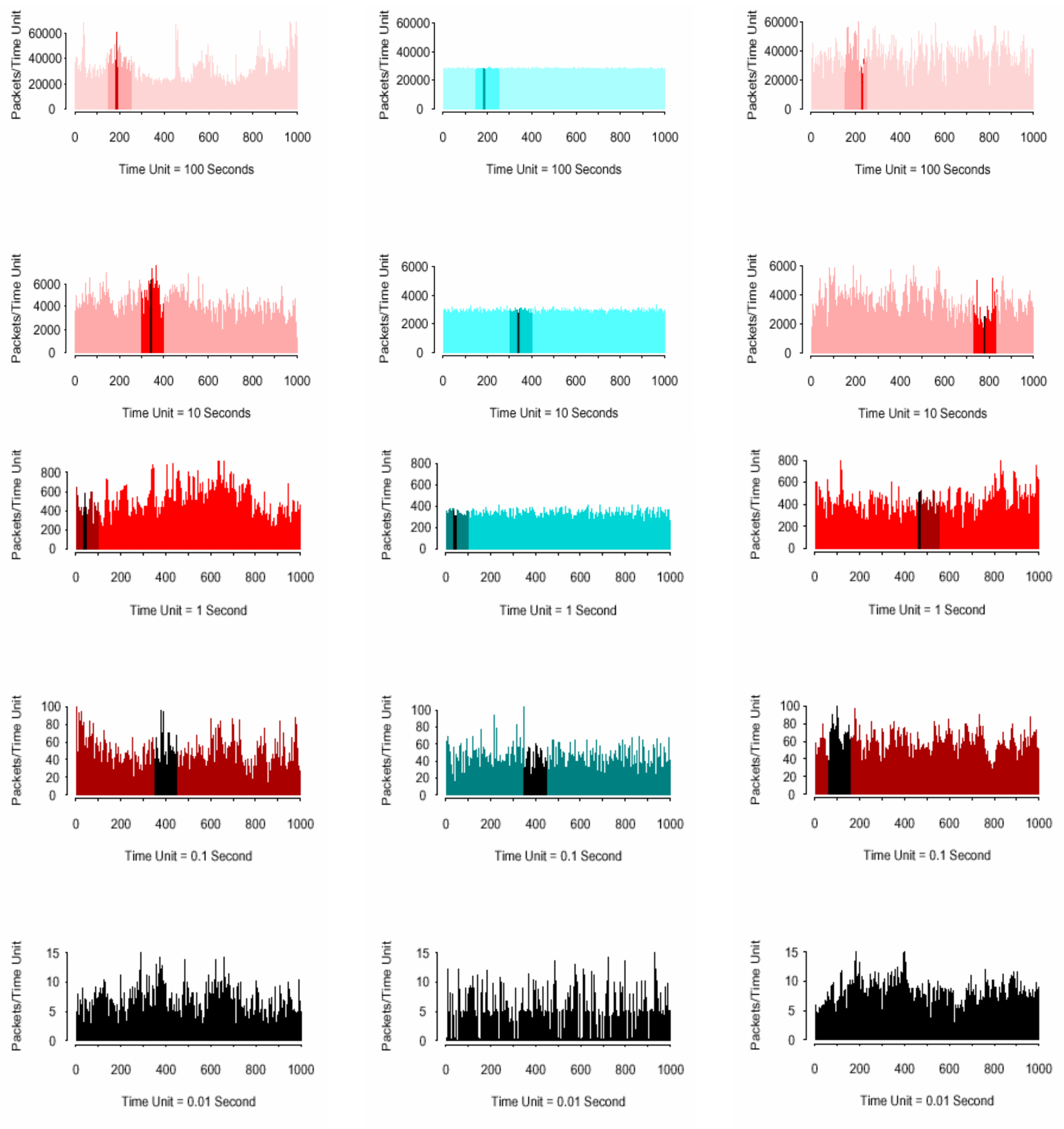
This chapter presents the mathematical definitions of the self similarity and also briefly discusses the models for generating self similar traffic.

3.2 The Existence of Self Similarity in the IP Traffic

The analysis of the Ethernet[7] traffic in [Erra00, Lela94] showed that the aggregated traffic of LAN traffic becomes more bursty as the number of active sources increases. This is in complete contrast to a Poisson process which rapidly smoothes to the mean value. The degree of self similarity, which typically depends of the utilization of the Ethernet, can be defined via the Hurst parameter (also known as parameter *H*). The term self similar was first introduced by Mandelbrot mainly through application, in areas of incomes and hydrology. Early application of the self similarity concept as applied to communication system is reported in [Mand65].

Self similarity is described as a traffic pattern that is bursty on many, or all, of the timescales. Contrast to the Poisson process which creates traffic that is bursty over a fine time scale but smooth over a large timescale. Figure 3.1 shows three sets of traffic traces: a) the actual Ethernet traffic, b) synthetic trace of Poisson traffic representing the

[7] Ethernet is a frame based computer networking technology for local area network (LANs)



a) Measured Traffic **b) Poisson Model** **c) Self Similar Model**

Figure 3.1: The comparison of the real and synthetic Ethernet traffic [Will97].

traditional traffic modelling and c) synthetic trace of self similar traffic. All of the traces are on five different time scales. We can see that the Ethernet traffic shows similar burst patterns on all of the time scales. The synthetic traces of the self similar traffic show the same behaviour as the measured traffic. Figure 3.1 also illustrates pictorially what is meant by the existence of self similarity in today's complex network traffic. Many subsequent researchers have taken this discovery very seriously [Lope00, Bera92].

3.3 Stochastic Self Similarity and Long-Range Dependence

The concept of self similarity is of a phenomenon that looks or behaves in some sense 'the same' when observed at different scales on a dimension. The dimension can be space (length, width) or time. In the case of data communication the dimension is time (seconds, timeslots). A Cantor set provides a graphical way to understand the concept. Figure 3.2 illustrates the construction of the Cantor set which obeys the following procedures [Stal98]:

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 line created by the preceding step.

These steps essentially form a recursive process that is more precisely defined by assuming S_i is the Cantor set after *i* levels of recursion. Starting with the unit interval:

$$
\begin{aligned}
S_0 &= [1,0] \\
S_1 &= [0, 1/3] \cup [2/3, 1] \\
S_2 &= [0, 1/9] \cup [2/9, 1/3] \cup [2/3, 7/9] \cup [8/9, 1]
\end{aligned}
$$

and so on.

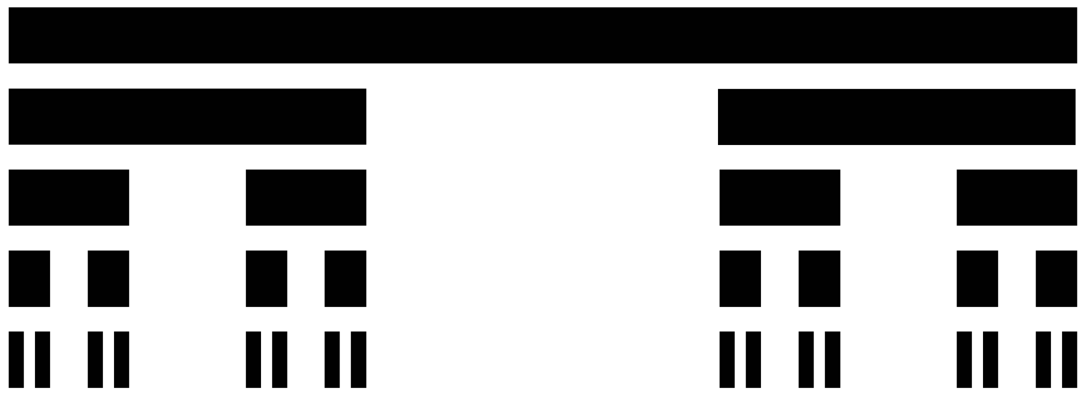


Figure 3.2: A Cantor set of four levels of recursion

The Cantor line can be used to represent a time series and each successive step magnifies the time scale by a factor of three. In Figure 3.2 we can see that at every step, the left and right portion of the set is a replica of the full set in the preceding step. The Cantor set reveals two important properties:

1. The structure has arbitrarily small scales. If a part of the set is magnified repeatedly, we will see a complex pattern of points separated by gaps of various sizes. However, continuous repeated magnification will make the structure becomes more featureless[8].
2. The structure contains smaller replicas of itself at all scales.

These properties do not hold indefinitely for real phenomena because at some point under magnification, the structure and the self similarity break down (e.g. an upper limit to hard disk size). However, over a large range of scales, many phenomena exhibit self similarity.

[8] The details of the structure are lost during the process of repeated magnification.

3.3.1 The Mathematics of Self Similarity

The self similarity concept can be described using mathematics. Considering a stationary discrete time stochastic process, let $X = (X_t : t = 1,2,...)$. A new time series $X^{(m)}$, can be obtained by averaging the original series *X* over non overlapping blocks of size *m* and $X^{(m)}$ may be express as

$$X_k^{(m)} = \frac{1}{m}\left(X_{km-m+1} + ... + X_{km}\right) \qquad k \geq 1 \tag{3.1}$$

$X^{(m)}$ is the aggregate process of the time series, *X*. The aggregated time series can be view as a technique for compressing the time scale. $X^{(1)}$ can be considered as the highest magnification of the highest resolution possible for this time series. For example X^3 can be express as

$$X_k^{(3)} = \frac{X_{3k-2} + X_{3k-1} + X_{3k}}{3} \tag{3.2}$$

This is the same process reduced in magnification by factor of 3. By averaging over each set of three points, the fine details available at the highest magnification are lost. To determined whether process *X* is self similar is by using statistical measure such as variance and autocorrelation.

The parameter *H*, is a key measure of self similarity, and parameter *H* is in the range of $0.5 < H < 1$. A value of *H* =0.5 indicates the absence of self similarity and, the self similarity becomes stronger as the *H* value is closer to 1. Parameter *H* is related to the heavy tailed property i.e. power law decay *a* where $H = (3-\alpha)/2$; therefore once *H* is measured *a* can be calculated (*H* can be measured using R/S plot [Bera95])

The process *X* is said to be <u>exactly second order</u> self similar with parameter $H = 1 - \beta/2$ if the corresponding aggregated process $X^{(m)}$ have the same variance and autocorrelation as *X*. This means that for all *m* = 1, 2, 3,… , we have variance of

$$Var\left(X^{(m)}\right) = \frac{Var(X)}{m^2} \tag{3.3}$$

and autocorrelation function of

$$r^{(m)}(k) = r(k) \tag{3.4}$$

where $k = 1, 2, \ldots$.

The process X is said to be <u>asymptotically second order</u> self similar with parameter $H = 1 - \beta/2$ if for all k large enough, we have variance of

$$Var\left(X^{(m)}\right) = \frac{Var(X)}{m^2} \tag{3.5}$$

and autocorrelation function of

$$\lim_{m \to \infty} r^{(m)}(k) = r(k) \tag{3.6}$$

where $m \to \infty$ and $k = 1, 2, \ldots$.

Hence, this definition of self similarity means that the autocorrelation process of the aggregated process has the same form as the original process. This would suggest that the degree of variability, or burstiness would be similar at different time scales.

A traffic stream with self similar characteristics is a produce of aggregated Long Range Dependence (LRD) sources. These LRD sources are exhibited by individual communication process with heavy-tailed distributions sojourn times.

3.3.2 Heavy Tailed Distributions

A distribution is said to be heavy-tailed if

$$P[X > x] = 1 - F(x) \approx x^{-\alpha} \tag{3.7}$$

where $x \to \infty$ and $0 < \alpha < 2$

The simplest heavy- tailed distribution is the Pareto distribution. The Pareto distribution (also referred to as the power-law distribution, the double-exponential distribution, and the hyperbolic distribution [Paxs95]) has been used to model frame sizes of variable-bit-rate (VBR) video [Garr94] and CPU activity process [Lela86]. Because of the simplicity and wide acceptance use, we make considerable use of the Pareto distribution.

The Pareto distribution has probability distribution function given by

$$F(x) = 1 - \left(\frac{\delta}{x}\right)^{\alpha} \tag{3.8}$$

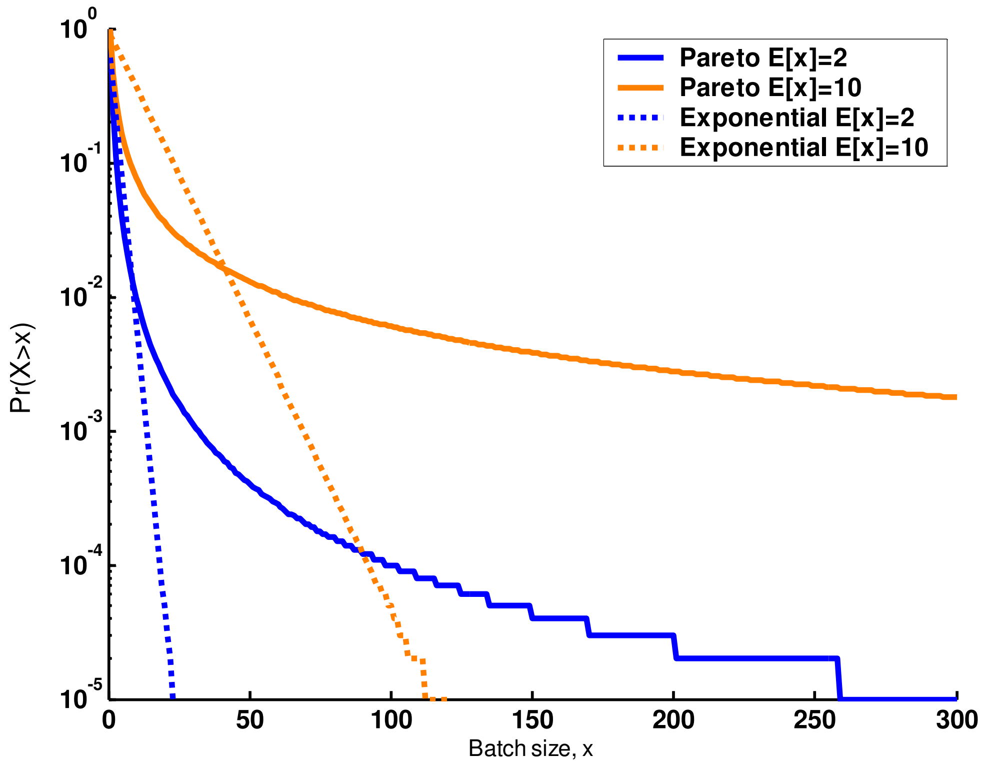
The probability density function given by

$$f(x) = \frac{\alpha}{\delta}\left(\frac{\delta}{x}\right)^{\alpha+1} \tag{3.9}$$

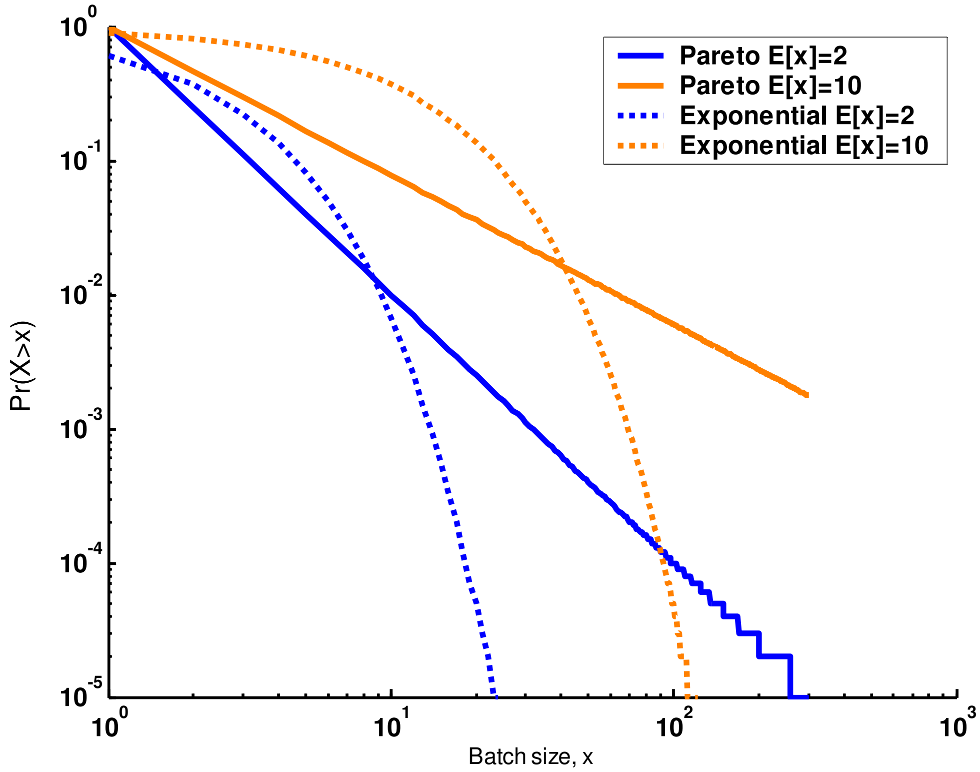
where α is the shape parameter which determines the characteristic 'decay' of the distribution (tail index) and δ is the location parameter which defines the minimum value of X (the random variable). The parameter α also determines the mean and variance of the distribution. The distribution has infinite variance and infinite mean if $1 \leq \alpha \leq 2$. The mean value of the Pareto distribution is

$$E[x] = \delta \cdot \frac{\alpha}{\alpha - 1} \tag{3.10}$$

Note that for this formula to be correct $\alpha > 1$ or otherwise the Pareto distribution has an infinite mean.



a)



b)

Figure 3.3: Comparison of heavy-tailed and exponential probability density function [Pitt00]

Figure 3.3 a) shows a comparison between Pareto and exponential probability density distributions on a log linear scale (reproduced from [Pitt00]). In this experiment, $\delta = 1$ and the mean value of the Pareto distribution, E[x] = 2/1 and 10/9. On this scale, the exponential density function (dashed lines) is straight, reflecting the exponential decay of the distribution. The Pareto distribution (solid lines) has tail that decays much more slowly than an exponential.

Figure 3.3 a) clearly illustrates the significance of considering highly variable traffic in modelling internet traffic or most importantly in buffer dimensioning. The practical significant is that the effect of using a Pareto distribution is that the buffer fill may become very large due to extremely long bursts, which is very different to classical queuing systems, i.e. with Poisson traffic. With Poisson traffic, queues in the buffer may build up over a short run but will clear out over the long run. Because buffers in system of servers and queues are designed in expectation of long-term smoothness, only modest-sized buffers are used. However, if the behaviour of the traffic is very bursty (i.e. self similar), then queues in the buffer may build up more than would be expected. In Figure 3.3 a) and b) the 'batch size' can also be interpreted as 'queue size'.

3.3.3 Power-Law Decaying Queues

The term 'power-law' is used to described the decay characteristics of these distributions (see Figure 3.3 b)), this is described by

$$\log y = -b \cdot \log x + \log c \tag{3.11}$$

b is the gradient or slope of the straight line which is restricted in range of $1<b<2$ (in order to generate self similar traffic), and *c* is a constant.

If on both sides of (3.11) we take the inverse log, we have,

$$\begin{aligned} y &= 10^{-b \cdot \log x + c} \\ &= 10^{-b \cdot \log x} \cdot 10^{c} \end{aligned} \tag{3.12}$$

since c is a constant, 10^c is also a constant and we obtain,

$$\begin{aligned} y &= c \cdot 10^{-b \cdot \log x} \\ &= c \cdot \left(10^{\log x}\right)^{-b} \\ &= c \cdot x^{-b} \end{aligned} \tag{3.13}$$

The gradient, b, therefore corresponds to the characteristic parameter, α of a power-law distribution. In this thesis most of the numerical results are plotted on a log-log scale for better visualization of the power-law decay.

3.4 Self Similar Traffic Modelling

There are several ways of modelling self similar packet traffic. Some of the common models in the literature are the M/G/∞ queue model, Fractional Gaussian Noise (FGN), Fractional AutoRegressive Integrated Moving-Average (F-ARIMA) process, chaotic maps and ON/OFF models.

In the M/G/∞ queue model, the customers arrive according to a Poisson process and have service times drawn from a heavy-tailed distribution with infinite variance. Solving the M/G/∞ model shows that multiplexing heavy-tailed distribution for connection lifetimes results in self similar traffic [Paru97]. With G as the service time distribution, many forms of time dependence can be modelled. The M/G/∞ model with an (integer) log-normal service time can be used to model traffic such as Telnet and FTP connections [Paxs95].

FGN and F-ARIMA processes are another two mathematical models that can represent self similar phenomena. In [Will95a], FGN and F-ARIMA processes are fitted to model traces of Ethernet traffic. FGN is characterized by its mean μ, variance δ^2 and H. F-ARIMA processes are much more flexible because they have the ability to capture both SRD and LRD correlation structure in time series modelling.

Chaotic maps can be use to model self similar ON/OFF sources. The maps generate traffic (in the ON or/and OFF periods) with a non- integer fractal dimension and long-range dependence. A map is a combination of a deterministic map and random process.

Modelling self similar traffic using a multiplex of ON/OFF sources was originally suggested by [Mand69], and then also applied by [Will97] to model Ethernet traffic. This approach is commonly considered in the communications literature and has been found to give good agreement with measured traffic [Will97]. The self similar process used in [Will97] has strict alternating ON and OFF periods, and may have different distributions for ON and OFF periods. The superposition of many ON/OFF source with heavy-tailed sojourn times produces a self similar aggregated process.

The disadvantage of using FGN and F-ARIMA is that they may involve tedious mathematical calculations. M/G/∞ in particular is not used because of the infinite server model which is not usually ideal in modelling packet networks (particularly abstraction of system is used in this thesis). In this thesis ON/OFF models are used, and the ON-OFF model is explained in more detail in the next section.

3.4.1 The ON/OFF model for Packet Traffic

Traditional ON/OFF source models typically assume an exponential or a geometric distribution for their ON and OFF periods. These models have been widely used in queuing and performance analysis, because of their mathematical simplicity [Jain86]. However, as we have seen more recent research has shown clearly that multiplexing a large number of traditional ON/OFF sources is inconsistent with the measured traffic from real networks [Ma04, Lela93, Lela94]. The superposition of many ON/OFF models with heavy tailed sojourn times however is consistent with measured packet traffic. These studies have motivated others in using ON/OFF models in modelling the modern traffic network [Crov97, Grib98, Schw01, Will97, Ma02c]. Figure 3.4 illustrates a view of an ON/OFF process.

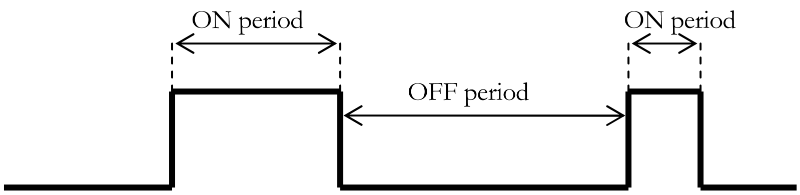


Figure 3.4: Illustration of the ON/OFF source.

The ON/OFF traffic model with a Pareto distribution for both the ON and the OFF periods may have separate *H* values for ON and OFF periods. However, 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 higher values of *H* in the OFF component. Therefore OFF periods dominate the final overall value of *H*. The *H* value also is related to the queuing performance as a high value of *H* is often associated with large queue size. However, if the measured LRD traffic has strong *H* in the OFF period but an insignificantly small *H* in the ON period, then the resulting queue length will not be power law. This is shown in [Neid98], where the value *H* itself may not accurately predict the queuing performance of self similar traffic. This is very important for the research reported in this thesis as the future work, extending the work in this thesis may have to parameterise the ON/OFF model from measured *H* values.

In this research, *H* is only used as an LRD index, which is maintained in the range of $0.5 < H < 1$. The effects towards the overall queuing behaviour are not predicted directly from *H* as this has been shown to be highly misleading and potentially incorrect. In fact, studies in [Mond01, Samu99] have discovered that the ON period is more important than *H* for teletraffic engineering in terms of overall queuing behaviour. This is essential in this research where we are more interested on the impact of the ON period to the queue size of the buffer. In this research, the ON period uses a Pareto distribution and the OFF period is an exponential distribution, this is justified by [Mond01, Samu99].

3.5 Generating Pareto Distributed Random Numbers

In traffic modelling, random numbers are generated from the uniform distribution on the interval of 0 to 1; this distribution is denoted *U* [0, 1]. In this thesis we use the inverse function method to produce an algorithm for generating Pareto distributed random numbers. The inverse function formula is given by

$$X_{par} = \frac{\delta}{U^{1/\alpha}} \tag{3.26}$$

where *U* is the uniform distribution U [0,1]. (Note: *U* [0, 1] random number generators are built into all simulation packages).

In theory, the Pareto distribution extends to infinity. However, in practice generating Pareto distributed, the random numbers gives a maximum value limited by the minimum value provided by the *U*[0,1]. Due to this, arbitrarily large values cannot be generated. However any true Pareto distribution of sufficiently large length will have values that exceed the range generated by computers. This is a problem for the Pareto (and not e.g. for the exponential) because of the fact that very large numbers will appear from a Pareto distribution (and not from an exponential).

Assuming *S* to be the smallest non-zero value that the uniform random generator U[0,1] may produce, then the generated Pareto distributed value will not exceed q_{par} [Ma03b]:

$$q_{par} = \frac{\delta}{S^{1/\alpha}} \tag{3.27}$$

The mean value of a Pareto distribution is obtained from:

$$E(x) = \int_{\delta}^{q_{par}} x \cdot f(x)dx$$

$$= \frac{\alpha\delta}{\alpha - 1}\left[\frac{1-\left(\frac{\delta}{q_{par}}\right)^{\alpha-1}}{1-\left(\frac{\delta}{q_{par}}\right)^{\alpha}}\right] \quad (3.28)$$

Substituting (3.27) into (3.28), the mean of the generated Pareto distribution random number is given by:

$$E(x) = \frac{\alpha\delta}{\alpha - 1}\left[\frac{1 - S^{\frac{\alpha-1}{\alpha}}}{1 - S}\right] \quad (3.29)$$

This is important and taken into account in this thesis.

3.6 Summary

This chapter presented the significance of self similarity in modelling traffic such as IP. It overviews self similarity in terms of mathematical definitions, and generating traffic through an appropriate ON/OFF model. This chapter also deals with the critical issue of practical limitation in generating random number generator (RNG) for heavy-tailed distributions, like the Pareto.

The next chapter will detail the simulation tools used in this research and describe the meaning of system performance analysis. The next chapter also provides a literature review for the existing accelerated simulation methods.

