Planning Simulation Run Length in Packet Queues in Communications Networks

By

Ling Xu

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Supervised by Dr. John A. Schormans

Department of Electronic Engineering and Computer Science,
Queen Mary, University of London

May 13, 2013

To my family

Declaration

I hereby certify that the work presented in this thesis is my own work and that to the best of my knowledge of the work is original, except for where referenced to corresponding authors.

Ling Xu

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. John Schormans, who has given me all the support, assistance and encouragement throughout my Ph.D. study. John spend huge amount of time to discuss ideas, carefully and in time written checking, as well as offering positive support when I feel frustrated. I would also like to thank my second supervisor, Athen Ma, and independent accessor, Karen Shoop, for their endless suggestions and discussions along four years of my study.

I will also thanks to Chinese Scholarship Council (CSC), funding me for three whole year study. Without CSC, I can not have the opportunity to study in this great University.

Moreover, my thanks will give to the colleagues and staff at Queen Mary, which has been a friendly and pleasant place to work. My big thanks to Ho Ko Huen, who gave me quick help for the systems and software when I needed.

Finally, my thanks will go to my families and friends. Thanks for the endless supports from my mother and my father, both emotionally and financially. Without you, I can not finish this huge work. Also my thanks will be given to all of my friends in the department, to those who have heated discussion with me.

Abstract

Simulation is a technique of growing importance and is becoming an indispensable tool applied in various academic industries, including packet networks.

Simulation provides an alternative research approach to implementing a real environment, owing to its features of scalability, flexibility and ease of setup. However, simulating large-scale networks can be very time and resource consuming. It can take several days to run one long simulation experiment, which may be expensive or even unaffordable. Therefore, planning simulation is important.

This research proposes to plan simulation run length through predicting the required shortest run length that approximates steady-state, in the form of mathematical and logical expressions, i.e. building an analytical model. Previously related research always focused on classical models, such as the M/M/1 queue model, M/G/1 queue model, and so on. This research expands the research base to include a packet multiplexing model of homogenous sources which is widely accepted and used. This thesis investigates different traffic types (Markovian/Pareto) and different QoS parameter (delay/losses), as well as applying them to end-to-end networks.

These scenarios are analysed and expressed, in terms of different desired precision level. Final results show that run length time is well predicted using the developed analytical model, which can be a guide for simulation planning in packet networks of the present and the future. This can be of great significance for performance evaluation studies.

Contents

List of Figures

List of Tables

Glossary

c.d.f.	cumulative density function
CI	confidence interval
FIFO	first-in first out
FT/BT	foreground traffic/ background traffic
i.i.d.	independent identically distributed
LRD	Long range dependency
Mbps	Mega bit per second
NOF	Non-OverFlow
OvFl	OverFlow
pdf/p.d.f./ PDF	probability density function
PLP	Packet Loss Probability
PMM	Packet Multiplexing Model
pps	packet per second
PSBS	Packet-scale/ burst-scale
QoS	Quality of Service
RBM	Reflected Brownian Motion
SCV	Squared coefficient of variation
SRLASS	shortest run length that approximates steady-state

Mathematical Expression

ΔT	mathematical expression for shortest run length that approximates steady-state
ε_a	Absolute width/absolute error
ε_r	Relative width/relative error
ε	Preset relative error/percentage of accuracy
η	the burst scale decay rate
μ	Parameter for exponential distribution
μ_p	Expected number of packets lost for each overflow periods
λ_s	measurement rate
σ_X^2	uncorrelated variance
A_p	Overall mean load for packet multiplexing model
c_2	mathematical expression for SCV
C	Service rate for packet multiplexing model
h	Packet generation rates in packet multiplexing model
N_{μ_p}	Required number of cycles for μ_p to reach steady state
N_a	required sample size for specified absolute width ε_a
N_{cycle}	Required number of cycles to reach steady state
$N_{p/cycle}$	Mean number of sending packets in one cycle
N_r	required sample size for specified relative width ε_r

N_{Tcycle}	Required number of cycles for T_{cycle} to reach steady state
p	Parameter for geometric distribution
$R(i)$	auto-covariance for discrete-time process
R_{off}	Overall mean rate of Markov packet multiplexing model in aggregate OFF state (overall load is less than service rate)
R_{on}	Overall mean rate of Markov packet multiplexing model in aggregate ON state (overall load is less than service rate)
$R(u)$	auto-covariance for continuous-time process
s_N^2	correlated variance for discrete-time process
s_t^2	correlated variance for continuous-time process
$T(on)$	Mean duration of packet multiplexing model in aggregate ON state (overall load exceeds service rate)
$T(off)$	Mean duration of packet multiplexing model in aggregate OFF state (overall load is less than service rate)
T_{on}	Mean duration of each packet source in ON state (sending packets)
T_{off}	Mean duration of each packet source in OFF state (idle, no packets sending)
T_{ovfl}	Mean duration of Markov packet multiplexing model in overflow (OvFl) state
T_{nof}	Mean duration of Markov packet multiplexing model in non-overflow (NOF) state
T_{cycle}	Mean of each cycle duration of one overflow/non-overflow cycle for packet multiplexing model
$Var(\mu_p)$	Variance of μ_p
$Var(T_{cycle})$	Variance of T_{cycle}
$\hat{X}_N$	Estimator for a discrete-time process
$\{X_i\}$	sample population for a discrete-time process

Chapter 1

Introduction

1.1 Research Motivation

Simulation is a technique of great importance in many fields, both theoretical and applied [31] [16]. It is becoming an indispensable tool used in various knowledge industries, such as weather forecasting, and manufacturing, as well as packet networking.

Simulation provides an alternative research approach to implementing a real environment, since a real testbed requires a large amount of investment on equipment [2]. Moreover, an inexpensive and flexible real test-bed for networks cannot get an equivalent scale and complexity to a real network [97]. Compared to a test-bed, simulation is much more economical and flexible. A network can be modelled at any scale for any research requirement.

Recently, the focus has been on large-scale networks [95][86][14], and simulating large-scale networks can be very time and resource consuming. It can take several days [53] to run one long replication of simulation experiment, which may be unaffordable. Large-scale network often have complex topology, combine various traffic pattern, multiple protocols, and applications, etc. Also, dimensioning and of large-scale network can be difficult, particularly in the presence of disparate traffic mixes. Under this circumstance, traffic partitioning [95] is of great importance, since this will improve corresponding network performance and QoS. Also, the new age of 4G technology is coming, which provides greater bandwidth, higher data rates, efficient spectrum use, etc [39]. Dimensioning 4G mobile will be even more challenging.

Because of all these factors, simulation to assist network dimensioning is more important than ever. Therefore, planning simulation is required. This is to "design the experiments, i.e., to determine what cases to consider, what statistical precision to aim for, and what experimental budget is appropriate, and whether to conduct the experiment at all" [89]. Moreover, when to stop the simulation is another problem involved in the simulation planning stages. Simulation stopped too late will result in a waste of time and computer resources with unnecessary high precision level achieved, while those stopped too early will lead to inaccurate results. In conclusion, all the issues mentioned above require simulation run length planning before the simulation is run.

1.1.1 Shortest Run Length Approximating Steady State

Network simulation is a numerical technique for conducting experiments on a digital computer [31], which involve statistical models that describe network behaviour, and finally outputs a series of results called 'metrics of interest' in our research. A simulation is implemented by setting up input parameters and running computer codes, until enough output metric is collected to achieve the desired precision level.

Any simulated output metric changes significantly over time[1] before it reaches steady state (detailed definition of steady state is in Section 2.1.1). For example, Figure 1.1 illustrates how a time series for an output metric changes over time. The blue line illustrates the collected output metric at the

[1] In Figure 1.1, The output metric looks like starting from a large value, rather than 0. This is judged by the naked eye. Actually, it starts from 0, and after a short simulation time, it grows largely since some events appear early, e.g. packet losses appear early when collecting PLP data. This phenomenon is normal, also in [20].

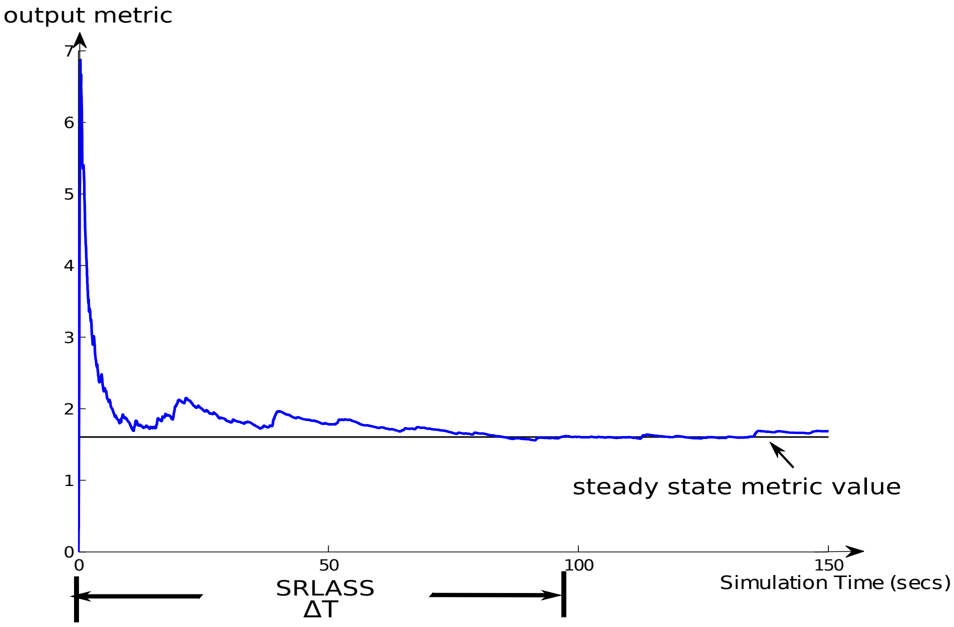


Figure 1.1: Illustration of SRLASS of steady state simulation

corresponding time instant. It is clear that the metric varies largely at the early stage of this replication (a single simulation run), and after a period, asymptotically converges to the so called steady state. This convergence period is defined as the shortest run length that approximates steady-state (SRLASS) in this thesis, represented as ΔT as illustrated in Figure 1.1. Planning simulation run length is to predict the required SRLASS, ΔT. As shown in Figure 1.1, the SRLASS is in the units of simulation time. This is different from wall clock time. Simulation time is used for the purpose of setting the run length for the simulation, while the wall clock time is the actual time consumed by one simulation run. This thesis will show the SRLASS results in the units of simulation time, and further maps this into wall clock time, with details given in Chapter 7.

Simulation run length planning can be studied by analysis and procedure design. The former always involves corresponding modelling of network behaviour, and approximates it as a stochastic process by a reflected brownian motion process [89], diffusion process [90], or a birth-and-death process [91]. Network behaviour is modelled by these processes through corresponding parameterisation and the SRLASS is further represented by mathematical or logical expressions. An alternative approach is to address this problem by designing procedures. Chen et al [13] designed a quasi-independent (QI) procedure, which increases the simulation run length until the number of samples satisfies the required precision level. Lada et al [43] designed a procedure called WASSP, focused on the M/M/1 waiting-time process, to determine a truncation point, i.e. the end of the warm-up period. The authors further compared the efficiency of three different procedures dealing with difficult test processes in [44].

1.2 Objectives of this research

This research proposes to plan simulation, i.e. to forecast the SRLASS, ΔT, using mathematical and logical expression in the early planning stages of the simulation, targeting queue models for packet buffering.

The objectives of this research are:

- To propose a more realistic queue model, packet multiplexing model (PMM), than classical queue models.
- To develop an analytical model for the SRLASS, ΔT, which will predict an accurate simulation run length before the simulation is run.
- To map the SRLASS into required wall clock time, which will give a guide how long the simulation will last, i.e. the real time.
- To focus on both Markovian traffic and Pareto traffic.
- To use UDP as a basic internet protocol.

- To use QoS parameters, packet loss probability (PLP) and network delay, as the targets of the simulation.
- Starting from a single access queue model, expand the research to an end-to-end network model.

1.3 Contribution of this thesis

Previous research in simulation planning is mainly focused on more general, classical queue models, e.g. the M/M/1 queue. This thesis expands the research into a more realistic queue model, a packet multiplexing model, which is a more realistic representation of packet queueing.

Besides, this research plans simulation SRLASS of PLP in Chapter 5. Previous work done by Whitt [89] concentrated on queue length or waiting time, but there is no research about PLP simulation planning.

Moreover, the research is expanded into an end-to-end network, which is a more complex network topology than a single buffer. Based on results for single access, the simulation planning in an end-to-end network is resolved.

Nowadays, some data traffic has been shown to be statistically self-similar [49], which is better modelled as a Pareto distribution, rather than a Markovian distribution (Voice traffic). However, the Pareto traffic is much more variable, which means it is harder (or impossible) for a simulation to reach steady state [20], as the Pareto has very large or even infinite variance. This makes simulation planning harder.

1.4 Thesis Outline

In chapter 2, previous work on simulation planning is reviewed. A standard statistical analysis is presented as a general analysis tool for simulation planning. The squared coefficient of variation (SCV) of the metric of interest (with more detailed explanation in Section 2.3.3) is proposed as a simulation planning estimator in this thesis. An introduction to run length indicators is discussed in detail.

Chapter 3 gives a brief introduction to simulation and the simulation models that will be used in this thesis. Traffic models are introduced, including both Markovian traffic and Pareto traffic. The main queue model that will be used in this thesis, the packet multiplexing model, is presented afterwards. The basic end-to-end network model is introduced in this chapter.

In chapter 4, simulation planning of mean delay for the packet multiplexing queue model with Markovian traffic is developed. Packet-scale/burst-scale (PSBS) characteristics are reviewed in chapter 4, and the formula for SRLASS for the mean delay is developed based on PSBS features. Results are shown for both single access and end-to-end networks. This result is then compared with previous work (by Ward Whitt [89]), which shows that the formula we developed is more accurate for run length prediction.

Chapter 5 further expands the research into the simulation planning of the PLP in the PPM queue model. PLP is an important QoS parameter. However, there is no previous work on planning simulation for PLP. The mathematical and logical expressions for SRLASS in this scenario is developed through finding the required sample size of Overflow/Non-Overflow (OvFl/NOF) cycles[2] for the PLP to reach steady state. Validation is achieved using Exponential bestfit. Results are shown with detailed discussions.

Chapter 6 investigates a more highly variable traffic type - Pareto traffic. The PLP is regarded as the parameter of interests in this chapter. Results show that the Pareto traffic is much more variable and requires much longer time for the simulation to reach the steady state.

[2] The PMM can be viewed into an aggregation OvFl/NOF two-state model (detailed introduction in Section 5.2). And one OvFl period, followed by an NOF period, is called one cycle. Number of cycles, or cycle number required for PLP to reach steady state is denoted as N_{cycle} in Section 5.3.

Chapter 7 A guideline of mapping from simulation run length (in the units of packet arrivals or simulation time) into a wall clock time is presented. Results show that in the simulation of the packet multiplexing model, number of packets simulated is the key parameter that will affect the wall clock time. Therefore, with some short experimental runs, real time consumed for longer replicates can be predicted.

Chapter 8 consists of a discussion and conclusion, as well as possible further work.

Chapter 2

Methodology

This Chapter will mainly review the methodology used in this thesis for planning simulation run length, which is a standard statistical analysis. A precision criteria - the standard coefficient of variation (SCV) is given as the estimator for SRLASS. Error, classified into absolute width and relative width, will also be discussed and this research mostly employs the latter. Finally, this chapter will review variance and bias, which are very important in statistical analysis.

2.1 Introduction

There are some important considerations before a simulation is designed and run. How to set the input parameters? When to stop the simulation? Will the output be accurate and reliable when the simulation is stopped? These issues need to be resolved when planning a simulation. The accuracy, precision and reliability of output results is defined for steady state simulations in this thesis.

As already discussed in Chapter 1, planning simulation run length is to predict the SRLASS. Therefore, the definition of steady state is very important.

2.1.1 Definition of Steady State

It is very important to get an accurate definition of steady state. The concept of steady state is used to describe the 'precision' and 'steadiness' of the simulation results.

As already introduced in Chapter 1, simulation uses models to describe network behaviour, reads in the input data, and finally outputs a series of results. The final result is usually obtained from the estimator of output data. Sample mean is a popular point estimator. However, sample mean is not ideal to define the 'steadiness', or 'steady state', as illustrated in Figure 2.1.

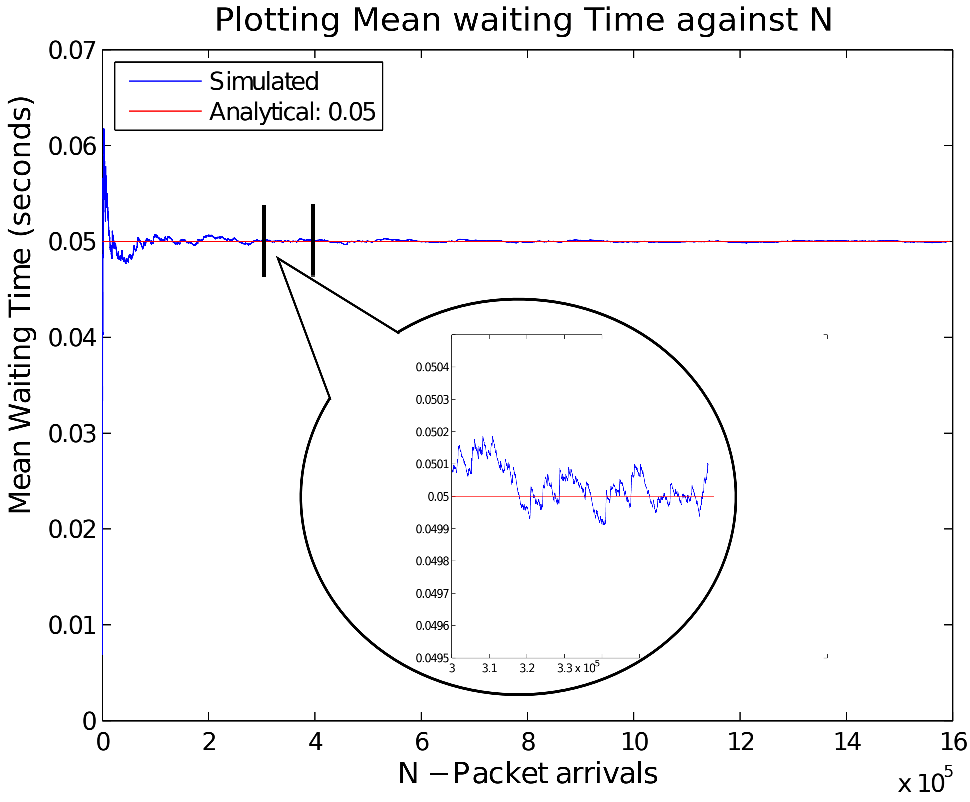


Figure 2.1: M/M/1 queue model - mean waiting time against packet arrivals

To generate the results shown in Figure 2.1, a simulation is run for an M/M/1 queue model, and the waiting time is collected as the output data for every packet arrival. The sample mean of each packet's waiting time is usually regarded as the estimated mean delay for the queue model. Figure 2.1 plots the mean waiting time against packet arrivals, and illustrates how the mean waiting time changes when more packets are generated.

As shown in Figure 2.1, the mean waiting time seems toe have already reached a state of 'steadiness', which is just varying slightly. However, when we zoom in, the data might still be changing significantly, as shown in the popping bubble. Therefore, whether or not the output data reaches the 'steadiness' (i.e. precision criterion) can not be judged by the naked-eye, or evaluated by a point estimator (e.g. the sample mean). Under this circumstance, steady state needs to be defined using an objective criterion.

In this thesis, such 'steadiness' is termed using 'steady state', and defined using a Confidence Interval (CI). The simulation is regarded to have reached steady state when the targeted CI has reached a desired smallness. CI is an interval estimator, which is composed of a sample mean with upper and lower bounds. Steady state, in this research, is defined as being when the output data's CI reach a preset target. A more formal and detailed definition of CI will be presented in Section 2.3.2.

2.2 Importance of Simulation Planning

In the early 1960's, Baran invented the concept of packet switching [7] [8] and Donald W. Davies researched similar ideas in 1967 [21]. Today, there is still a lot of work focusing on packet networks and related applications/protocols research [48] [10] [71] [96]. Networks possess features of large scale, and a high degree of complexity, and extend worldwide [53]. Research aiming at higher transmission speed, wider broadband and larger coverage normally requires a large amount of testing before being officially adopted. Testing in a real environment gives an equivalent performance, but requires a large amount of investment, which can be a barrier for the majority of researchers [2]. On the other hand, an inexpensive and flexible real test-bed for networks cannot get equivalent scale and complexity as a real network [97]. Therefore, simulation is used and plays a nontrivial role in the study of packet networks.

As introduced in Chapter 1, Simulation, compared to a real test-bed, is cheaper and it is easier to use it to build flexible and scalable network. However, it also has drawbacks, e.g. the time and computer resources consumed by one simulation can be very large, and the simulation may not reach steady state yet when stopped.

Therefore, planning simulation is of great importance. Planning simulation normally requires analysis against targeted scenario and metric of interest before the simulation is run.

2.2.1 Previous Work on Simulation Run Length Planning

There are two main methods in simulation run length planning research: analysis and procedure design. Since research on steady-state is a generic topic, the majority of previous papers are focused on classic queue models, and M/M/1 is one frequently used classical queue model, e.g. in [89] [90] [92] [91] [43] [44] [85] [13]. The M/M/1 is very widely used as it is both completely random, and yet simple to analyse, and therefore provides a well known model against which simulation run length planning techniques can be tested.

Analysis

In 1989, Ward Whitt first proposed to plan queuing simulations. He provided a formula for the required simulation run lengths in the early planning stages, and proposed to use this "to design the experiments, i.e., to determine what cases to consider, what statistical precision to aim for, and what experimental budget is appropriate, and whether to conduct the experiment at all"[89]. In [89], he researched the queue length long run behaviour of a GI/G/1 model, and approximated the stochastic process by a reflected Brownian motion process. He continued to approximate the model into a diffusion process [90] and a birth-and death process [91]. More recently, he examined simulation planning work in 2005 [92] again, proposing to use SCV as metric of interest (e.g. delay/ buffer length) as the general indicator for planning simulations.

Procedure design

An alternative approach is to address run length planning problem based on simulation methods. Ref. [13] designed a quasi-independent (QI) procedure, which increases the simulation run length until the number of samples is such that the estimator reaches the required precision level. Ref. [43] designed a procedure called WASSP, which focused on the M/M/1 waiting-time process, to determine a truncation point, i.e. the end of the SRLASS. The authors of [43] further compared the efficiency of three different procedures dealing with difficult test processes in [44]. In [85], the authors are also interested in the steady-state waiting time of the M/M/1 model. They argued that a sequential procedure is more efficient than fixed-interval designs[1]. They designed their procedure based on input/output behaviour, and use bootstrapping[2] to predict the variance. The variance is then used as the main estimator to find the required run length in [85].

[1] The sequential design procedure controls events in the simulation. It is well known that the sequential design procedure is more efficient than fixed-interval simulation.

[2] Bootstrapping is a computer-based technique used to estimate properties of an estimator (e.g. its variance) from an approximating distribution. One standard way is to use the empirical distribution as the approximating distribution. Simplicity is one main advantage of bootstrapping.

2.3 Statistical Analysis

There is no lack of research in modelling a simulation replication process into a statistical process, [89] [90] [91] which use existing mathematical models to solve related problems. In this section, a standard statistical analysis is introduced, and this will be further explained for how it can be applied to simulation planning research.

In this thesis, the discrete-time method is used for the following reasons: 1) NS2 is the simulator tool used, which is a discrete event-driven simulator. Therefore, all the data collected are in a discrete pattern; 2) PLP and delays are the output parameters of interest in this thesis, and they will be analysed also in a discrete way. The PLP will be collected by cycles[3], and delay will be collected for each individual packet arrival (The discrete-time waiting time is also used in 3.2 and 5.2 in [89]). The number of cycles and the number of packet arrivals are all discrete.

2.3.1 A Discrete-Time Process

Applied to a statistical analysis, the metric of interest of a simulation is usually regarded as a stochastic process, $\{X_i\}$ where $i = 1, ..., N$, with mean $\bar{X}$ and variance σ^2. Assumption of strict stationarity is usually made on the condition that $\sigma^2 < \infty$. Let $\bar{X}$ represent the true value of the metric of interest while $\hat{X}_N$ represent the estimator of the true mean, calculated by the sample mean from the simulation results, where

$$\hat{X}_N = \frac{1}{N}\sum_{i=1}^{N} X_i, \tag{2.1}$$

[3] The PMM can be viewed into an Overflow/ Non-Overflow cycles (with detailed introduction in Section 5.2). When it is in Overflow state, packets losses occur. Therefore, PLP is collected by cycles. Section 5.3 will also explain how PLP is collected by cycles.

Based on the Central Limits Theorem (CLT), the sample mean converges as

$$\sqrt{N}(\hat{X}_N - \bar{X}) \sim N(0, \sigma^2) \tag{2.2}$$

where the process is uncorrelated, and σ^2 is called the asymptotic variance of the process $\{X_i\}$. From Equation (2.2), we use the approximation

$$\hat{X}_N \approx N(\bar{X}, \frac{\sigma^2}{N}) \tag{2.3}$$

for sufficiently large N, which means that the sample mean is asymptotically Normally distributed with mean $\bar{X}$ and variance σ^2/N.

This approximation of Equation(2.3) is valid based on four assumptions:

- The distribution of the sample mean $\hat{X}_N$ is Normal
- The mean is $\bar{X}$ with no bias (detailed discussion about bias, see Section 2.3.4)
- The variance of the sample mean is approximated by $Var(\hat{X}_N) = \sigma^2/N$, which also provides a method to calculate the variance of the sample mean through the variance of the sample population.
- The run length is sufficiently large.

In conclusion, a simulation is run in order to estimate the stochastic metric of interest. The sample mean of interest is conventionally regarded as the estimator, to simulate the true value. For a stationary process, the sample mean will converge to the true mean, which is also the steady-state value. The sample size, N, is required to be large enough so that the sample mean could be used to represent the true mean, while Equation (2.3) to be a reasonable approximation. So, the required sample size, N, is also an important factor in this research.

2.3.2 Confidence Intervals (CIs)

Confidence intervals (CIs) are used in relevant research, e.g. simulation planning [84] [89][90] [91] and sample size analysis [29] [77]. In our research, CIs are applied in describing the 'steadiness', or the desired precision level.

The CI, as a kind of interval estimator of a sample population, is used to express the precision. Assume that N observations are independent and identically distributed (i.i.d.) , denoted as $X_1, X_2, \ldots, X_N$, with a finite mean $\bar{X}$ and a finite variance σ^2, where sample mean $\hat{X}_N$ is given by

$$\hat{X}_N = \frac{\sum_{i=1}^{N} X_i}{N} \tag{2.4}$$

and variance σ^2 given by

$$\sigma^2 = \frac{\sum_{i=1}^{N} (X_i - \hat{X}_N)^2}{N(N-1)} \tag{2.5}$$

A CI is a range between an upper bound $C1$ and a lower bound $C2$, where

$$Pr\{C1 \leq \mu \leq C2\} = 1 - \beta$$

so that the probability of the estimated mean lying within the range $(C1, C2)$ is $100(1-\beta)\%$. $(C1, C2)$ is called CI , where β represents the significance level, $100(1-\beta)\%$ the confidence level, and $1-\beta$ confidence coefficient. Usually, a CI is represented using a percentage, normally 90% or 95%. Thus, CIs are used to indicate the steadiness and precision of the sampled mean. How likely the interval is to contain the estimator is determined by the confidence level or confidence coefficient.

In order to find β, define a random variable z_n, where

$$z_n = \frac{\hat{X}_N - \bar{X}}{\sqrt{\sigma^2/N}} \tag{2.6}$$

Based on the CLT [11], if N is sufficiently large, the random variable z_n will usually have a standard Normal distribution with mean 0 and variance 1, regardless of the underlying distribution of X_i. Define $\phi(z)$ to be the distribution function of a standard normal random variable, given by

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-y^2/2} dy \tag{2.7}$$

Then, the sample mean $\hat{X}_N$ for sufficiently large samples is approximately Normally distributed with mean $\bar{X}$ and standard deviation $\sigma/\sqrt{N}$:

$$\hat{X_N} \sim N(\bar{X}, \sigma/\sqrt{N}) \tag{2.8}$$

Therefore, z_n follows

$$P(-z_{1-\beta/2} \leq \frac{\hat{X}_N - \bar{X}}{\sqrt{\sigma^2/N}} \leq z_{1-\beta/2}) = P(\hat{X}_N - z_{1-\beta/2}\frac{\sigma}{\sqrt{N}} \leq \bar{X} \leq \hat{X}_N + z_{1-\beta/2}\frac{\sigma}{\sqrt{N}}) = 1-\beta$$

Thus, for a sample population $X_1, X_2, \ldots, X_n$, given that n is sufficiently large, an approximated $(1-\beta)(100)\%$ confidence interval for $\bar{X}$ can be written as:

$$(\hat{X}_N - z_{1-\beta/2}\frac{\sigma}{\sqrt{N}}, \hat{X}_N + z_{1-\beta/2}\frac{\sigma}{\sqrt{N}})$$

By setting the confidence level, $z_{1-\beta/2}$ can be calculated. The confidence level can be set as required. Normally, the confidence level is 90% or 95%. In this research, a 95% confidence interval (where z=1.96) is generally used since it is widely used in [45] [28] [89] [43] [44].

Assume that the sample mean of an i.i.d. discrete-time process $\{X_i\}$ is $\hat{X}_N$ and its variance is σ^2. From Equation (2.3), a $(1-\beta)(100)\%$ CI for true value $\bar{X}$ is

$$(\hat{X}_N - z_{1-\beta/2}\frac{\sigma}{\sqrt{N}}, \hat{X}_N + z_{1-\beta/2}\frac{\sigma}{\sqrt{N}})$$

where

$$P(-z_{1-\beta/2} \leq N(0,1) \leq z_{1-\beta/2}) = 1 - \beta$$

Take Figure 2.2 for example, a CI is constructed from the critical value (which can be any metric of interest). The blue zone is the constructed CI with the upper limit bound and lower limit bound. The interval gives as an interval estimator which means that the probability that the critical value lies within the blue zone is 95%.

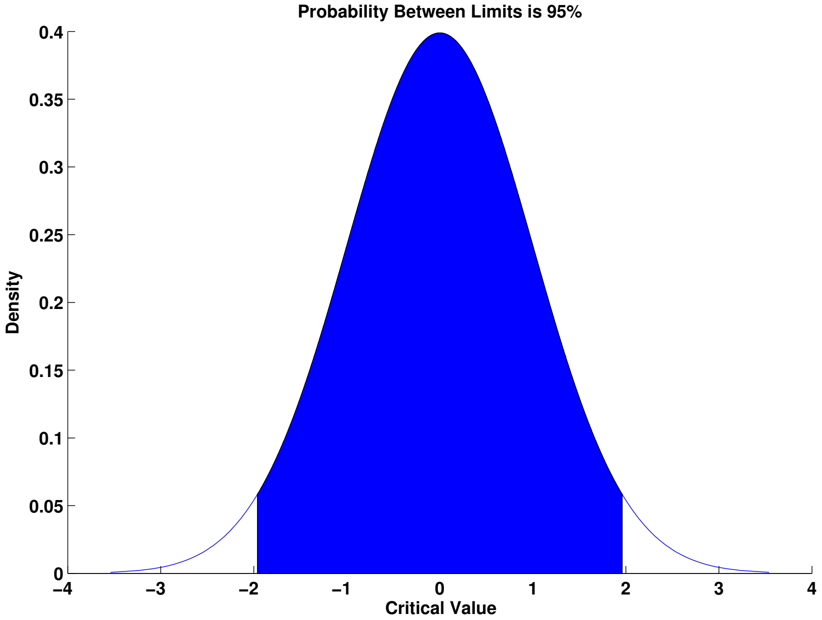


Figure 2.2: Illustration of Confidence Interval

In this research, the CI is used in an alternative way, when CI is used to describe the steadiness of the simulation output metric of interest. The simulation is regarded to reach the steady state when the targeted CI has reached a desired smallness (which is set according to the simulation requirement). The mathematical formula will show how it works in Section 2.3.3.

2.3.3 Squared Coefficient of Variation (SCV)

Absolute Error and Relative Error

We consider two kinds of error, the absolute error and relative error (also called absolute width and relative width for CIs). Relative error is defined as the ratio of the simulation standard error to the simulation estimator of the mean. Relative error is preferred to be a better practical measure of statistical precision for evaluating approximations [89] as it is independent of the measuring units, which can be chosen arbitrarily.

However, there often is a measuring unit that is naturally meaningful in an application, so that independence of the measuring unit is not always desirable. For example, the queue length could be in units within the range from units of 0 to 10^6 customers. Thus, if a mean buffer length is 0.01, then we might prefer to measure precision of an estimate by the absolute error instead of the relative error.

In conclusion, it is believed that relative error is usually a better measure of statistical precision, but not always. In this research, relative error is used as it is independent of the units and more suitable for planning simulation run lengths for the PMM model [89].

Continuing with the analysis in section 2.3.2, we define the absolute width, ε_a, of the confidence interval, given by

$$\varepsilon_a = 2z_{1-\beta/2} \cdot \frac{\sigma}{\sqrt{N}} \tag{2.9}$$

and the relative width, ε_r, of the confidence interval, given by

$$\varepsilon_r = 2z_{1-\beta/2} \cdot \frac{\sigma}{\bar{X}\sqrt{N}} \tag{2.10}$$

Thus, for specified absolute width ε_a and specified level of precision β, the required sample size correspondingly, N_a, is given by

$$N_a = \frac{4\sigma^2 z_{1-\beta/2}^2}{\varepsilon_a^2} \tag{2.11}$$

Applying the same to a specified relative width ε_r and specified level of precision β, the required sample size correspondingly, N_r is given by

$$N_r = \frac{4\sigma^2 z_{1-\beta/2}^2}{\varepsilon_r^2 (\bar{X})^2} = 4 \cdot \frac{z_{1-\beta/2}^2}{\varepsilon_r^2} \cdot \frac{\sigma^2}{\bar{X}^2} \tag{2.12}$$

Define squared coefficient of variation (SCV) as

$$c^2 = \frac{\sigma^2}{\bar{X}^2} \tag{2.13}$$

Therefore, the required sample size for relative width is finally given by

$$N_r = 4 \cdot \frac{z_{1-\beta/2}^2}{\varepsilon_r^2} \cdot c^2 \tag{2.14}$$

In the above equation, $z_{1-\beta/2}^2$ is the confidence parameter, usually set to be 1.96 with 95% confidence level. ε_r is the precision width, which is defined according to the precision requirements (In this thesis, set to 10%, 20% and 50% respectively). From the Equation (2.14), if the analytical model of SCV is known, then the required sample size N_r can be obtained. Therefore, the SCV is regarded as the standard simulation planning estimator in this thesis.

In this thesis, relative error is also used to validate the accuracy of the developed models, which will be discussed in detail in section 3.1.4.

2.3.4 Variance and Bias

Variance and bias are two important aspects in evaluating the precision and accuracy of simulation output data. In this section, both of them are reviewed, and their relationship with the research objectives are addressed. Since this research studies the transient behaviour (how long does it take to reach steady state) for a single simulation replication, variance is much more relevant than bias. Although not in a research objective, bias is still of great importance in evaluating the simulation performance.

Bias

The objective of this research is to predict the SRLASS, ΔT, which is more relevant to variance (indicating precision), rather than bias (indicating accuracy). However, this is not to say bias is not important when analysing output results. In this section, we will first define bias, and analyse bias using both analysis and simulation to show bias is negligible when planning simulation.

In statistics, bias is systematic favouritism that is presented in the data collection process resulting in misleading results [90]. There are several types of statistical bias. The bias of an estimator is defined as the difference between an estimator's expectation and the true mean value of the parameter being estimated.

In [90], Whitt analysed bias using mathematical expressions to show that bias is asymptotically negligible compared to the relative width of the confidence interval, when simulation time t is sufficiently large. Because bias is approximately in the order of t^{-1}, where t is the time required. And the relative width of confidence interval is proportional to $t^{-\frac{1}{2}}$.

In [68], in order to analyse the effects of the bias, the author implemented M/D/1 experiments and collected the waiting time as the metric of interest. He implemented different experiments by using different starting points, including starting the simulation experiments from empty, from a random point, and from a point chosen using the distribution. Results of [68] show that different starting points produced no significant different means of waiting time, which proves that the bias is negligible.

In conclusion, both analysis of [90] and experimental results of [68] show that initial condition bias can be assumed to be negligible.

Since the objective of this research is about predicting SRLASS, ΔT, which is mainly related to variance, rather than bias, as illustrated by Equation (2.11) and (2.12). In this research, simulations are all set to start from empty, with slight initial condition bias, which is assumed to be negligible.

Variance

The variance of a random variable or distribution, on the other hand, is the expectation, or mean, of the squared deviation of that variable from its expected value or mean. It is used as one of several descriptors of a distribution [32]. It describes how far values lie from the mean.

Assume that the sample mean of random variables $\{X_1, X_2, X_3, \ldots, X_N\}$ are μ where

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$$

and the variance is the expectation of the standard squared deviation, given by

$$Var(X) = E[(X - \mu)^2]$$

In our simulation, an unbiased variance is calculated using

$$Var(X) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \mu)^2$$

2.4 Discussion of SRLASS indicators

The objective of this thesis is to find the ΔT, which is the SRLASS for the parameter of interest to reach the steady state. As already argued in section 1.1, there are two main motivations for this research: 1) The requirement of knowing how to set the run length parameter for each simulation experiments. Stopping too early will get inaccurate results, while stopping too late will waste unnecessary time and resources. In this case, the SRLASS need to be represented using the simulation time, which is the setup parameter for the simulation. 2) The requirement of knowing how actual time will be consumed for the simulation, so that it can be judged whether or not the time consumed can be affordable. This is about the wall clock time, which is also the real time that it takes for the simulator to run.

In this thesis, the SRLASS is measured in the units of simulation time. Finding the relationships between the simulation time and the wall clock time will also be an objectives of this research.

2.4.1 Simulation Time and Packet Arrivals

There are a couple of things in queuing systems, for which we may wish to find the SRLASS, including simulation time and packet arrivals.

simulation time

Simulation time, compared to wall clock time, is the virtual time unit maintained by the simulator [40]. This time is used to schedule the events in the simulator. It is not necessarily the same as, or in any easy sense to be related to, the wall clock time. The simulation time is used to keep track of the simulation progress. For example, one wants to simulate packets arriving at one specific time point. This time point is controlled by the simulation time.

As all the simulation in this thesis, is run on NS2. In NS2, all the events are advanced and controlled by the simulation time. Therefore, simulation time in NS2 is one important setup parameter. This will control how long a simulation will last. Also, the simulation time is very important to other simulation software, e.g. MatLab. The results generated in the units of simulation time can be also used by other simulation softwares.

Packet Arrivals

In queueing systems, packets are generated at a source, sent to the server to be queued, and wait until served, and then to be received at the destination, as shown in Figure 2.3. The customers (packets in network) arrive and queue to be served in the buffer, with the mean arrival rate λ. In the buffer, time consumed for waiting here is called the waiting time, which is represented by T_w. When the server is ready, the packet will be served by the server, and the service time is represented by s. After service, the packet leaves the queueing system, for which the service rate is μ.

Data can be collected for each individual packet, so that the sample mean of them can be regarded as a proper estimator.

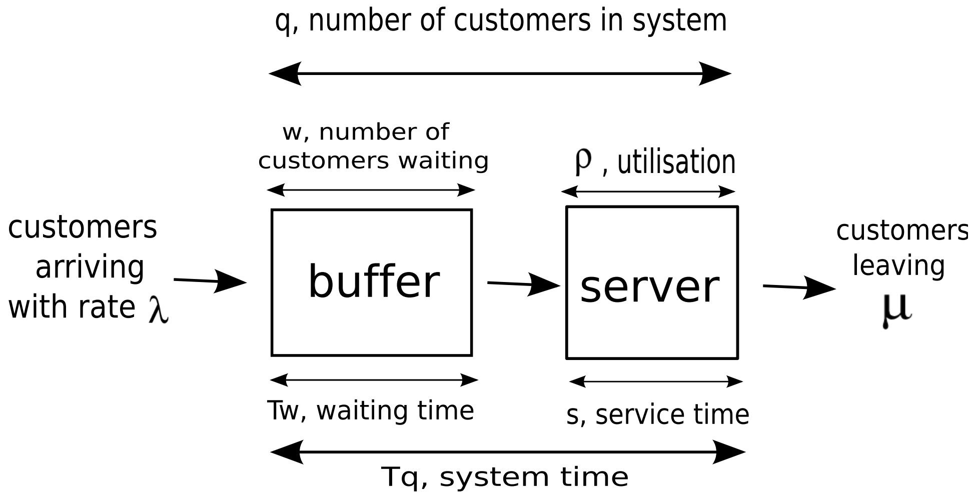

q, number of customers in system
w, number of customers waiting
ρ , utilisation
customers arriving with rate λ
buffer
server
customers leaving
μ
Tw, waiting time
s, service time
Tq, system time

Figure 2.3: Illustration of Queuing Systems

Data can be collected for each individual packet, so that the sample mean of them can be regarded as a proper estimator. For example, Roughan [77] collected the delay and queue length data for each individual packet. In this way, the SRLASS can be described in the units of packet arrivals. In other words, we plan simulation by finding x so that a simulation run lasts for x packet arrivals, when the mean delay or queue length reaches the steady state.

2.4.2 Wall Clock Time

Wall clock time is the real time consumed by the processor when the simulation is running. This parameter will not only depend on how the simulation models are designed, but also depends on the speed of the computer.

The design of the simulation models is important. Simulation can be at an arbitrary level of complexity, according to different system requirements. Therefore, when the requirements are satisfied, using more effective modelling of simulator models means less time is consumed for the simulator to run. However, with the increasing requirements of simulating complex networks, the design of the simulation models is getting more complex, and therefore, a powerful computer is required.

The speed of the computer processor is an essential factor affecting the wall clock time. With powerful computer processor and more memory allocated to the simulator, the simulation should run faster. However, as designing the computer processor is not in the scope of this research, how the computer processor affects the wall clock time consumed for the simulator will be discussed later in this thesis.

2.4.3 Summary

In this thesis, the methodology used is statistical, so the most direct analytical model developed is related to simulation time. Also, with the setup parameters for each queuing system, the packet arrivals are equivalent to simulation time.

Wall clock time, on the other hand, not only depends on the network model, but also relies essentially on the computer hardware parameters. In chapter 7, the simulation time will be mapped into the wall clock time, so that both of them can be planned before the simulation is run.

2.5 Conclusion

This chapter introduced the methodology used in this research to plan the simulation run length, which is a standard statistical analysis. Precision criteria, the confidence interval is crucial to this analysis. Relative and absolute errors were discussed, and relative error will be generally used in this research. The SCV is introduced and regarded as the standard estimator for planning simulation run length. Variance and bias were also reviewed.

From the standard statistical analysis, it can be concluded that, given the desired precision level, confidence level, and the SCV model of the parameter of interest, SRLASS, ΔT, can be analysed and predicted.

Also, a discussion about the indicators for the SRLASS is also given in this chapter. This mainly consists of two parts: the virtual simulation time and wall clock time.

In the following chapters, the standard statistical analysis will be further developed and applied to specific network scenarios and models.

Chapter 3

Simulation and System Models

This chapter addresses the definition of simulation, different types of simulation models, validation methods, as well as the chosen simulation tool, NS2. Also, this chapter gives an introduction to the relevant simulation models, including the network scenario used in this research, the packet multiplexing model, and the traffic models. Finally, an end-to-end simulation model is reviewed.

3.1 Simulation

3.1.1 Definition

Simulation modelling is a technique for using computers to imitate, or simulate, the operations of various kinds of real-world facilities or processes [47]. The studies on those facilities or processes are usually based on assumptions, which take the form of mathematical or logical relationships.

The most basic concepts for simulation consists of 'system', 'state' and 'events'. 'System' can be defined as the collection of hardware, software and firmware components [40]. The 'state' are the variable's values, describing the state of the system components at a particular time. An 'event' is defined as an instantaneous occurrence that may change the state of the system [47].

Take a queuing system for example. The whole queuing system, with packets injected into the input port, being served, and then leaving the queue at the output port, can be viewed as a system, as shown in Figure 2.3. The 'number of packets in queue' can be regarded as the queue state, and the state probability distribution is one of the most useful characteristic. Similar state variables are 'delay' of a packet, and if the packet is dropped or served, etc. The 'event' can be the arrival of a packet, or a departure of a packet after being served, or a packet loss event.

Simulation is designed using various models, representing the real systems. For example, for the queuing system, packets can be designed as objects, which are generated when a packet arrives in a queue, and killed when the packet leaves the queue. And the queue can be modelled using other objects, with the state variables of the buffer capacity and service rate. In this way, a simple queuing system can be characterised using simulation models/computer code.

3.1.2 Types of Simulation Models

Static vs. dynamic simulation model

A static simulation model is a representation of a system at a particular time, where such a representation does not change over time [47]. All of the model elements do not change during the execution of the model and remain constant. The elements of the model may be fixed in the simulation implementation, or be read in during the initialisation phase of the simulation, but will remain the same during the entire execution.

On the contrary, a dynamic simulation model represents a system as it evolves over time. The elements of a model of this type may change their properties or attributes during the model execution. One form of simulation based on dynamic simulation model is an interactive simulation[1], where users

[1] This approach could be useful for simulation of wireless networks, which is beyond the scope of this thesis.

can make modifications during runtime.