Chapter 4

SIMULATING PACKET NETWORKS

4.1 Introduction

This chapter presents the techniques used in this research to simulate a queuing system. It explains briefly the types of the simulation model in general, the random number generators used and the simulation output analysis. This chapter also covers some of the existing accelerated simulation methods. The validation and verification methods use in this thesis are explained at the end of this chapter.

4.2 System Performance Analysis

The definition of a 'system' depends very much on the particular objectives under study. It can be as simple as a queue of waiting customers and a serving teller machines or as complex as different types of customer communicating via a WAN. Whatever the system might consist of, the main idea is to study the performance of the system in certain situations of interest.

Many simple systems can be solved using an analytical approach [Pitt00]. As the networks being studied become more complex, analytical solutions become intractable and simulation is the only alternative. The advantage of simulation is that any complicated network, whether it is a small or large scale, can be simulated. Simulation is more flexible than analysis and simulation may support the generation of random numbers with practically any distribution. A few of the well developed and publicly available software packages include OPNET[9], NS2[10] and MATLAB[11]; they can provide a platform for network modelling using either built-in standard models, or user-defined models.

---

[9] OPNET® is a commercial simulation tool with established libraries developed by Mil3.

[10] NS2 is the 2nd version of NS. Developed by UC Berkeley, USC/ISI, Xerox PARC and LBNL. The software is available at www.isi.edu/nsnam/ns/

[11] MATLAB® is a product of Mathworks Inc.

However, simulation modelling has its drawbacks, such as possible excessive memory requirements and of particular interest in this thesis, too much time consumed in computation (in order to reach steady state).

4.2.1 Static vs. Dynamic Simulation Models

A static[12] simulation model represents the system in such a way that time plays no substantive role. In this type of simulation, most of the elements and parameters in the model do not change during the simulation run. The elements are set at the initial stage of the simulation and remain the same during the entire simulation run; thus they may be set to represent 'busy hour' values of e.g. load.

A dynamic[13] simulation model represents a system as it evolves over time. In this type of simulation some or most of the elements in the model changes their properties or attributes during simulation.

4.2.2 Deterministic vs. Stochastic Simulation Model

Any simulation model that does not contain any random or probabilistic element is called a deterministic simulation model. The characteristic of this type of simulation model is that the output is determined when the set of input elements and properties in the model have been specified. For example, a deterministic simulation model can represent a complicated system of differential equations.

Many simulation models however, have at least one element that is random, which gives rise to the stochastic simulation model. In most simulation models randomness is important to mimic the real scenario, for example user connections to the internet arise 'randomly' when a person pressing a key. However, for any stochastic simulation model that has random output, the output (numerical results) can only be treated as an estimate of the true output parameters of the model[14].

[12] Static-is defined as not active or moving; stationary

[13] Dynamic is defined as anything of or concerned with force that produce motion (as opposed to static)

[14] Accuracy in these contents is quantified by confidence intervals

4.2.3 Continuous vs. Discrete Simulation Model

A continuous simulation model is used to model a system as it changes over time by using a representation in which the state variables change continuously with respect to time. Continuous simulation usually involves differential equations that give relationships for the rates of changes of the state variable with time.

A discrete simulation model is about the modelling of a system as it evolves over time by using a representation in which the state variables change instantaneously at discrete points in time. These points in time are the only ones where events occur. An event is defined as an instantaneous occurrence that may change the state of the system. Even though discrete-event simulation could be done by hand calculations, the amount of data that must be stored and manipulated for most real-world systems is so large that it is best done on a computer.

4.2.4 The Simulation Model Used in This Thesis

The simulation tool used is the commercial OPNET software. Simulations described in this thesis are classified as discrete event-simulation, stochastic and static.

The status of the buffer queues will be monitored at specific points in time as the simulation is executed. An event in a communication network model may be any of a wide variety of happenings such as packet loss or arrival. *Instantaneous batch arrivals*[15] are the fundamental events that drive our simulation models. In this case, whenever a batch arrives at the buffer, the state of the buffer is recorded. In this research the traffic sources are defined as 2 states ON/OFF models with Pareto ON and exponential OFF periods.

As is usually the case, we are concerned with the steady state values of loss and delay for packets. Steady state behaviour of the queue implies that none of the system's parameters should be changed during the simulation execution. The role of time is this research therefore has no impact on the simulation it but will specify the length of the simulation. The simulation needs to run a sufficiently long time, and this will be address this in section 4.3.2.

[15] instantaneous batch arrivals - more than 1 packet arrival in a batch is allowed

The simulation model is a discrete time simulation model an in this model the ER batch will be send in a timeslot e.g. ER batch per timeslot.

4.2.5 Simulation Clock

The variable in a simulation model that gives the current value of simulated time is called the simulation clock. It is necessary to keep track of the current value of the simulation internal clock as the simulation executes. Discrete event simulation has a mechanism to advance simulated time from one value to another.. There are two commons approaches for advancing the simulation clock: next-event time advance and fixed-increment time advance. The first approach is used by most of major simulation software and also used in this research.

4.2.5.1 Next-event advance mechanism

With the next-event time advance approach, the simulation clock is initialised to zero. The times of occurrence of the most imminent of these future events is found, and at this instant the system is updated to account for the fact that an event has occurred. Figure 4.1 illustrates the concept:

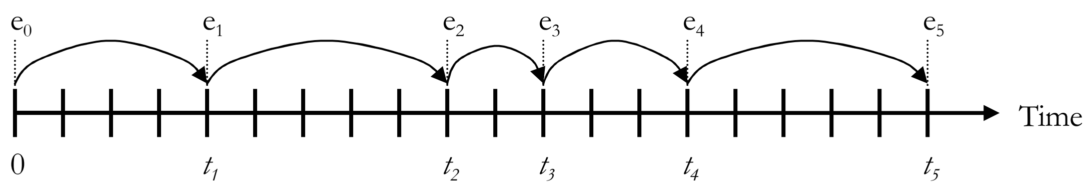


Figure 4.1: Next event time advance

The process of advancing the simulation clock from one event, e_i (i=0,1,2,…), to another is continued until eventually some pre-specified stopping condition is satisfied or until the distribution of interest becomes 'steady state' (see section 4.3.2). Since all state changes occur only at discrete event times, t_i (i=0,1,2,3,…), for a discrete event simulation model, periods of inactivity are skipped by jumping the clock from an event time to another event time. In this research all of the OPNET models are based on the next event time advance concept.

4.2.5.2 Fixed-Increment Time advance mechanism

With fixed-increment time advance, the simulation clock is advanced in increments of exactly Δt. After each update of the clock, a check is made to determine if any events occurred during the previous Δt. If any events were scheduled to have occurred during this interval, they are considered to occur at the end of the interval. The fixed-increment time advance approach is illustrated in Figure 4.2.

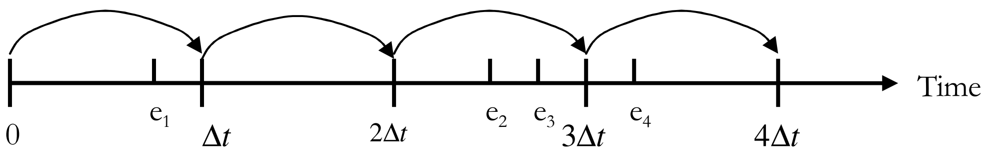


Figure 4.2: Fixed-increment time advance

In Figure 4.2, the curved arrow represent the advancing of the simulation clock and e_i (i=0,1,2,….) is the actual time when the *i*th event occurred.

4.2.6 Random Number Generators

In this research, all the models involve many stochastic elements, and random number generators (RNG) were extensively used to provide e.g. the burst length of a traffic source. All RNGs are based on specific mathematical algorithms which are repeatable and sequential.

A good RNG should be independent interval distributions (IID) in the interval of 0-1 and generated values should not exhibit any correlation with each other or otherwise the validity of the results will be questionable. It also has to be fast, not need much storage space and must be reproducible [Pawl02].

The length of the sequence of numbers, called the 'period', must to be sufficiently long (depending on the simulated traffic involved, i.e. power law or exponential, or the scale of the network i.e. long enough to be larger than the total number of event used in the whole simulation run). This is vital to avoid reaching the end of the period and repeating numbers in a single simulation run, leading to unwanted correlations in the results. RNG that have an extremely long period are important especially in modelling power-law traffic when small probabilities of loss or delay are considered.

RNGs begin to generate a sequence by using a seed. Different seeds produce different sequences of random numbers. The seed can be a default value, where entering the same number of seed each time before simulation is executed, will allow the same sequence of random number to be produced again. In some simulations, where independent streams of random numbers required, different seed value are initially generated. This is to obtain non-overlapping sub-sequences.

4.2.7 Stochastic Modelling

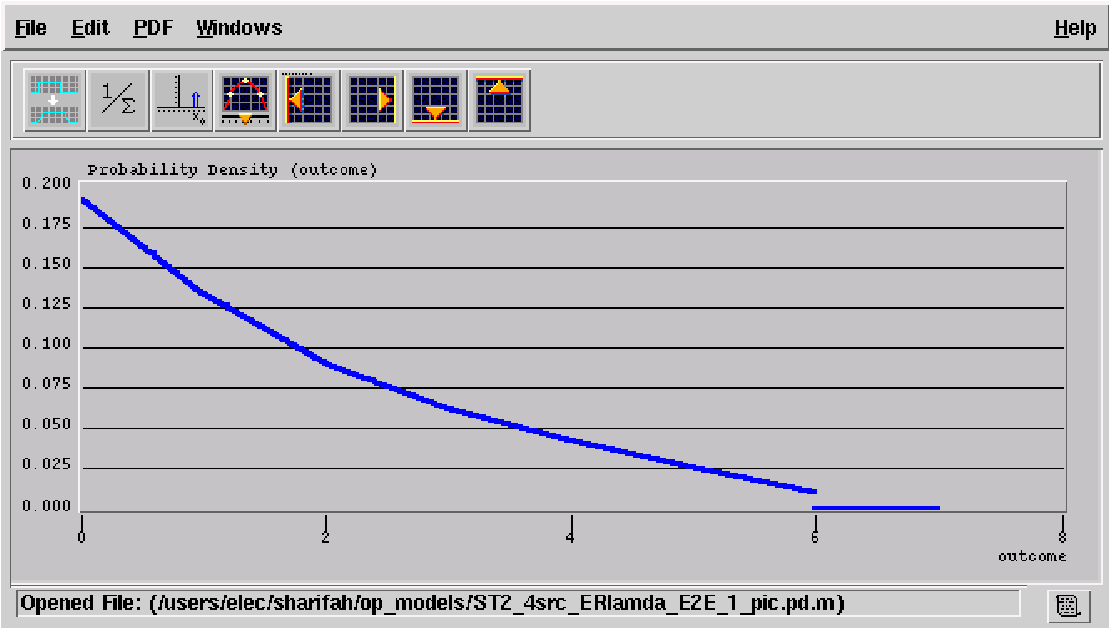
Stochastically modelled elements depend on a random number source on which to based their behaviour. In particular, OPNET provides commands or Kernel Procedure (KP) which supports the use of pre-defined probability distributions.

OPNET supports two types of distributions:

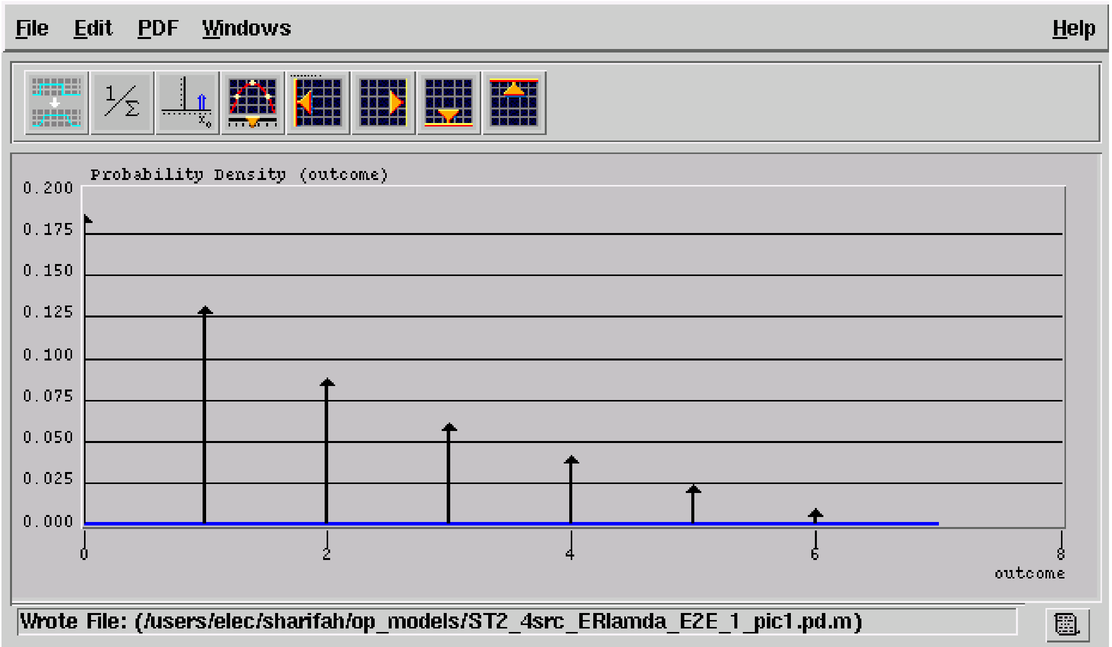
- ❖ Predefined distribution i.e. Pareto, Poisson, Exponential and etc. (supplied by OPNET): actual algorithms are invoked to provide values when needed.
- ❖ User-defined distribution (created by user using PDF editor): the PDF editor creates distribution based on tabular collections of data.

The predefined distributions in OPNET generate values according to a specified probability distribution. Both predefined and user-defined probability distributions are obtained using two steps:

- o ***op_dist_load()*** KP : this KP takes a specification of a probability distribution (for predefined distribution) or specified PDF editor files (for user-defined distribution) and prepares this distribution for efficient use by future processes when this KP is called.
- o ***op_dist_outcome()*** KP: once a distribution has been loaded, a stochastic outcome can be obtained according to the distribution via this KP. KP ***op_dist_outcome()*** used the RNG as its basis for selection of a random generator.



a) An example of a PDF graph.



b) An example of a PMF graph.

Figure 4.3: The PDF editor in OPNET

4.2.7.1 *Creating a New Distribution (PDF Editor)*

The PDF editor in OPNET allows a user to create, edit and view continuous or discrete probability functions. Continuous probability function refers to probability density functions (pdf) and discrete probability function refers to probability mass function (pmf). For pdfs, the PDF editor defines the probability weighting for every possible outcome over a range of possible outcomes (see Figure 4.3 a)). For pmfs, an operation called an *impulse* is added to the pdf. An impulse is a fixed amount of probability mass associated with a single outcome value.

For this research we use only pmfs and Figure 4.3 b) shows as sharp narrow spikes in the pdf that gives one particular outcome when the specified probability occurs. For example, a pmf might model the probability associated with packet arrival times or the packet waiting times.

pmfs are referred by node[16] within simulation models by calling to the ***op_dist_load()*** KP . To represent a pmf, OPNET uses a table (a list of discrete outcome-property pairs). This table can be manipulated via a graphical display of points as shown in Figure 4.3.

4.3 Simulation Output Analysis

The options available for analyzing simulation output depend on the type of the simulation of interest. As illustrated in Figure 4.4, simulations may be either terminating or steady-state.

4.3.1 Terminating simulations

A terminating simulation requires one or more conditions that specify the end point of a simulation run. Different simulation runs use independent random numbers and the same initialization rules. This implies that comparable random variables from different simulation runs are IID. The condition is usually specified at the beginning of the simulation run, and the times these occur in a particular simulation run may be a random variable. The initial conditions for terminating a simulation usually affect the desired

[16] Nodes in OPNET simulation models represent physical sites in the real-world network which may either represent the source, destination or router.

measures of performance, so it is essential to understand these conditions of the actual system.

4.3.2 Steady State Simulations

In a steady state simulation, no simple condition is required to specify the length of a simulation run. This type of output analysis usually applies when designing a new system or changing an existing system where the behaviour of the system in a long run (when it is operating normally) are of interest. With respect to output analysis, the steady-state simulations are divided into three categories; steady state parameters, steady state cycle and other parameters (see Figure 4.4).

- Steady-state parameters – the output parameter of interest must actually have a steady state distribution. For example, if random variable Y has a steady state distribution, then it might be interesting to estimate the steady-state mean $v = E(Y)$ or a probability $P(Y \leq y)$ for some real number y.

  The stochastic processes in most of the real systems do not have steady state distributions because the characteristic of the system changes over time (i.e. the load during the off peak and the peak hours varies, the number of customer that are using the system might change). However, a simulation model may have a steady state distribution provided that the characteristics of the particular model are assumed not to change over time (refer to section 4.2.1).

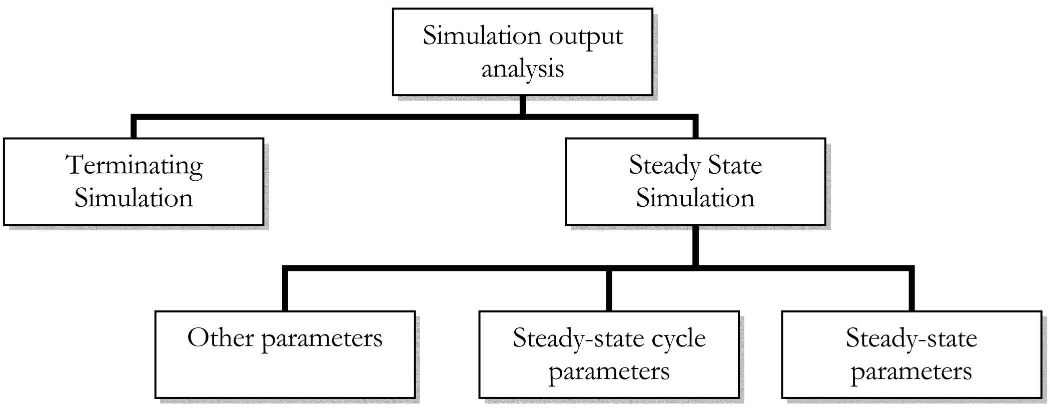


Figure 4.4: Types of simulations regards to output analysis [Law00]

- Steady-state cycle parameters - for a steady-state simulation that does not have a steady state distribution, the time axis can be divided into equal lengths, contiguous time intervals called cycle. For example Y_i^C is a random variable defined on the *i*th cycle and assume that $Y_1^C, Y_2^C, \ldots$ are comparable. If process $Y_1^C, Y_2^C, \ldots$ has a steady-state distribution F^C, and that $Y^C \sim F^C$, a measure of performance is said to be a steady-state cycle parameter if it is a characteristic of Y^C such as the mean $v^C = E(Y^C)$. Thus a steady-state cycle parameter is just a steady-state parameter of the appropriate cycle process Y_1^C, Y_2^C, ...

- Other parameters- for steady-state simulation that have no appropriate steady state cycle distribution there may still be a fixed amount of data describing how input parameters change over time. This can provide a terminating event for the simulation. The analysis technique for terminating simulations is then appropriate.

In this research, the long-term behaviour of the buffer is required as this is almost always the focus of QoS guarantees such as packet delays and losses. So steady-state simulation is used exclusively.

4.4 Literature Review of Accelerated Simulation Techniques

Accelerated simulation techniques have been proposed to reduce simulation time. These techniques can be categorized into three different types (see Figure 4.5): computational power, simulation technology and simulation model.

4.4.1 Computational Power

Parallel and Distributed Simulations (PADS) falls into this category because simulations are speeded up by using more powerful and faster machines. Originally the term parallel referred to simulations executed on tightly coupled parallel computers, and distributed simulation referred to geographically distributed simulation [Fuji03]. However, with the integration of computing networks, such as workstation clusters and grid computing, the difference has become less clear.

PADS technology had been developed largely in high performance computing and defence military work. In high performance computing PADS synchronization algorithms were designed in order for the distributed simulators to produce exactly the same results as the sequential simulators of the simulation program but faster.

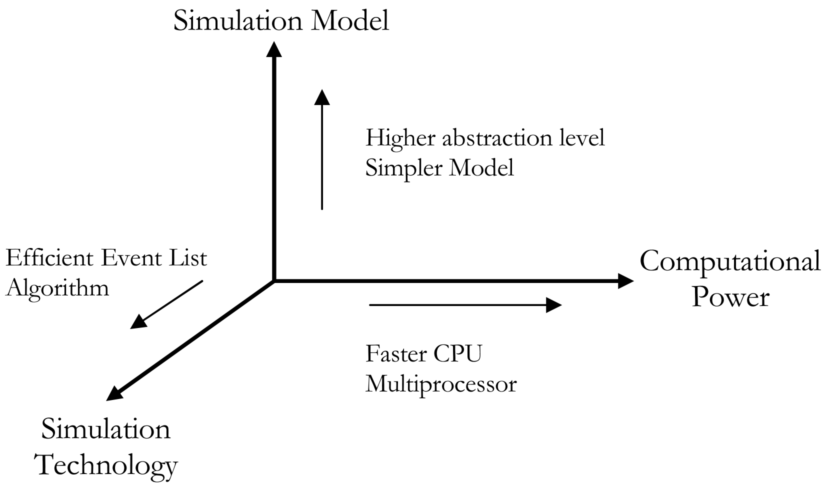


Figure 4.5: Different ways to speed up simulation [Liu01]

In defence and military work, virtual environments are created by PADS. These are to allow execution in real time, i.e. the simulator must be able to simulate a second of activity in a second of wall clock time. This is to ensure a realistic virtual environment that acts like the actual scenario. There are two types of synchronization techniques 1) conservative – where the algorithm guarantees that dependent events never occur and 2) optimistic – where the algorithm allows synchronization errors to occur but provide a mechanism to recover. These algorithms are concerned with the time management in PADS, to ensure that the execution of the simulation is properly synchronized.

4.4.2 Simulation Technology

In these methods, new enhanced algorithms are used to accelerate simulations which involve changes in the statistical behaviour to provoke rare events of interests to occur more often. One example of this approach is the RESTART (Repetitive Simulation Trials After Reaching Thresholds) [Vill94] mechanism.

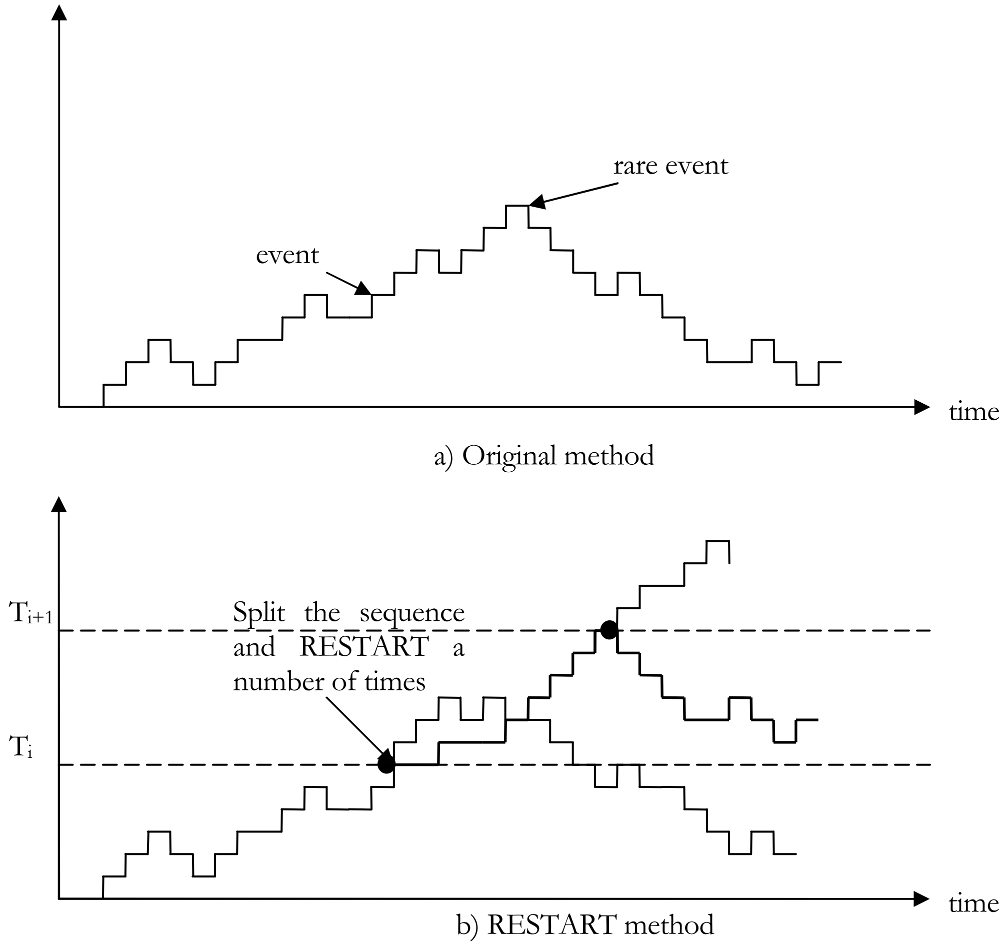


Figure 4.6: The rare events occurrence in the original simulation method and the splitting of sequence at the threshold in RESTART method

The original approach was proposed by [Kahn51] and was called the Splitting Technique. Other studies using this approach include [Glass96, Glass98, Schr96]. In the conventional method, the occurrences of rare events are as in Figure 4.6a. However in RESTART method the occurrences of these rare events can be made less rare by defining a subspace where rare events occur more often, and for each entrance to this subspace current sequences are split and restarted a number of times (see Figure 4.6b).

In [Garv98] the RESTART method is divided into two: i) Fixed Splitting (FS) method and ii) Fixed Effort (FE) method. FS method is referred to as the original RESTART, where at every restart stage each run is split into fixed number of copies. An implementation of the FS method is sensitive to the choice of the RESTART parameters. In the FE method the total simulation effort is fixed for each run of the simulation. This way the number of runs that hit the next level will remain approximately the same despite the choice of the RESTART parameters [Ma04].

Another example in this category is Importance Sampling (IS) in which the basic concept comes from Monte Carlo simulation [Hamm64]. In simulations, certain values in the input random variable have more impact on the parameter being estimated than others. The idea of IS is to reduce the variance of a given simulation estimator by sampling more frequent of these 'important' values [Held95, Town98]. A system that used IS chooses a new set of probability value that bias the important values by a likelihood ratio.