Deterministic vs. stochastic simulation modelling

A system may be regarded either as deterministic or stochastic, depending upon the relationship between input and output. The output of a deterministic system can be predicted completely if the input and the initial state of the system are known [31]. In other words, the model is not described by random variables and the same input always leads to the same output. The main characteristic of this type is that the inputs of the system determine the output as soon as they are fed in to the simulator, even though the process might not calculate the results immediately. A typical example is a communication system entirely represented by analytical models in which appropriate mathematical or logical expressions are used, and the corresponding outputs are calculated once the input parameters are given.

For a stochastic system, given the input and the state of the system it is possible to predict only the range within which the output will fall and the frequency with which various particular outputs will be obtained over observations. As the output produced by a stochastic simulation is random, statistical methods are often necessary to analyse the output data.

Continuous vs. discrete simulation models

A simulation model is classified as a continuous or as a discrete-event model, based on how the state variables in the model are updated throughout the simulation. A discrete system is one for which the state variables change instantaneously at discrete clock ticks, separated by a constant period of time. While a continuous system is one for which the state variables change continuously with respect to time [47].

It is worthwhile to distinguish the means of simulation modelling from the properties of real-world systems. Either continuous models or discrete-event models can model a continuous system; discrete systems such as transaction systems in a financial market or transportation systems are not restricted to discrete-event simulation models, appropriate continuous simulation models can also characterise them.

3.1.3 Simulation Clock

The simulation clock is used in a dynamic simulation. As for a dynamic simulation, state and variables of simulators are changed in an interactive way with the system. So, it is important to trace the simulation time throughout the whole simulation. Simulation time, different from the real time (a.k.a wall clock time), is always used to control when specific events happen. There are mainly two methods for advancing the simulation clock: constant simulation time advances and next-event advances.

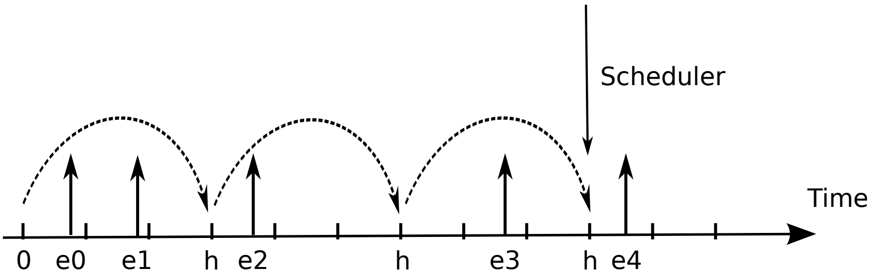


Figure 3.1: Constant Time-driven Advancement [53]

In simulation time advance method, the time clock is advanced to the next unit of simulation time. The time duration controlling the simulation advance is always constant. Figure 3.1 illustrates the concept of the constant simulation time advancement, where e_i is the time points when events happen, and the simulation clock is advanced regularly at a constant time interval h.

In the other technique, the clock is advanced event by event, as shown in Figure 3.2. When the next event happens, the simulation time will be incre-

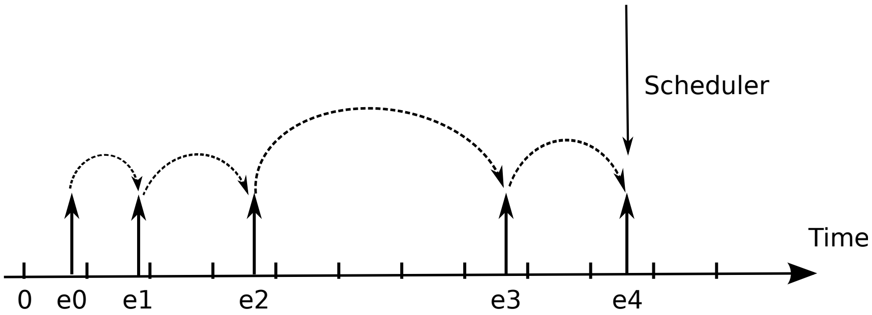


Figure 3.2: Event-driven Advancement [53]

mented correspondingly at the time point e_i. This is called the event-driven advancement. This method is currently mainly used by most simulation software, including NS2, which is used in this research. Events in the real system won't occur at a constant rate, and this will save computer resources and wall clock time by avoiding simulating unnecessary periods. A simulator designed using next-event time advance is called an event-driven simulator.

3.1.4 Validation of simulation results

Validation is the process of comparing the model's output with the behavior of the phenomenon, in other words, comparing model execution to known reality (physical or otherwise). Validation needs to be differentiated from verification which is the process of comparing the computer code with the model to ensure that the code is a correct implementation of the model [5].

Validation is of great importance in simulation since it is a measure of the extent to which it satisfies its design objectives [31]. This can be a difficult task, since a very general program that is capable of analysing a large number of scenarios will be impossible to test in all of them, especially as it would probably have been developed to solve systems that have no analytical solution to check against. However, even for the most general simulators it will be possible to test certain simple models that do have analytical solutions [72].

In this section, two methods of validation will be introduced, and will be used in Chapter 4, Chapter 5 and Chapter 6: Exponential Bestfit, and the Relative Width.

Exponential Bestfit

As in much research, the analysis in this research is frequently based on the assumptions of certain types of distribution of related variables. Therefore, a validation of such distributions is important. The Exponential Bestfit concept is introduced for this purpose.

Exponential bestfit is used because the overflow period and cycle time (to be introduced in Section 5.2) are both assumed to be exponentially distributed in this research. The assumption is reasonable, and is also used in [35]. The assumption is validated using Exponential bestfit.

The probability density function (p.d.f.) of an exponential distribution appears to be a straight line in log-linear scale [24] [72] [6]. However, since the raw data obtained from simulation results can not be judged by the naked eye, the bestfit of line is introduced to facilitate the validation [57].

For the exponential distribution, the p.d.f. is $f(x) = \lambda e^{-\lambda x}$. Figure 3.3 is an validation example of how the exponential distribution is validated. The figure shows the comparison between raw data and exponential bestfit of line. The raw data is obtained from random variables of an exponential distribution. The blue line plots the p.d.f. of the raw data. The blue line shows an approximately straight line in the log-linear scale. A line is fitted and plotted in red triangles, using exponential bestfit algorithm (see Appendix B.2).

As shown in Figure 3.3, the line is best fitted to the raw data. There are some 'noisy' tails which is usual because those are rare events, which require longer runs to obtain steadiness.

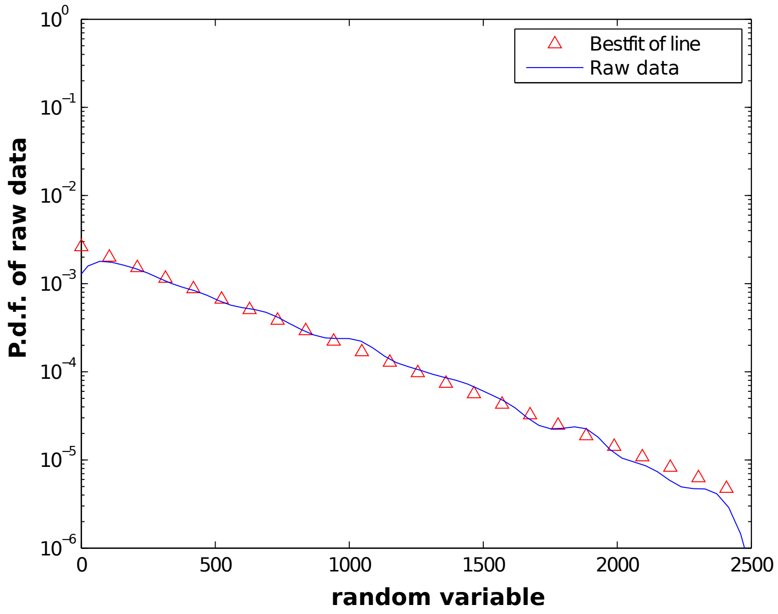


Figure 3.3: Exponential bestfit of line with raw data

Relative Width

Relative Width is introduced already in section 2.3.3. It can also be used to validate whether the developed analytical model is accurate.

Recall that the equation for the required sample size is related to the relative width according to the relationship:

$$N_r = 4 \cdot \frac{z^2_{1-\beta/2}}{\varepsilon^2_r} \cdot c^2 \tag{3.1}$$

From this equation, we can know that if the analytical model of SCV, c^2 is known, the required sample size for the target parameter to reach the steady state can be obtained from them. And the equation of the relative width is:

$$\varepsilon_r = 2z_{1-\beta/2} \cdot \frac{\sigma}{\bar{X}\sqrt{N}} \tag{3.2}$$

Therefore, plotting the relative errors against the sample size, as supplied in

Equation (3.2) can be a way to validate the accuracy of the analytical model.

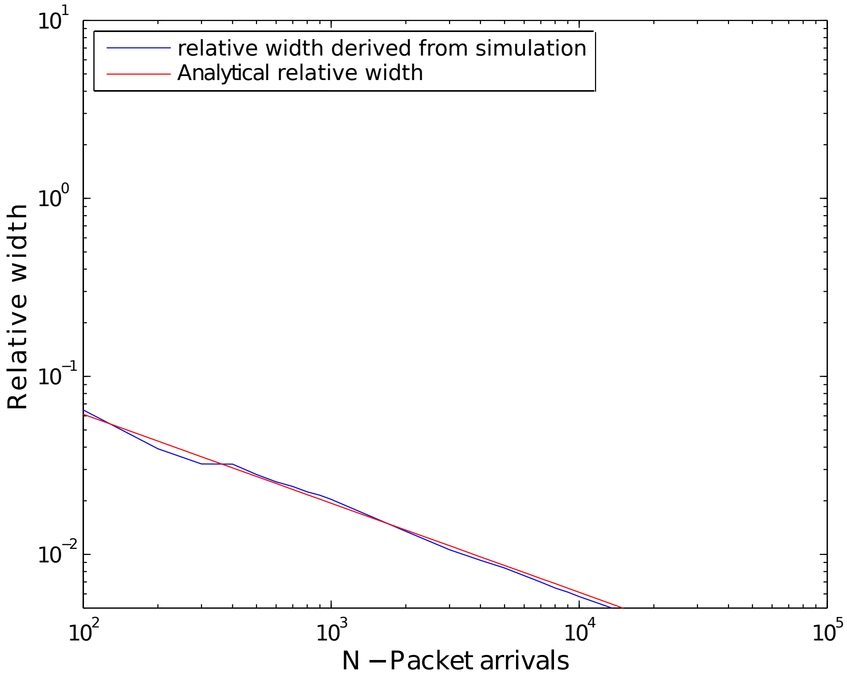


Figure 3.4: An example of comparison between simulated and analytical relative width

As shown in Figure 3.4, an M/M/1 queue is simulated and the interval between packet arrivals is collected. The blue line plots the simulated values, while the red line is the relative error comparing to the analytical model. The closeness between these two lines show how accurate the model will be. Using this method, this research will be validated well before the main results are given.

3.1.5 Simulation Tools: Network Simulator 2

In this research, Network Simulator 2 (NS2) [17] is employed since it is widely accepted, and is open source software. NS2 is a discrete event network simulator that derived from the REAL network simulator.

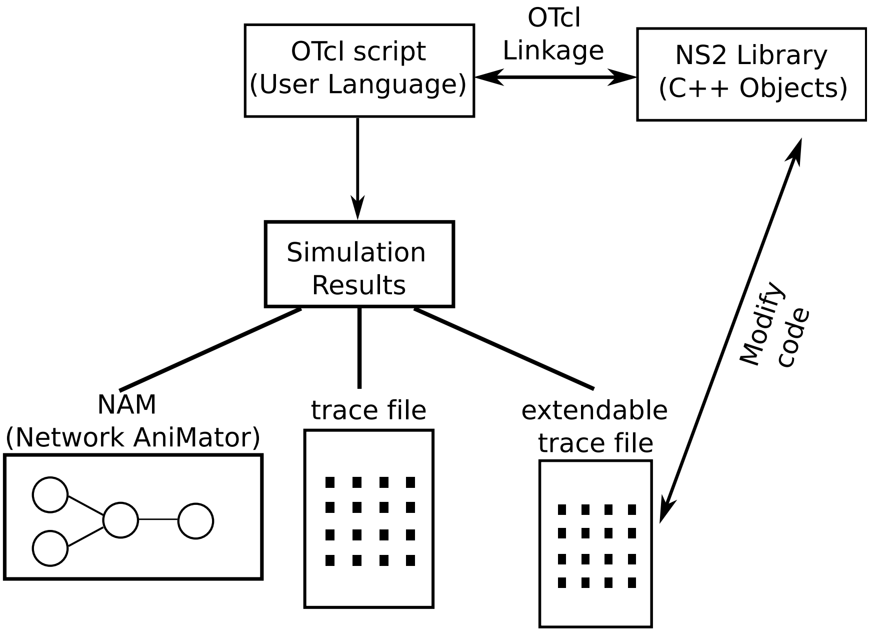


Figure 3.5: Functional Layout of NS2

As shown in Figure 3.5, NS2's source code is split into two parts: OTcl, an object oriented version of Tcl9 for configuration and simulation scripts[2], and C++ for its core engine. The combination of the two languages achieve both performance and ease of use. OTcl scripts, as a user language, can be edited directly by users to call the corresponding models in the NS2 library. These two parts are linked using OTcl linkage, which is all edited and contained in an xml file.

After running the codes through OTcl script, there are two kinds of simulation results: NAM and trace file. Network AniMator (NAM) is a graphical tool used to represent a visual topology of networks and animation of events, e.g. packet flows and packets drops. An alternative choice is the trace file which is appropriate for statistical analysis. A standard trace file traces all events and all relevant information in a well defined format, on which can be further carried out post-scripting, e.g. MatLab, Awk and Perl etc. This

[2] A user language used in NS2.

standard trace file records all detailed events, but for the majority of occasions is a waste of memory. Thus, for different simulation requirements, users can define their own trace file using existing functions, or further edit models inside C++ codes. Furthermore, a new protocol can be implemented in NS2 by adding C++ code and updating OTcl configuration files in order for NS2 to recognise the new parameters and methods for the new protocol. The C++ code also defines those parameters and methods which are available for OTcl scripts.

Because of the open source nature of NS2, all the models can be simply read, edited, and upgraded for new research requirements. Users can modify the essential codes, and output their self-regulated trace file as the simulation results.

In conclusion, there always are 5 steps to implement and simulate in NS2:

- Analysing the research requirement and implementing it by updating C++ code for new functionalities.
- Describing the simulation scenario in an OTcl script, and calling functions defined in NS2 library (C++ source codes) using OTcl linkage;
- Running the simulation;
- Carrying out post-scripting using Awk, Perl, or MatLab, etc.
- Analysing the generated trace files.

Some disadvantages of NS2 come from its open source nature. For example, documentation is often limited and out of date w.r.t the current release of the simulator. Fortunately consulting the highly active newsgroups and browsing the source code directly solves most problems. Another disadvantage is the lack of tools to describe simulation scenarios and analyse or visualise simulation trace files.

In this research, NS2 is employed as the simulation tool, and C++ codes are modified for self-regulated trace files. New added parameters are defined in the OTcl linkage, as well as within C++ codes. Post-scripting and results generation is carried out using MatLab.

3.2 Traffic Models

In network simulation, accurate traffic models are very important. A model can be based on real network trace samples, or created using analytical models, where the traffic models are described using mathematical tools, to mimic the real network traffic features. In this thesis, analytical traffic models are used because they are most commonly used in most simulation studies [38][3][53][57].

Analytical traffic models generate pseudo random data, which follows some pattern, defined using mathematical tools. This will make the data maintain statistical features of the real network traffic. However, it is universal that different types of traffic follow different traffic patterns. For example, it is well known that the voice traffic follows a largely Markovian pattern, while the data-dominant traffic is often self similar with heavy-tailed features[49][69][62][67], which is usually described by the Pareto distribution [49][19][30].

A good traffic model plays an important role in capturing the key features of the real queuing system. If the arrival process is not modelled accurately, the network performance may be overestimated or underestimated [74]. In this section, the traffic models to be used in this thesis will be introduced, mainly focusing on the ON/OFF traffic model [4], since it is widely accepted and used [55][56][80][59][60][22][98].

3.2.1 Short-Range Dependent Traffic Model

The first performance models for telecommunication systems were based on the assumption that aggregate packet arrival processes follow a Poisson process [9][34][72], while sources are always modelled as an ON/OFF model, as this captures the talkspurt feature of voice for example.

ON/OFF sources

There is no lack of research using the ON/OFF model, and the bursty property of network traffic can be captured by the ON/OFF traffic model [4][38] [73][78][93][22][60][59][87][15]. The ON/OFF model captures the network feature using the two state: ON state and OFF state. Figure 3.6 illustrates how it works.

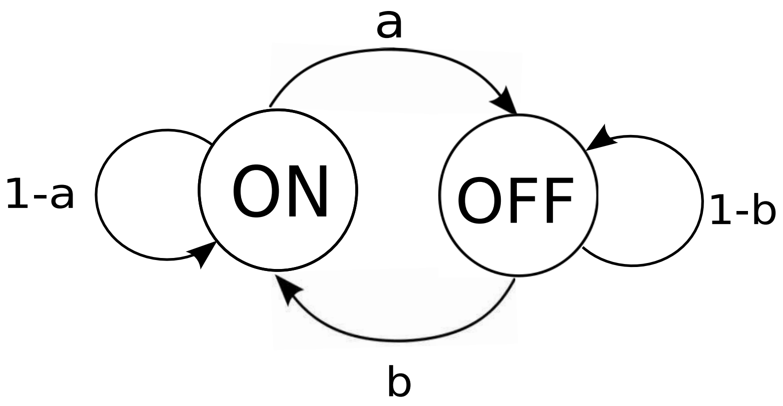


Figure 3.6: ON/OFF state

In this model, when the source is in the ON state, the packets are generated at a constant rate[3]. On the other hand, when the source is in the OFF state, there will be no packets generated. In a time slotted time base, the probability that the state changes at the end of each time slot from ON to OFF is 'a', otherwise from ON to ON is '1-a'. Similarly, the probability that

[3] Alternatively, this rate could be variable, e.g. a Poisson process.

the state changes from OFF to ON is ‘b’, otherwise from OFF to OFF is ‘1-b’.

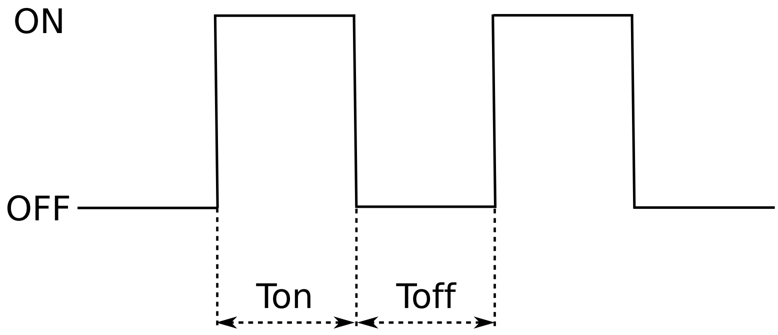


Figure 3.7: Illustration of a single ON/OFF source model

The switch between the ON and OFF state can also be viewed in another way, shown in Figure 3.7. In this figure, the time duration of the state keeping in the ON state and OFF state are tagged as T_{on} and T_{off} respectively, and they are all modelled as exponentially distributed[4] for the Markovian modulated arrival process. This is widely used to model the voice traffic.

3.2.2 Long-Range Dependent Traffic Model

Significant research shows that some data traffic in networks has features that do not follow the exponential distribution, but features heavy-tailed distributions. It is better to use the Pareto distribution to capture this heavy-tailed feature [49] [67] [30] [62] [19]. And the LRD traffic models are highly variable, and hard to predict, sometimes, very difficult to reach steady state in simulation, which makes this research difficult. In the ON/OFF model, the sojourn times of LRD traffic are all modelled as the Pareto distribution.

[4] Exponential distribution is formed for the continuous time view of this model. This is the continuos version of a Geometric distribution when the time slot duration → 0.

Pareto Model

A random variable x is defined as being Pareto distributed if

$$Pr\{X > x\} = 1 - F(x) \approx \frac{1}{x^{\alpha}} \tag{3.3}$$

as $x \to \infty$ and $0 < \alpha < 2$. The formula shows the probability that the random variable X is larger than x. The shape of it is heavy-tailed, and this can have a high or even infinite variance.

The cumulative distribution function (cdf) is

$$F(x) = 1 - (\frac{\phi}{x})^{\alpha} \tag{3.4}$$

and the probability density function (pdf) is given by

$$f(x) = \frac{\alpha}{\phi} \cdot (\frac{\phi}{x})^{\alpha+1} \tag{3.5}$$

and the mean value of the Pareto distribution is

$$E(x) = \phi \cdot \frac{\alpha}{\alpha - 1}, \qquad \text{when } \alpha > 1 \tag{3.6}$$

the variance of Pareto distribution is given by

$$Var(x) = \begin{cases} \infty & \text{for } \alpha \in (1{,}2] \\ \frac{\phi^2 \alpha}{(\alpha-1)^2(\alpha-2)} & \text{for } \alpha > 2 \end{cases} \tag{3.7}$$

Figure 3.8 shows the comparison between the Pareto distribution and the Exponential distribution.

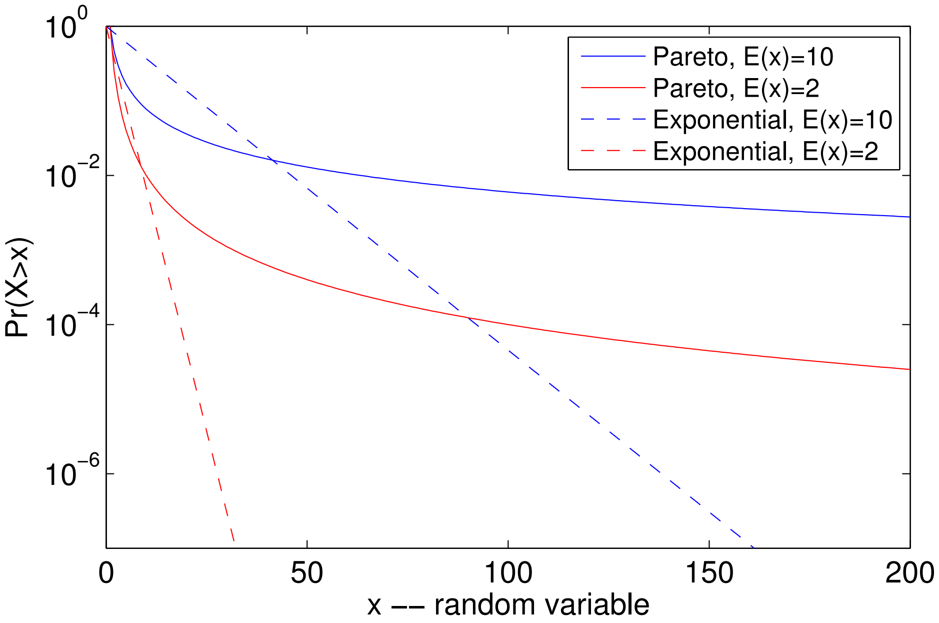


Figure 3.8: Comparison between Exponential and Pareto distribution

Figure 3.8 plots the probability that the random variable X is larger than x, with different mean value of different distribution type. The solid lines represent the Pareto lines, while the mean value of the blue line equals 10, while the red one equals two. Similarly, the dotted lines represent the Exponential distribution. With the same mean value, the Pareto distribution gives a much larger probability of rare events (i.e. the tail in Figure 3.8), compared to the Exponential distribution, and falls with a large portion of random variables in the tail. Those extremely large values can not be ignored. With this pattern, traffic with Pareto features will have heavy tails, which means high or even infinite variance.

3.3 Packet Multiplexing Model Used in This Research

This thesis expands research into simulation run length planning for classical queue models, into a more realistic model, the Packet Multiplexing Model (PMM). In this section, the PMM will be introduced in detail. It is based on the single ON/OFF traffic source, as already introduced in section 3.2, by multiplexing N of them into a FIFO buffer.

The packet multiplexing model used in this research is shown in Figure 3.9 [72][3][37][57][22][74][33][54].

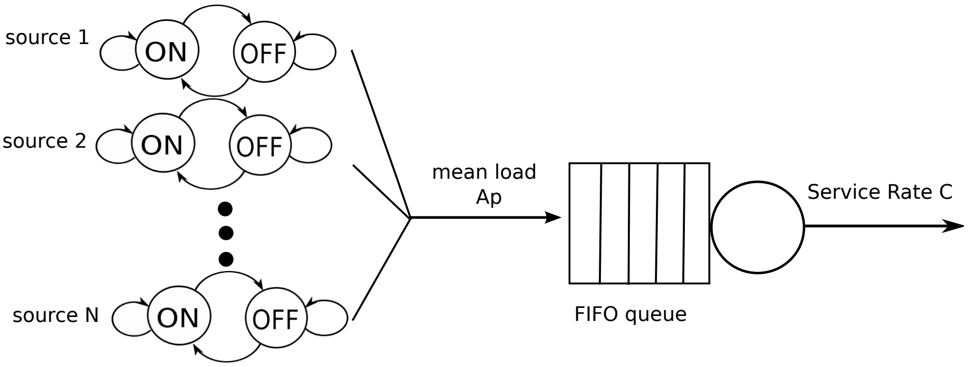


Figure 3.9: Packet multiplexing model

N homogenous ON/OFF VoIP packet sources are multiplexed into a FIFO queue, with service rate C (in pps). Each ON/OFF packet source generates packets with rate h (in pps) when active (the ON state), and sends no packets when it is idle (the OFF state). The duration in the ON state and OFF state are denoted as T_{on} and T_{off} respectively. The sojourn times in the states can be modelled as an Exponential distribution for Markovian traffic source, or the Pareto distribution for the LRD traffic source in this thesis. When the instantaneous overall arrival rate exceeds the service rate C for an amount of time until the buffer is overflowed, packet losses occur.

3.4 End-to-End Network Model

3.4.1 FG/BG Network Model

It is well-accepted to research an end-to-end network using Foreground Traffic (FT)/ Background Traffic (BT) model [82] [79] [50] [57] [35] [46] [42], and has been proven to be a reliable modelling method. This method gives the possibility of treating (when coding) the foreground traffic (with more detailed description since it is the focus of research) and background traffic separately. In [46], this network model is used in a FIFO scheduling queue, while in [42], the FG/BF method is proven to be a successful and reliable model, even for Fair Queueing Scheduling(FQS).

Figure 3.10 illustrates how this model works.

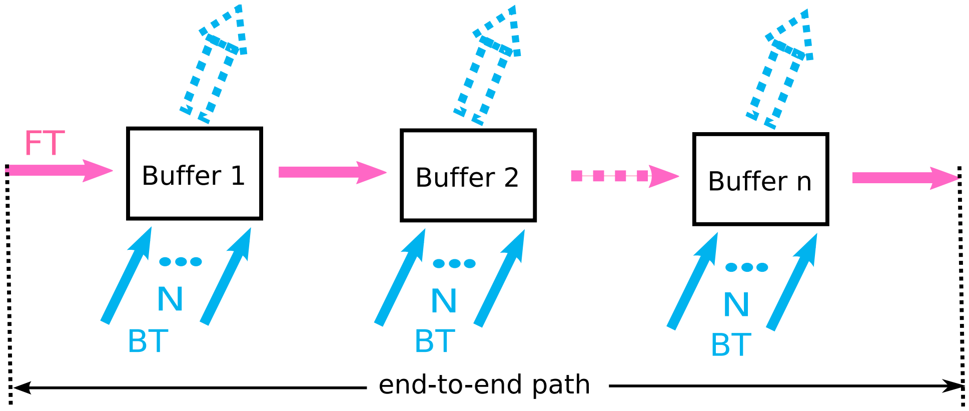


Figure 3.10: End-to-end FG/BG network model

As shown in Figure 3.10, the traffic is divided into FT and BT. The FT is the traffic flow of interest, which is injected into the network, and passed through every buffer in series in the network. BT flows are all independent traffic sources, which are multiplexed with the FT at each buffer, and routed elsewhere in the network.

As shown in Figure 3.10, FT flow traverses n identical buffers throughout the network.

3.5 Conclusion

Simulation techniques and simulation models are introduced in this chapter, including all relevant tools, e.g. the validation methods, simulation tools, as well as simulation clock.

Also, the system models, which will be used throughout this thesis are reviewed. Traffic models are introduced, followed by the network scenario, and the PMM. This model is widely accepted, and it is a good start to begin the simulation planning research into a more realistic network model. End-to-end network model is discussed finally in this chapter, and an FG/BG model is used.

Chapter 4

Simulation Planning for Delay in Markovian Source PMM

4.1 Overview of simulation planning for delay

Delay is a very important QoS parameter in network research. There is no lack of such research about delay, see [60][83]. This chapter investigates simulation planning for delay in a Markovian source PMM. This chapter provides an analytical model for the SCV, which directly gives how to plan simulation for delay in the PMM.

The PMM model is adopted here mainly due to its wide acceptance, application in the area of packet networks research [61] [22] [98] [74] [33] [3] [37] [72]. This chapter uses the PMM introduced in section 3.3, and applies statistical analysis, as introduced in section 2.3, to the PMM scenario.

This chapter uses the packet-scale/burst-scale queue length characteristics (detailed introduced in Section 4.2). Delay is collected from each individual packet arrival. If packet i suffers a long waiting time, it is more probable that also packet $i + 1$ will experience a long waiting time. In this case, there are correlations between adjacent packets [77]. Therefore, the data needs to be sampled. This research uses a formula (to be introduced in Section 4.3) to define the sampling interval, in the units of packet numbers, i.e. how many packets between samples is used. In this way, the correlation between adjacent packets is removed.

The required sample size for the delay to reach steady state, is then in the units of the sampled data. Using the calculated required sample size and the sample interval, finding a formula for the SRLASS ΔT is achievable.

4.2 Packet-scale/ Burst-scale Queuing Behaviour

The PMM has two specific queue length characteristics [41] [72] [75] [76] , which are as shown in Figure 4.1. This figure gives a log-linear plot, which shows the distribution of the queue length state probability. The x-axis shows the number of packets in the buffer, which is also called the state of the queue length. While the y-axis gives the corresponding probability of the specific state.

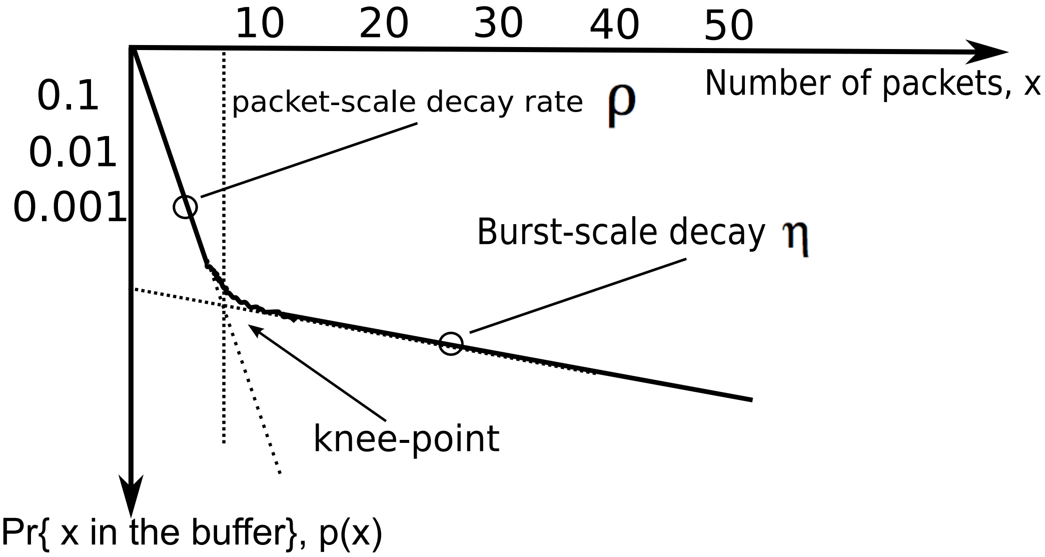


Figure 4.1: Packet and burst-scale queuing

This state probability distribution is composed of two components, packet-scale and burst-scale. As shown in Figure 4.1, queues have 'exponential' type decay rate. In the packet-scale region, if the overall arrival rate is less than the service rate [76], then the average queue length will be in the order of tens of packets [72]. The slope of this part coincides with corresponding distribution of the M/D/1 queue if the packet sizes are fixed. Therefore, the so called decay-rate (the slope) for this part can be obtained using only the utilisation ρ. The packet-scale component is often referred to as the 'smooth traffic' component [61].

It is widely accepted that Internet traffic cannot be modelled just using the 'smooth traffic' component. It has its inherent feature of 'burstiness' [70]. The second parts in Figure 4.1 shows the result of non-negligible burst-scale components in the traffic. Burst-scale queuing occurs when the instantaneous overall arrival rate exceeds the service rate over a substantial time duration[1]. This will make the queue grow at a higher rate, and the average queue length of this part may be in the order of hundreds of packets [72]. The decay rate for this part is called burst-scale decay rate and denoted as η. η has an accurate analytical model given in [3], and details are given in Appendix A.

The intersection of the two parts is called the 'knee point', as shown in Figure 4.1, where just enough sources are active to use all the service rate [61].

Since the queue length has a direct relationship with the delay, therefore, the distribution of delay always has the same pattern as the distribution of queue length [50].

4.2.1 SCV Model of Delay

As the two separate parts of the queue length distribution all follow a Geometric distribution, the combination of two different Geometric distributions can be used so that the SCV model of the delay is obtained.

For any Geometric distribution with parameter p, the following formulas apply:

[1] This tends to happen more regularly in the PMM as ρ gets larger (typically > 0.5)

$$\text{Mean of the Geometric distribution is}\frac{1-p}{p}$$

$$\text{Variance of the Geometric distribution is } \frac{1-p}{p^2}$$

$$\text{Mean square for the Geometric distribution } \frac{1-p}{p^2}+(\frac{1-p}{p})^2 = \frac{(p-1)(p-2)}{p^2}$$

Let P_B be the probability that the queue is experiencing burst-scale queueing, using the above Geometric distribution formula, the expectation of the queue length, $E[Q]$ can be obtained from:

$$E[Q] = (1 - P_B) \cdot \frac{\rho}{1-\rho} + P_B \cdot \frac{\eta}{1-\eta} \tag{4.1}$$

where parameter η and burst probability P_B are shown in [76] (also reviewed in Appendix A)

$$\eta = \frac{1 - [ln(h/C)/ln(\rho) + (h^2 Ton\rho)/(C(1-\rho)^2)]^{-1}}{1 - [\rho(1-\rho)^2/(h/C) \cdot Ton \cdot [(1-\rho)C + h \cdot \rho]]} \tag{4.2}$$

and

$$P_B = \frac{1}{(1-\rho)^2 \cdot (C/h)} \cdot \frac{\rho \cdot (C/h)^{(C/h)}}{(C/h)!} e^{-\rho \cdot (C/h)} \tag{4.3}$$

Also, from the Geometric distribution, the mean square for queue length $E[Q^2]$ is:

$$E[Q^2] = (1 - P_B) \cdot \frac{\rho + \rho^2}{(1-\rho)^2} + P_B \cdot \frac{\eta + \eta^2}{(1-\eta)^2} \tag{4.4}$$

from which the variance of queue length can be obtained by

$$Var[Q] = E[Q^2] - (E[Q])^2 \tag{4.5}$$

Since the delay $E[WT]$ has the following relationship with the queue length:

$$E[WT] = \frac{E[Q]}{A_p} \tag{4.6}$$

and

$$Var[WT] = \frac{Var[Q]}{A_p^2} \tag{4.7}$$

And the SCV of delay, $c^2(WT)$ can be found from:

$$c^2(WT) = \frac{Var(WT)}{E[WT]^2} \tag{4.8}$$

From the model of SCV, the required sample size for the delay to reach steady state can be found, as already introduced in section 2.3.3 using Equation (4.9), with the pre-defined relative width ε_r, and preset confidence level, described using $z^2_{1-\beta/2}$.

$$N_r(WT) = 4 \cdot \frac{z^2_{1-\beta/2}}{\varepsilon_r^2} \cdot c^2(WT) \tag{4.9}$$

The data collected for the delay of each adjacent individual packet arrival has correlations. Therefore, sampling is required in this stage, so that the SRLASS period ΔT can be known through the required sample size $N_r(WT)$ times the sampling interval (explained in Section 4.3).

4.3 Sample Interval

Statistical analysis is based on the samples being identically distributed and independent to each other. However, data collected by each packet arrival will be correlated. For example, if the delay is the parameter of interest, if packet i suffers a long waiting time, it is more probable that also packet $i + 1$ experience a long waiting time [77]. Therefore, correlations need to be accounted for when proceeding to other analysis work in this thesis.

Correlation between adjacent samples can be reduced and eliminated by increasing the sampling interval, the time duration between two consecutive samples. Therefore, we aim to find the minimum sample interval to remove correlation between adjacent samples.

In [81], an idea for the minimum time interval for two samples from different regeneration cycle[2] is given, which can be used as a guide to choosing the sampling interval in this model. In [1], the c.d.f. of busy period[3] is given, which is the probability of time t is less than a busy period. If we set the probability of t less than a busy period to approach 0, it means the probability that two adjacent samples from the same busy period will be approximately 0. This will ensure that two consecutive samples are from different regeneration cycles. In conclusion, we inversely use the c.d.f. in [1] and find the minimum sample interval t to satisfy that the c.d.f to approach 0.

Define the c.d.f. of the busy period duration $B^c(t)$, to be the probability that t is less than a busy period, Pr(t<a busy period). The formula of $B^c(t)$ is given in [1] as:

$$B^c(t) = 2\alpha t^{-1}\gamma(t/\beta) \tag{4.10}$$

where

$$\alpha = (1-\rho)^{-1}(1+(1-\rho)(1-\xi)+O((1-\rho)^2)) \tag{4.11}$$

$$\xi = m_3/3m_2^2 = 0.074 \tag{4.12}$$

$$\beta = \frac{(1+\rho^{1/2})^2}{4} \tag{4.13}$$

$$\gamma(t) = (2\pi t)^{1/2}e^{-t/2} \tag{4.14}$$

[2] Regeneration cycle is time cycle over which the queue is alternatively busy then idle.

[3] The busy period is the time duration when the queue is in the busy state, when normally the overall arrival rate is larger than the service rate.

$B^c(t)$ is required to be close to 0, since $B^c(t)$ is the probability that sampling interval t is less than a busy period (meaning two adjacent samples are coming from the same regeneration cycle). Therefore, the minimum sample interval t is determined by taking $B^c(t)$<0.0001 (which is assumed to be sufficiently small, but a smaller value could be used if desired).

In conclusion, sample interval t is determined by finding minimum t which makes $B^c(t)$<0.0001.

4.4 Extending Analysis to an End-to-End Network

The previous sections aim to predict the SRLASS, ΔT of the mean delay based on the PMM for a single access link. This section extends the analysis to an end-to-end network, using the FG/BG end-to-end network model (already introduced in Section 3.4.1).

Let n represent the number of nodes. If all the nodes are assumed to be identical [35], the delay for the end-to-end $E_{e2e}[WT]$, can be obtained from the individual buffer case:

$$E_{e2e}[WT] = n \cdot E[WT] \tag{4.15}$$

Since the variance of summation has the general rules:

$$Var(aX + bY) = a^2Var(X) + b^2Var(Y) + 2abCov(X, Y) \tag{4.16}$$

for two random variables X and Y.

Then, the variance of the end-to-end network can be obtained from

$$\begin{aligned} Var_{e2e}(WT) &= Var(WT_1 + WT_2 + \cdots + WT_n) \\ &= Var(WT_1) + Var(WT_2) + \cdots + Var(WT_n) \\ &= n \cdot Var(WT) \end{aligned} \tag{4.17}$$

by assuming all the nodes are independent [25] [26] [51] [23] [35] to each other. And the $c^2_{e2e}(WT) = Var_{e2e}(WT)/E_{e2e}[WT]^2$.

4.5 Previous Work by Whitt to Use for Validation

As already reviewed in Chapter 2, Ward Whitt was first to plan queueing simulations. In [92], Whitt proposed to approximate queue models to Diffusion Process and shows applications of this method, including M/M/1, M/M/∞, and G/G/∞. However, they are too general to be compared for PMM. Fortunately, Whitt plans simulation for a packet queue in his earlier paper [89]. In this case, planning simulation run length for our PMM can be also done using his formula. His formula will be compared with our analytical model through numerical examples in Section 4.6.3.

4.5.1 Overview

In [89], Whitt proposed to plan a simulation before the simulation is run. He focused on the classical queue models, including the M/M/1 queue, G/G/1 queue, etc. He used a statistical analysis (also used in this research, as introduced in section 2.3). In [89], the general formula for the simulation run length is given by

$$t_a(\epsilon, \beta) = \frac{4\sigma^2 z^2_{\beta/2}}{\epsilon^2} \tag{4.18}$$

$$t_r(\epsilon, \beta) = \frac{4\sigma^2 z^2_{\beta/2}}{\epsilon^2 (\bar{X})^2} \tag{4.19}$$

corresponding to absolute width and relative width of CI, respectively. The notations are:

- t_a: required run length using absolute width

- t_r: required run length using relative width
- ϵ: the target error width defined according to targeted precision level
- β: confidence interval parameter, which defines the level of precision.
- σ^2: variance of metrics, e.g. waiting time or queue length.
- $\bar{X}$: mean of the metric, e.g. true mean of waiting time or queue length.

He approximates the general queue models into a Regulated Brownian Motion (RBM) process, and finds the SCV of the waiting time to be

$$\frac{\sigma_W^2}{(E[W_0])^2} = \frac{2(c_A^2 + c_S^2 - 2c_{AS}^2)}{(1-\rho)^2} \tag{4.20}$$

This formula is suitable for comparing results with our results from Chapter 4, since it also uses the mean waiting time as the metric of interest. As a general formula, Equation 4.20 can be applied to different queue models by finding the values for c_A^2, c_S^2 and c_{AS}^2. Finding these values for the PMM is described in subsection 4.5.2.

4.5.2 Methodology for Planning Simulation of PMM

Whitt discusses a packet queue model in [89] in Section 5.3, which can be applied to our PMM. This section will review the methodology used in [89] in detail.

The packet queue model is a single server queue, with unlimited waiting room, using first-come first-served (FCFS) discipline.

The model described in Whitt's paper has k customer classes. For each class i, the customers arrive in batches. Each batch consists of independent adjacent packets. Therefore, each batch can be interpreted as the 'ON' state in our research, and the idle period is the 'OFF' state. This is all shown in Figure 4.2.

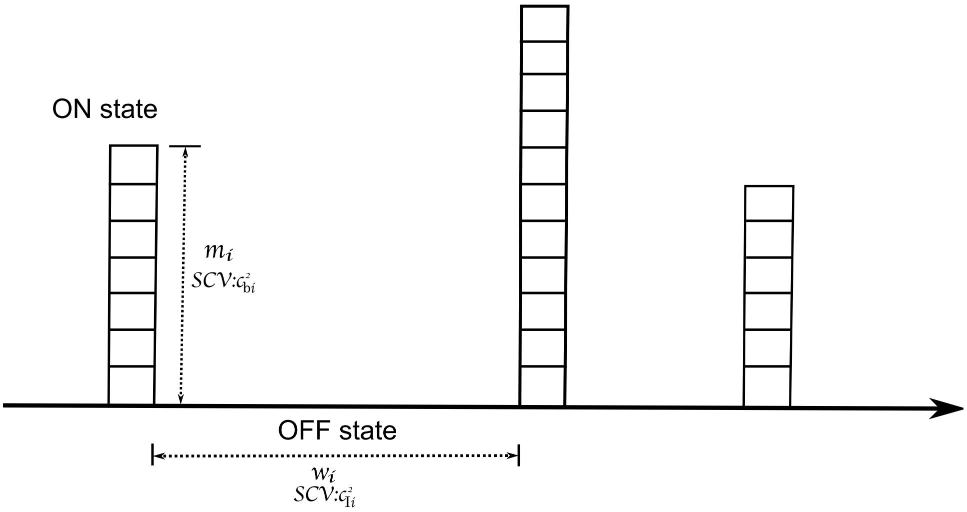


Figure 4.2: Illustration of Parameters in the Packet Queue Model

The batch size, service times, space between customer arrivals in one batch, and the idle period are all described in Table 4.1.

	Mean	**SCV**	**Description**
Batch Size	m_i	c_{bi}^2	Each batch consists of a random number of customers.
Service Times	τ_i	c_{si}^2	Service time is the time used to serve one customer.
Space between arrivals	ξ_i	c_{xi}^2	The spaces between the arrivals of the customers in one batch. No spacing in PMM, since the customers coming in one group where the spaces can be ignored.
idle period	w_i	c_{Ii}^2	The sojourn time in the 'OFF' state, as shown in FIgure 4.2.

Table 4.1: Parameter Description in the Packet Queue Model

For each class i, let λp_i be the arrival rate of batches, where $p_1 + p_2 + \cdots + p_k = 1$. Therefore, the arrival rate of customers for each class i is

$$\bar{\lambda} q_i = \lambda p_i m_i \tag{4.21}$$

where q_i is the proportion of all customers of class i, given by

$$q_i = \frac{p_i m_i}{\sum_{i=1}^{k} p_i m_i} \tag{4.22}$$

and $\bar{\lambda}$ is the total mean arrival rate of customers.

Let r_i be the proportion of service time of class i, given by

$$r_i = \frac{\tau_i}{\tau} \tag{4.23}$$

where τ is the service time for all customers, and τ_i is the service time for customers of class i.