4.4.3 Simulation Model

The third type of the accelerated simulation technique simplifies the simulation model and improves its efficiency. An example of this is the 'packet-train' simulation technique, which was first proposed in [Anic82, Jain86] to model data network traffic. In this technique, the network traffic is modelled in terms of a continuous packet flow rather than discrete packet instances and the overall number of simulation events is reduced by marking an event, at the end of the packet-train instead of each individual packet (see Figure 4.7).

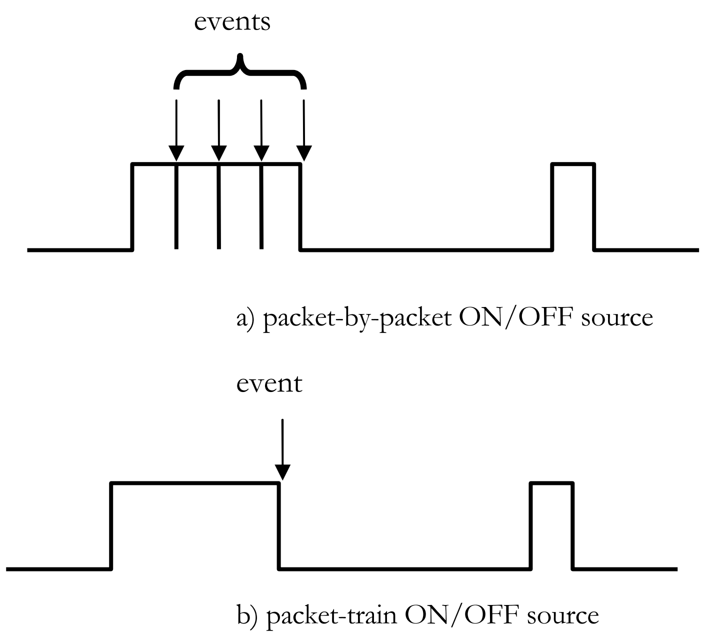


Figure 4.7: The simulation events in packet-by-packet and packet-train technique.

In some studies this technique is called cell-rate or packet-rate modelling where an event is defined as the change of rate in the connection [Pitt95]. Using this technique, rare event probabilities, e.g. cell loss probabilities, can be obtained in less time.

Another example in this category is Hybrid methods. A Hybrid method is a combination of other techniques, which could be an analytical model with a simulation model, or two different kinds of simulation models. The advantage of these is that they highlight the best features of each method and faster execution times can be obtained. Hybrid methods in current research have shown promising results in accelerating simulations [Ma02c, Scho01]. [Ma02c] combined analytical and simulation methods to predict the buffer overflow probabilities based on simulation of the input traffic, and then applied this to obtain buffer overflow probability faster than the pure simulation method can. The method in [Scho01] is called HQ method (Hybrid Queuing) where it distinguishes between the foreground and the background traffics. The foreground traffic is the traffic of interest in the investigation and the background traffic is represented using an analytical method. This method can be used for both cell and burst scale queuing. The key difference between the work in [Ma02c] and [Scho01] is that [Ma02c] involved power law traffic and [Scho01] involved Markovian traffic.

Apart from hybrid methods, [Ma02c] also used the Traffic Aggregation (TA) method to further reduce the number of simulation events. The TA concept has been applied for Markovian traffic and proved to be highly accurate [Scho00]. TA was further developed for self similar traffic [Ma03a, Ma02a, Ma02b, Ma02c, Ma01]. TA has been recently proposed as a generic technique for packet buffers that significantly reduces the number of simulation events required to achieve steady state in a simulation experiment. However it is currently limited to First-In-First-out (FIFO) buffer scheduling.

The acceleration methods in the first and second category mentioned above have achieved simulation speed up. However, they involved complex mathematics which can be difficult to implement. This research proposes an accelerated simulation method involving power-law traffic with priority buffer, based on an extended version of the TA method.

4.5 Burst and Packet Scale Queuing Behaviour

Packet scale and burst scale queuing behaviour will both result when multiplexing a number of ON/OFF sources together. For packet scale queuing behaviour, in 1 timeslot, one or more packets may arrive and this will exceed the queue's service rate for a short period of time. Figure 4.8 illustrates packet scale queuing behaviour and in this example [Pitt00] it shows two packet streams with utilization of 50% (*stream A*) and 25% (*stream B*). The third row shows the combination of the two packet streams resulting in 75% utilization. In the first 12 timeslots, packets from *stream A* and *stream B* do not arrive in the same timeslots; hence the packets are served immediately. However, the second set of 12 timeslots shows the arrival of more than one packet in a timeslot, and due to this some packets have to wait before being served. The number of packets waiting in the queue is shown in the graph of Figure 4.8 where vertical axis represents the number of packets in the queue and the horizontal axes is in timeslots.

If a third packets stream (*stream C*) is added to the scenario of Figure 4.8 with utilization 33%, the total rate now exceeds the service rate. Over a longer period the number of packets waiting to be served will build up. Figure 4.9 illustrates this scenario and this longer-term queuing behaviour is called burst scale queuing (this is shown as a solid line in the graph of Figure 4.9). In this example the input rate exceeds the service capacity by 8% (i.e. 0.08 packets per timeslot), and the queue size increases over 24 timeslots to 0.08 x 24 ≈ 2 packets. During this period (24 timeslot) there were 26 packet arrivals with an excess of 2 packets. These 2 packets are called 'Excess-Rate' (ER) packets because they represent a number of arrivals in excess of the server capacity, aggregated over significantly longer than a single timeslot. The work in this thesis makes considerable use of the ER concept, as do TA before.

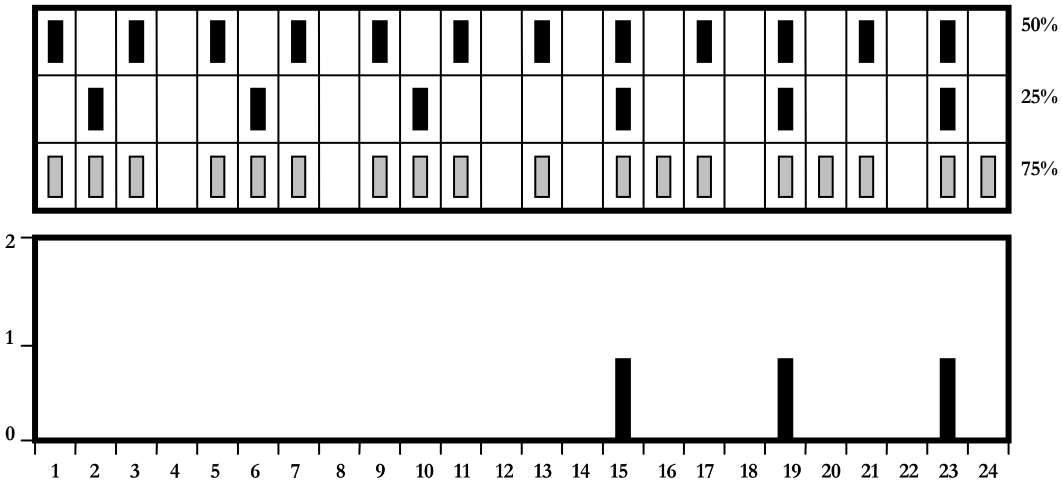


Figure 4.8: Packet scale queuing behaviour [Pitt00].

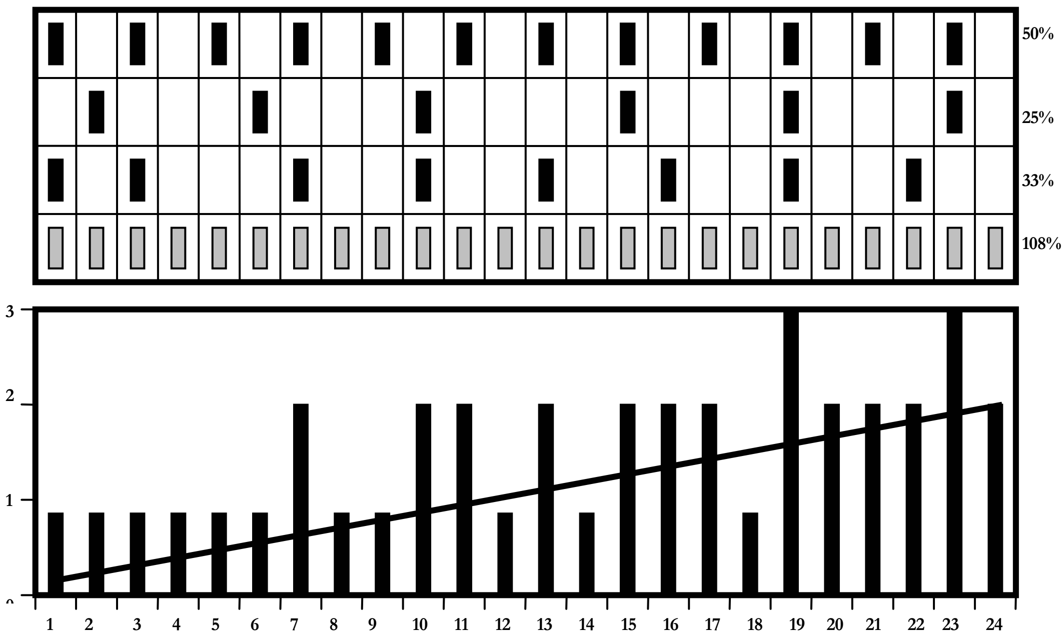


Figure 4.9: Burst-scale queuing behaviour [Pitt00].

4.5.1 Excess-Rate Queuing for A Single ON/OFF Source

A single ON/OFF source can model the scenario of burst scale queuing. When the source is ON, packets are produced at rate R, overloading the service capacity C and causing burst-scale queuing. This scenario is illustrated in Figure 4.10 [Pitt00].

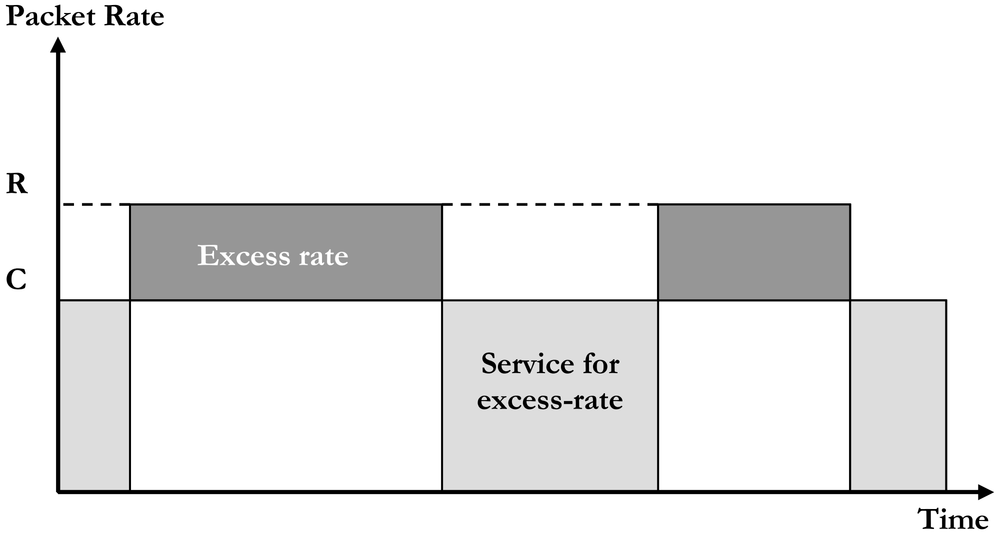


Figure 4.10: Burst scale queuing with a single ON/OFF source.

When OFF, the source sends no packets and the buffer can recover from the accumulated packets by serving these buffered ER packets. The ER arrival rate approach has been used successfully in the analysis of queuing systems which involves multiplexing in communication networks [Scho00a, Scho00b, Scho94, Scho98]. According to [Pitt00], ER packets are those which must be buffered as they represent an excess of '*instantaneous*' arrival rate over service rate. ER packets cause the queue to increase in size and the queue decrease only when there are no arriving packets in any timeslot. Hence, in our model, ER packets are responsible for building up queues in buffers.

4.6 Traffic Aggregation

Studies of queuing theory in [Scho00, Scho94] have discovered that N x 2 state process can be replaced by a single 2 state process accurately. The resulting 2 state process is either in the ON state, or in the OFF state. Using this traffic aggregation any N multiplexed sources can be replaced by a single equivalent ON/OFF source with equivalent queuing behaviour in the buffer. This concept has been successfully applied to Markovian ON/OFF sources [Scho01b, Scho03] and traffic sources with power law sojourn times [Ma02a, Ma02b, Ma02c, Ma03a, Ma03b]. In this thesis aggregation involving power law traffic will be what we refer to as Traffic Aggregation (TA.).

Consider a conventional[17] traffic model, comprising N ON-OFF sources with Pareto distributed sojourn times (see Figure 4.11a). These individual traffic sources transmit packets at a rate of *R* packet/timeslot during the ON periods and *R*=0 during the OFF periods. The mean ON and OFF periods are denoted by *Don* and *Doff* respectively. The mean arrival rate for an individual source is given by:

$$\lambda_{Conv} = R \cdot \alpha \tag{4.1}$$

and

$$\alpha = \frac{Don}{Don + Doff} \tag{4.2}$$

where α is the probability that the individual source is active. In this work queuing system use a deterministic service rate which transmits packet at *C* packets/timeslot[18]. The randomly[19] switching of states (ON to OFF and vice versa) from this individual source results in a traffic pattern that is similar to the illustration of Figure 4.11c. The First-In-First-Out (FIFO) queuing mechanism is used and the utilization of the system, ρ, is given by:

$$\rho = \alpha \cdot N \cdot R \cdot \frac{1}{C} \tag{4.3}$$

---

[17] In this thesis conventional refers to a simulator without acceleration.

[18] In this thesis, C is usually equal to 1.

[19] Random switch here means the decision to be in ON or in OFF state is determined by a RNG. In this case the sojourn ON time is not random (i.e. exponential)but Pareto distributed.

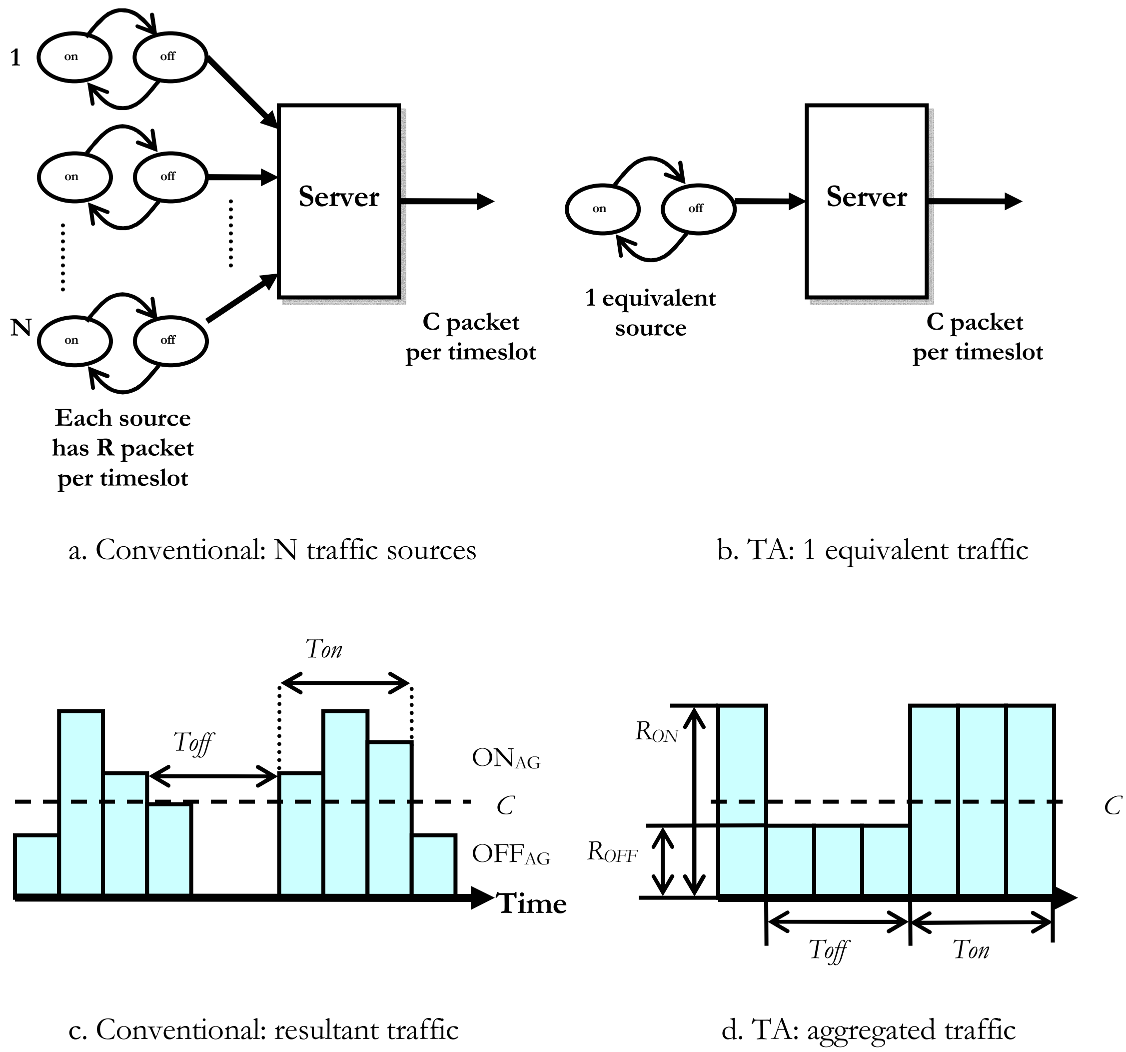


Figure 4.11: Conventional and TA concept.

TA introduced an acceleration technique that reduces the multiplexed scenario to only one single equivalent source (see Figure 4.11b.). The aggregated 2 state process of the equivalent 2 state model is either in the ON state or the OFF state; these are defined by reference to the resultant traffic pattern from the N original sources (see Figure 4.11c). The ON and OFF states of the aggregated processes are denoted ON_{AG} and OFF_{AG} respectively, and these (ON and OFF states) are derived with reference to the service rate, *C*. An ON_{AG} period occurs when the packet arrival rate is greater than the service rate, i.e. $NR>C$, and the mean duration of this aggregated state is denoted by *Ton*. An OFF_{AG} period occurs when there is a continuous number of timeslots in which the packet arrival rate is less than or equal to the service rate, and the mean of this duration is denoted as *Toff*. Figure 4.11d shows that the rate of packet arrivals during the OFF period is always less than the service rate, and therefore the accumulated packets can be served. The packet arrival rate that is equal to the service rate, *NR*=*C*, is considered as OFF because no effect on the buffer size, and no further accumulation of packets.

Ron denotes the arrival rate during the ON state, *Roff* is denotes the arrival rate during the OFF state and both had a value >0 calculated for them. During the early development of TA, both *Ron* and *Roff* were taken into consideration [Ma02a, Ma02b, Ma02c, Ma03a]. When TA was developed, the relationship between parameters (*Don, Doff, N, C* and *R*) was observed. Changes in *Doff* values had very little effect in the overall queuing behaviour of the buffer. This affected the parameterization process of TA in the sense that the effect brought by *Doff* can be neglected. The later development of TA [Ma03b] showed that *Roff* is virtually redundant and therefore setting *Roff* to zero abridges complexity and further reduces the total event count. Hence, the single equivalent source representing *N* Pareto ON/OFF traffic sources has only three parameters, that is; *Ton, Toff* and *Ron* and this is illustrated in Figure 4.12. This is very important for the work reported in this thesis as setting *Roff* = 0 directly implies the possibilities of E_TA.

The discovery of *Roff* equal to zero had lead to E_TA technique where there are no packet arrivals between the ER batches. The arrival distribution of the ER batches is exponential and the service rate distribution is Pareto which is the M/G/1 queuing system.

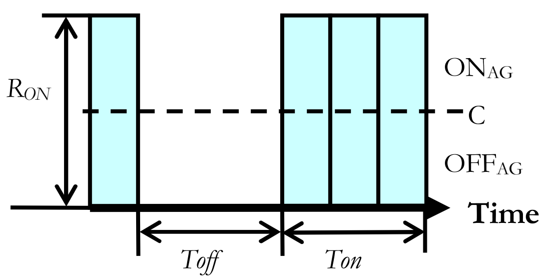


Figure 4.12: The later version of TA

4.7 Validation and Verification

Verification is about whether the simulation model has been correctly translated into the computer program. Validation is important in determining whether a simulation model is an accurate representation of the system under study and usually uses comparison with known results, where available. In this research validation is done using the power-law best fit to verify the trend of the power law distribution. For validation, in this thesis we compare queuing behaviour results of E_TA with conventional model and other existing accelerated simulation method e.g. TA. One way to justify the accuracy is by using theoretical data i.e. confidence intervals. However, for heavy-tailed distributions, the accuracy in comparing the actual data and theoretical data using confidence intervals had been said to be questionable. This is because in confidence interval, the standard error of the mean which is defined by the standard deviation divided by the square root of the number of samples comes from the Central Limit Theorem (CLT). CLT says that as the number of sample size increases, the sampling distribution of the sample means approaches that of a normal distribution. This usually needs enough samples and it is hard to apply to LRD data [Roug05, Will97]. Hence in this thesis we do not use confidence interval to validate the results.

4.7.1 Power-law Best-fit and Exponentially Wider Bins

Power-law best fit is used to graphically display the trend of power law distributions. The best fit is obtained by calculating the least square fit through points by using (3.13). Power law traffic has a characteristic feature of being very bursty. To obtain meaningful results, long simulation runs are essential. However, the problem with output analysis of power-law traffic lies in the tail of the distributions (see Figure 4.13) where the 'messy' tail produced as the raw output of the simulation. This is because the tail probabilities are much less rare with power-law than with e.g. exponential distribution. This gives rise to the fundamental problem in simulating power-law distribution where you cannot get enough data points. For example, Figure 4.13 shows the result of a certain simulation run involving power-law traffic, the probability of the buffer having queue size of 10^3 was not as large as the probability of the buffer having queue size of 10^1. Hence, by simply fitting the best fit line (the straight line in black) to the data in Figure 4.13 gives a slope that is too shallow and looks wrong.

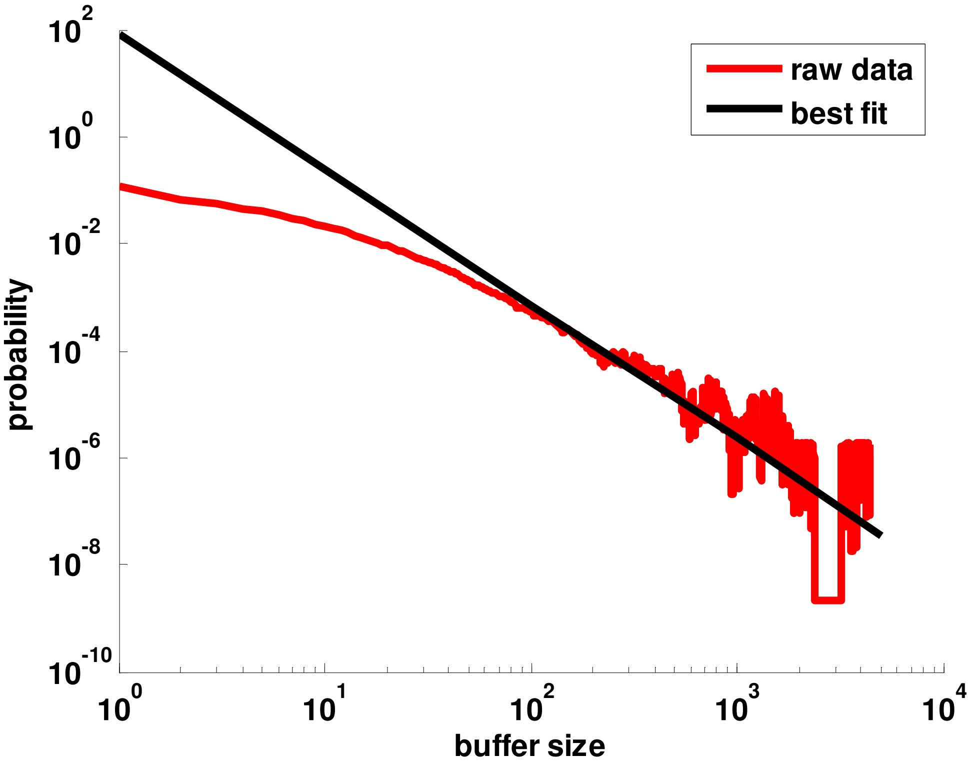


Figure 4.13: Raw data from simulated power law traffic with best fit line.

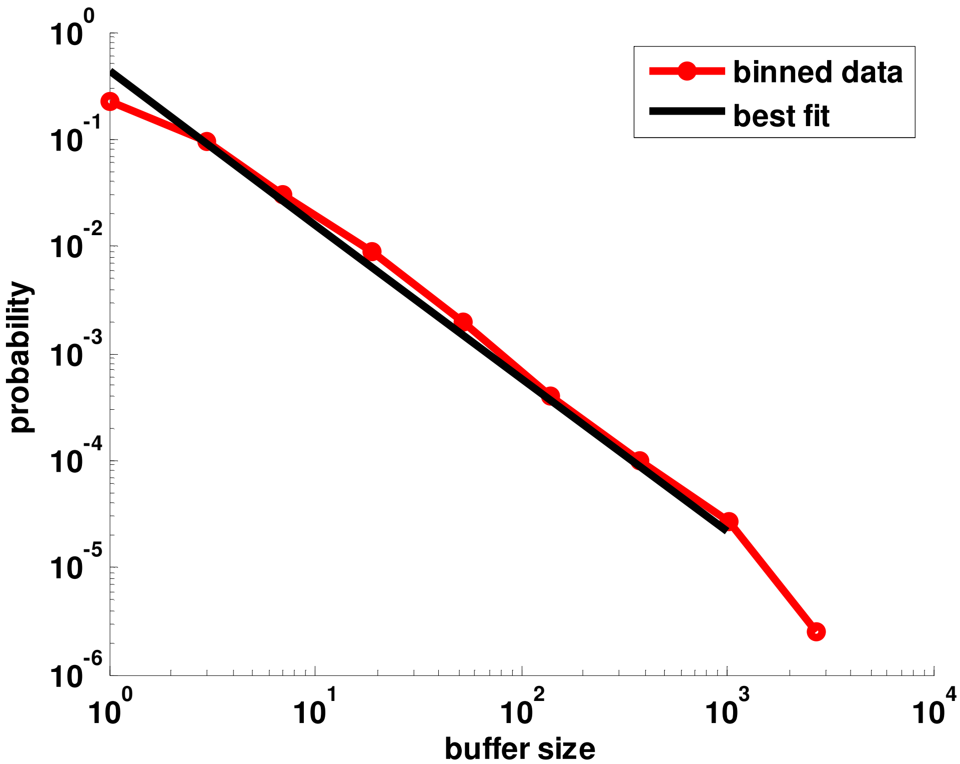


Figure 4.14: The binned data and the best fit line.