Let β_i be the proportion of busy time in each cycle, given by

$$\beta_i = \frac{m_i \xi_i}{m_i \xi_i + w_i} \tag{4.24}$$

Let c_{Ai}^2 be the SCV for the ith arrival process, given by

$$c_{Ai}^2 = m_i(1 - \beta_i)^2(c_{bi}^2 + c_{Ii}^2) + \beta_i^2 c_{xi}^2 \tag{4.25}$$

Based on the above parameters, c_A^2, c_S^2 and c_{AS}^2 are given by

$$c_A^2 = \sum_{i=1}^{k} q_i c_{Ai}^2 \tag{4.26}$$

$$c_S^2 = \sum_{i=1}^{k} q_i [r_i^2 c_{si}^2 + (r_i - 1)^2 c_{Ai}^2] \tag{4.27}$$

$$c_{AS}^2 = \sum_{i=1}^{k} q_i (1 - r_i) c_{Ai}^2 \tag{4.28}$$

4.5.3 Application of Whitt's Formula to PMM

In the previous subsection, planning run length simulation of waiting time for a k-class packet queue model is introduced. The values of the parameters all depend on the queue model. Therefore, in this section, the formula will be applied to our PMM (with detailed introduction in Section 3.3).

The PMM is with deterministic service pattern (constant service rate). So, there is no variation of service time, which means $c_{si}^2 = 0$.

There is approximately no spacing between the packets (customers) when in the 'ON' state, since the packets are coming consecutively. So $\xi_i = \beta_i = 0$ according to [89].

Since there is only one customer class, it is obvious that $r_i = 1$ and $q_i = m_i = m$ (mean batch size).

In Section 5.4.2, Batch Size (μ_p in section 5.4.2) and Idle Period are both approximately exponentially distributed, which with the feature of SCV=1. So, $c_{bi}^2 = c_{Ii}^2 = 1$.

In this case, $c_S^2 = \sum_{i=1}^k q_i[r_i^2 c_{si}^2 + (r_i - 1)^2 c_{Ai}^2] = 0$, since $r_i = 1$ and $c_{si}^2 = 0$. $c_{AS}^2 = \sum_{i=1}^k q_i(1 - r_i)c_{Ai}^2 = 0$, because $r_i = 0$.

And

$$c_{Ai}^2 = m_i(1 - \beta_i)^2(c_{bi}^2 + c_{Ii}^2) + \beta_i^2 c_{xi}^2 = 2m \tag{4.29}$$

where m is the batch size.

In this case,

$$c_A^2 = \sum_{i=1}^k q_i c_{Ai}^2 = m \cdot c_{Ai}^2 = 2 \cdot m^2 \tag{4.30}$$

Taking values of c_A^2, c_S^2 and c_{AS}^2 into Equation 4.20, the corresponding run length will be obtained, which can be compared with our results.

4.6 Numerical Examples

4.6.1 Simulation Set-up Parameters

In this section, numerical examples are provided by using the widely used PMM parameters [88] as follows:

- T_{on}=0.96s, T_{off}=1.69s,
- ON rate h=170packets/s,
- packet size=100 bytes,
- utilisation ranging from 0.6 to 0.9 [4].
- Number of sources ranging from 50, 70, 100 and 120.
- Unlimited buffer capacity to remove the effects of the lost packets
- model type including single access, and end-to-end (Foreground Traffic and Backgroud Traffic are set to be same as shown in Table 4.2).

And these parameters are listed in Table 4.2.

N	T_{on}	T_{off}	**h**	**a**
50	0.96s	1.69s	170pps	0.3623
70	0.96s	1.69s	170pps	0.3623
100	0.96s	1.69s	170pps	0.3623
120	0.96s	1.69s	170pps	0.3623

Table 4.2: Set-up Parameters for Markovian Traffic Source in Delay Evaluation

[4] Use of utilisation in the range [0.6, 0.9] is because: 1) This is a typical load range on an access node; 2) If utilisation is under 0.6, the arrival process will approximately tend to a Poisson process, i.e. it won't exhibit burst scale queueing, and standard classical simulation run length planning techniques [89] can be used instead.

4.6.2 Validation

The validation is done using the methodology introduced in Section 3.1.4. A comparison between the simulated relative width (the solid lines) and the analytical relative width (the dotted lines), is shown from Figure 4.3 to Figure 4.10. The x-axis is the sample size corresponding to the number of samples collected. And the relative width is in the units of percentage, representing the respective precision level. The analytical relative width is calculated using Equation 3.2, 4.1, 4.4, 4.5 and 4.9.

Relative width is set to 10% in the validation part, and a 95% confidence level is used, as this is most commonly chosen in the literature, although any value could be chosen.

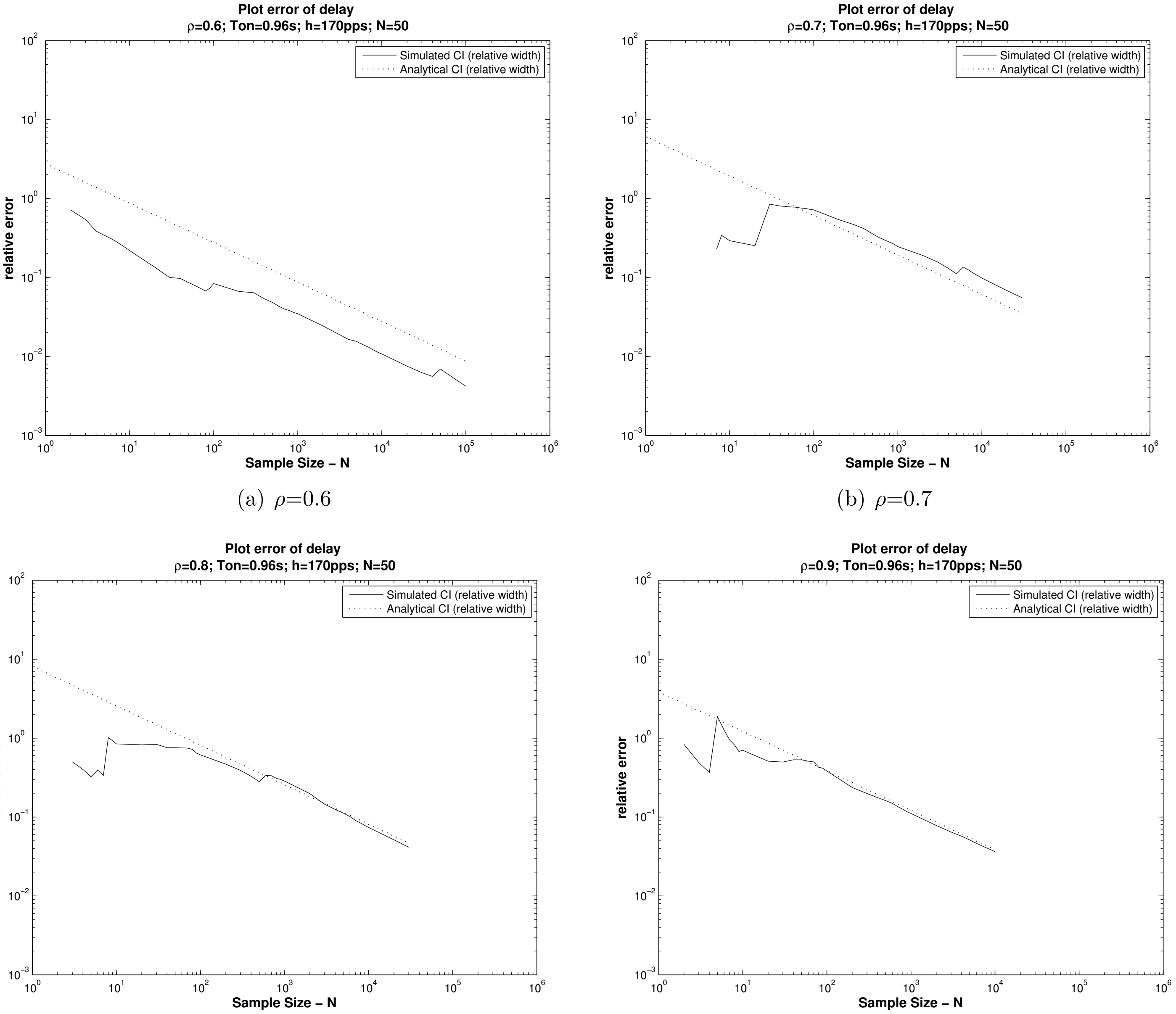
For validation of the single access node, as the utilisation ρ increases, the developed model works better, i.e. the gap between the simulated relative width and analytical relative width is smaller.

The working range of the analytical model is for utilisation to be from 0.6 to 0.9. We are not targeting $\rho < 0.6$ as when load is lower than 0.6, the aggregated traffic is not bursty, therefore looks a lot like Poisson. In ref [89], there are good simulation planning analytical models for classical queue and Poisson arrival process, therefore when the utilisation is lower than 0.6, we can use [89] to plan simulation.

For the end-to-end network, although the model is not as perfect as for the single access, for the high utilisation, e.g. $\rho = 0.9$, it still provides a better guide to simulation planning than existing methods (comparison results are given in Section 4.6.3). Therefore the best working range of our analytical model for the end-to-end network is for utilisation from 0.8 to 0.9, and it overestimate the case when load is 0.6 and 0.7, which is acceptable[5].

[5] Overestimation of SRLASS is always preferred than underestimation. Since if we underestimate SRLASS, the output results will be unreliable. Overestimation will lead to waste time and computer resources. However, if it is within tolerable range, the results are acceptable.

In conclusion, the results are all well validated.



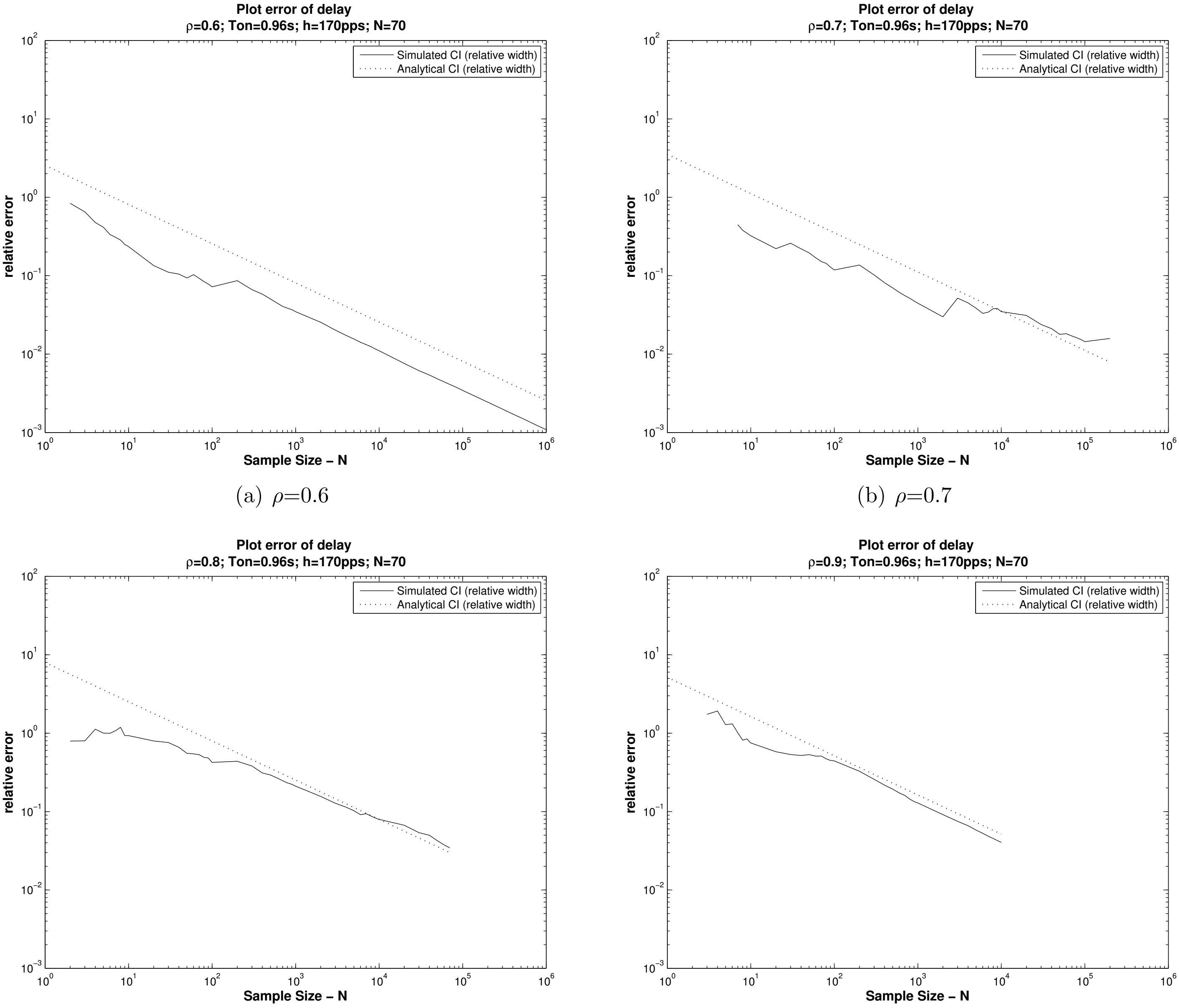
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.3: Delay error v.s. Required Sample Size (N=50)



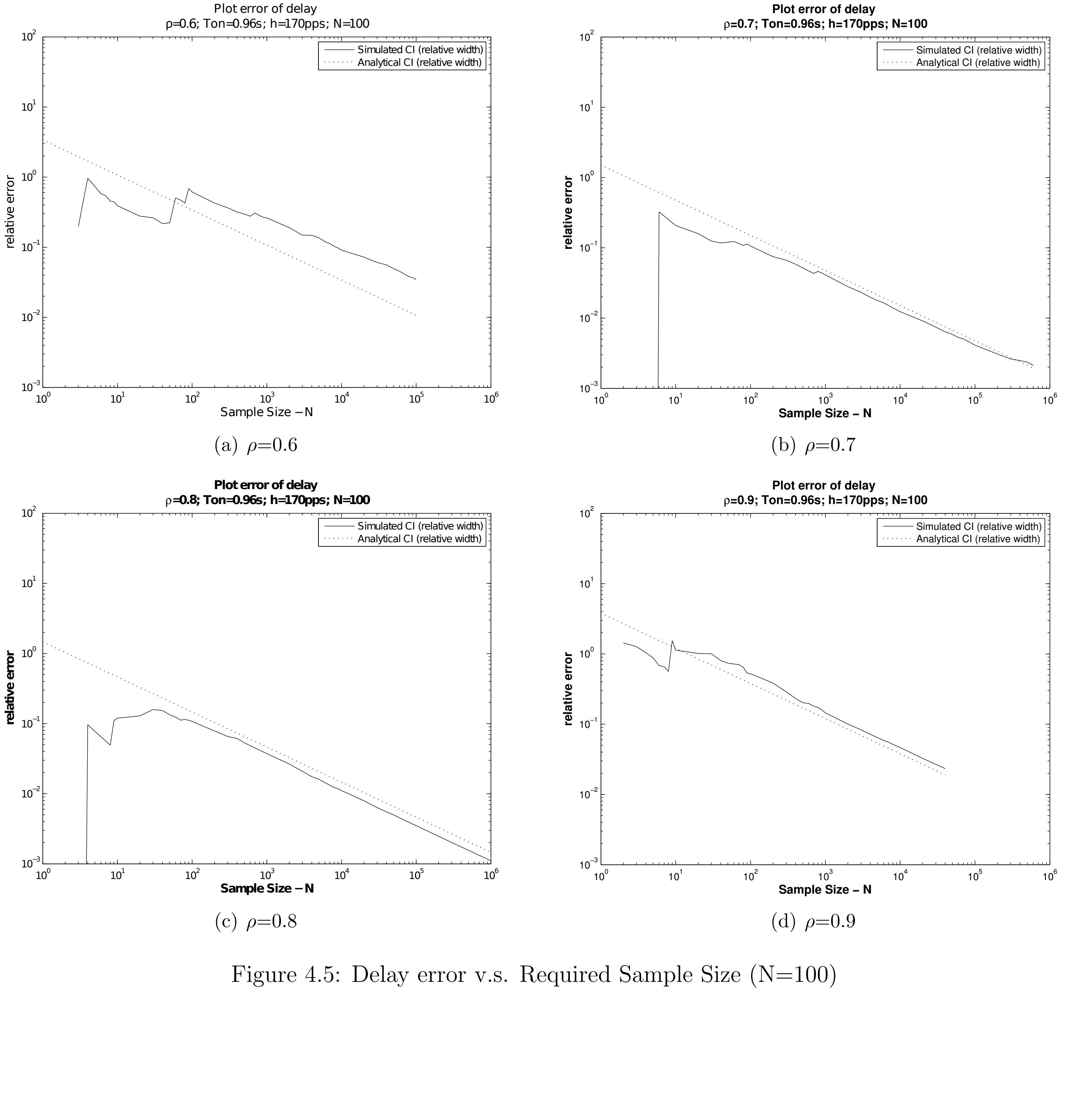
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.4: Delay error v.s. Required Sample Size (N=70)



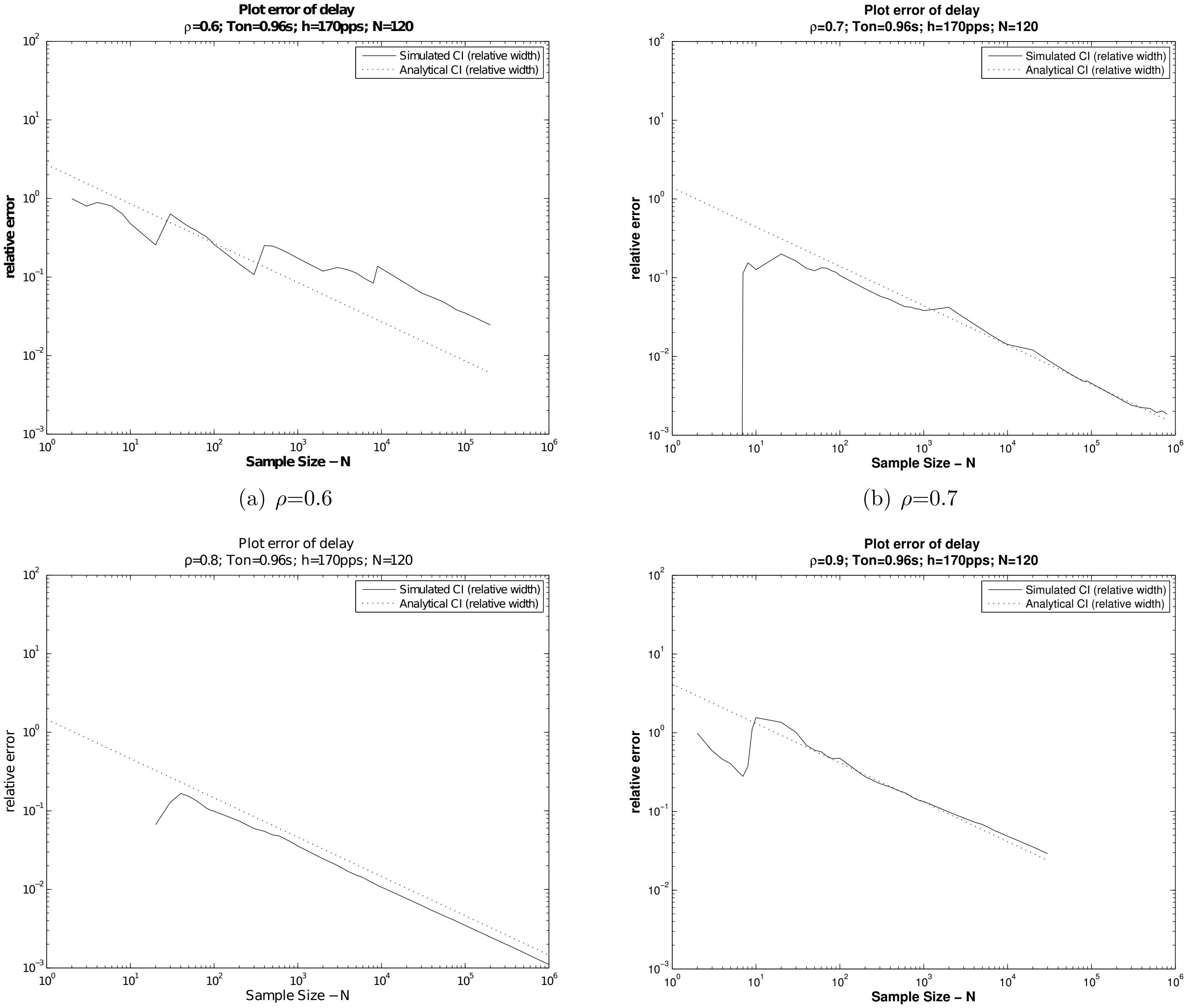
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.5: Delay error v.s. Required Sample Size (N=100)