To achieve a proper fit, the data needs to be binned into exponentially wider bins [Adam] where the graph will appear evenly spaced on a log-log scale as shown in Figure 4.14. In exponentially wider bin the width of the bin are not equal but exponentially distributed depending on the amount of data in the range of the bin. In Figure 4.14 we can see that the 'noise' in the tail has been smoothed out and the best fit better reflects the queuing behaviour of the power law traffic. Both the power-law best fit and exponentially wider bins techniques will be used throughout this thesis to show the results of the simulations.

4.7.2 Active Measurement

Active measurement (e.g. using Cisco IOS Service Assurance Agent (SAA)) is a technique that injects artificial probe traffic into a network to measure round trip end-to-end or one-way end-to-end performance [Cisc13]. Typical use of the active measurement method is to obtain end-to-end statistics such as delay, loss and route availability [Paxs99, Pasz01]. Active measurement has increasingly become important in network measurement due its great flexibility. The benefits of using active probes are that they can run virtually anywhere in the network and give an end-to-end perspective of network conditions. There are various sampling techniques that can be used for measurement such as Simple Random, Systematic and Stratified Random [Claf93], see Appendix 2 for details.

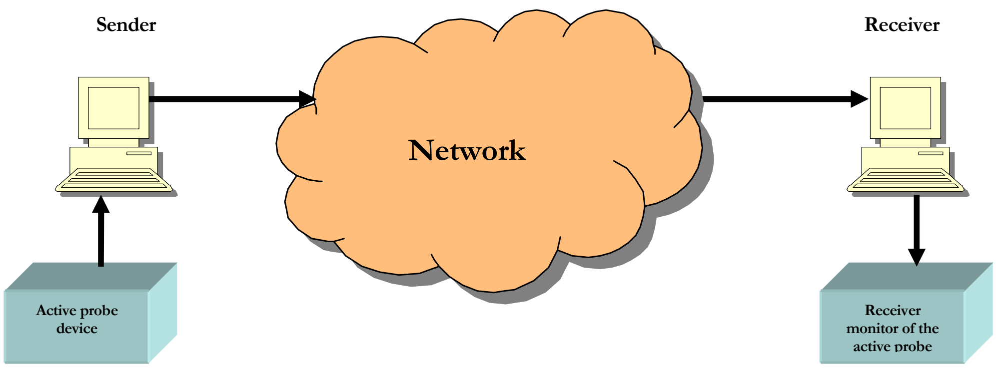


Figure 4.15: The basic component in active probing measurement

In this thesis, active measurement is used in order to ensure the accuracy of the proposed accelerated simulation method compared to the conventional simulation model. This is done by comparing the packet delay distribution of the parent population collected using both simulation models. The probes are generated from the sending end and monitored at the receiving end. These probes are time-stamped so that their delay can be measured at the receiver monitor, as illustrated in Figure 4.15. Comparing the distributions in E_TA with the conventional by use of active probing should provide a completely fair and widely agreed method of comparison.

However, the intrusive character of active probes can ruin the results if the probe rate is too high. Hence, in this thesis the mean arrival rate of the probes is set to be 1% of its parent traffic. The possible effect of the probes on the queuing behaviour of the buffer is detailed in Appendix 2.

4.8 Summary

This chapter covers the tools used in this research, and briefly explains the classification of the simulation models. It also looked into the two mechanisms available for discrete-event simulation. Random number generator is also discussed where this is important in simulating stochastic models.

A literature review of the existing accelerated simulation methods is also covered in this chapter. At the end of this chapter, the validation techniques that are used in this research are discussed. The next chapter will present the numerical results generated by using E_TA to accelerate simulation of packet networks.

Chapter 5

ENHANCED TRAFFIC AGGREGATION WITH FIFO BUFFER (E_TA-FIFO)

5.1 Introduction

This chapter present in detail the further development of the TA technique, called Enhanced TA (E_TA). The E_TA technique involves generating traffic that only consists of the Excess Rate (ER) packets from the single equivalent ON/OFF source. These ER packets arrive in a batch instead of packet-by-packet. The batch compresses the effect of the aggregated ON period. The periods between batches are empty of arrivals, and this works because it is already known that aggregated *Roff* can be set to zero [Ma03]. This chapter concentrates on the development of E_TA with a FIFO scheduler.

At the end of this chapter, the results collected from the E_TA are validated by comparing with the results collected from the conventional, and the prior accelerated simulation method (TA). Results are collected for the packet delay probability (from the sending end to the receiving end) and the queuing behaviour in the buffer of the E_TA model. Both state probabilities collected using conventional simulation and TA are found to be are accurately reproduced using E_TA.

5.2 The Concept of E_TA-FIFO

In E_TA, the ER arrivals represent the excess of arrivals during the ON period of the aggregated source (i.e. those arrivals that force a queue to build up in the buffer). In a packet-by packet model (i.e. conventional), a simulation event occurs when there is an arrival of a packet, but in E_TA a simulation event occurs only when there is a change of state, i.e. from the ON state to the OFF state or vice versa. Moving from state OFF to ON in E_TA will add an 'ER batch' to the queue. This forms an acceleration technique that is more advanced, and requires fewer simulated events than TA. For example, in TA a particular ON period might send M packets and the total number of events to complete one cycle of ON/OFF periods is M + 1 (including the end of the OFF period) events. Using E_TA, the total number of events in 1 ON/OFF cycle is 2 (the end of the ON period and the end of the OFF period), this is regardless of the number of packets in the ON period. The definition of events in E_TA and TA is illustrated in Figure 5.1.

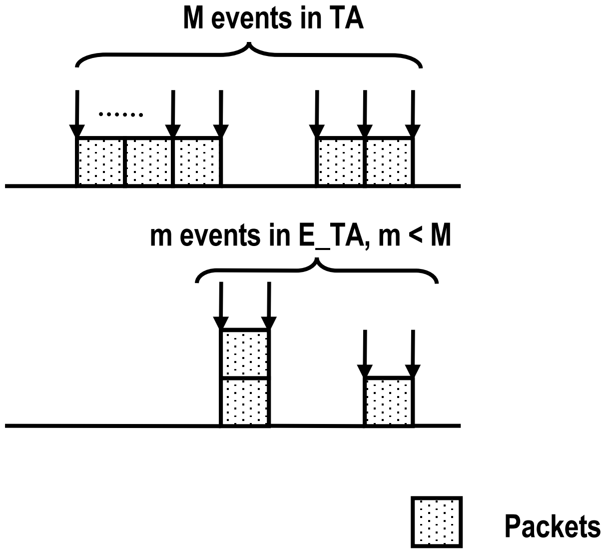


Figure 5.1: Simulation events in packet-by-packet model i.e. TA and E_TA model.

The mean size of an ER batch is given by:

$$\overline{ER} = Ton(Ron - C) \tag{5.1}$$

where Ton - The mean ON period of the aggregated traffic

Ron - The mean arrival rate in the ON period of the aggregated traffic

C – Service rate (packet/ timeslot)

The mean OFF period in E_TA, $Toff_2$, which is also the inter-arrival time of the ER batches, is found from the following equation:

$$Toff_2 = \frac{\overline{ER}}{\rho \cdot C} \tag{5.2}$$

where ρ is the utilization of the system. This is because ON periods in E_TA last for 1 timeslot.

As already described in section 4.2.2 time in an E_TA simulation model is divided into slots where each is of duration equal to the fixed service time of a packet. In a conventional packet-by-packet simulator, if the service rate is C, the service time of each individual packet will be 1/C. However in E_TA, Pareto distributed batches of ER arrivals will arrive in any slot with probability q and the FIFO buffer holds ER batches instead of individual packets. Hence the mean service time of each ER batch in the FIFO buffer, $\overline{ST_B}$, is given by:

$$\overline{ST_B} = \frac{\overline{ER}}{C} \tag{5.3}$$

5.3 Validations

All simulation experiments in this research used Optimum Network Performance (OPNET) version 8.1. OPNET runs on an Intel Pentium 4 at 1.70GHz and about 512Mb of RAM. In the following the results of E_TA-FIFO model will be validated using packet delay distribution and queuing behaviour of the FIFO buffer.

The packet delay distribution is collected using the active probe measurement (Section 4.7.2 and Appendix 2). This procedure is done without significantly interfering with the parent traffic if the proportion of the probes is low enough. The packet delay experienced in the conventional model is compared with the packet delay distribution collected from the E_TA-FIFO model to validate the accuracy of the E_TA-FIFO model.

Another way to validate the E_TA-FIFO model is by using the state probabilities distribution. The state probability of interest in this research is the state probability as seen by an arriving packet.

The probability distribution of a discrete random variable is represented by a probability mass function (pmf). The pmfs are obtained by measurement using histogram.

5.3.1 Numerical Results -Packet Delay Probability (for a single node)

The concept of the ETA-FIFO method is first validated for a single node network. The simulated scenarios are illustrated in Figure 5.2, where the probes are generated at the sending end and the delay distributions are observed at the receiving. The steady state probability of packet delay is defined as:

Pr (*t*) = Pr (the delay of a packet probe is *t* timeslot)

Individual parameters of Don and N are varied, and the details of the simulations are listed in Table 5.1 and 5.2. The data set of TA in Table 5.1 and 5.2 are equivalent to the conventional using TA algorithm in [Ma03b]. The data set of E_TA in Table 5.1 and 5.2 are equivalent to the conventional and TA using equation (5.1) and (5.2). These data sets (Conventional, TA and E_TA) do not look equivalent; they are equivalent but only in sense that they produce equivalent network performance from their respective algorithms. The traffic values in these tables are chosen mainly because these values can be generated accurately (due to restricted range of random numbers generated by the Pareto distribution, see Section 3.5), and not specifically to represent any particular 'real' traffic or application type.

The comparisons are presented in two versions; 1) direct[20] comparison of the conventional N source model with E_TA-FIFO; 2) comparison of the power-law best fit line of the conventional with E_TA-FIFO. The best fit lines were obtained when conventional simulations were run much longer periods of time. Power law best fit line of the conventional model is an important reference for the validation of E_TA-FIFO (see section 4.7.1)

[20] Direct comparison of the different models (i.e. conventional and E_TA) using the raw data which are binned using exponentially wider bin.

Each result is an average of 10 simulation runs. For better visualization, these results were binned into exponentially wider bins [Ma03a, Ma03b]. Results are collected at the end of 10^6 timeslots.

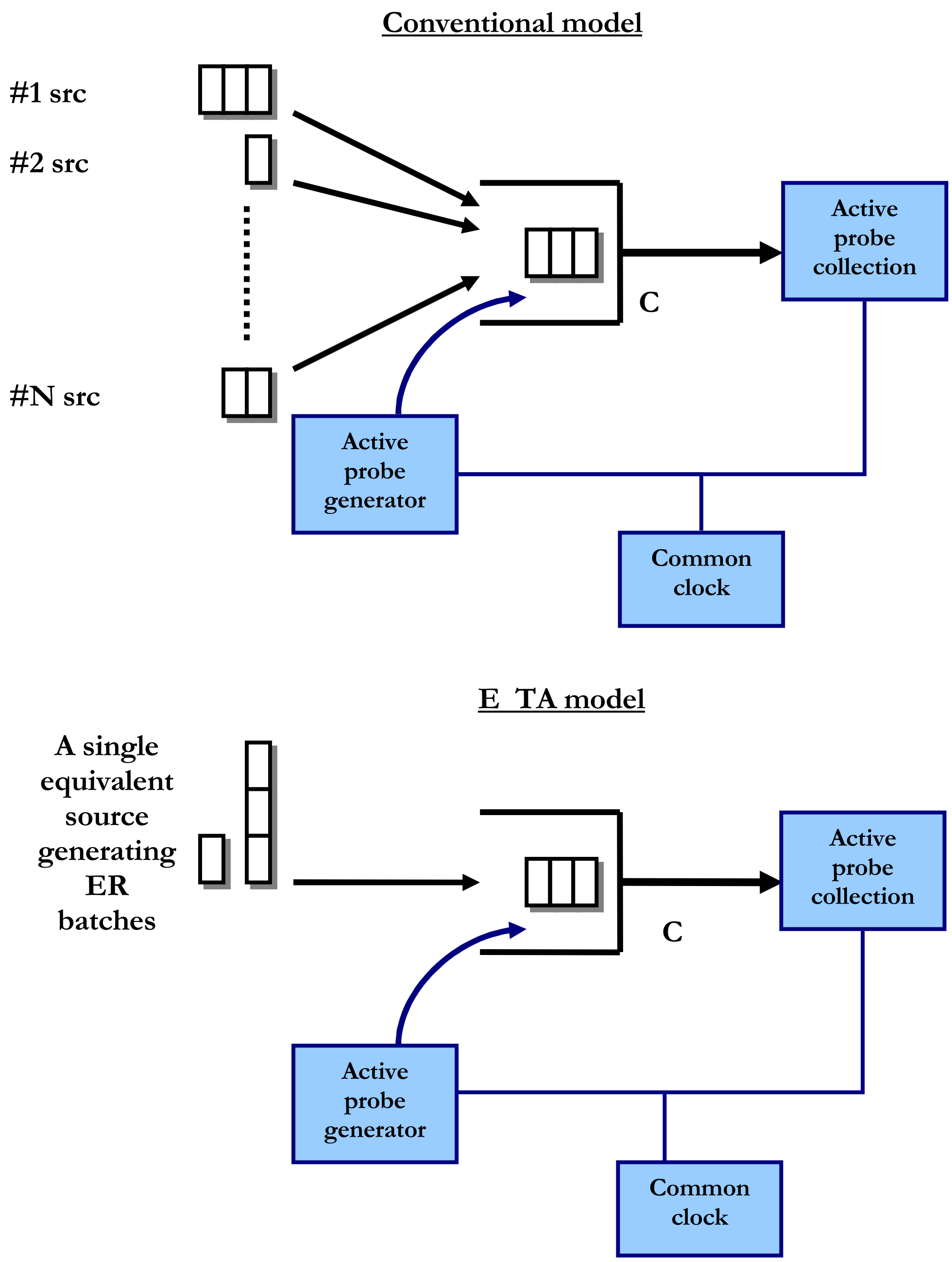


Figure 5.2: The sending end of both conventional and E_TA model.

Figure 5.3 shows the packet delay probabilities for E_TA-FIFO, conventional and TA. As the mean ON period is increased the packet delay probabilities tends to flatten Further validation is given in Figure 5.4, which is for of E_TA_FIFO against best fit Power law from conventional simulation.

E_TA-FIFO is tested as the number of the aggregated traffic sources is varied. Figure 5.5 shows that the packet delay probabilities of E_TA -FIFO is accurate compare to conventional and TA Moreover better coverage of the probabilities was achieved in the same length of time. Figure 5.6 show the comparison of E_TA-FIFO with conventional best fit data.

Table 5.1: The parameters values for E_TA, TA and conventional model used in Figure 5.3, 5.4 and 5.7

Conventional					TA				E-TA		
Doff	Don	N	R	C	Toff	Ton	Ron	C_{TA}	$\overline{ER}$	$Toff_2$	C_{ETA}
10	3.1	10	1	5	2.3258	1.8059	6.2258	5	2.2137	0.9354	5
10	3.9	10	1	5	2.2292	2.3086	6.2871	5	2.9714	1.2556	5
10	5.0	10	1	5	2.2022	2.9998	6.3569	5	4.0704	1.7200	5

Table 5.2: The individual and equivalent traffic sources value used in Figure 5.5, 5.6 and 5.8

Conventional					TA				E_TA		
Doff	Don	N	R	C	Toff	Ton	Ron	C_{TA}	$\overline{ER}$	$Toff_2$	C_{ETA}
10	3.9	16	1	8	2.0658	1.9218	9.3149	8	2.5369	0.5628	8
10	3.9	18	1	8	2.1551	2.4844	9.94314	8	3.5562	0.7042	8
10	3.9	20	1	8	2.3774	3.3764	9.5627	8	5.2763	0.9403	8

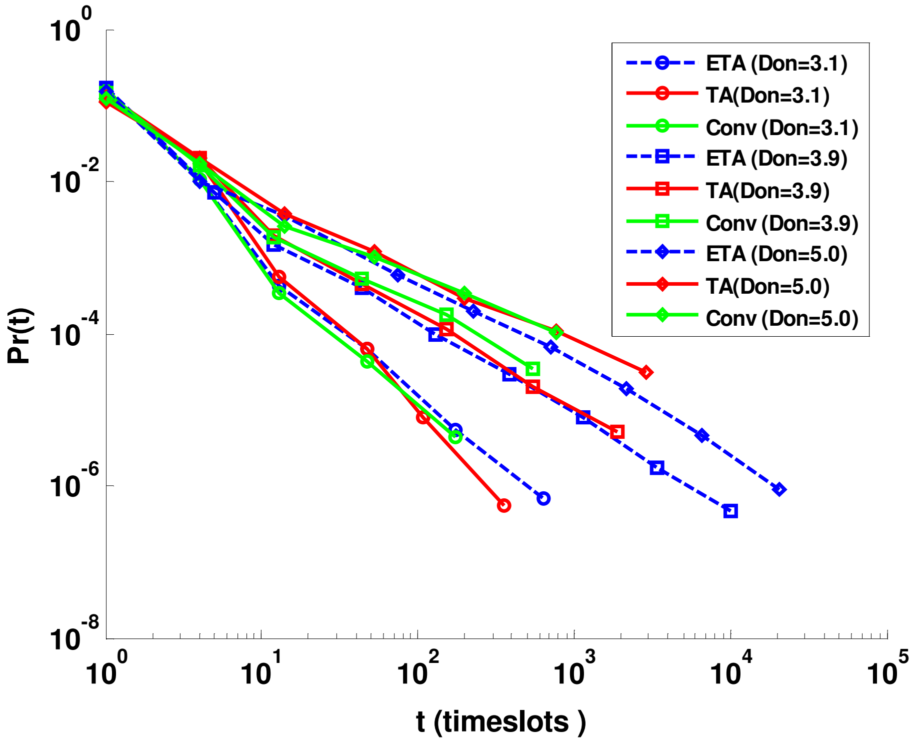


Figure 5.3: PMF of the packet delay

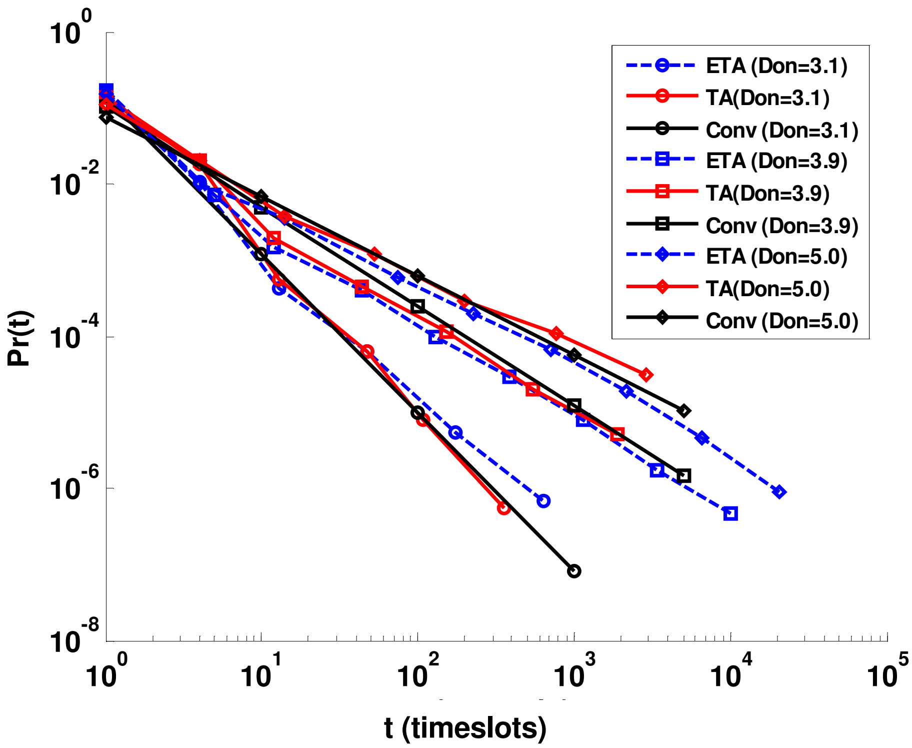


Figure 5.4: PMF of the packet delay with conventional best fit data

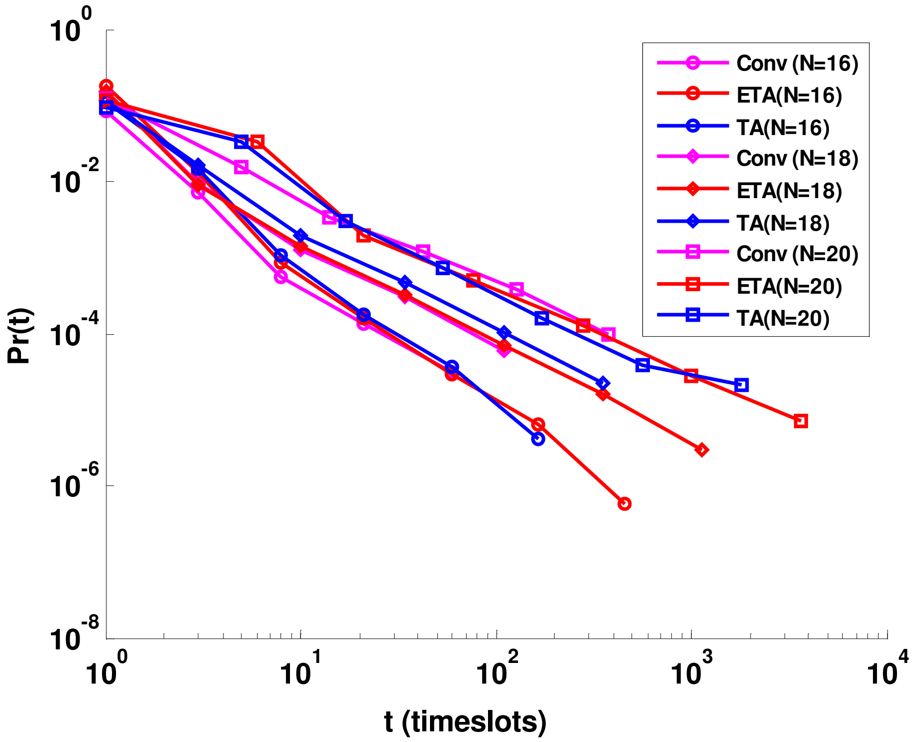


Figure 5.5: PMF of the packet delay

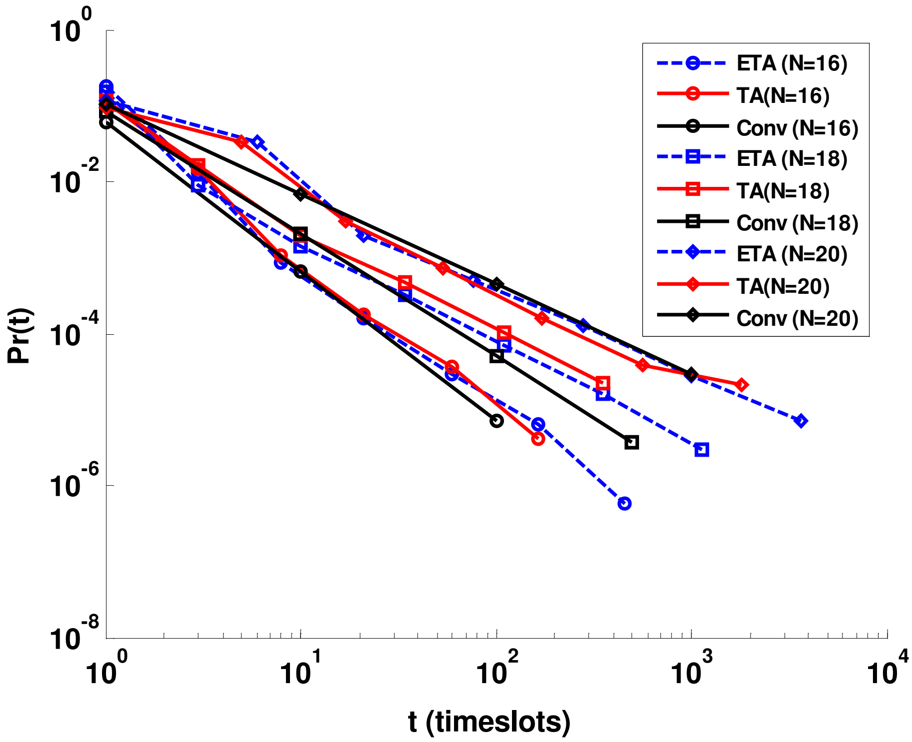


Figure 5.6: PMF of the packet delay with conventional best fit data

5.3.2 Numerical Result- the Queuing Behaviour

In this section the queuing behaviour of the FIFO buffer is observed to further test E_TA. For this purpose, we are not using active probing but we use passive monitoring to collect the statistics. The queuing behaviour is observed at the end of the ON period for E_TA, and this is compared to that collected from TA. Results were obtained for the instant of the arrival of the last packet in the ON period in the TA model and at the arrival instant of the ER batch in the E_TA model. Therefore, we defined the queue state probability in the FIFO queue at the end of the ON periods as:

Pr (k) = Pr (k packets in the queue at the end of the ON period)

Therefore, the results from TA will be comparable to those collected from E_TA-FIFO. We have two sets of results to evaluate the accuracy of the queuing behaviour in E_TA-FIFO model; all results were collected at the end of 10^6 timeslots.

Figure 5.7 shows the results when varying the mean ON period, Don, of the individual traffic source. High variability of the power law traffic can be observed in the tail of the results. This explains why comparison between E_TA and TA queuing behaviour is better from k= 10^1 to k= 10^2 but the results of E_TA and TA tend to deviate towards the tail. Despite that, Figure 5.7 shows E_TA-FIFO can accurately reproduce the queuing behaviour compared to TA and moreover, for the same length of processing time E_TA-FIFO gives better coverage of the queue states. In Figure 5.8 the number of traffic sources is varied: 16, 18 and 20 number of traffic sources and the results from E_TA-FIFO mimic the queuing behaviour of TA.