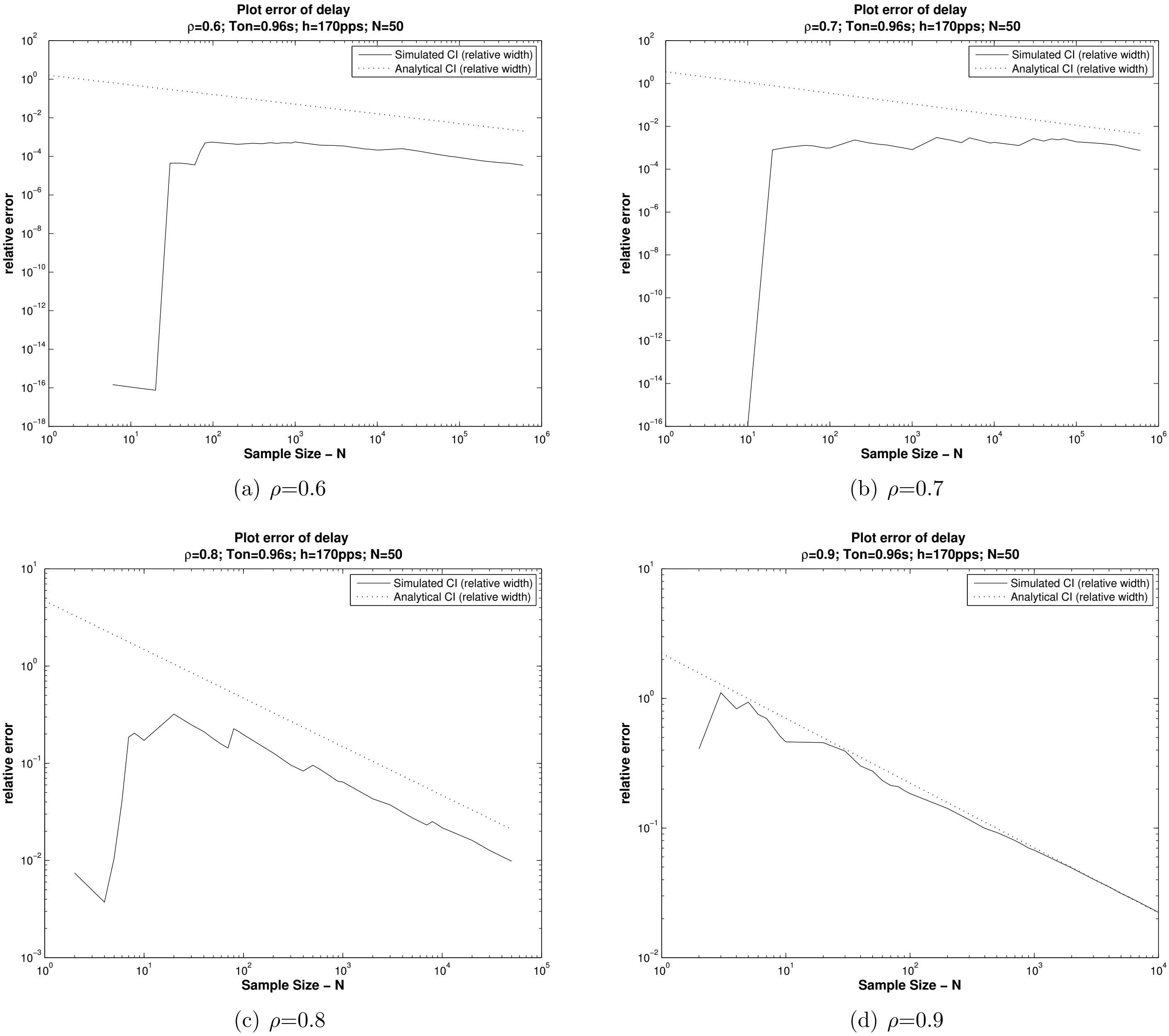
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.6: Delay error v.s. Required Sample Size (N=120)



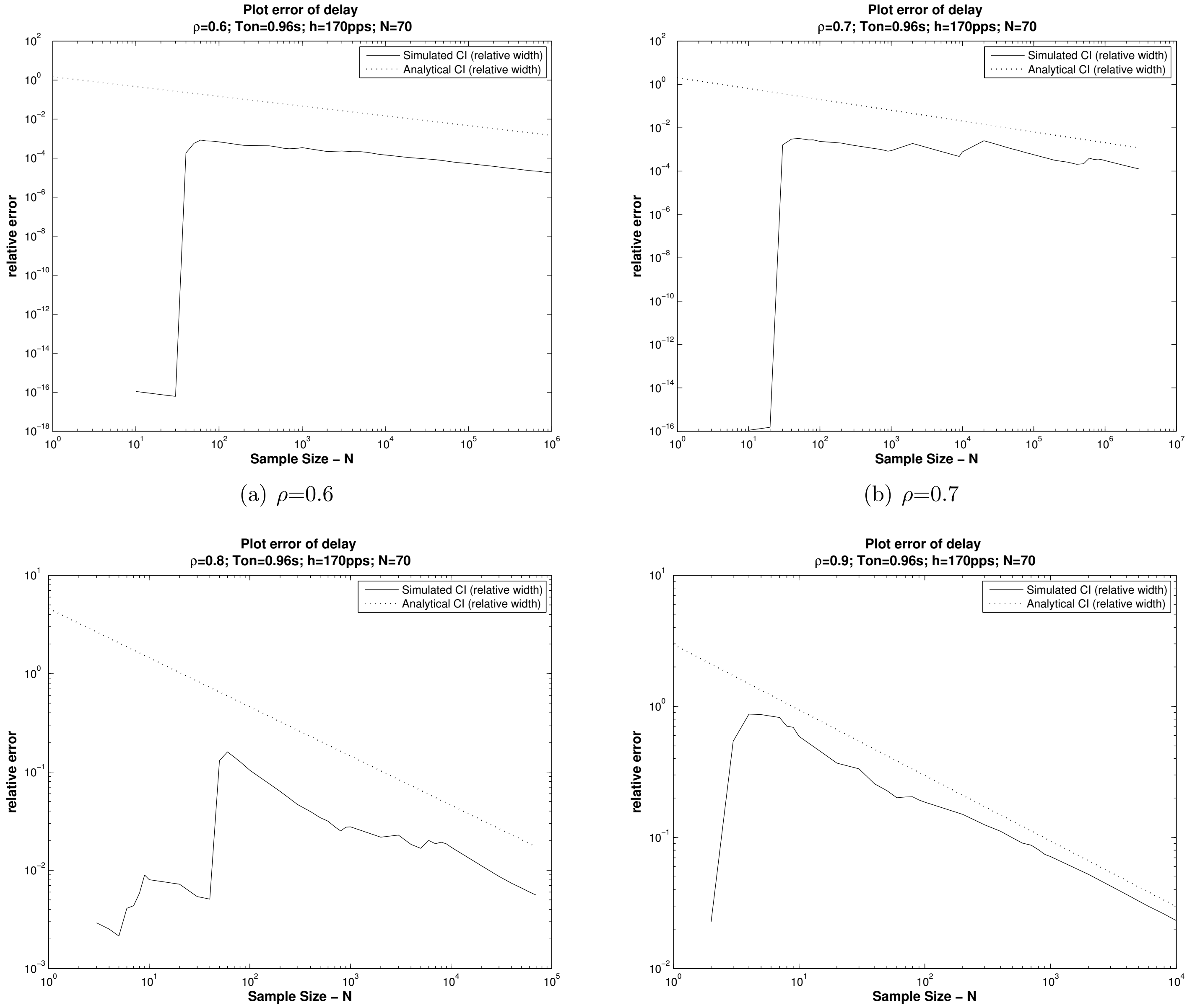
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.7: Delay error v.s. Required Sample Size (node=3; N=50)



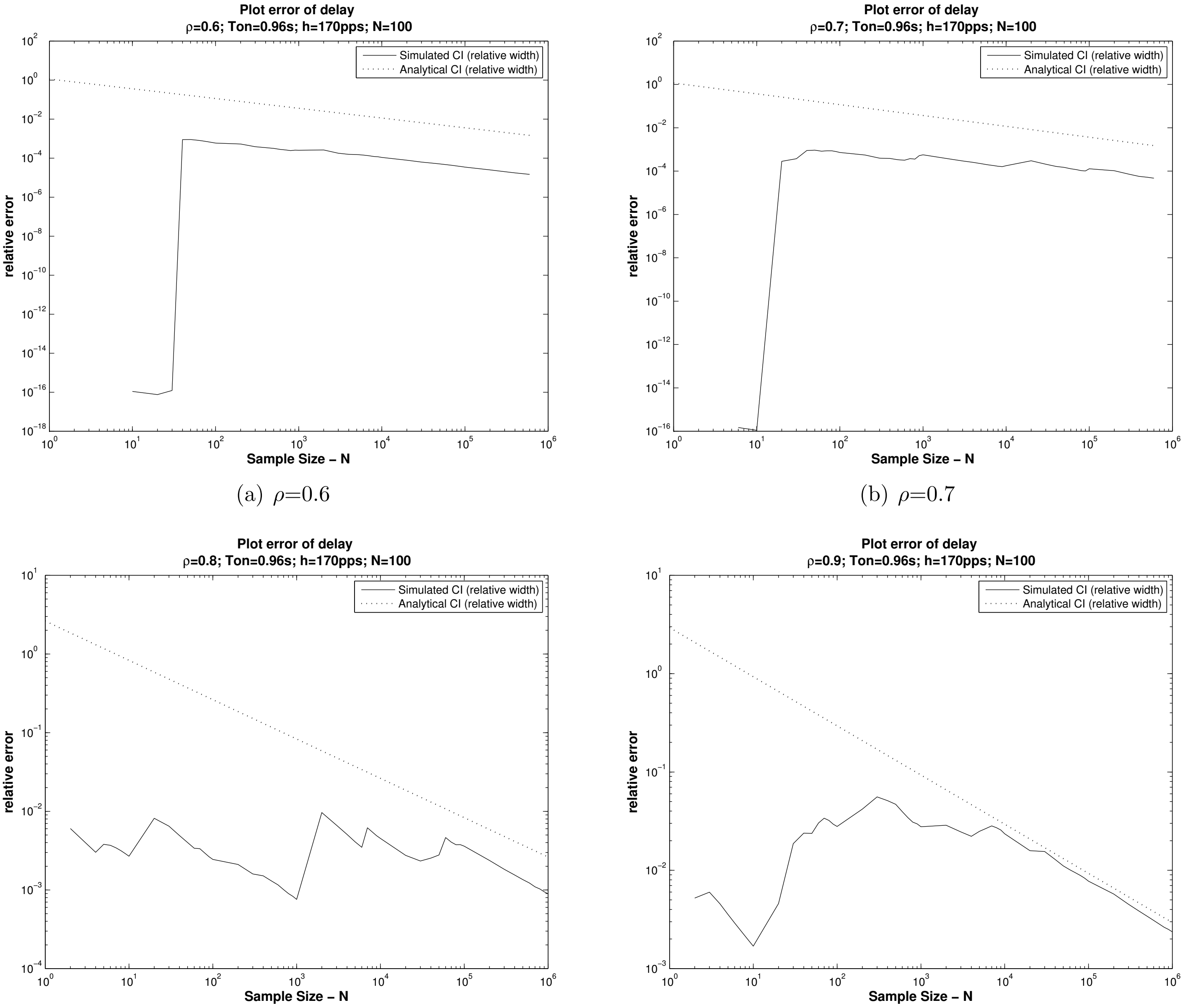
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.8: Delay error v.s. Required Sample Size (node=3; N=70)



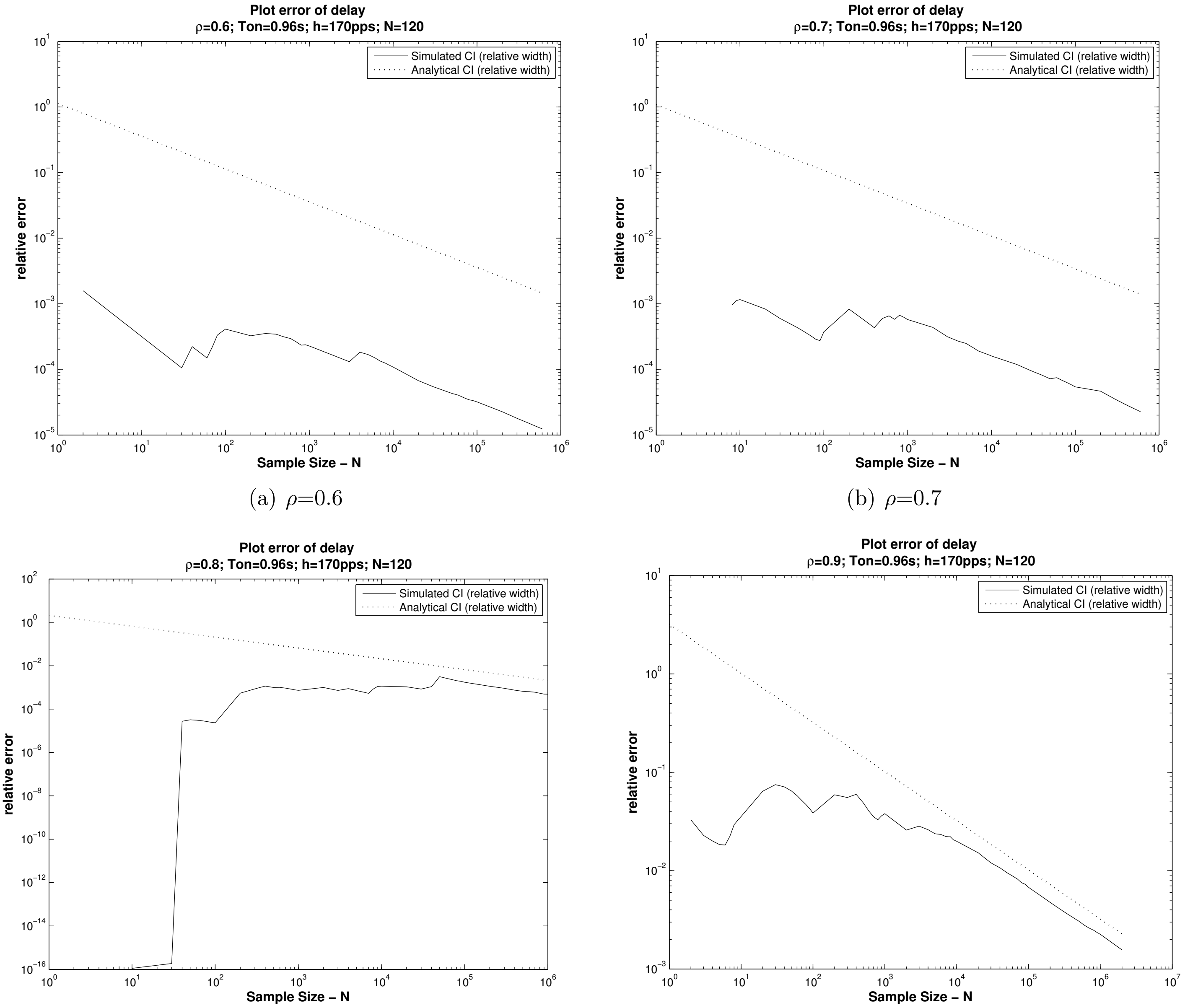
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.9: Delay error v.s. Required Sample Size (node=3; N=100)



(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 4.10: Delay error v.s. Required Sample Size node3 (node=3; N=120)

4.6.3 Results and Comparison with Whitt's Work

Results

In this section, results are shown by plotting the SRLASS period ΔT, in the units of simulation time against different utilisation $\rho = 0.6, 0.7, 0.8$ and 0.9 by changing the number of sources among $N = 50, 70, 100$, and 150 to acheive the required utilisation.

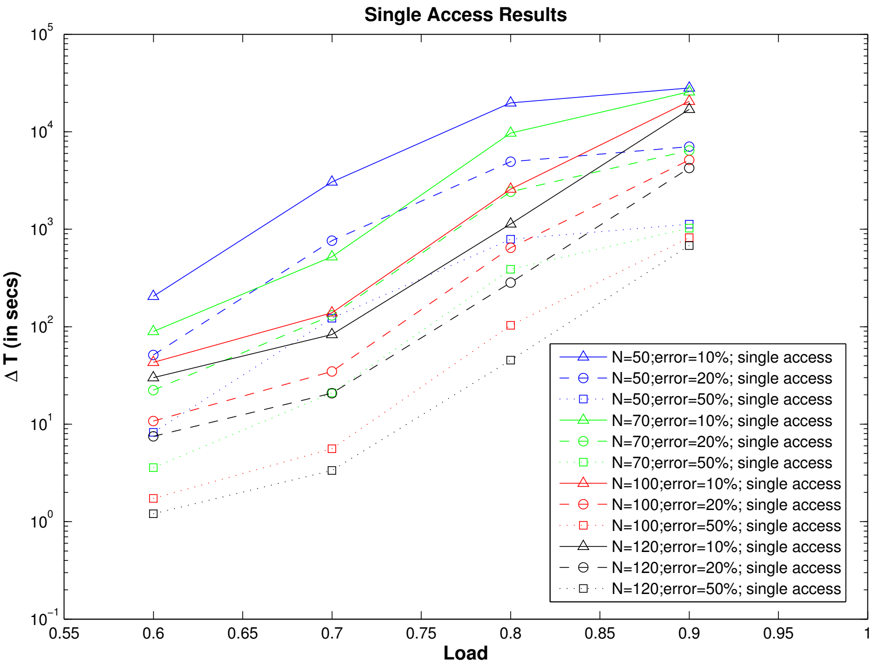


Figure 4.11: Plotting simulation time with different parameters

As shown in Figure 4.11, as the utilisation increases, the required SRLASS period for the delay to reach steady state also increases. With the increase of the number of sources, the required SRLASS period decreases generally, as shown in Figure 4.11. This is because more sources means larger service rate 'C', which leads to less net burstiness, so the traffic looks more Poisson ('Poisson Limit'). Larger service rate with more sources will make the process less variable, as the results show.

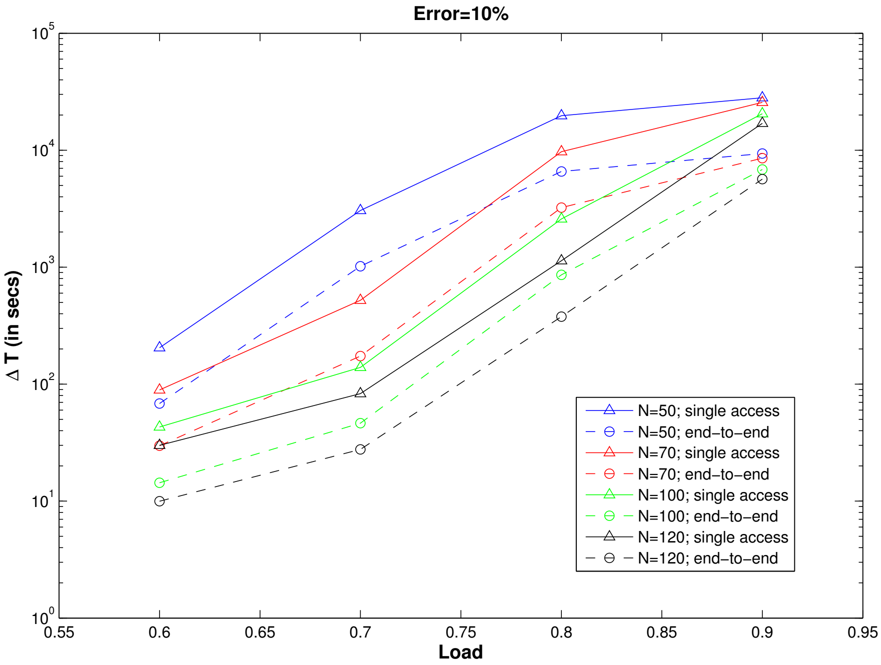


Figure 4.12: Comparison of simulation time between single access and end-to-end network

Figure 4.12 compares SRLASS results between single access and an end-to-end network. As shown in Figure 4.12, it takes less time for the end-to-end network to reach steady state than for single access node. We look back at Equation 4.15 and Equation 4.17, both mean delay and variance of delay increases for an end-to-end network. However, when calculating SCV of delay, mean delay is squared, while variance kept the same. As a result, SCV of delay decreases, which leads to a smaller SRLASS being required.

Comparison Results with Previous Work

In this section, the results of [89] and our analytical model are compared, as shown in Figure 4.13 (Varying Number of sources N) with parameters given in Table 4.3 and Figure 4.14 (Varying traffic parameters, as also used in [3]) with parameters given in Table 4.4.

N	T_{on}	T_{off}	**h**	**error**
50	0.96s	1.69s	170pps	10%
70	0.96s	1.69s	170pps	10%
100	0.96s	1.69s	170pps	10%
120	0.96s	1.69s	170pps	10%

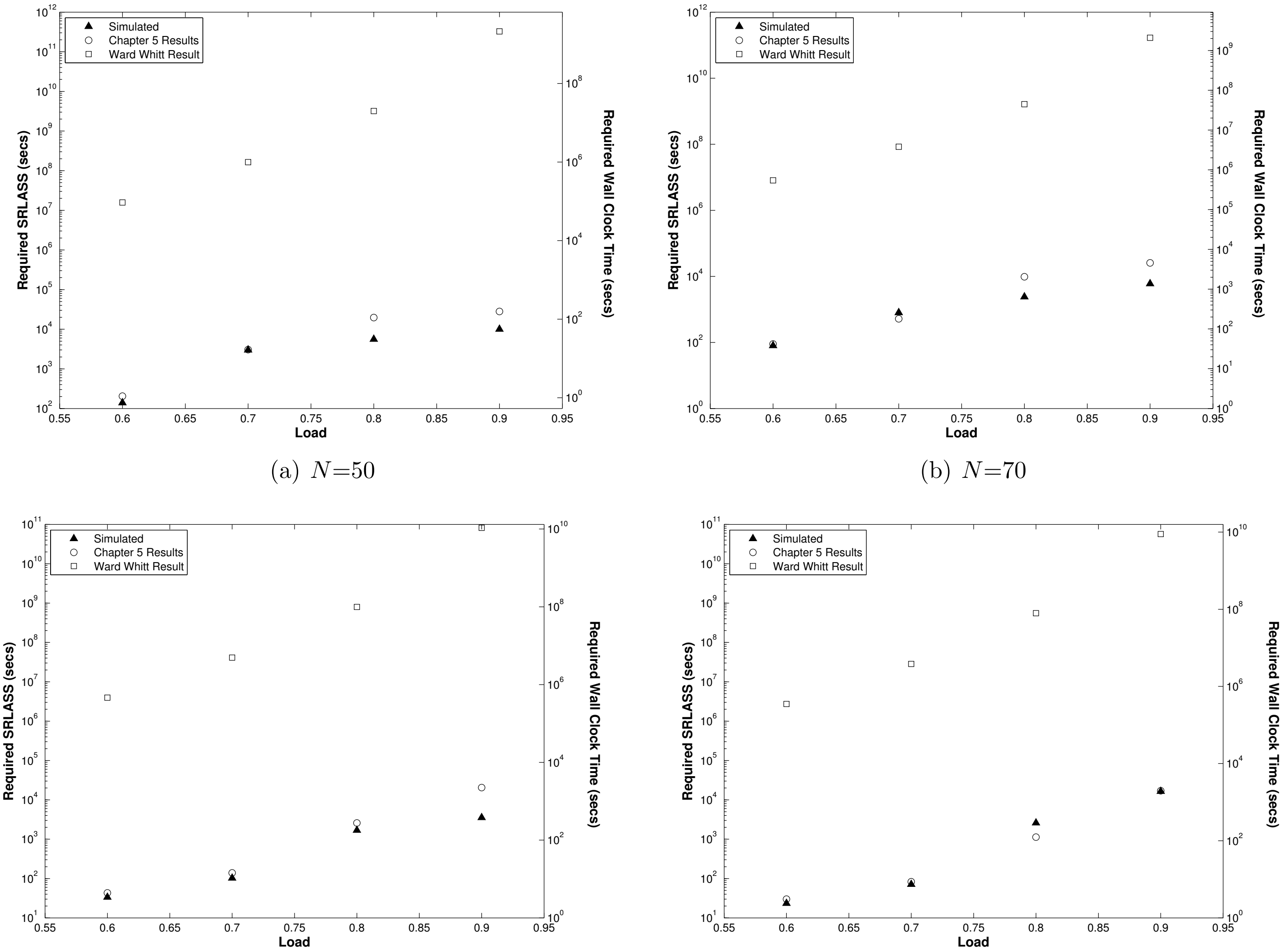
Table 4.3: Set-up Parameters for Markovian Source - Varying N

N	T_{on}	T_{off}	**h**	**error**
100	0.96s	1.69s	170pps	10%
100	2.4s	5.39s	200pps	10%
100	4.8s	12.35s	220pps	10%
100	9.6s	16.9s	250pps	10%

Table 4.4: Set-up Parameters for Markovian Source - Varying T_{on}

The simulation results (methods of obtaining simulated results are shown in Appendix D) are represented by solid triangles, while the results from Chapter 4 is plotted using circles. Whitt's results are plotted in squares, as shown in Figure 4.13 and Figure 4.14.