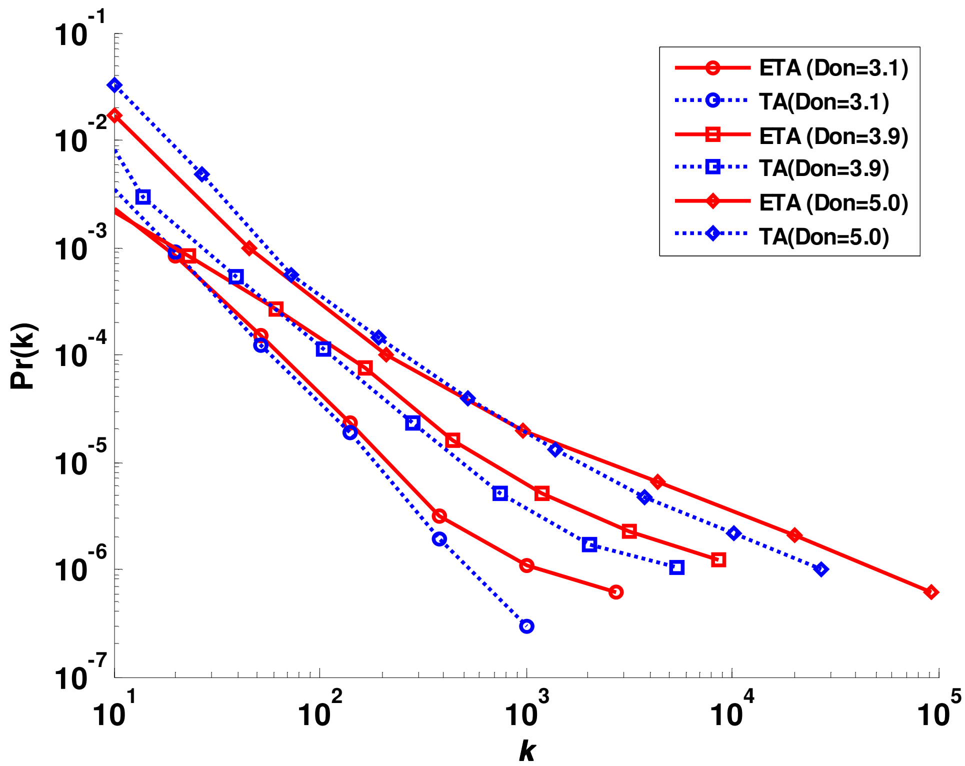


Figure 5.7: The queuing behaviour of different mean ON period

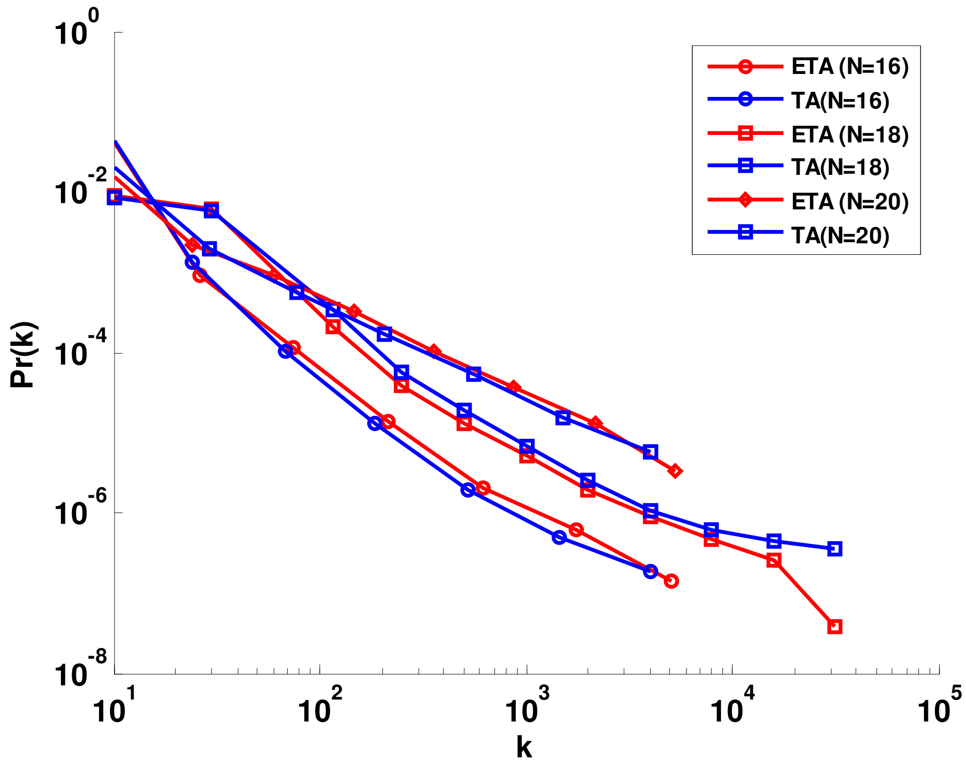


Figure 5.8: The queuing behaviour for different number of aggregated traffic sources

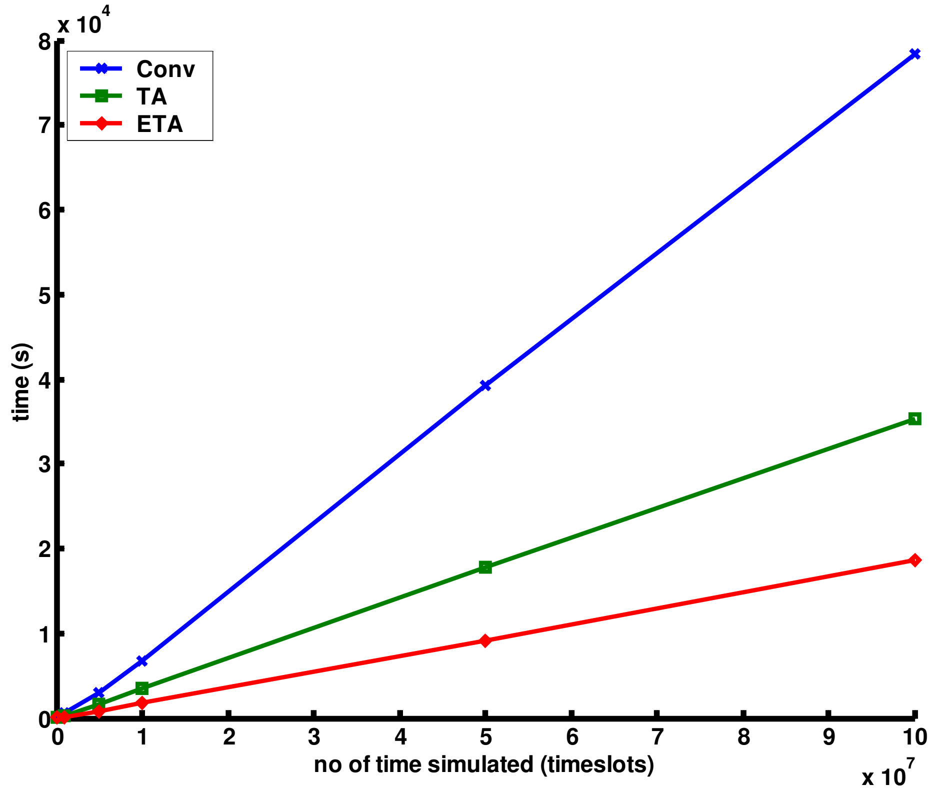


Figure 5.9: Reduction of time in E_TA-FIFO

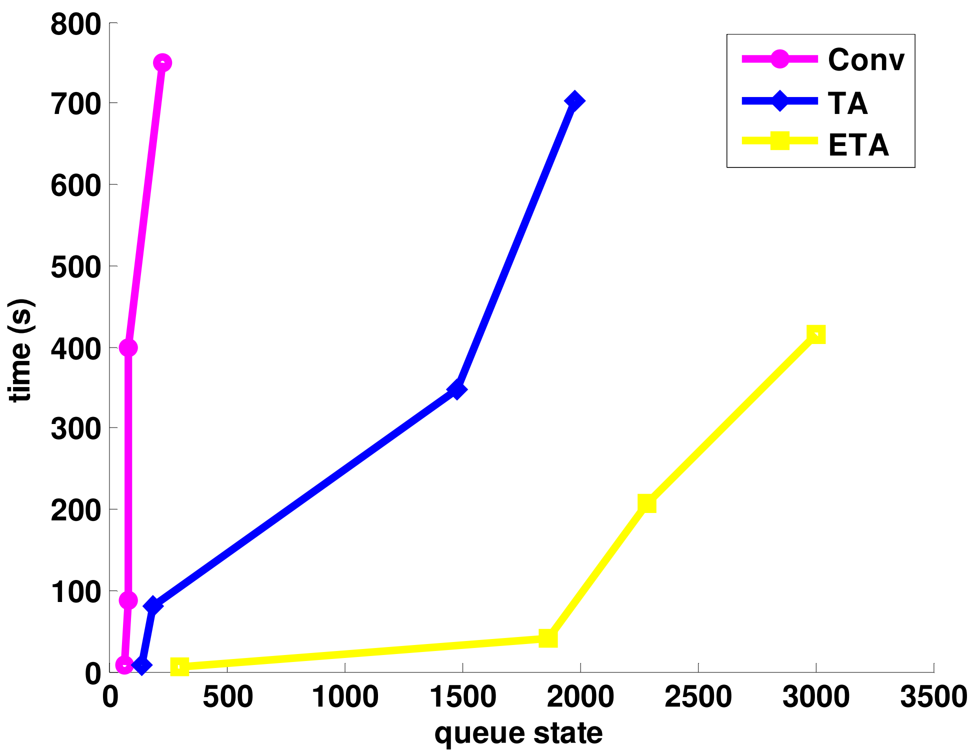


Figure 5.10: The queue state coverage.

The amount of acceleration for E_TA-FIFO found by comparison with conventional and TA. Compare in terms of the speed up, and the state coverage. Figure 5.9 presents the comparison of the processing time taken (i.e. CPU processing time) by E_TA-FIFO, TA and conventional for the same duration i.e. the same number of timeslots. A conventional scenario was set up with Doff=10, Don=5, N=10, R=1 and C=5. An equivalent E_TA-FIFO was derived having $\overline{ER}$=4.0704, $Toff_2$=1.7200 and C=5. The seed values of all simulation are fixed in order to get fair comparison results. The timeslots starts at 10^6 until 10^8 is set before the simulation is executed. In this experiment, the result shows that E_TA-FIFO runs in significantly less processing time compared to conventional and even TA. For duration of 10^8 timeslots E_TA-FIFO takes only 18000 seconds but for conventional simulation needs 60000 seconds processing time.

Figure 5.10 shows the state coverage for E_TA-FIFO achieved for the same simulation run duration (timeslots). The state coverage plots the number of states covered in the output queuing results. A scenario was set up with Doff=10, Don=3.1, N=10, R=1 and C=5. An equivalent E_TA-FIFO was derived having $\overline{ER}$=2.2137, $Toff_2$=0.9354 and C=5. Comparison with conventional and TA clearly shows that E_TA has achieved significantly greater coverage in less processing time. This also indicates that E_TA-FIFO will be able to reach steady-state in less time compare to conventional. (See section 4.3.2 for details of the steady state simulations)

5.4 Summary

This chapter presented an accelerated simulation technique called Enhanced TA (E_TA) which was developed from a previous accelerated simulation technique called Traffic Aggregation (TA). It lays out the details concerning the Excess-Rate (ER) packets concept which is used to determine the size of the 'instantaneous ER batch' arrivals, and describes how this idea is used to create more simulation speedup.

Active probe measurement is used to compare the E_TA model with a conventional simulation and with TA. The packet delay probability is observed and the experimental results presented for different scenarios. The queuing behaviour of E_TA-FIFO is also compared to TA. The results of the packet delay experiments indicate that E_TA-FIFO is accurate compared to conventional and TA simulation methods.

Then the processing times are compared between E_TA-FIFO, TA and conventional when same the duration of timeslots is executed. Significant reduction of processing time has been achieved. This chapter concentrates on simulating FIFO packet buffer with 1 priority traffic, in the next chapter a priority scheduler will be introduced.

Chapter 6

ENHANCED TRAFFIC AGGREGATION WITH PRIORITY BUFFER (E_TA-PQ)

6.1 Introduction

This chapter presents the details of E_TA-PQ[21]. In this chapter the traffic has two levels of priority: high priority and low priority. It is a critical aspect of E_TA-PQ that uses a hybrid method which involves the Generalized Ballot Theorem (GBT) to get the waiting time for the lower priority traffic. The waiting time includes busy periods of the high priority traffic, and this means that the high priority traffic can be removed completely from the simulation, giving extra speedup.

The numerical results for the packet delay probability and the queuing behaviour of the low priority traffic are presented. The results are used to validate the accuracy of the E_TA-PQ compared to conventional simulation.

6.2 The Concept of E_TA-PQ

The E_TA algorithm for a priority scheduler supports two priority levels (see Figure. 6.1); the priority type is non-preemptive, because all real packet scheduling systems are non-preemptive. The high priority traffic is set to represent VoIP, and, as an aggregate of Constant Bit Rate voice traffic streams (for EF PHB), is well modelled by a Poisson process [Bona01], we set the high priority traffic model to be Poisson. In contrast, the low priority traffic is for AF and BE PHB groups, which includes: FTP, TELNET and SMTP (email) traffic. Studies [Paxs95] have shown that over larger time scales most of the AF and BE traffic appears to be bursty, and dominated by self-similar characteristics.

The scheduler in PQ works as follows: when all the packets in the busy period of the high priority sub queue are served, the server will switch to low priority sub queue. Low priority packets may arrive after the busy period of high priority ends, in which case these

[21] The priority scheduler is a HOL static priority and it is explain in detail in Section 2.4.1

packets are served right away (when the high priority sub queue is idle). However, if low priority packets arrive during a busy period of high priority, then these packets will have to wait until the busy period ends.

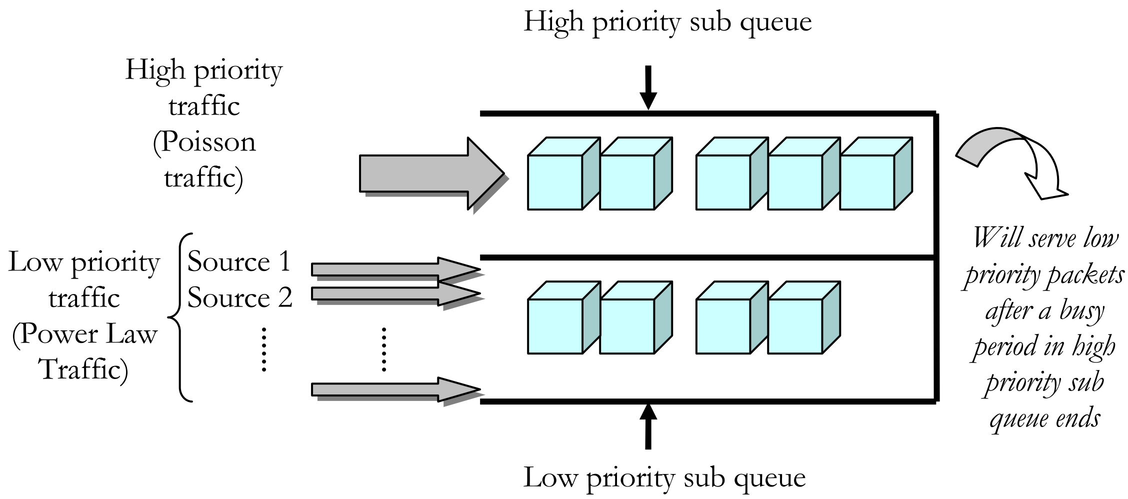


Figure 6.1: The priority buffer of two priority levels.

Studies of queuing analysis, e.g. in [Scho90a, Scho90b, Scho92, Scho93] used the powerful GBT [Gure00] to analyze the steady state waiting time probabilities in PQ through busy period analysis. In the PQ, the waiting time of the low priority packets depends very much on the high priority traffic, (see Figure 2.2). The waiting time for any packet is defined as the total time from entry to the system to entry to the server (being a non-preemptive scheduler, a packet will then complete transmission after its service time with probability 1).

The analysis of the waiting time probabilities in PQ by [Scho90a] was originally for Poisson and renewal traffic. In this thesis the earlier analysis of the waiting time probabilities is applied to power law traffic, and then used as part of the kernel of a simulation algorithm.

In section 4.1 we explained in detail the structure of ER batch model. The mean service time of each ER batch in a FIFO scheduler is given by equation (5.3). For the E_TA-PQ model, the waiting time of the low priority traffic is equivalent to the total time each ER batch spends in the buffer. To find the waiting time probabilities of the low priority traffic requires the use of convolution. Based on the waiting time probability for two priority levels in [Scho90], a packet or in this case an ER batch, say P , arriving in timeslot i, will have to wait behind a number of low priority ER batches and high priority packets. This total wait has 3 essential components:

1. the total number of packets of equal or higher priority that were already present in the buffer at the end of timeslot *i*-1
2. all the packets of higher priority that ***P*** that arrive in timeslot *i*
3. higher priority packets that arrive subsequently, but before ***P*** enters service.

Define component 1 the unfinished work- as *u (k),*
component 2 the wait caused by high priority arrival in timeslot *i*,
component 3 the extra wait cause by the subsequent arrivals of the high priority traffic.

6.3 The Waiting Time Probability for the Low Priority Traffic

The waiting time probability of the low priority traffic consist of

1. the service time of ER batch itself.
2. the extra wait cause by subsequent arrivals of the high priority traffic.

In this research, * denotes the convolution operator and all packet sizes are fixed.

Definitions: High Priority will be denoted 1 and Low Priority 2

$a_1(k)$ = Pr(k packets of high priority arrive in any timeslot)

$a_2(k)$ = Pr(k packets of low priority arrive in any timeslot)

$a_c(1,k)$ = Pr(k units of work of high or low priority arrive in any 1 timeslot)

ρ_1 = load of high priority traffic

ρ_2 = load of low priority traffic

$u(k)$ = Pr(an arriving batch of low priority packets sees k packets of unfinished work of high or low priority already in the queue)

$V_2(k)$ = Pr(a low priority batch waits k timeslots due to the unfinished work of other high and low priority arrivals already ahead of it when it joins the queue)

$W_2(k)$ = Pr(a low priority batch has a total wait of k timeslots)

Due to our model of VoIP as priority 1, the high priority traffic is Poisson, so $a_1(k)$ is simply:

$$a_1(k) = \frac{\rho_1^{\ k}}{k!} \cdot e^{-\rho_1} \tag{6.1}$$

The probability of low priority batch arrivals is a discrete distribution whereas the Pareto is a continuous distribution. Using the discrete-time queuing model for long-range dependent (LRD) traffic in [Pitt00], the probability that a Pareto batch is size of *k* can be found using (see also Appendix 4):

$$gp(k) = \left(\frac{1}{k-0.5}\right)^{\alpha} - \left(\frac{1}{k+0.5}\right)^{\alpha} \tag{6.2}$$

and

$$\alpha = \frac{B}{B-1}$$

where *B* is the mean size of the ER batch arrivals in the E_TA model. This gives the probability that there is an ER batch (of size *k*) arrivals in any timeslot as:

$$a_2(k) = \begin{cases} 1-q & k=0 \\ q \cdot gp(k) & k>0 \end{cases} \tag{6.3}$$

$$q = \frac{\rho_2}{B}$$

where *q* is the probability of there being a batch in any random timeslot.

Because the distribution of the high and low priority traffic are different the unfinished work, $u(k)$, is actually found from the convolution of the distribution of the high and low priority traffic which gives:

$$a_c(1,k) = \begin{cases} a_1(0) * a_2(0) & k = 0 \\ a_1(k) * a_2(k) & k > 0 \end{cases} \tag{6.4}$$

$$s_2(k) = \begin{cases} s_2(0) \cdot \dfrac{1 - a_c(1,k)}{a_c(1,0)} & k = 1 \\ \dfrac{s_2(k-1) - s_2(0) \cdot a_c(1,k-1) - \sum_{i=0}^{k-1} s_2(i) \cdot a_c(1,k-i)}{a_c(1,0)} & k > 1 \end{cases} \tag{6.5}$$

where $s_2(0) = 1 - E[a]$, $E[a]$is the mean number of arrivals per timeslot of both high and low priority arrivals.

$$E[a] = \rho_1 + \rho_2 \tag{6.6}$$

Hence,

$$u(k) = \begin{cases} s_2(0) + s_2(1) & k = 0 \\ s_2(k+1) & k > 0 \end{cases} \tag{6.7}$$

The virtual waiting time for the low priority batch can be found using

$$V_2(k) = a_1(k) * u(k) \tag{6.8}$$

as all two distributions are independent.

The GBT is applied to find the extended waiting time caused by subsequent arrivals of high priority packets. Stated in the form that is applied, the GBT is:

$$pr\{y = k | x = i\} = \frac{i}{k} \cdot \left(pr(k - i)\right) \tag{6.9}$$

where y – number of packets in the busy period of high priority packets.

x – number of packets initially in the system.

$pr(k-i)$ - Probability that k high priority packets arrive in (k-i) timeslots

The waiting time probability for the low priority batch is found using:

$$W_2(k) = \begin{cases} V_2(0) & k = 0 \\ \sum_{i=1}^{k} \left[V_2(i) \cdot \frac{i}{k} \cdot pr(k-i) \right] & k > 0 \end{cases} \tag{6.10}$$

where:

$W_2(k) =$ Pr (a low priority batch must wait k timeslots before it enters service)

The waiting time probability of the low priority traffic, $W_2(k)$, is calculated before any simulation is executed. All values of $W_2(k)$ are written in the PDF editor table of OPNET (see section 4.2.6.1 for more detail explanation of the PDF editor). In the E_TA-PQ simulation, when an ER batch arrives, the simulation will randomly generate a value using implementation of $W_2(k)$ in OPNET.

6.4 Validations

In this section the concept of E_TA-PQ for a single node network is validated. The numerical results of interest will be for the low priority traffic. The waiting time probability (equation (6.10)) is implemented in E_TA_PQ in order to get accurate results compared to conventional simulation but with fewer simulated events.

This section is divided into two, where the results of the packet delay and the queuing behaviour experienced by the E_TA-PQ are collected. These results will be compared to conventional or/and TA.

6.4.1 Numerical results – Low Packet Delay Probability (for a single node)

In this section the packet delay of the low priority traffic will be of interest. Active measurement is used to collect the packet delay probability of the low priority sub queue. The probes are generated at the sending end and the delay probability is observed at the receiving end of the system (details of active measurement is in section 4.7.). The arrival rate of the probe is set to be 0.001 probe/ timeslot. This low arrival rate is essential in order for the system to work without any significant interruption from the active probes. The steady state packet delay probability is defined as:

$$\mathbf{Pr_{low}}\ (t) = \mathbf{Pr}\ (\textbf{the delay of the low priority traffic is } t \textbf{ timeslot})$$

The parameter values of the conventional and E_TA-PQ low priority traffic (used for experiments in this sub section) are shown in Table 6.1 and Table 6.2. The high priority is Poisson and the arrival rate, λ_1, is equal to 0.2 packet/timeslot [Cisc79]. The comparisons are presented in two versions; 1) direct[22] comparison of the conventional N source with E_TA-PQ; 2) comparison of the power-law best fit line of the conventional with E_TA-PQ. Power-law best fit line of the conventional model is an important reference for the validation of E_TA-PQ (see section 4.7.1).

[22] Direct comparison of the different models (i.e. conventional and E_TA) using the raw data which are binned using exponentially wider bin.

In general we can see that simulation using E_TA-PQ with fewer events can reach lower probabilities in the same length of time. Figure 6.2 and 6.3 shows that the E_TA-PQ model has reproduced the packet delay accurately for different mean ON period of the individual low priority traffic. These results are for aggregated of 4 low priority traffic sources.

Figure 6.4 and 6.5 shows the distribution of the packet delay for aggregated of 6 low priority traffic sources. Here, the mean ON period is varied and the queuing behaviour is observed. Again it shows accuracy and more coverage in the time delay.

Table 6.1: The parameter values of the low priority traffic used in Figure 6.2, 6.3 and 6.6

Doff	Don	N	R	C	$\overline{ER}$	$Toff_2$	Ron	C_{ETA}
10	3.1	4	1	2	2.6904	2.8421	3.0719	2
10	3.5	4	1	2	2.8852	2.7821	3.0805	2
10	3.9	4	1	2	3.0821	2.7462	3.0888	2

Table 6.2: The individual and equivalent traffic sources value used in Figure 6.4, 6.5 and 6.7.

Don	Don	N	R	C	$\overline{ER}$	$Toff_2$	Ron	C_{ETA}
10	3.1	6	3	3	2.5188	1.7740	4.1210	3
10	3.3	6	3	3	2.6455	1.7770	4.1286	3

6.4.2 Numerical Results - The Queuing Behaviour of the Low Priority Traffic

The queuing behaviour is observed at the end of the ON period of E_TA-PQ, due to the modelling structure of ER batch arrivals. As mentioned in section 5.5.2, the queuing behaviour observed at the end of the ON period for E_TA is best compared to the one collected from TA at an equivalent point. Therefore, for the experiment in this section a packet-by-packet, PQ model is developed which has aggregated low priority traffic and we called in this thesis as TA-PQ (this is referred to as TA1 in Figure 6.6 and 6.7). As same as in the previous section, in this section we also focus exclusively on the low priority traffic of Figure 6.1. The steady state probability is defined as

Pr2(k) = Pr (k packets in the low priority sub queue end of the ON period)

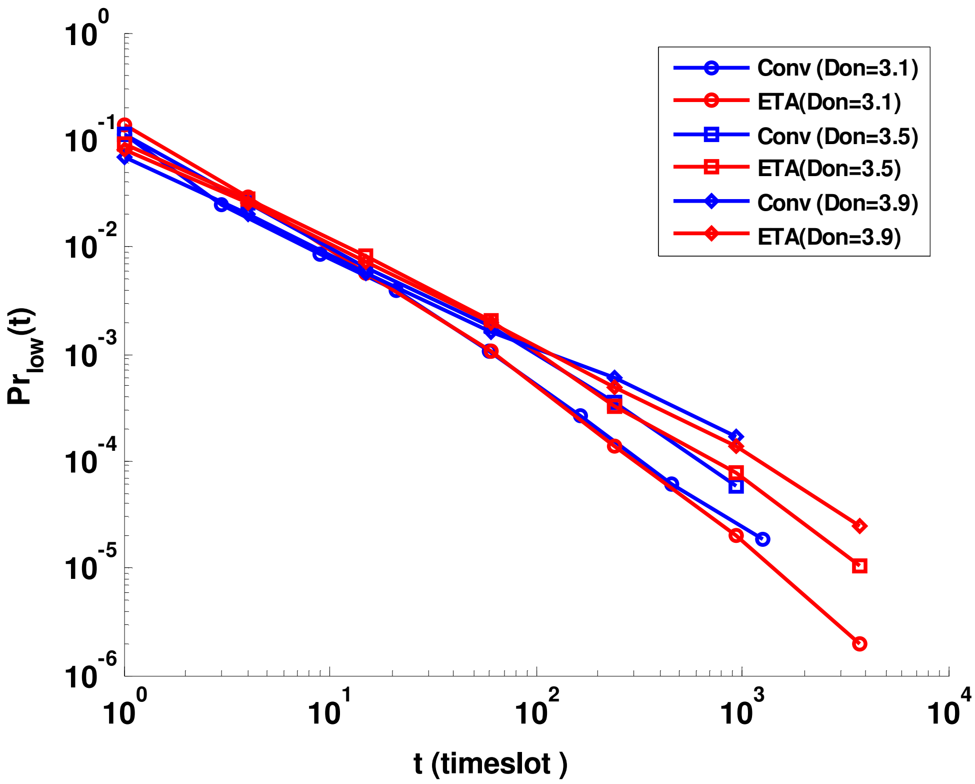


Figure 6.2: The PMF of low priority traffic packet delay for actual data(N=4)

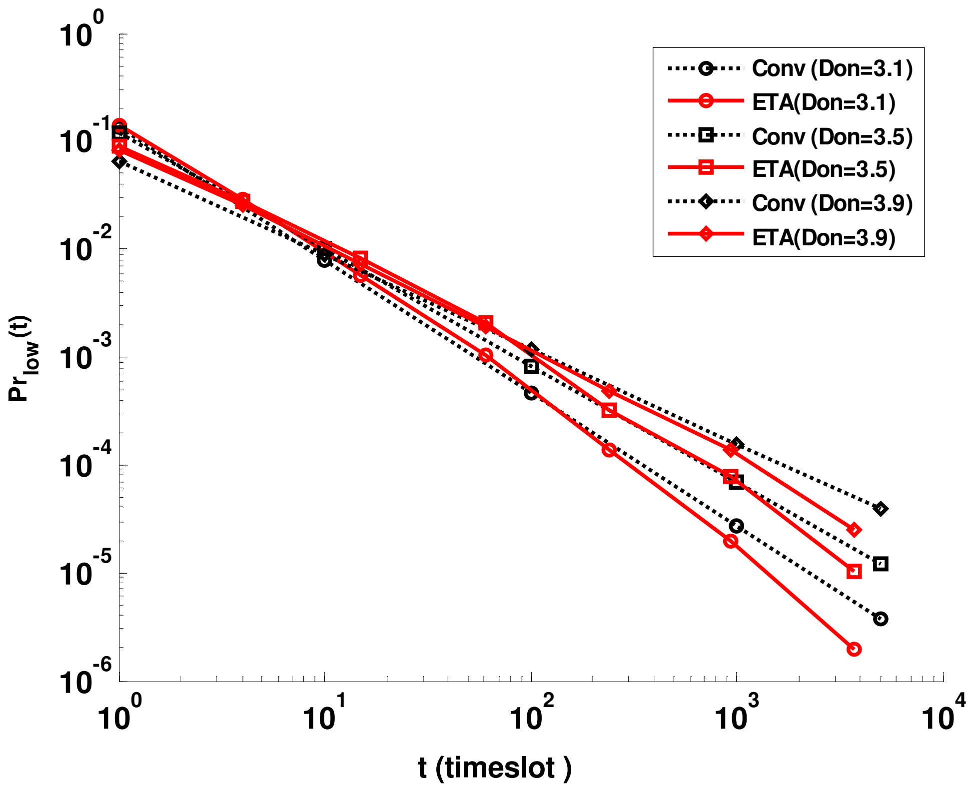


Figure 6.3: The PMF of low priority traffic packet delay for best fit data (N=4).

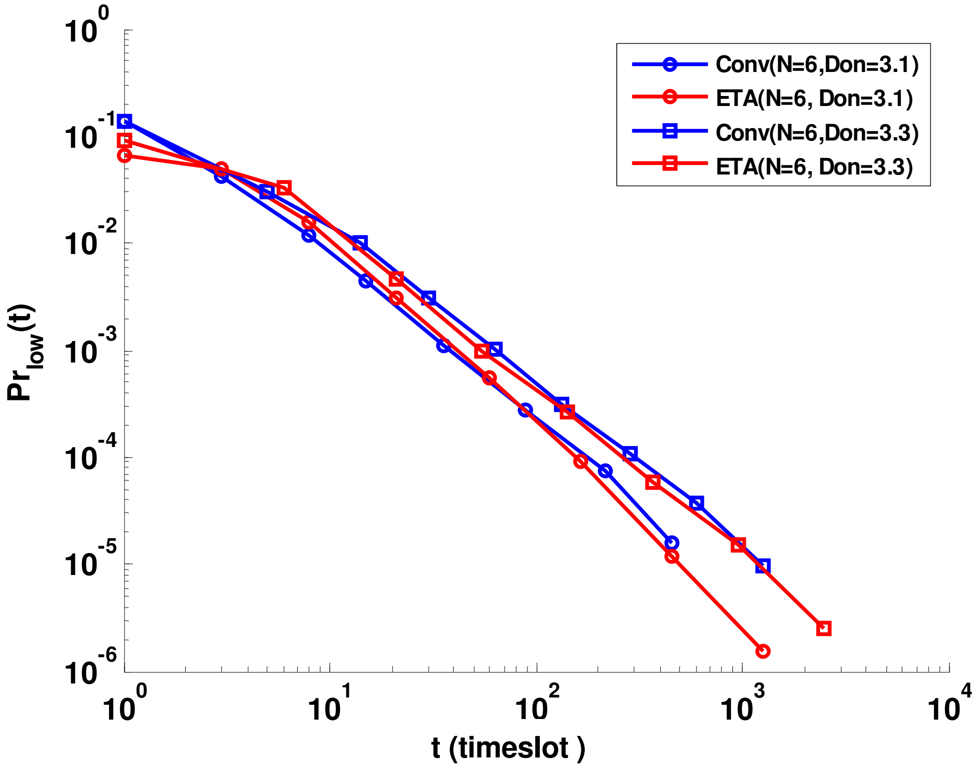


Figure 6.4: The PMF of low priority traffic packet delay for actual data (N=6).

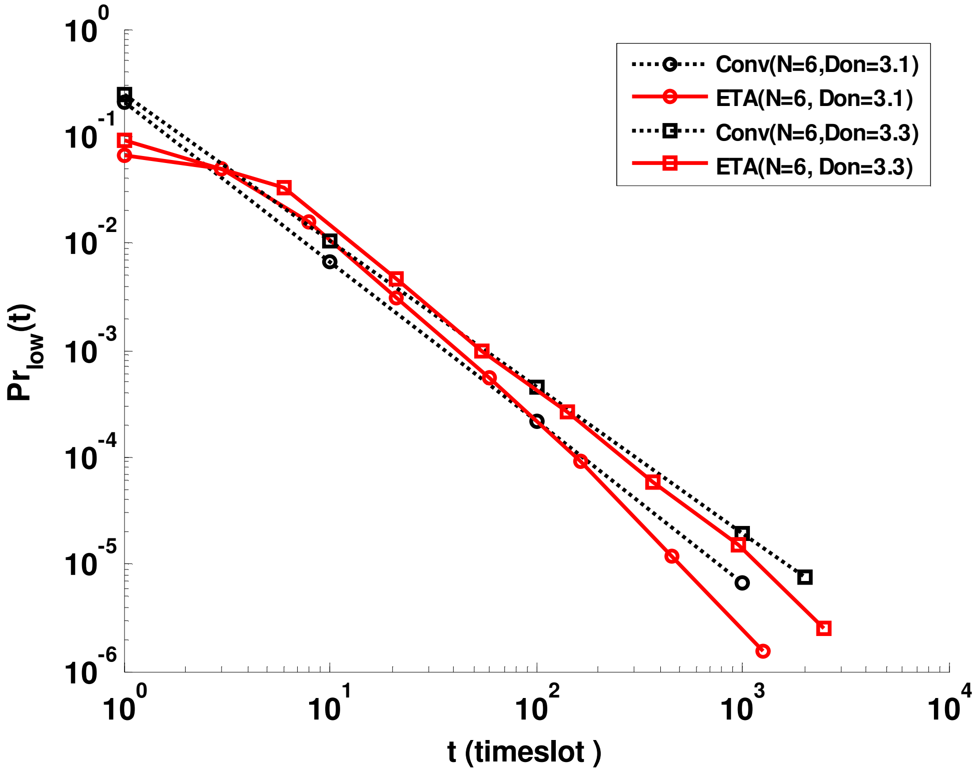


Figure 6.5: The PMF of low priority traffic packet delay for best fit data (N=6)

The high priority traffic is set to have mean arrival rate of 0.2 packet/ timeslot. Two sets of results are presented. One is E_TA-PQ with 4 low priority traffic sources and the other one is with 6 sources of low priority traffic. Figure 6.6 shows the reproduction of the queuing behaviour in the low priority traffic is very accurate compared to the one from TA-PQ model. Figure 6.7 also shows that E_TA-PQ can reproduce accurate queuing behaviour.

Then the processing time taken for N events for TA-PQ and E_TA_PQ is compared in Figure 6.8. All OPNET simulations are terminated using end-simulation interrupts which are delivered to all processors and queues that have the "`endsim intrpt`" attribute enabled. In Figure 6.8, it clearly shown the significant reduction of processing time using the E_TA-PQ model compared to TA-PQ model.

Figure 6.9 shows the coverage of the queue state for the low priority sub queue against the processing time *t*. It shows that E_TA-PQ cover more queue state than TA-PQ in much less time.

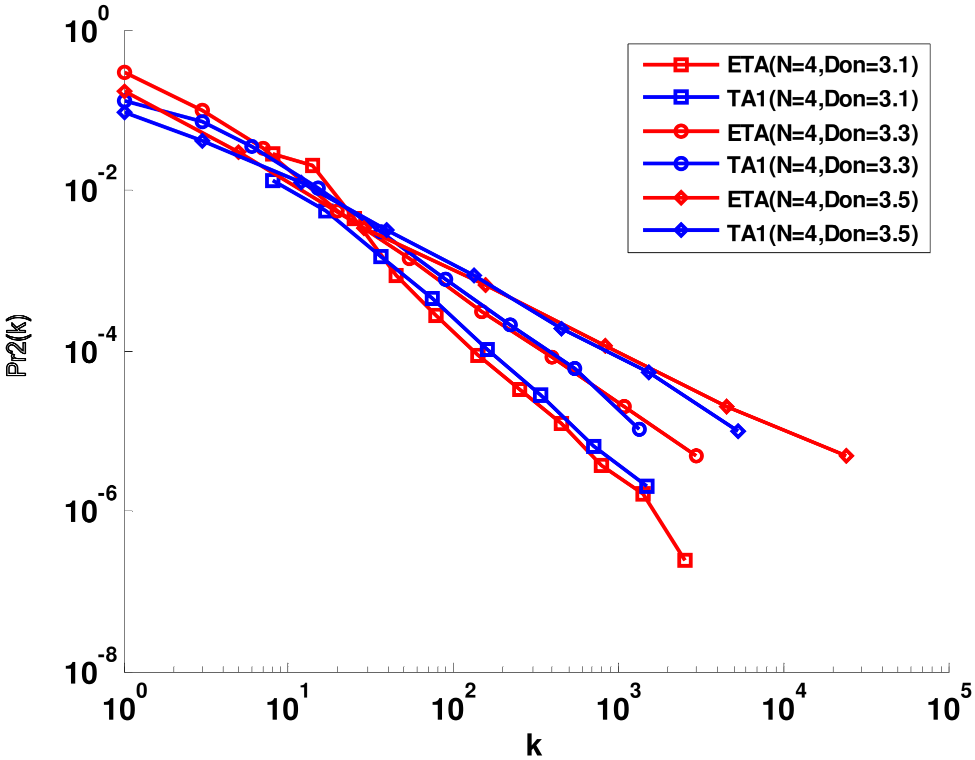


Figure 6.6: The queuing behaviour with 4 the low priority traffic sources

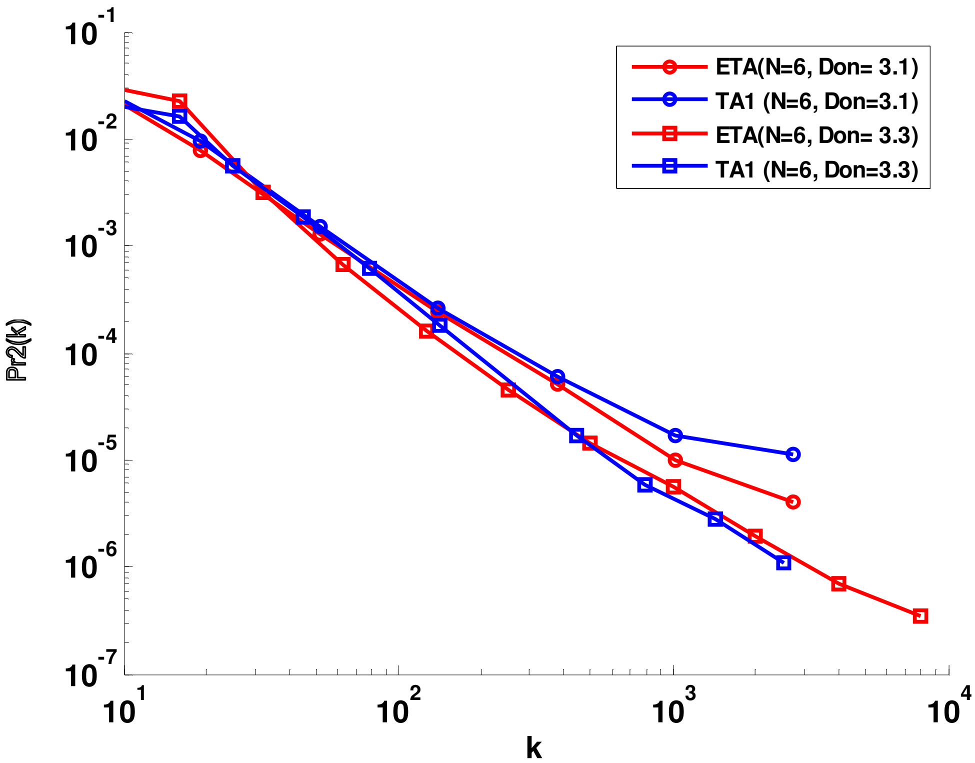


Figure 6.7: The queuing behaviour with 6 the low priority traffic sources

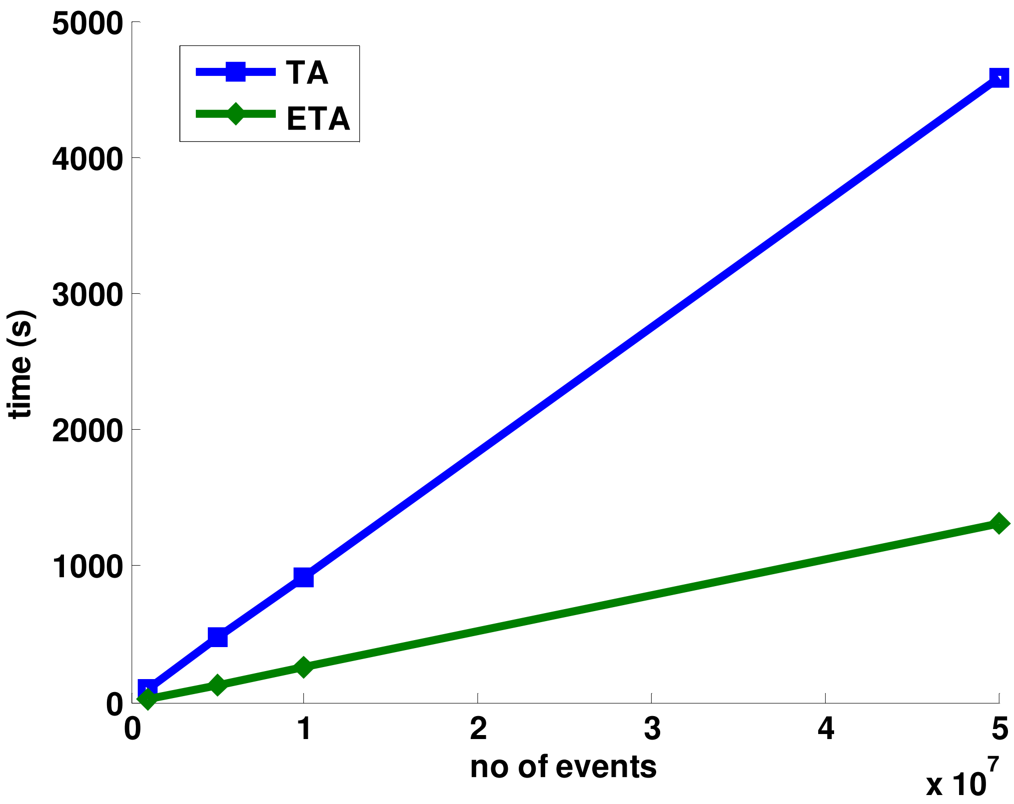


Figure 6.8: Time reduction in E_TA-PQ

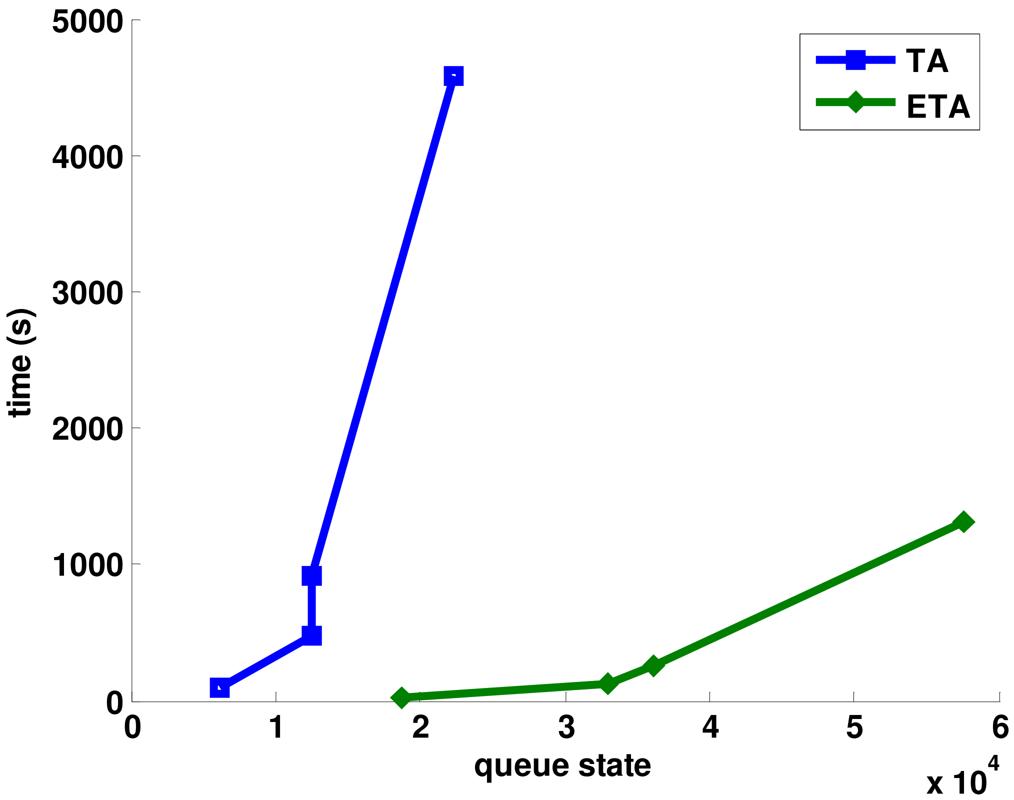


Figure 6.9: Queue state coverage

6.5 Summary

This chapter has detailed the acceleration method for two priority levels. This involves the use of ER batch arrivals and the removal of the high priority traffic. In order to do this the waiting time probability of the low priority traffic is calculated prior to simulation runs to achieved accurate buffer performance. The result of this calculation is then reproduced in a PDF editor during the simulation of E_TA-PQ.

The results of the queuing behaviour and packet delay probability of E_TA-PQ and conventional show excellent accuracy. Processing time reduction has also been achieved and E_TA-PQ achieves more queue state coverage when compared to TA-PQ over an equal number of events (see section 5.2 for events definition for E_TA and TA).

Chapter 7

APPLYING E-TA OVER END-TO-END NETWORK PATHS

7.1 Introduction

This chapter develops the E_TA-FIFO and the E_TA-PQ methods for end-to-end network paths with foreground (FG) and background (BG) traffic. The content of this chapter is divided into two, where the first part of the chapter concentrates on scenarios that have no priority traffic and hence FIFO scheduling. The number of BG traffic sources in the conventional method is replaced by a single equivalent E_TA source where traffic passes through every buffer. The second part is of this chapter presents end-to-end connections with priority buffers. In this second part the E_TA-PQ concept is apply where the traffic of interest (the low priority power law traffic) as the FG traffic. In this case the high priority traffic will be the BG traffic.

7.2 E_TA-FIFO Model with Foreground and Background Traffic

Network simulation has many applications where only a small fraction of the traffic is of specific interest. However, the traffic of interest which we call the foreground traffic (FG) is affected by other, background traffic (BG). This chapter concentrates on lowering the computational requirement to simulate scenarios in networks with FG and BG traffic by using the E_TA method.

In WAN the end-to-end network has ambient traffic from various applications. In particular the BG traffic going through the ingress and egress routers will be heavier than the BG traffic going through the routers inside the WAN. This is illustrated in Figure 7.1. However in this thesis for simplicity all BG traffic is assumed to be uniform.

In the model, the queuing behaviour of the FIFO schedulers is affected by not only the traffic of interest but the other traffic that pass through the same buffer. Figure 7.2a) illustrates the conventional simulation model with FG traffic and N BG traffic sources.

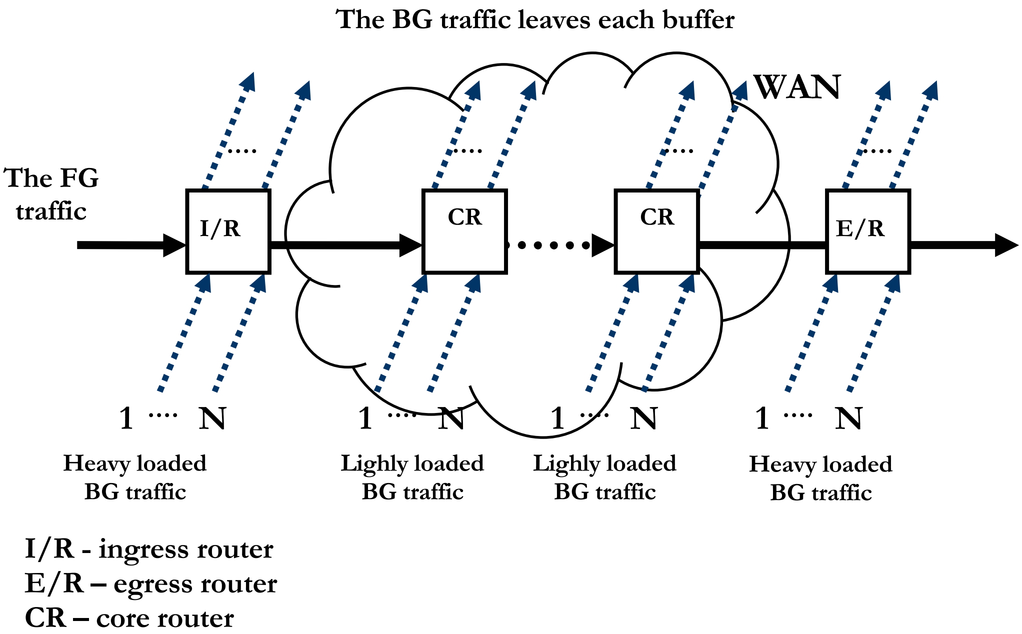
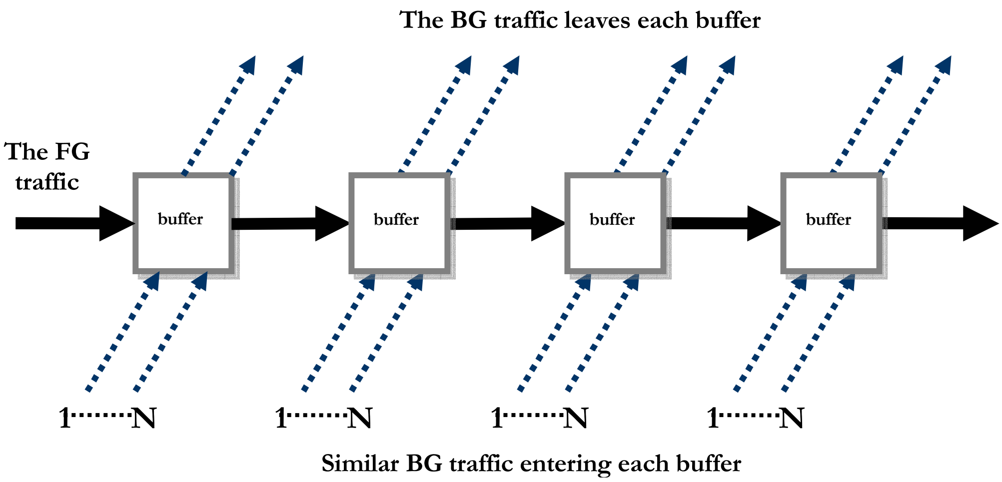
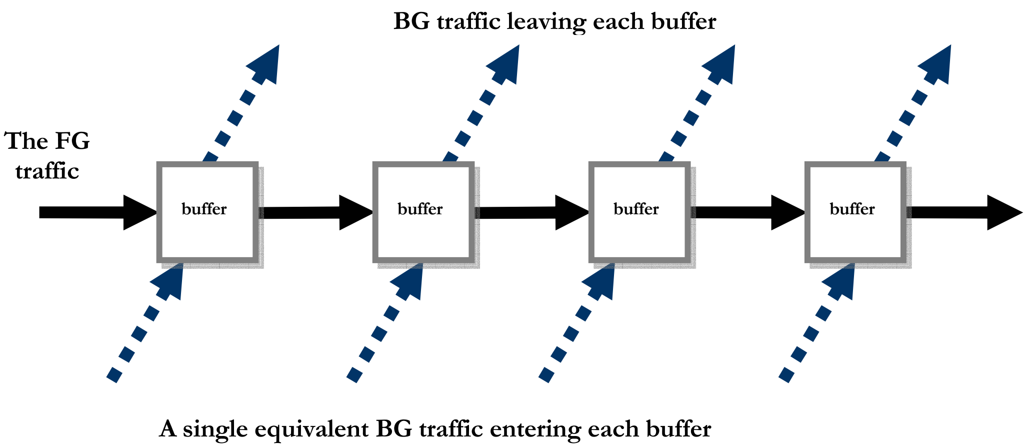


Figure 7.1: The end-to-end-path of WAN network



a) The FG and BG traffic in conventional end-to-end model



b) The FG and BG traffic in E_TA end-to-end model

Figure 7.2: The conventional and E_TA-FIFO end-to-end model

The service rate for all the nodes is set at C packets /timeslot. All the nodes in the end-to-end connections have FIFO schedulers.