Figure 4.13 and Figure 4.14 both illustrates that, for the whole group of parameters, Whitt's formula overestimates the required SRLASS, while our results are much closer to the simulated one. The is mainly because Whitt's method is a general method, which approximates the queue behaviour to a RMB processes. And our approach uses the analytical modelling especially focused on the PMM.



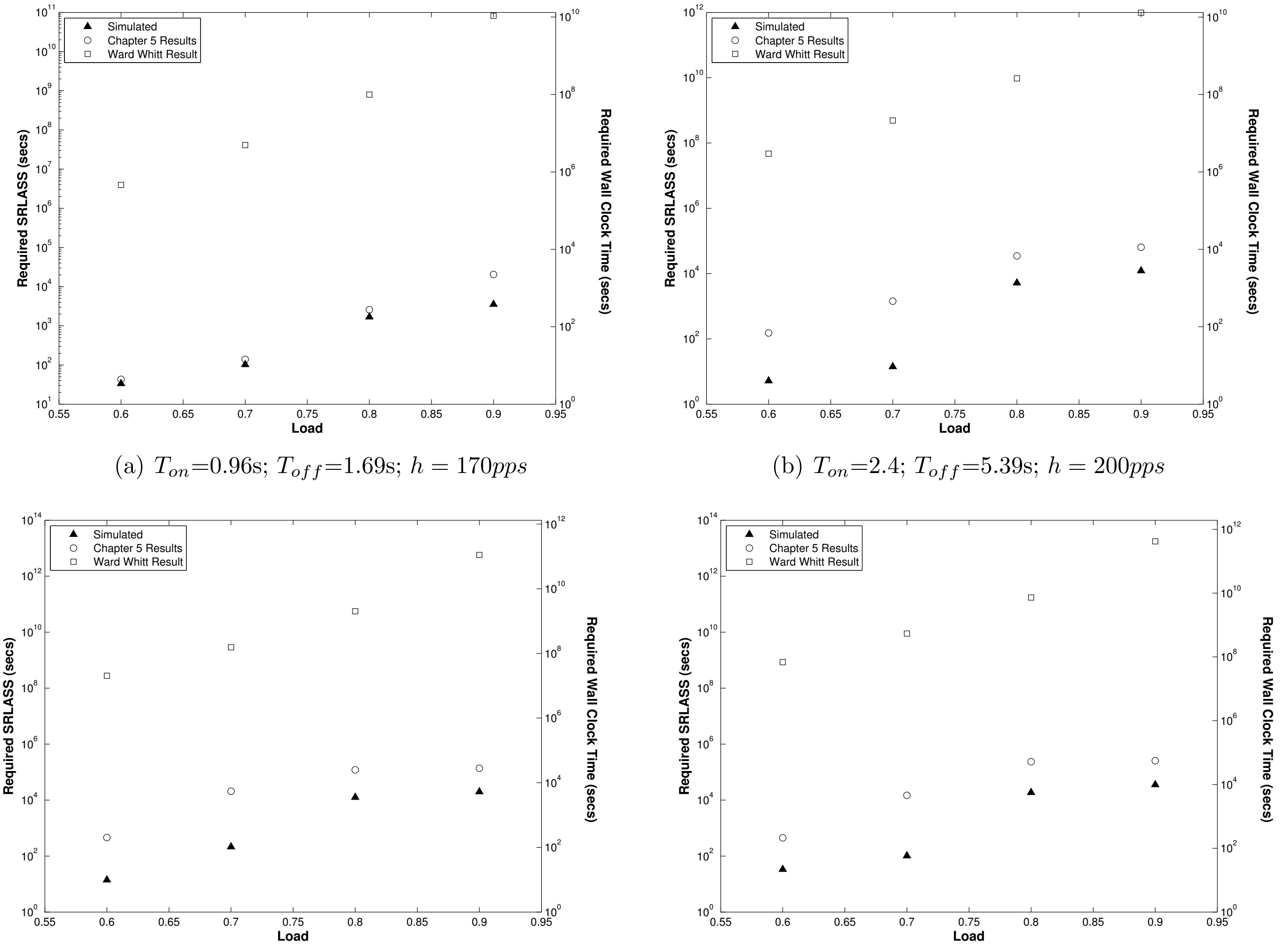
(a) N=50

(b) N=70

(c) N=100

(d) N=120

Figure 4.13: Comparison of Simulated and Analytical SRLASS - Varying N



(a) T_{on}=0.96s; T_{off}=1.69s; $h = 170pps$

(b) T_{on}=2.4; T_{off}=5.39s; $h = 200pps$

(c) T_{on}=4.8; T_{off}=12.35s; $h = 220pps$

(d) T_{on}=9.6; T_{off}=16.9s; $h = 250pps$

Figure 4.14: Comparison of Simulated and Analytical SRLASS - Varying T_{on}

4.7 Conclusion

In this chapter, run length simulation planning is done for delay in Multiple Markovian ON/OFF multiplexing model, using analytical modelling. Results show that the developed analytical model can be used to predicted the required run length, as a guide when to stop the simulation. Run length simulation planning is very model dependent. SCV can be used as a standard estimator for any model to predict the simulation run length.

This chapter also reviews Whitt's work, and applies it to the PMM used in this research. In this case, the required run length of waiting time for planning simulation for PMM can be obtained using Whitt's formula. This is further compared with our formula.

Our results are much closer to the simulated result, as shown in Section 4.6.3 with numerical results comparison.

Chapter 5

Planning Simulation of PLP in the Markovian Source PMM

This Chapter plans simulation of PLP for the PMM, which is consistent with the delay simulation planning done in Chapter 4. Previous work by Ward Whitt on simulation planning [89] concentrates on queue length or mean waiting time, and there is no research about PLP simulation planning. Since there is no direct SCV model for PLP, this chapter proposes to plan simulation of PLP by how many cycles for the PLP to reach the steady state. The cycle concept is based on the Overflow/Non-Overflow (OvFl/NOF) cycle analysis of the PMM (to be introduced in section 5.2). Finally, validation and results are given in this chapter.

5.1 Overview of Simulation Plan in PMM

As introduced in section 3.3, the PMM multiplexes multiple packet sources (Markovian sources in this chapter), into one finite buffer. All the multiplexed sources are identical and independent of each other. In this chapter, Packet Loss Probability (PLP), as the parameter of interest, is the key parameter. This chapter proposes to plan simulation for PLP in PMM. In realistic networks, the PLP is targeted in the magnitude of 10^{-4} [64][58] (as a value widely set in Service-Level Agreements (SLAs) [35] [27]), and is viewed in

OvFl/NOF cycles. The OvFl/NOF cycle is introduced and defined later in section 5.2. Simulation planning for PLP in PMM is solved by proposing how many cycles are required for the PLP to reach steady state. Therefore, the sample size is in the units of cycle number, when applying standard analysis to PMM scenario. Number of cycles is then translated to number of packet arrivals and in Chapter 7 into wall clock time.

5.2 Overflow Analysis of PMM

The basic PMM introduced in section 3.3, as shown in Figure 3.9, can be simplified to a single aggregate OvFl/NOF process, as first used in [37]. This is as shown in Figure 5.1: the aggregate process is either in the overflow (OvFl) state, buffer is overflowing, and packet losses occur, or in the non-overflow (NOF) state, where there are no packet losses.

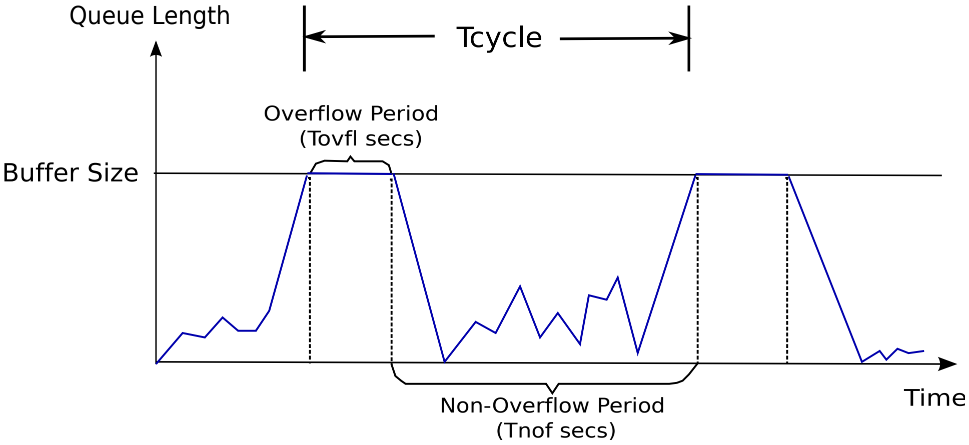


Figure 5.1: Overflow/ non-oveflow analysis

The mean duration of an OvFl period is denoted as T_{ovfl}, while the duration of a NOF period is denoted as T_{nof}. The duration of OvFl periods is modelled as an exponential distribution, which has successfully been shown to be accurate for Markovian traffic in [36]. This thesis employs the same idea for T_{ovfl} and T_{nof}, as well as T_{cycle}, which is validated in Section 5.4.2.

One OvFl period, followed by another NOF period is called one cycle, the mean time for which is denoted T_{cycle}. It is intuitive that

$$T_{cycle} = T_{ovfl} + T_{nof}. \tag{5.1}$$

5.2.1 Parameterisation

In this section, related variables of the overflow analysis are parameterised through relating to an aggregate ON/OFF model, since this aggregated ON/OFF analysis has an existing, well-developed analytical model.

5.2.2 An Introduction to the Aggregate ON/OFF Model

In many papers [61] [9] [34], the packet multiplexing model is simplified into an aggregate ON/OFF model, reducing the number of possible states from 2^N to just 2 states. As shown in Figure 5.2, the aggregate process is either in the ON state, when the overall input rate exceeds the service rate, C, or in the OFF state when the overall input rate is less than the service rate, but not normally zero.

In the ON state, the overall mean rate is denoted as R_{on}, and the expected time spent in the ON state denoted as $T(on)$. Since the overall input rate is larger than service rate C, the buffer will fill in the rate of $R_{on} - C$.

Similarly, in the OFF state, the overall mean rate is denoted as R_{off}, and the expected time spent in the OFF state denoted as $T(off)$. In the OFF state, the queue length decreased at a rate of $C - R_{off}$.

From [72], the analytical models of these parameters[1] are given by

$$R_{on} = C + h \cdot \frac{A_p}{C - A_p} \tag{5.2}$$

[1] h is the sending rate of a single ON/OFF traffic source, when the source is in the 'ON' state; A_p is the aggregate overall arrival rate of the whole access node/queue. For for information about PMM, please refer to Section 3.3.

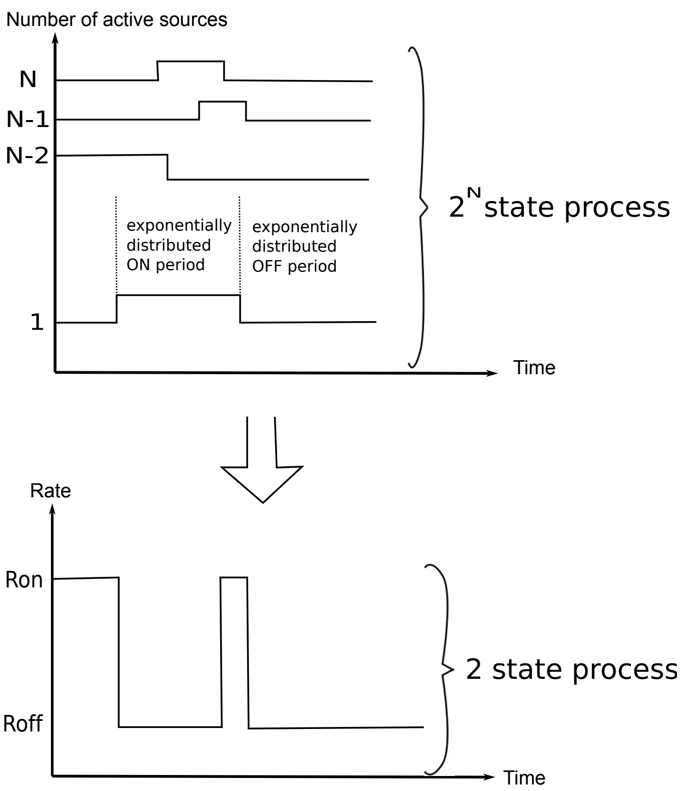


Figure 5.2: Aggregated ON/OFF Model for the Packet Multiplexing Model [72]

$$T(on) = \frac{h \cdot T_{on}}{C - A_p} \tag{5.3}$$

$T(on)$ can be used as an approximation for the expected OvFl period, T_{ovfl}[36], given by

$$T_{ovfl} \approx T(on) \tag{5.4}$$

5.2.3 Analytical Models for Overflow Analysis

Let $N_{p/cycle}$ be the mean number of packets in one cycle, and μ_p be the expected number of packets lost per OvFl period. The PLP can be obtained from

$$PLP = \frac{\mu_p}{N_{p/cycle}} \tag{5.5}$$

The expected number of packet lost per OvFl period, μ_p, can be obtained from the time period T_{ovfl} multiplied by the loss rate. Since when the system is in the OvFl state, it's necessarily in the ON state (overall input rate exceeds the service rate), the packet loss rate is given by $R_{on} - C$. Thus μ_p is

$$\mu_p = (R_{on} - C) \cdot T_{ovfl} \approx (R_{on} - C) \cdot T(on) \tag{5.6}$$

It is intuitive that

$$T_{cycle} = \frac{N_{p/cycle}}{A_p} \tag{5.7}$$

since the mean arrival rate times the cycle time give the total number of packet arrivals in one cycle.

Therefore,

$$T_{cycle} = \frac{\mu_p}{A_p \cdot PLP} = \frac{(R_{on} - C) \cdot T(on)}{A_p \cdot PLP} \tag{5.8}$$

Taking Equation (5.2) and (5.3) into Equation (5.8), T_{cycle} is given by

$$T_{cycle} = \frac{h^2 \cdot T_{on}}{PLP \cdot C^2 \cdot (1 - \rho)^2} \tag{5.9}$$

5.3 Packet Loss Probability in the Overflow Analysis

Since the objective of this research is to find the SRLASS, ΔT, for the PLP to reach steady state, the PLP should be applied to the discrete-time process analysis. This research proposes to address this: how many OvFl/NOF cycles required, denoted as N_{cycle}, for the PLP to reach steady state? In this way, the raw sample size is viewed in the units of number of cycles.

In order to get N_{cycle}, the SCV c^2 of the PLP is needed. However, it is still hard to find the analytical model of the variance of the PLP since $c^2 = \sigma_2/PLP$. Therefore, this research proposes to find the c^2 of μ_p and T_{cycle}, since they are related according to (as already discussed in Section 5.2.3)

$$PLP = \frac{\mu_p}{N_{p/cycle}} = \frac{\mu_p}{A_p \cdot T_{cycle}} \tag{5.10}$$

based on the existing distribution type of μ_p and T_{cycle}, as well as the constant value of A_p for each group of parameters. In other words, the PLP is regarded as reaching steady state when both μ_p and T_{cycle} reach steady state. Therefore, N_{cycle} is achieved by finding the maximum value of N_{μ_p} and N_{Tcycle}, given by

$$N_{cycle} = max\ (N_{\mu_p}, N_{Tcycle}) \tag{5.11}$$

where N_{μ_p} is the sample size required for μ_p to reach steady state, and N_{Tcycle} for T_{cycle}. As a result, SRLASS, ΔT , is obtained by

$$\Delta T = N_{cycle} \times T_{cycle}. \tag{5.12}$$

Therefore, the objective problem is to find the required sample size for N_{cycle} (to be discussed in section 5.3.1) and the analytical model of variance for T_{cycle} and μ_p (to be discussed in section 5.3.2).

5.3.1 Sample Size Analysis

In this section, statistical analysis (as already discussed in Section 2.3) is applied to find the required sample size, in the units of cycle number.

Recall from Section 2.3, assume the expected packets lost per OvFl period μ_p and cycle time T_{cycle} collected from each cycle are the samples of a population $\{X_i\}$, measured from the simulator for $i = 1, 2, \ldots, N$.

As discussed in Section 2.3.3, relative width is used in this research. Define relative width to be

$$\varepsilon = 2 \cdot z_{1-\beta/2} \cdot \frac{\sigma}{\bar{X}\sqrt{N}} \tag{5.13}$$

Thus, for specified relative width ε and specified precision level of β, the required sample size, N, is given by

$$N = \frac{4\sigma^2 z^2_{1-\beta/2}}{\varepsilon^2(\bar{X})^2} = 4 \cdot \frac{z^2_{1-\beta/2}}{\varepsilon^2} \cdot c^2 \tag{5.14}$$

Applying μ_p and T_{cycle} to this analysis, the required sample sizes are given by

$$N_{\mu_p} = 4 \cdot \frac{z^2_{1-\beta/2}}{\varepsilon^2} \cdot c^2(\mu_p) \tag{5.15}$$

$$N_{Tcycle} = 4 \cdot \frac{z^2_{1-\beta/2}}{\varepsilon^2} \cdot c^2(T_{cycle}) \tag{5.16}$$

5.3.2 SCV Model of μ_p and T_{cycle}

The OvFl and NOF periods are modelled as approximately an exponential distribution, as already discussed in Section 5.2. Furthermore, the expected packets lost per OvFl period are regarded as following a Geometric distribution, an approximation which has been used successfully [37][35], and further tested in Section 5.4. Further validation of this is illustrated in numerical examples, as shown in Figure 5.4 - 5.7.

In fact, the expected duration of NOF period, T_{nof}, is much larger than the expected duration of OvFl period, T_{ovfl}[35]:

$$T_{nof} >> T_{ovfl},$$

therefore, the effect of T_{ovfl} on the distribution of T_{cycle} is negligible. A more substantial evaluation of this is given in Table 5.1, where at e.g. 60% load, the mean duration of NOF period, T_{nof} is 850 times the mean duration of

OvFl period, T_{ovfl}.

Load	T_{ovfl}	T_{nof}
0.1	0.036266667	13.70074074
0.2	0.0408	17.34
0.3	0.046628571	22.64816327
0.4	0.0544	30.82666667
0.5	0.06528	44.3904
0.6	0.0816	69.36
0.7	0.1088	123.3066667
0.8	0.1632	277.44
0.9	0.3264	1109.76

Table 5.1: Calculation and comparison between T_{ovfl} and T_{nof}

Because of this, the assumption is made that the cycle time, T_{cycle}, is exponentially distributed. Further detailed validation will be presented in numerical examples in section 5.4, as shown in Figure 5.4 and 5.6.

Since for any Geometric distribution:

$$\text{Mean of the Geometric distribution} = \frac{1}{p}$$

$$\text{Variance of the Geometric distribution} = \frac{1-p}{p^2}$$

where p is a parameter of the Geometric distribution.

Similarly, for any Exponential distribution:

$$\text{Mean of the Exponential distribution} = \frac{1}{\mu}$$

$$\text{Variance of the Exponential distribution} = \frac{1}{\mu^2}$$

where μ is the parameter of an Exponential distribution.

Therefore,

$$c^2(\mu_p) = Var(\mu_p)/\mu_p^2 = (\mu_p^2 - \mu_p)/\mu_p^2 = 1 - 1/\mu_p \tag{5.17}$$

and

$$c^2(T_{cycle}) = Var(T_{cycle})/T_{cycle}^2 = T_{cycle}^2/T_{cycle}^2 = 1 \tag{5.18}$$

Take them into Equation (5.15) and (5.16), N_{μ_p} and N_{Tcycle} can be written as:

$$N_{\mu_p} = 4z_{\beta/2}^2 \cdot \frac{\mu_p^2 - \mu_p}{(\varepsilon \cdot \mu_p)^2} = \frac{4z_{\beta/2}^2}{\varepsilon^2} \cdot (1 - \frac{1}{\mu_p}) \tag{5.19}$$

$$N_{Tcycle} = 4z_{\beta/2}^2 \cdot \frac{T_{cycle}^2}{(\varepsilon \cdot T_{cycle})^2} = \frac{4z_{\beta/2}^2}{\varepsilon^2} \tag{5.20}$$

N_{Tcycle} will always be much larger than N_{μ_p}, therefore, the approximation

$$N_{cycle} \approx N_{Tcycle}$$

is used in this analysis.

Based on the above analysis, SRLASS is given by

$$\Delta T = N_{cycle} \times T_{cycle} \approx N_{Tcycle} \times T_{cycle} = \frac{4z_{\beta/2}^2}{\varepsilon^2} \cdot \frac{h^2 \cdot T_{on}}{PLP \cdot C^2 \cdot (1-\rho)^2} \tag{5.21}$$

which is the required analytical model for SRLASS, with a preset PLP and the desired precision level.

5.3.3 Extending the Analysis to an End-to-End Network

The previous sections aim to predict SRLASS, ΔT, for the PLP based on a packet multiplexing model for a single access link. This section extends the analysis to an end-to-end network. Unlike the single access link, the overall PLP for an end-to-end network is governed by multiple buffers throughout the network. The PLP at each buffer across the entire link is called the individual PLP, which contributes to the overall PLP. In this research, all the individual buffers are assumed to be identical [35], as already introduced in Section 3.4.1.