E_TA-FIFO was used to model the BG traffic so the process of generating packet-by-packet sources (as in the conventional model) is eliminated and only a single source is used as a substitute. This concept is illustrated in Figure 7.2 b), where the N sources in Figure 7.2 a) are replaced by a single BG traffic source. This accelerated simulation model has not just aggregated the BG traffic (from N sources to a single ON/OFF source), but with the aid of E_TA, instead of packets being fed into the all the network nodes, the ER batch arrival method was used to feed all the network nodes. This replacement not only simplifies the model but also gives significant speedup as fewer events are simulated.

As in previous chapters, evaluation of the end-to-end delay is studied for the actual end-to-end delay using active measurements.

7.2.2 Deriving an Equivalent E_TA-FIFO Model for BG Traffic Substitution

To apply the E_TA-FIFO method to the BG traffic, a suitable service rate for the BG traffic has to be defined because, using a FIFO scheduler, the service rate is shared between the FG and BG traffic. However the amount of capacity used by the BG traffic must be known to provide the parameters defining E_TA.

Given the parameters of the conventional model end-to-end connection setting:

$Doff_{FG}$	Mean OFF sojourn time of the foreground traffic (timeslot)
Don_{FG}	Mean ON sojourn time of the foreground traffic (timeslot)
N_{FG}	The number of foreground traffic sources
R_{FG}	The packet arrival rate of the foreground traffic (packet/ timeslot)
T_{FG}	The mean total packet rate of the foreground traffic (packet/ timeslot)
$Doff_{BG}$	Mean OFF sojourn time of the background traffic (timeslot)
Don_{BG}	Mean ON sojourn time of the background traffic (timeslot)
N_{BG}	The number of background traffic sources
R_{BG}	The packet arrival rate of the background traffic (packet/ timeslot)
T_{BG}	The mean total packet rate of the background traffic (packet/ timeslot)
C	The overall service rate of the buffer (packet/timeslot)

Before the amount of capacity of the background traffic is calculated, the mean of the packet rate of the FG and BG traffic need to be known. For the FG traffic, the mean packet generation rate can be calculated using:

$$T_{FG} = \frac{Don_{FG}}{Don_{FG} + Doff_{FG}} \cdot R_{FG} \cdot N_{FG} \tag{7.1}$$

For the BG traffic, the mean packet generation rate can be calculated using:

$$T_{BG} = \frac{Don_{BG}}{Don_{BG} + Doff_{BG}} \cdot R_{BG} \cdot N_{BG} \tag{7.2}$$

The proportion of capacity for the BG traffic is the ratio of the BG traffic to the total mean;

$$C_{BG} = \frac{T_{BG}}{T_{BG} + T_{FG}} \cdot C \tag{7.3}$$

This C_{BG} is then used to calculate the parameter of E_TA-PQ. Table 7.1 shows the value of the FG and BG traffic used in the conventional and E_TA-PQ. Active measurement is used to measure the end-to-end delay performance. The delay probability is defined as

Pr_{FG} (*t*) = Pr [the end-to-end delay of the FG traffic is *t* timeslot]

Table 7.1: The values of the FG and BG traffic for Figure 7.3

Conventional			E_TA		
	FG	BG		FG	BG
Don	2.5	5	Ton	2.5	2.5954
Doff	7.5	10	$Toff_2$	7.5	0.9713
R	6	1	Ron	6	7.1227
C_{BG}	-	-	C_{BG}	-	6
N	1	9	N	1	1

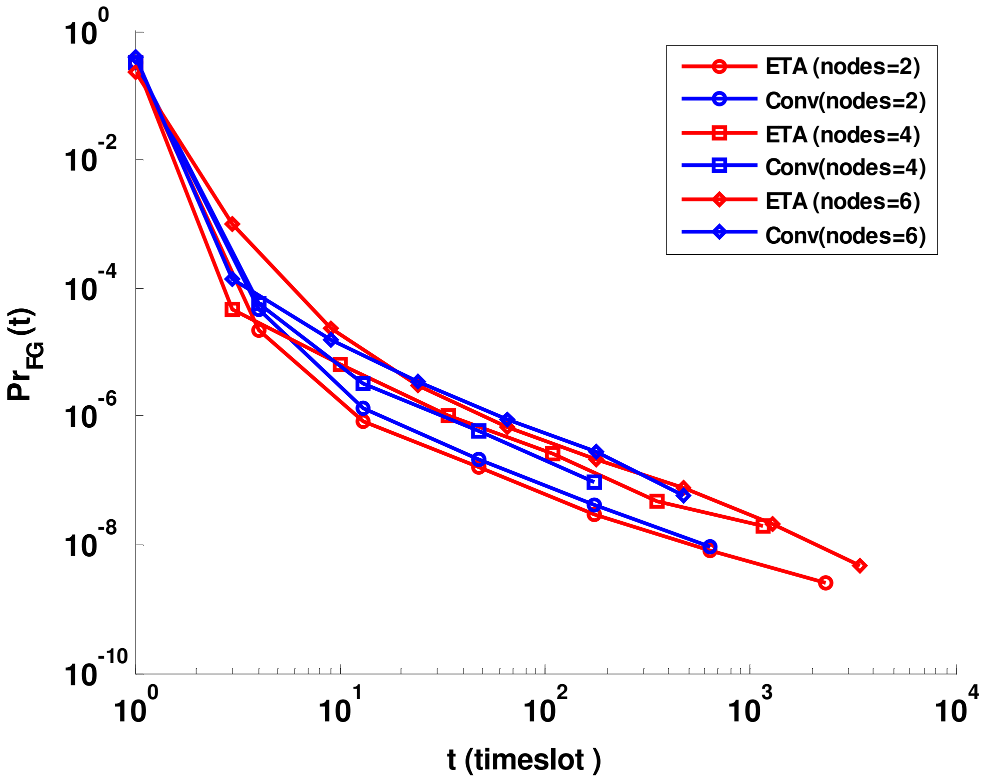


Figure 7.3: The PMF of the FG end-to-end delay

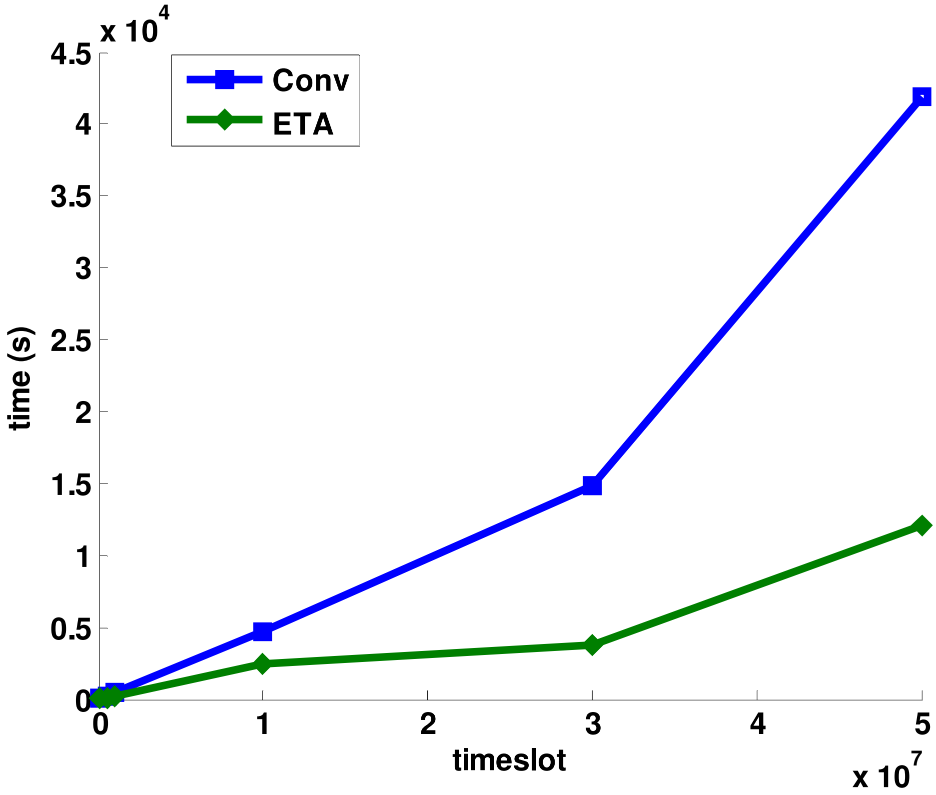


Figure 7.4: Processing time for E_TA-FIFO end-to-end model.

Figure 7.3 shows the results of the E_TA-FIFO and conventional for 2, 4 and 6 number of nodes. The results shows excellent accuracy of E_TA-FIFO compared to conventional.

Then the processing time taken for the end-to-end delays across 4 nodes is compared, where simulations are run for different lengths of time, measured as timeslots. Figure 7.4 shows that E_TA requires much less processing time compared to the conventional method for the same number length of timeslot.

7.3 E_TA_PQ Model in High and Low Priority Traffic

Now the E_TA-PQ is apply over end-to-end connections, where each node's buffer has a PQ scheduler. In this section the traffic is divided into two: high priority traffic and low priority traffic. The conventional model has N low priority power law traffic sources and one high priority traffic source (which is model by a Poisson process) as the BG traffic.

With a single equivalent low priority traffic, and high priority traffic removed, E_TA-PQ is simplified for priority traffic. This is illustrated in Figure 7.5 and 7.6.

In the following experiments the high priority traffic arrival rate is again 0.2 packet/ timeslot. The active probe measurement is again used to collect the low priority packet delays. Table 7.2 shows the arrival rate of the low priority traffic individual sources (in the conventional model), and the arrival rate of the equivalent low priority source (in E_TA-PQ). The steady state probabilities we are collecting using the probes is defined as:

$$\mathbf{Pr_{low}\ (\mathit{t}) = Pr\ [the\ end\text{-}to\text{-}end\ delay\ of\ the\ low\ priority\ traffic\ is\ \mathit{t}\ timeslot]}$$

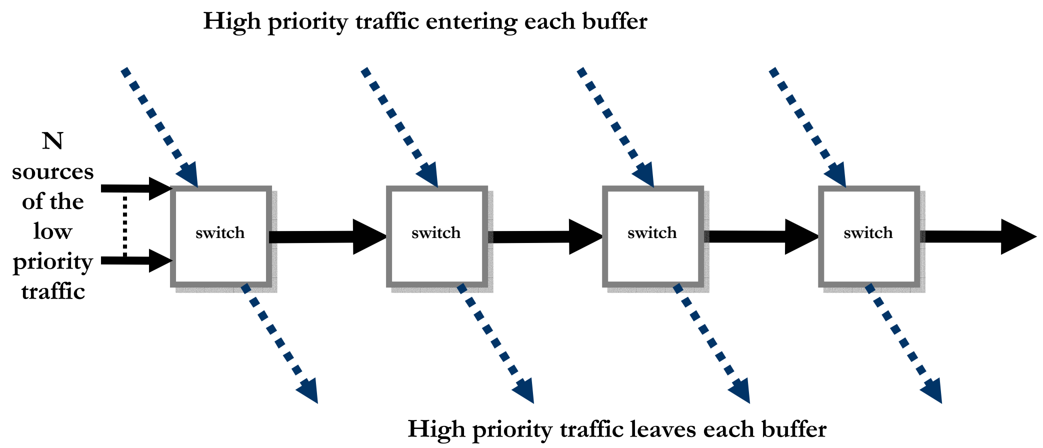


Figure 7.5: The high and low priority traffic in Conventional model with non-FIFO scheduler

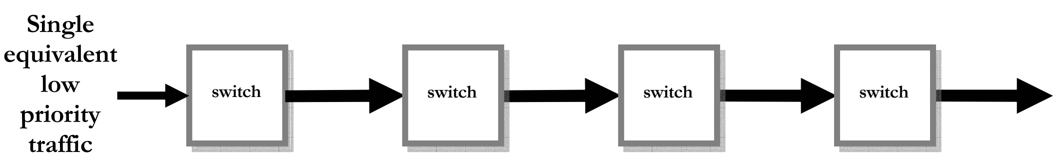


Figure 7.6: The E_TA model with non-FIFO scheduler for end-to-end connections

Table 7.2: The values of the low priority traffic for Figure 7.7

Conventional FG		ETA FG	Conventional BG	
Don	3.1	-	λ_{BG}	0.2
Doff	10	-	-	-
R	1	-	-	-
N	4	-	-	-
$\overline{ER}$	-	2.6904	-	-
$Toff_2$	-	2.8421	-	-
Ron	-	3.0719	-	-
C_{FG}	2	2	C_{BG}	1

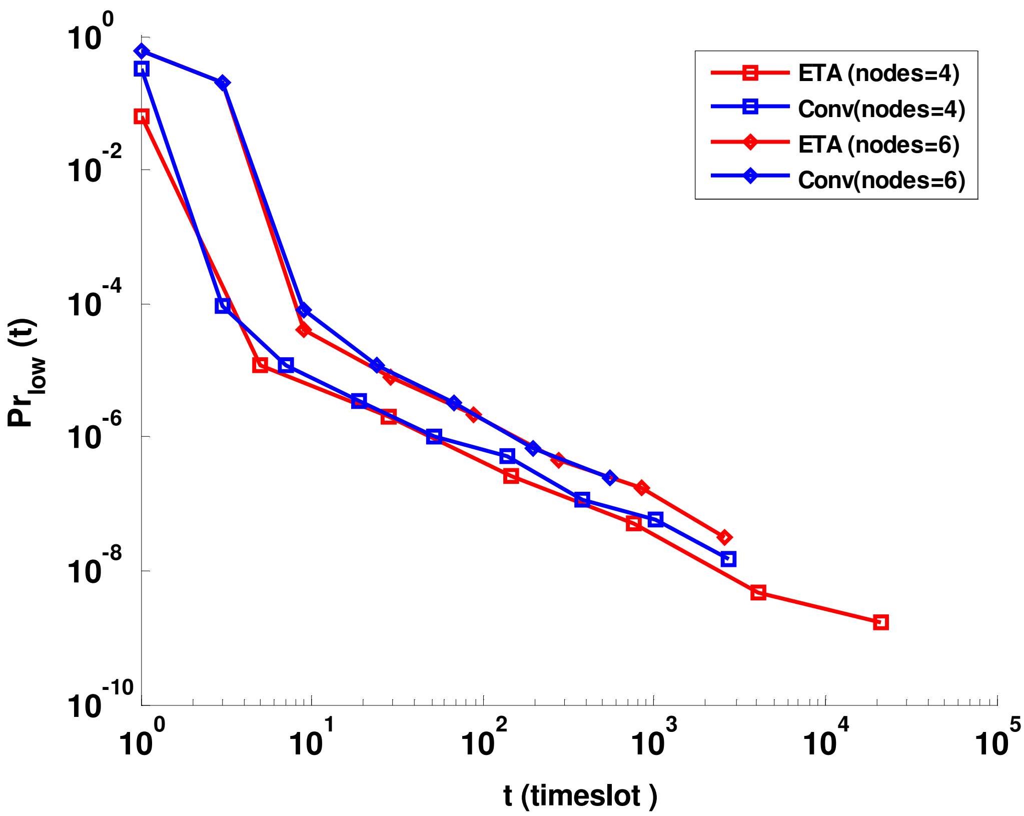


Figure 7.7: The PMF of the end-to-end delay of the low priority traffic

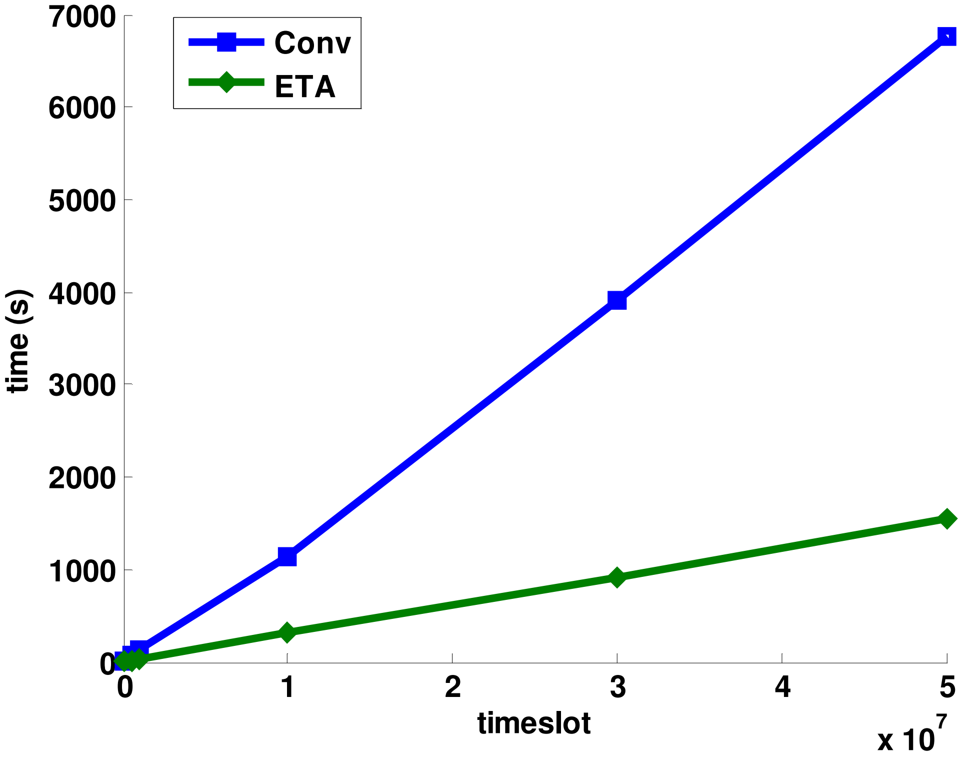


Figure 7.8: Processing time E_TA-PQ end-to-end model

The experiments were done for 4 and 6 nodes. Figure 7.7 shows that the end-to-end delay probability of the low priority traffic can be reproduced accurately by E_TA-PQ.

In Figure 7.8, the simulations are set to run at different length of timeslot and the processing time (in seconds) is observed. Compared to conventional simulations, E_TA-PQ provides a very significant processing time reduction.

7.4 Summary

This chapter has presented E_TA used for end-to-end connections involving both a FIFO scheduler and a PQ scheduler. The results show the end-to-end delay probability for E_TA compared to the conventional model. The accuracy of these results gives confidence that the E_TA method can form the basis of an acceleration simulation technique for PQ scheduling. Further more very significant reductions in simulation time have been achieved.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusion

This thesis reports research into a new accelerated simulation method for packet multiplexing. This new method, E_TA, is a further development of the existing technique called TA. E_TA is an improvement to TA in the following ways:

- E_TA further reduces the event count for a FIFO buffer
- E_TA, unlike TA, also supports priority queuing

There were a few difficulties faced in the process of this research and one of them was applying power law process to the analysis of the waiting time probabilities for priority queuing. The waiting time probability analysis for PQ originally used Poisson input or renewal processes. Power law processes e.g. the Pareto distribution in this case are generally continuous and a discrete version had to be computed before it was used in the analysis. Another difficulty was applying the analysis (the outcome of the waiting time probability) to the simulator.

The novelty of the E_TA method is that the original TA single source model is further minimized until it becomes simply a series of arrival instants. At these instants a batch of packets arrive: the size of this batch is such as to represent the ER arrivals only. Furthermore priority scheduling is achieved by use of the GBT. Validations of E_TA have showed accuracy in both the queuing behaviour and speed up in the processing time.

In this research E_TA has also been applied to end-to-end paths for nodes with FIFO buffers and nodes with priority buffers. In the former case the total traffic is divided into FG and BG traffic flows, and E_TA is applied to replace the BG traffic. Active probes are used to measure the end-to-end delay of the FG traffic in the conventional model and the E_TA-FIFO. In the latter the E_TA-PQ is tested over end-to-end paths. E_TA has shown surprisingly good results in terms of packet delay probabilities and queuing behaviour compared to conventional and TA. Also very importantly, significant simulation event reduction has been achieved.

The E_TA accelerated simulation method can be use in future network planning for cost estimation or network modelling, where it should provide faster evaluation with accuracy. However considerable further work has to be done in order to apply E_TA to real network applications. This is now considered in the next section.

8.2 Future Work

The work reported in this thesis is a significant further development of that in the earlier thesis of the Ma [Ma03]. Where Ma's work showed that TA has application to power-law traffic and FIFO buffers, this thesis has developed TA into E_TA. E_TA can form the basis of an accelerated simulation technique for power-law traffic and priority scheduling. There are still important tasks to be done to create a practical simulator for E_TA. These tasks include:

- The method of E_TA developed in this thesis depends on the parameters of the aggregated traffic, TA. The next step for the future work is to parameterize E_TA with traffic models from real applications. To fit into the real parameters of the real network traffic, parameter *H* can be measured using $H = (3-\alpha)/2$ and α calculated to represent the heavy-tailed property of the ON periods in the ON/OFF sources.

- TA was originally developed for fixed packet lengths. E_TA can be extended to work with variable packet lengths. However this will need modification in the TA algorithms and in the use of the GBT specifically.

- This research has divided the traffic into high priority traffic for voice and video streaming protocol, and low priority traffic such as FTP and emails. Further work could generalize to more than 2 priority levels, and consider schedulers such as WRR as well as PQ. To apply other schedulers such as PQ-pre emptive will need modifications in the algorithms in chapter 6 to calculate the waiting time probabilities of the low priority traffic in order to take account for the high priority traffic (in the work of chapter 6 non pre-emptive PQ was used). To consider more than 2 priority level the extra traffic of the other priorities must be combined by convolution in calculating the unfinished work of the waiting probabilities of the low priority traffic.

- In the work of chapter 7 uniform BG traffic is used to represent the ambient load on the end-to-end network path. This was done in order to test the E_TA-FIFO and E_TA-PQ over an end-to-end network path. However, in a real network, the BG traffic will not be uniform, and might vary depending on the network path. Hence for further work, varied BG traffic load should be applied to better represent real networks.

Appendix 1

A1.1 Priority in Queuing Systems

A queuing discipline decides which packet should be served next in a queue and according to [Klei76] this depends on any or all of the following:

i) Some measure to the relative arrival times for those packets in the queue;
ii) Some measure (exact value, estimate, pdf) of the service time required or the service so far received;
iii) Or some function of group membership.

Queuing disciplines of type i) which depend upon arrival time include First-In-First-Out (FIFO), First-In-Last-Out (FILO), and random order of service. Queuing disciplines in ii) discriminate packets based on service time, and include: shortest-job-first (SJF), longest-job-first (LJF) or similar rules based on averages. Queuing disciplines of type iii) are usually referred to as priority queuing disciplines.

Static priority defines that the priority of a packet remains constant as it travels through the network and can be divided into two types: pre-emptive and non pre-emptive. Pre-emptive scheduling interrupts the service of the lower priority packet and non pre-emptive does not interrupt. The former type is further divided into two categories which are pre-emptive resume (service of the lower priority packet will resume where it left off) and pre-emptive restart (where a packet that is interrupted must start the service activity from the beginning again when the server next become available). Priority queuing is diagrammatically represented in Figure A.1

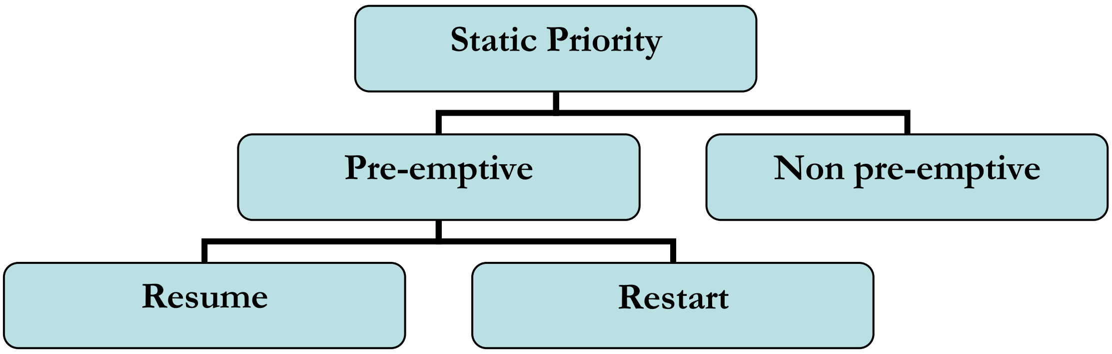


Figure A1.1:Types of Priorities

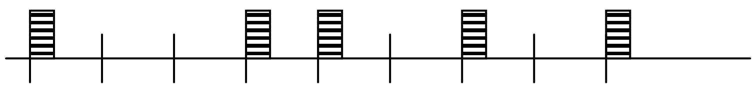
A p p e n d i x   2

A1.1 Sampling Techniques in Active Measurement

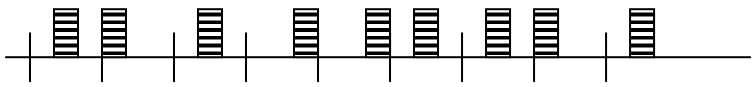
Simple Random - generates test packets randomly during the simulation run. The time interval between the test packets may have an exponential distribution in order to minimize correlation between the samples. Figure B1a illustrates an example of the generated test packet sequence in Simple Random.

Stratified Random – it splits the population into N sub populations which do not over lap and covers the whole original population. Test packets are generated randomly during each sub population. Figure B1b. illustrates an example of the generated test packet sequence in Stratified Random.

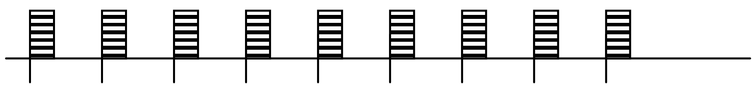
Systematic – generate a test packet at exactly intervals of *t* time. Figure B1c. illustrates an example of the generated test packet sequence using Systematic sampling.



a) Simple Random



b) Stratified Random



c) Systematic Random

Figure A2.1: Sampling method for active probes

A2.2 The Effect Of The Probes On The Queuing Behaviour Of The Buffer

This Appendix reports experiments that have been performed to observe the effect of the probes queuing behaviour in the buffer. In this experiment the conventional models has parameters values as follows, Doff=10, Don=5, N=10, R=1 and C=5. An equivalent E_TA-FIFO was derived having $\overline{ER}$=4.0704, $Toff_2$=1.7200 and C=5. The queuing behaviour in the buffer of simulation models that have probes measurement packets is compared to one without probes. This is done in order to verify that the collected results of the simulations are significantly not affected by the probes and Figure A2.2 shows that the queuing behaviour of the buffer remains the same. The queue state probability is defined as:

Q1 (k) = Pr (k packets in the queue at packet arrival instant)

and

Q2 (k) = Pr (k packets in the queue at ER batch arrival instant)

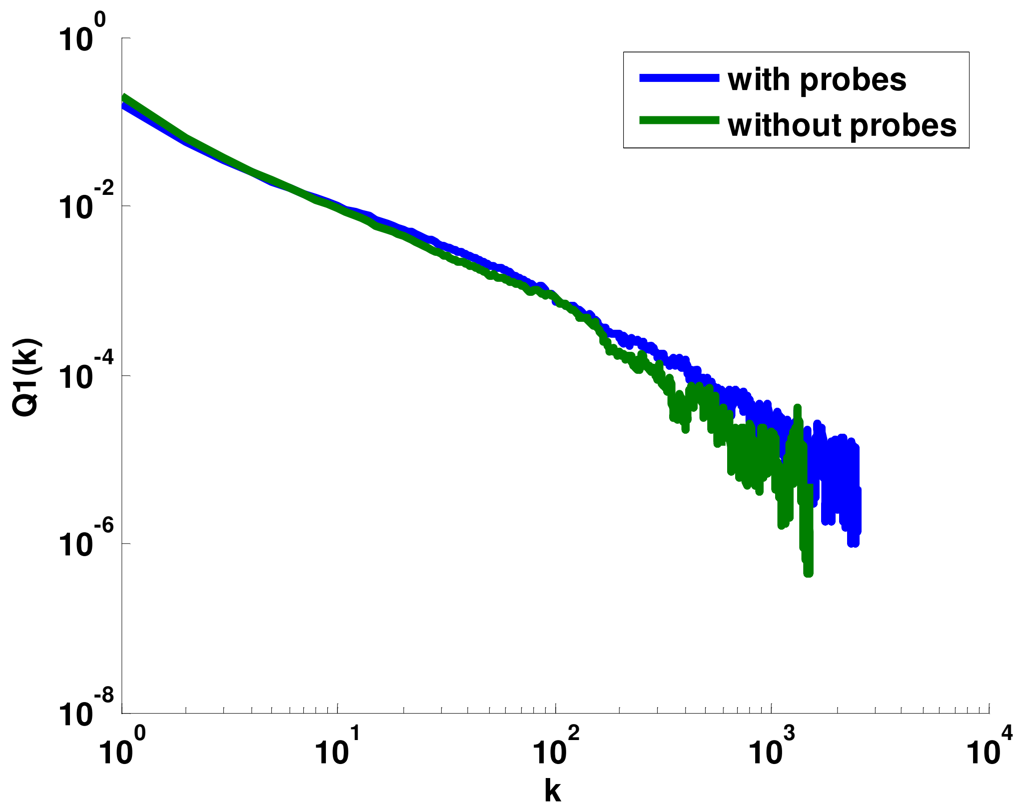


Figure A2.2: The queuing behaviour in the buffer of the Conventional model

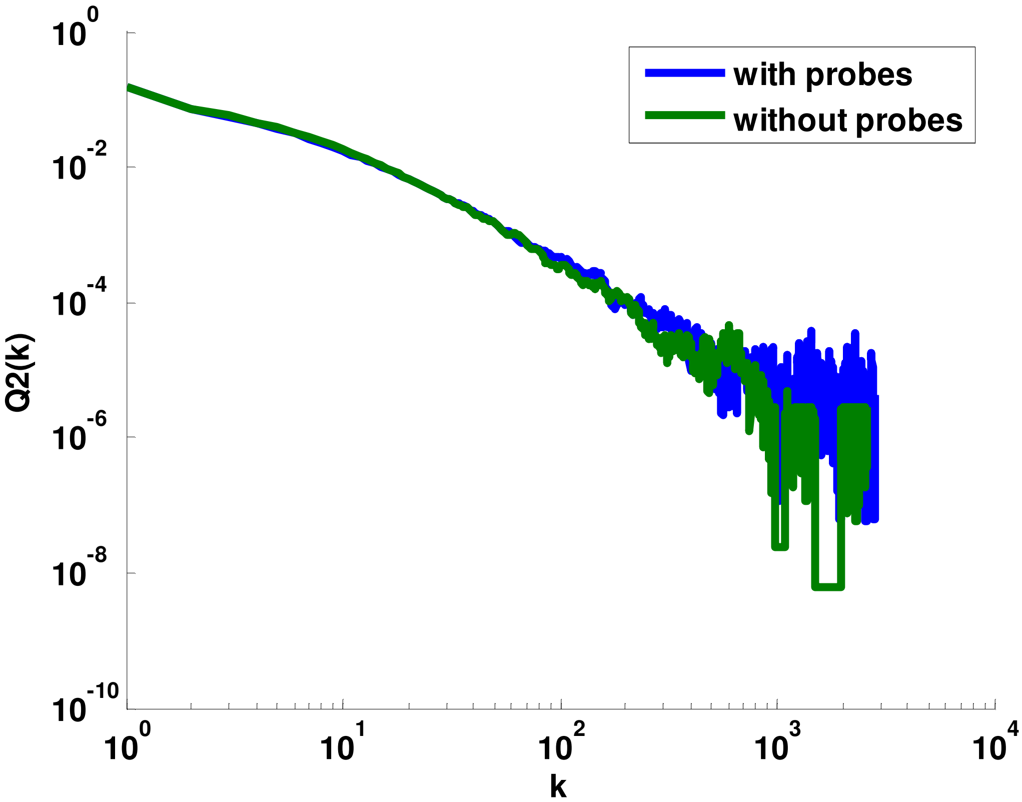


Figure A2.2: The queuing behaviour in the buffer of the E_TA model

Appendix 3

A3.1 ITU-T Recommendation Y.1541 QoS Signalling Requirement

Table A3

Class	Mean delay Upper bound	Delay Variance	Loss Ratio	Application Examples
0	100ms	<50ms	$<10^{-3}$	VoIP, video teleconference
1	400ms	<50ms	$<10^{-3}$	VoIP, video teleconference
2	100ms	unspecified	$<10^{-3}$	signalling
3	400ms	unspecified	$<10^{-3}$	signalling
4	1s	unspecified	$<10^{-3}$	Short transactions, bulk data, video streaming
5	unspecified	unspecified	unspecified	Default IP networks

Appendix 4

A4.1 The Geo/Pareto/ 1 Queue [Pitt00]

The Geo/Pareto/ 1 queuing analysis is used for a queue model that has batches of packets arriving at random i.e. as a Bernoulli process, and the number of packets in a batch is Pareto distributed. The Bernoulli process has a basic time unit, and a probability *q* which is the probability a batch arrives during that timeslot.

The probability that there are *k* arrivals in any timeslot is denoted by $a(k)$, where,

$$\begin{aligned} a(0) &= 1 - q \\ a(1) &= q \cdot b(1) \\ a(2) &= q \cdot b(2) \\ &\vdots \\ a(k) &= q \cdot b(k) \end{aligned} \qquad \text{(A4.1)}$$

where *b(k)* is the probability that an arriving batch contains k packets. To compute the discrete version of the Pareto distribution (as the Pareto distribution is continuous), the cumulative form is used:

$$F(x) = 1 - \left(\frac{1}{x}\right)^{\alpha} \qquad \text{(A4.2)}$$

In order to calculate *b(k),* the interval of [k-0.5, k+0.5] is used on the continuous distribution and we have:

$$b(k) = F(x+0.5) - (x-0.5) = \left(\frac{1}{x-0.5}\right)^{\alpha} - \left(\frac{1}{x+0.5}\right)^{\alpha} \qquad \text{(A4.3)}$$

Since $F(k)$ is a continuous distribution; for $F(1)=0$, i.e. the probability that an arriving batch contains less than 1 packet is zero, since in this system a batch has to contain at least 1 packet. The probability that a batch size is of one packet is given by

$$b(1) = F(1.5) - F(1) = 1 - \left(\frac{1}{1.5}\right)^{\alpha} \tag{A4.4}$$

Therefore, *b(k)* is the conditional probability distribution for the number of packets arriving in a timeslot, given that there is a batch. *a(k)* is the unconditional probability distribution for the number of packets arriving in a timeslot.

Author's Publications

1. ARIFFIN S.H.S, SCHORMANS J. A., 'Efficient Accelerated Simulation Technique for Packet Switched Networks: A Buffer with Two Priority Inputs', *IEEE International Conference on Communication*, ICC04, Paris, June 2004.
2. ARIFFIN S.H.S, SCHORMANS J. A., 'Accelerated Simulation Method Involving Markovian And Self-Similar Traffic Sources With Non-FIFO Scheduler', *UKSIM04*, Oxford, March 2004.
3. ARIFFIN S.H.S, SCHORMANS J. A., 'Accelerated Simulation Method for Power-law Traffic and Non-FIFO Scheduling', *London Comms Symposium, LCS'03*, London, August 2003.
4. ARIFFIN S.H.S, SCHORMANS J. A., 'ON/OFF Pareto Distributed Source At Burst Level: A Study Of Fast Simulation Methods For Packet Network', *PGNET2002*, Liverpool, June 2002.

REFERENCES

[Adam] ADAMIC L.A., 'Zipf, Power-Laws and Pareto- a Ranking Tutorial', Xerox Palo Alto Research Centre, Palo Alto, CA, USA, (or http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html)

[Anic82] ANICK D., MITRA D. and SONDHI M. M., 'Stochastic Theory of a Data-Handling System with Multiple Sources', *The Bell System Technical Journal,* Oct., 1982, **61**, (8).

[Bera92] BERAN J., 'Statistical Method For Data with Long-Range Dependence', 1992, Statistical Science, 7 (4), pp.404-427.

[Bera95] BERAN J., SHERMAN R, TAQQU M. S. and WILLINGER W., "Long-Range Dependence in Variable-Bit-Rate Video Traffic", *IEEE Transactions on Communications*, **43**, (2/3/4), February/March/April 1995.

[Bona01] BONALD T., PROUTIERE A., and ROBERTS J. W., 'Statistical Performance Guarantees for Streaming Flows Using Expedited Forwarding', IEEE INFOCOM, 2001, 2, pp. 1104- 1112.

[Buch01] BUCHLI M., DE VLEESCHAUWER D., JANSSEN J., VAN MOFFAERT A. and PETIT G.H., 'Resources Allocation and Management in DiffServ Networks for IP Telephony', *NOSSDAV 01'*, pp. 33-39.

[Carp02] CARPENTER B. E. and NICHOLS K. 'Differentiated Services in the Internet', *Proceedings of the IEEE*, 2002, **90**, (9), pp 1479-1494.

[Cisc13] 'Using Cisco Service Assurance Agent and Internetwork Performance Monitor to Manage Quality of Service in Voice over IP Networks (IP Application Services', Document ID: 13938, www.cisco.com .

[Cisc49] 'Quality of Service Networking', Cisco white papers, www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.pdf

[Cisc79] 'Voice Over IP-Per Call Bandwidth Consumption', Document ID: 7934, www.cisco.com

[Claf93] CLAFFY K. C., POLYZOS G.C. and BRAUN H., 'Applications, Technologies, Architecture and Protocols for Computer Communication', *ACM SIGCOMM* 1993, 23, (4), pp. 194-203.

[Cobh54] COBHAM A., 'Priority Assignment in Waiting Line Problem', *Operations Research*, 1954, **2**, pp. 70-76.

[Cox84] D. R. Cox, 'Long-range Dependence: A review,' *Statistics: An Appraisal*, H, A, David and H, T. David (Eds.). Ames, Iowa: Iowa State University Press, 1984, pp. 55-74.

[Crov97] CROVELLA, M.E. and LIPSKY L., 'Lost-Lasting Transient Condition in Simulations with Heavy Tailed Workloads', *WSC `97*, pp 1005-1012.

[Erra00] ERRAMILLI A., NARAYAN O. and NEIDHARDT A., 'Performance Impacts of Multi-Scaling in Wide Area TCP/IP Traffic', 2000, INFOCOM.

[Floy95] FLOYD S. and JACOBSON V., 'Link-sharing and Resource Management Models for Packet Networks', IEEE/ACM Transactions on Networking, August 1995, **3**, (4), pp.365-386.

[Fuji03] FUJIMOTO R.J., 'Distributed Simulation Systems', *Winter Simulation Conference*, 2003, **1**, pp 124-134.

[Garr94] GARRET, G. and WILLINGER, W., 'Analysis Modelling and Generation of Self Similar VBR Video Traffic'*, ACM SIGCOMM*, 1994, **24**, (4), pp 269-280.

[Garv98] GARVELS M. J. J. and KROESE D. P., 'A Comparison of RESTART Implementations', *Winter Simulation Conference*, 1998, **1**, pp. 601-608

[Glass96] GLASSERMAN P., HEIDELBERGER P. and SHAHABUDDIN P, 'Splitting for Rare Event Simulation: Analysis of Simple Cases', *Winter Simulation Conference*, 1996, pp. 302-308.

[Glass98] GLASSERMAN P., HEIDELBERGER P. and SHAHABUDDIN P., 'A Large Deviations Perspective on the Efficiency of Multilevel Splitting', *IEEE Transactions on Automatic Contro*l, **43**, (12), December 1998, pp. 1666-1679.

[Grib98] GRIBBLE S., MANKU G., ROSELLI D., BREWER E., GIBBON T. and MILLER E., 'Self Similarity in File Systems', *ACM SIGMETRICS '98*, 1998, **26**, (1), pp. 141-140.

[Gure00] GUREWITZ O., SIDI M. AND CIDON I, 'The Ballot Theorem Strikes Again: Packet Loss Process Distribution', *IEEE Transaction, on Information Theory*, 2000, 46, (7), pp.

[Hamm64] HAMMERSLEY J. M. and HANDSCOMB D. C., '*Monte Carlo Methods'* Methuen, London, 1964.

[Heid95] HEIDELBERGER P., 'Fast Simulation of Rare Events in Queuing and Reliability Models', *ACM Transactions on Modeling and Computer Simulation*, **5**, (1), 1995, pp. 43-85.

[Hein99] HEINANEN J., BAKER F. WEISS W. and WROCLAWSKI J., 'Assured Forwarding PHB Group'*, IETF RFC 2597*, Jun 1999, www.ietf.org/rfc.html

[Jaco99] JACOBSON V., NICHOLS K. and PODURI K., 'An Expedited Forwarding PHB Group' *IETF RFC 2598*, Jun 1999, www.ietf.org/rfc.html

[Jain86] JAIN R. and ROUTHIER R. A., 'Packet Trains- Measurements and a New Model for Computer Network Traffic', *IEEE JSAC*, September 1986, **4**, (6), pp. 986-995.

[Kahn51] KAHN H. and HARRIS T.E., "*Estimation of Particle Transmission by Random Sampling*", National Bureau of Standards Applied Mathematics Series.

[Kesh97] KESHAV S., '*An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephony Network*', 1997, Addison-Wesley.

[Klei76] KLEINROCK L., '*Queueing Systems*: Vol. 1', 1976.

[Law00] LAW A. M. and KELTON W. D., '*Simulation Modelling and Analysis*', 2000, McGraw-Hill.

[Lela86] LELAND W. and OTT T., 'Load Balancing Heuristics and Process Behaviour', *ACM SIGMETRIC*, 1986, pp. 54-69.

[Lela93] LELAND W., TAQQU M., WILLINGER W. and WILSON D.: 'On the Self-Similar Nature of Ethernet Traffic', *ACM SIGCOMM Computer Review,* 1993, **23**, (4), pp.183-193.

[Lela94] LELAND W., TAQQU M., WILLINGER W. and WILSON D.: 'On the Self-Similar Nature of Ethernet Traffic (Extended version)', *IEEE/ ACM Transaction on Networking*, 1994, **2**, (1), pp.1-15.

[Liu01] LIU B., FIGUEIREDO D. R., GUO Y. KUROSE J. and TOWSLEY D., 'A Study of Networks Simulation Efficiency: Fluid Simulation vs. Packet Level Simulation'*, IEEE INFOCOM 2001*, **3**, pp 1244-1253.

[Lope00] LOPEZ-ARDAO J.C., LOPEZ-GARCIA C., SUAREZ-GONZALEZ A., FERDANDEZ-VEIGE M. and RODRIGUEZ-RUBIO R., 'On the Use of Self-Similar Processes in Network Simulation', 2000, *ACM TOMACS*, **10**, (2).

[Ma01] MA A. H. I. and SCHORMANS J. A., 'A Fast Simulation Method for Modelling IP Networks', *17th UK Teletraffic Symposium on Networks: A Service Centre Approach*, May 2001.

[Ma02a] MA A. H. I. and SCHORMANS J. A., 'Accelerated Simulation Modelling Of Power-Law Traffic via Aggregation', *ICT Beijing, June* 2002.

[Ma02b] MA A. H. I. and SCHORMANS J. A., 'Efficient Simulation Modelling for Long Range Dependent Traffic', *Fourth Workshop on Rare Event Simulation and Related Combinatorial Optimization Problems (RESIM / COP'02)*, 2002.

[Ma02c] MA A. H. I. and SCHORMANS J. A., 'Hybrid Technique for Analysis of Multiplexed Power-Law Traffic in Broadband Networks', *IEE Electronic Letters*, 2002, **38**, (6), pp. 295-297.

[Ma03a] MA A., SCHORMANS J. and CUTHBERT L., 'Aggregation Technique for Networks with Power Law Traffic and Application to Accelerated Simulation', *IEE Proc. Comm.*, June 2003, **150**, (3), pp. 177-183.

[Ma03b] MA, A. H. I.: '*Accelerated Simulation of Power-Law traffic in Packets Networks*', PhD Thesis, Queen Mary University of London, 2003.

[Ma04] MA A., SCHORMANS J, 'Fast, Stable Simulation of Power-law Packet traffic Using Concatenated Acceleration Technique', *IEE Comm. Proc.*, 2004, **152**, (4), pp. 420-426.

[Mand65] MANDELBROT B., 'Self-Similar Error Clusters in Communication System and the Concept of Conditional Stationary', 1965, *IEEE Trans. Comm. Technology,* **43**, pp 71-0.

[Mand69] MANDELBROT B., 'Long-run Linearity, Locally Gaussian Processes, H-Spectra and Infinite Variances', 1969 Int. Economic Rev., 10, pp. 82-113.

[Mond01] MONDRAGON R. J., ARROWSMITH D.K. AND PITTS J.M., 'Chaotic Maps for Traffic Modelling and Queuing Performance Analysis', Performance Evaluation, 2001, **43**, pp. 223-240.

[Neid98] NEIDHARDT A. L and WANG J. L., 'The Concept of Relevant Time Scales and Its Application to Queuing Analysis of Self-Similar Traffic (or Is Hurst Naughty or Nice?)' *ACM SIGMETRICS Performance Evaluation Review Proceedings*, 1998, **26**, (1), pp. 222-232.

[Nich98] NICHOLAS K., BLAKE S., BAKER K., and BLACK D., 'Definition of The Differentiated Service Field (DS Field) in The IP4 and IP6 Header', *IETF RFC 2474,* December 1998, www.ietf.org/rfc.html

[Paru97] PARULEKAR M. and MAKOWSKI A. M., 'Tail Probabilities for M/G/∞ input Processes (i): Preliminary Asymptotic', 1997, Queuing Systems, **27**, pp. 271-295.

[Pasz02] PASZTOR A. and VEITCH D., 'Active Probing Using Packet Quartets', ACM, 2002, pp. 293-304.

[Pawl02] PAWLIKOWSKI K., JOSHUA JEONG H. –D. and RUTH LEE J.S., 'On Credibility of Simulation Studies on Telecommunications Networks', *IEEE Comm. Magazine*, 2002, **40**, (1), pg 132- 139.

[Paxs95] PAXSON V. and FLOYD S.: 'Wide Area Traffic: The Failure of Poisson Modelling', *IEEE/ACM Transactions on Networking*, 1995, **3**, (3), pp.226-244.

[Paxs99] PAXSON V., 'End-to-End Internet Packet Dynamics', *IEEE/ ACM, Trans. On Networking*, 1999, **7** (3), pp. 277-291.

[Pitt00] PITTS J. M. and SCHORMANS J. A.: 'Introductions *to IP and ATM Design and Performance'*, John Wiley and Sons, Chichester, 2000.

[Pitt95] PITTS J. M., 'Cell-Rate Modelling for Accelerated Simulation of ATM at the Burst Level', *IEE Comm.*, Dec., 1995, **142**, (6), pp. 379-385.

[Quan03] QUAN ZHI and JONG-MOON CHUNG, 'Queue Length Analysis of Non-Preemptive DiffServ Networks', Int. Journal of Electronics and Communications (AUE), 2003, 57, (5), pp. 338-340.

[Roug05] ROUGHAN M., 'Fundamental Bounds on the Accuracy of Network Performance Measurements', *ACM SIGMETRICS*, 2005, **33**, (1), pp. 253-264.

[Samu99] SAMUEL L. G., '*The application of non linear Dynamics to Tele traffic Modelling'*, PhD. Thesis, 1999, Queen Mary , University of London.

[Scho00a] 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', *IEE Electronic Letters*, March 2000, **36**, (6), pp. 523-524.

[Scho00b] SCHORMANS J. A., PITTS J., WILLIAMS K. and CUTHBERT L. G. 'Equivalent Capacity for ON/OFF Sources in ATM', *IEE Electronics. Letters*, 2000, **30**, (21), pp. 1740-1741.

[Scho01] SCHORMANS J. A., LIU E., CUTHBERT L. G. and PITTS J. M., 'Hybrid Technique for the Accelerated Simulation of ATM Networks and Networks Elements', *ACM Transaction Modelling and Computer Simulation (TOMACS)*, April 2001, **11**, (2), pp. 182-205.

[Scho03] SCHORMANS J., LIU E., STEWART R. and CUTHBERT L., **'**Analytical Technique for Accelerating the Simulation of Packet *Networks' IEE Proc. Comm.*, Volume **150**, (5), 2003, pp. 341-346.

[Scho90a] SCHORMANS J. A.: '*Discrete Time Priority Queues in Telecommunication Networks'*, PhD Thesis, Queen Mary University of London, 1990.

[Scho90b] SCHORMANS J. A. and PITTS J. M.: 'Analysis of Telecommunications Switch Model (GEO/D/1) with Time Priorities', *Electronics Letters*, 1990, **26**, (5), pp.325-326.

[Scho92] SCHORMANS J. A., SCHARF E. M. and PITTS J. M.: 'Prioritized Geo/D/1 telecommunication Switch Model', *Electronics Letters*, 1992, **28**, (6), pp.597-598.

[Scho93] SCHORMANS J. A, SCHARF E. M., PITTS J. M.: 'Waiting Time Probabilities in a statistical multiplexer with priorities', *Communications, Speech and Vision, IEE Proceedings I*, 1993, **140**, (4), pp.301-307.

[Scho94] SCHORMANS J., PITTS J. and CUTHBERT L.: 'Exact Fluid-Flow Analysis of Single on/off Source Feeding an ATM buffer', *IEE Electronics Letters,* 1994, **30**, (14), pp.1116-1117.

[Scho98] SCHORMANS J. A. and PITTS J. 'Decay Rate (ER) Modelling of G/D/1 queue, and Results for ATM Telecommunications', *IEE Electronics. Letters*, 1998, **34**, (10), pp. 943-945.

[Schr96] SCHREIBER F. and GORG C., 'The RESTART/LRE Method for Rare Event Simulation', *Winter Simulation Conference* 1996, pp. 390-397.

[Schw01] SCHWEFEL H. P., 'Behaviour of TCP-like Elastic Traffic at a Buffered Bottleneck Router', *IEEE INFOCOM* 2001, **3**, pp. 1698-1705.

[Stal98] STALLINGS W., '*High Speed Networks TCP/IP and ATM Design Principles'*, Prentice-Hall Inc, 1998.

[Taqq97] TAQQU M. S., WILLINGER M. and SHERMAN R., 'Proof of a Fundamental Result in Self Similar Traffic Modelling', *ACM SIGCOM Computer Comm. Review*, 1997, 27, (2), pp. 5-23.

[Town98] TOWNSEND J. K., HARASTZI Z., FREEBERSYSER J. A. and DEVETSIKIOTIS M., 'Simulation of Rare Events in Communication Networks', *IEEE Comm.*, August, 1998, **36**, (8), pp. 36-41.

[Vill94] VILLEN-ALTAMIRANO M. and VILLEN-ALTAMIRANO J., "RESTART: A Straight Forward Method for Fast Simulation of Rare Events", *Winter Simulation Conference*, 1994, pp.282-289.

[Will95a] WILLINGER W., TAQQU, M. S., LELAND, W. E., and WILSON, D. E., 'Self Similarity in High Speed Packet Traffic: Analysis and Modelling of Ethernet Traffic Measurements', 1995, *Statistical Science*, **10**, (1), pp 67-85.

[Will95b] WILLINGER W., TAQQU M. SHERMAN R. and WILSON D. V., 'Self Similarity Through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level', *Computer Communication Reviews*, 1995, **25**, pp. 100-113

[Will97] WILLINGGER 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 Transactions on Networking*, 1997, **5**, (1), pp. 71-86.