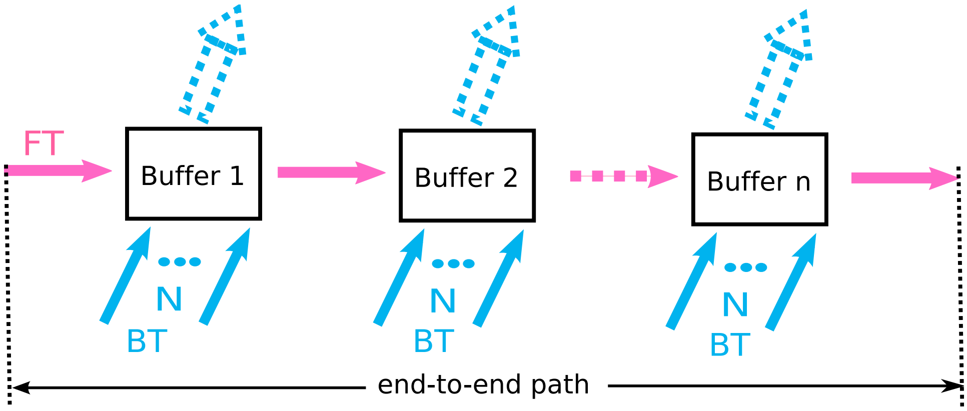


Figure 5.3: End-to-end FG/BG network model

As discussed in section 3.4.1, this research employs the FG/BG end-to-end network model. In Figure 5.3, FT is the traffic flow of interest, which is injected into the network, and passed through every buffer in series. BT flows are all from independent traffic sources, which are multiplexed with the FT at each buffer, and routed elsewhere in the network. The FT flow traverses n identical buffers throughout the network. In order to differentiate the PLP for every individual buffer from the overall PLP, we denote them as $IPLP$ and $TPLP$, respectively. Therefore,

$$TPLP = 1 - \prod_{i=1}^{n}(1 - IPLP_i)$$

when the nodes are independent of each other [25] [26] [51] [23] [35]. For small and identical $IPLP_i$, we can use the approximation that

$$1 - \prod_{i=1}^{n}(1 - IPLP_i) \approx \sum_{i=1}^{n} IPIP_i \approx n \cdot IPLP,$$

therefore, the relationship between $TPLP$ and $IPLP$ is shown as

$$TPLP \approx n \cdot IPLP \tag{5.22}$$

Applying this to cycle time T_{cycle}, we get

$$T_{cycle} = \frac{h^2 \cdot T_{on} \cdot n}{TPLP \cdot C^2 \cdot (1-\rho)^2} \tag{5.23}$$

In this case, the SRLASS, ΔT_{e2e}, for an end-to-end network is given by

$$\Delta T_{e2e} = \frac{4z^2_{\beta/2}}{\varepsilon^2} \cdot \frac{h^2 \cdot T_{on} \cdot n}{TPLP \cdot C^2 \cdot (1-\rho)^2} \tag{5.24}$$

5.4 Numerical Examples

5.4.1 Simulation Set-up Parameters

In this section, we provide the validation results, and evaluate the SRLASS, of the ΔT needed for the PLP to reach steady state.

A standard multiplexing model of N homogenous Markovian ON/OFF VoIP packet sources, is used in this thesis. Some popularly used parameters [88] for voice over packet are:

- T_{on}=0.96s,
- T_{off}=1.69s,
- ON rate h=170packets/s,
- packet size=100 bytes,
- N is adjusted to give different loads. Buffer size is set to target PLP approximately equal to 10^{-4} (as a proper value according to Service-Level Agreements (SLAs) [35] [27]).

To maintain a target PLP to be 10^{-4}, a larger buffer size is needed as traffic utilisation ρ increases. This is due to the fact that the service rate is set constant, where high utilisation ρ makes traffic sources more bursty and requires a longer buffer to keep the same PLP. It is widely accepted that the buffer overflow probability $Q(x)$ for an infinite buffer is an excellent approximation for the PLP. And $Q(x)$ always in the form of

$$Q(x) = P_B \eta^x, \tag{5.25}$$

where P_B is the probability of experiencing burst-scale queuing (with detailed explanation in Appendix A). Equation (5.25) reveals that the PLP and buffer size, x, to be a log-linear relationship [72] [75] [76].

This research employs the analytical model for P_B and η as shown in Equation (5.26) [76] and (5.27) [3]

$$P_B \approx \frac{1}{(1-\rho)^2 \cdot (C/h)} \cdot \frac{(\rho \cdot (C/h))^{\lfloor (C/h) \rfloor}}{\lfloor (C/h) \rfloor !} \cdot e^{-\rho (C/h)} \tag{5.26}$$

$$\eta \rightarrow \frac{1 - [ln(h/C)/ln(\rho) + (h^2 T_{on} \rho)/(C(1-\rho)^2)]^{-1}}{1 - [\rho(1-\rho)^2/(h/C) \cdot T_{on} \cdot [(1-\rho)C + h \cdot \rho]]} \tag{5.27}$$

Equation (5.25) - (5.27) will be used for the calculation of the buffer size for the packet multiplexing model reported in this section, so that buffer length is changed to adjust PLP to be 10^{-4}. Full derivation of these equations are given in Appendix A.

Table 5.2 shows the buffer size and number of sources for different load (varying from 0.6 - 0.9), single access and different service rate 2Mbps and 4Mbps.

C=4Mbps			C=2Mbps		
Load	N	**Buffer Size**	**Load**	N	**Buffer Size**
0.6	49	59.65	0.6	24	219.36
0.7	57	234.67	0.7	28	570.43
0.8	65	834.32	0.8	32	1.78E+03
0.9	73	5.12E+03	0.9	37	1.82E+04

Table 5.2: Parameter Table of Markovian Source for Single Access Link

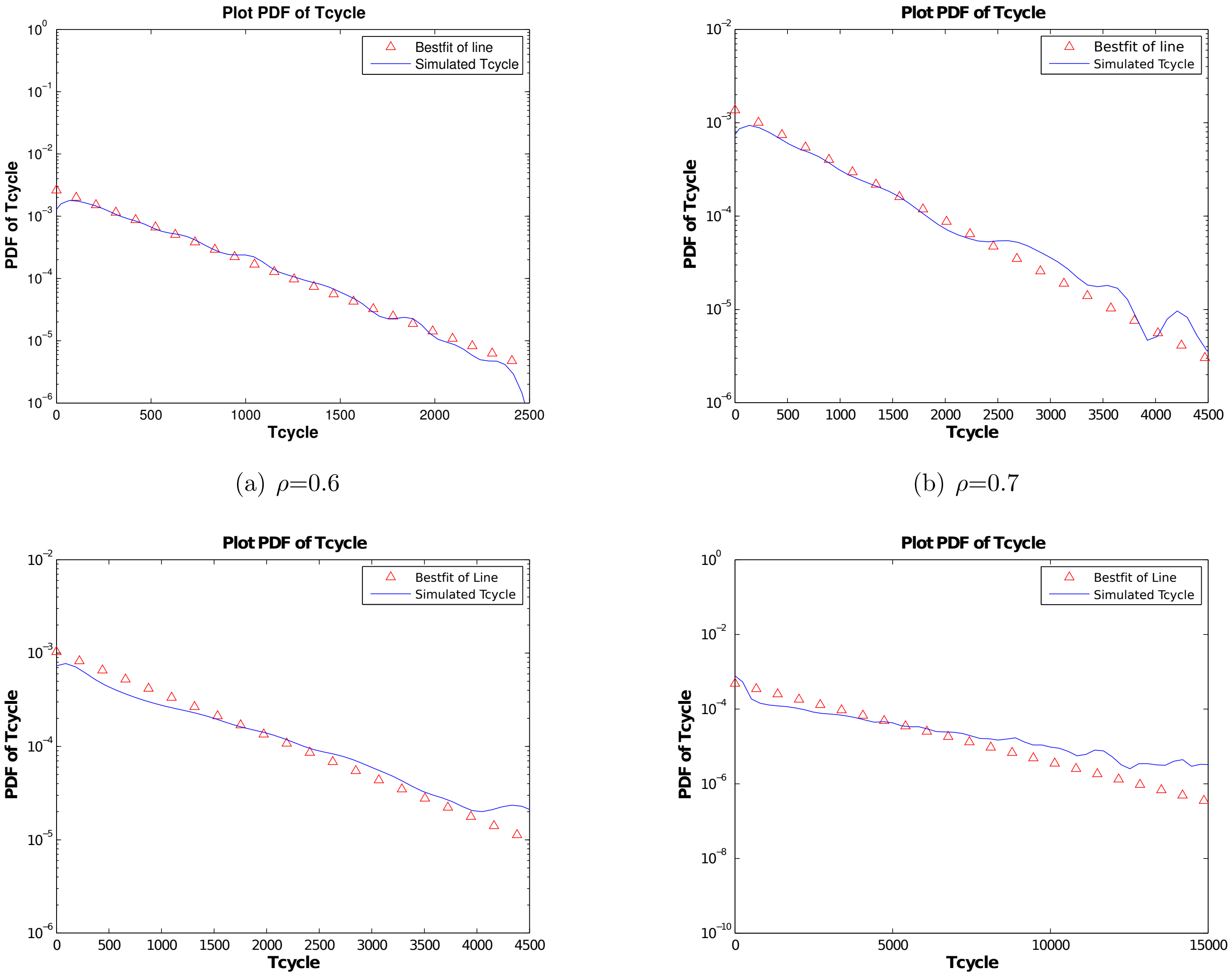
C=4Mbps			C=2Mbps		
Load	N	**Buffer Size**	**Load**	N	**Buffer Size**
0.6	49	94.04	0.6	24	275.55
0.7	57	302.87	0.7	28	685.6
0.8	65	1.01E+03	0.8	32	2.08E+03
0.9	73	5.92E+03	0.9	37	2.06E+04

Table 5.3: Parameter Table of Markovian Source for End-to-end (n=3)

5.4.2 Validation of Distribution of T_{cycle} and μ_p

As already discussed in section 5.3.2, the analysis is based on the assumption that T_{cycle} is exponentially distributed and μ_p is geometrically distributed. Therefore, the validation of these distributions is crucial. Validation is done using parameters from Table 5.2 for T_{cycle} and μ_p. Exponential bestfit is used to validate the distribution, which was discussed in Section 3.1.4 and the detailed algorithm given in Appendix B.

Figure 5.4 - 5.7 are all plotted log-linear, in which a straight line indicates an exponential or geometric distribution. Figure 5.4 shows the distribution of T_{cycle} for different loads from 0.6 to 0.9 when service rate C is 2Mbps by applying bestfit technique to its pdf. The same applies to μ_p in Figure 5.5. Similarly, when the service rate C is 4Mbps, bestfit of T_{cycle} and μ_p distribution are shown in Figure 5.6 and Figure 5.7, respectively.



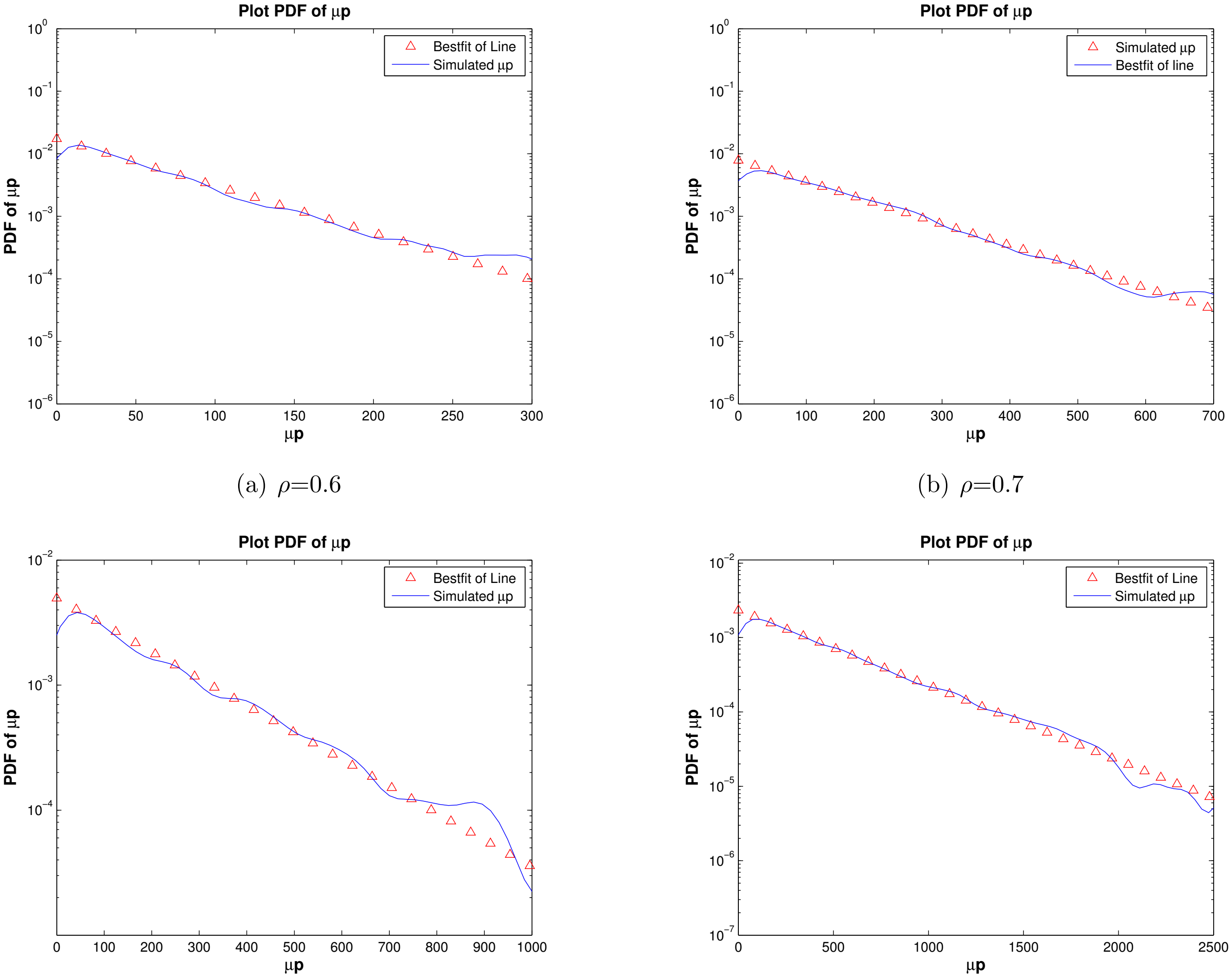
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 5.4: Validate Distribution of Cycle Time, T_{cycle}, with C=2Mbps



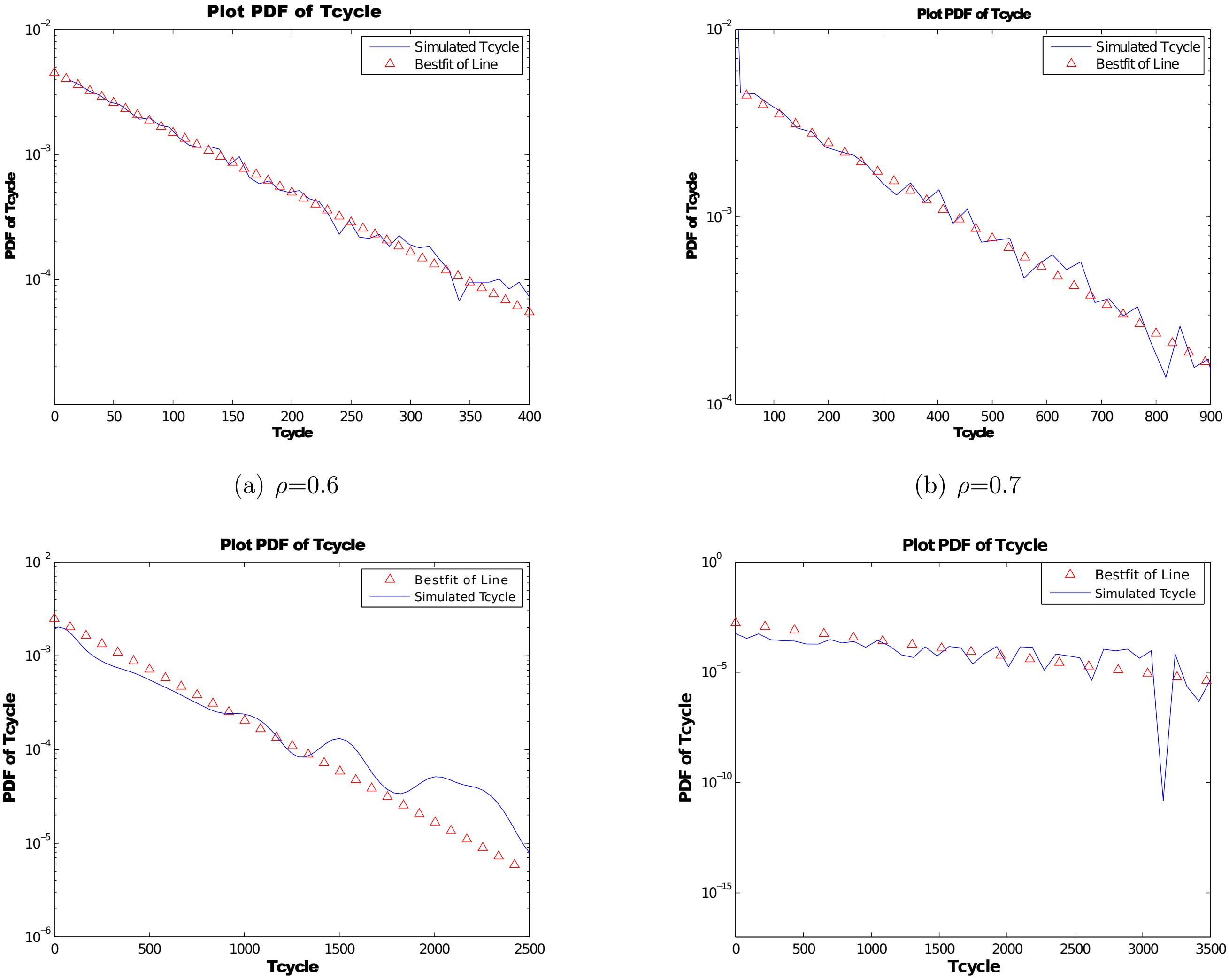
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 5.5: Validate Distribution of expected packets lost per OvFl period, μ_p, with C=2Mbps



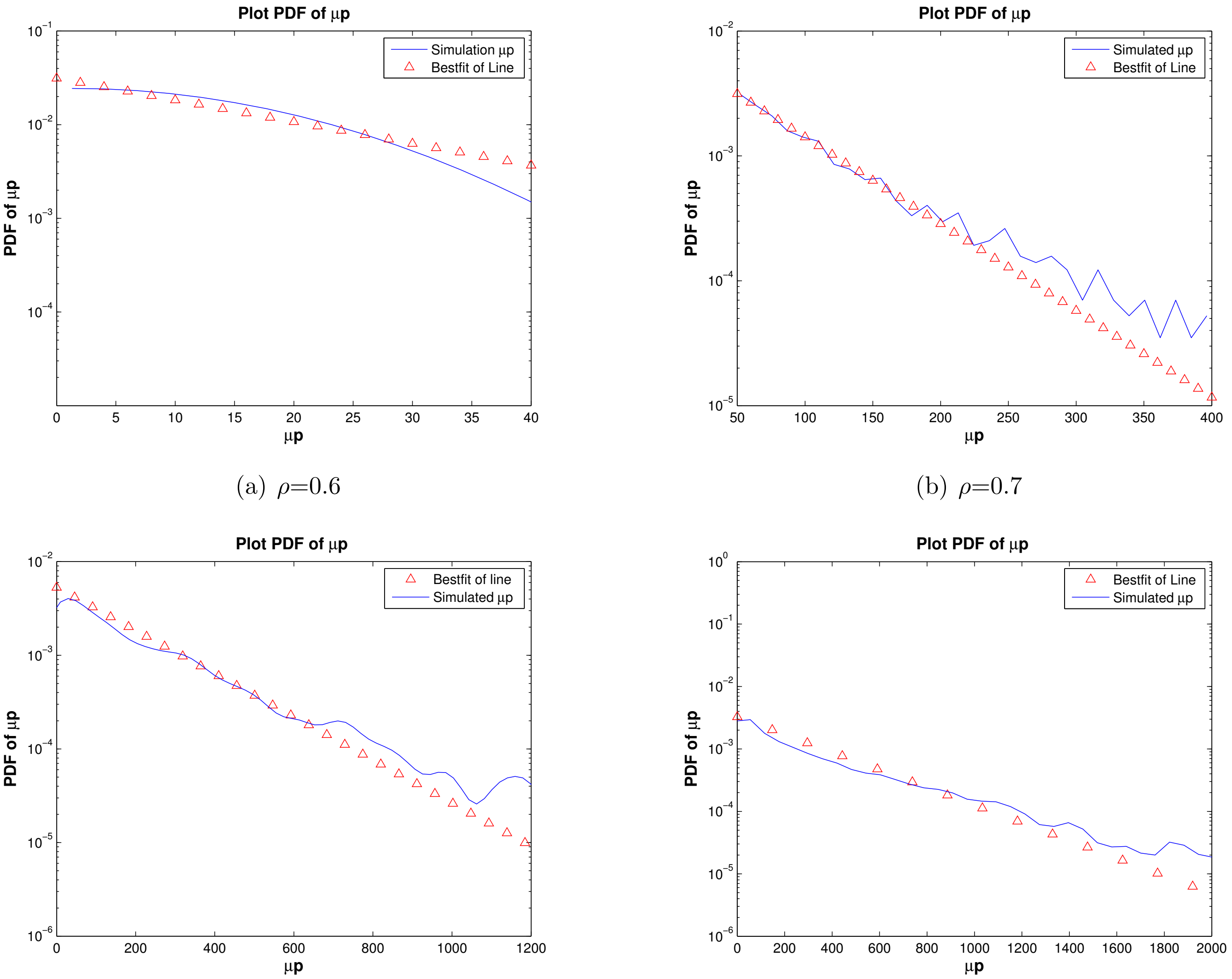
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 5.6: Validate Distribution of Cycle Time, T_{cycle}, with C=4Mbps



(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 5.7: Validate Distribution of expected packets lost per OvFl period, μ_p, with C=4Mbps

From Figure 5.4 - Figure 5.7, results show approximately straight lines for the majority of cases, from which it can be concluded that cycle time T_{cycle} is well modelled as being exponentially distributed and the number of the expected packets lost per OvFl period μ_p is well modelled as being geometrically distributed.

For some cases, there are distorted tails in those distributions caused by rare occurrence of events, which is usual. These tails of the distributions can be found more accurately by using longer simulation runs.

5.4.3 Results

In this section, the required SRLASS, ΔT, will be shown in two groups. All the results are plotted log-linear and are shown by varying different target relative width, 10%, 20% and 50%. The utilisation ρ ranges between 0.6 to 0.9 since they are typical loads on an access node[2]. In Figure 5.8, results for a single access link are shown, with comparison between different service rate, 4Mbps and 2Mbps. In Figure 5.9, results are compared between single access and an end-to-end network. The plotted points in both Figures have all been validated as falling within the relative error C.I.'s, for the ΔT limit, as defined in this thesis.

As shown in Figure 5.8, ΔT are plotted with preset precision target, 10%, 20% and 50%, for a single queue model for PMM, over load ranging from 0.6 to 0.9 and service rate of 4Mbps and 2Mbps. Results are also shown with respect to different relative width, where smaller width requires longer SRLASS, ΔT.

As shown in Figure 5.8, the required ΔT increases at least exponentially as the load increases. High load, compared to low load, requires larger buffer length in order to achieve the same PLP target, so it takes more time to fill up the buffer, and so longer simulation runs are required.

Low service rate also requires longer SRLASS, compared to high service rate, because the required buffer length for 2Mbps link to achieve a target PLP of 10^{-4} is larger than that of 4Mbps. Reduced service rate means that for a unit of time, the number of packets that the buffer can serve reduces,

[2] Compared to access node, core node has the feature of high bandwidth, and low utilisation, where can be approximated by M/D/1 queue. Therefore, the classical model (Work by Whitt) can be used.

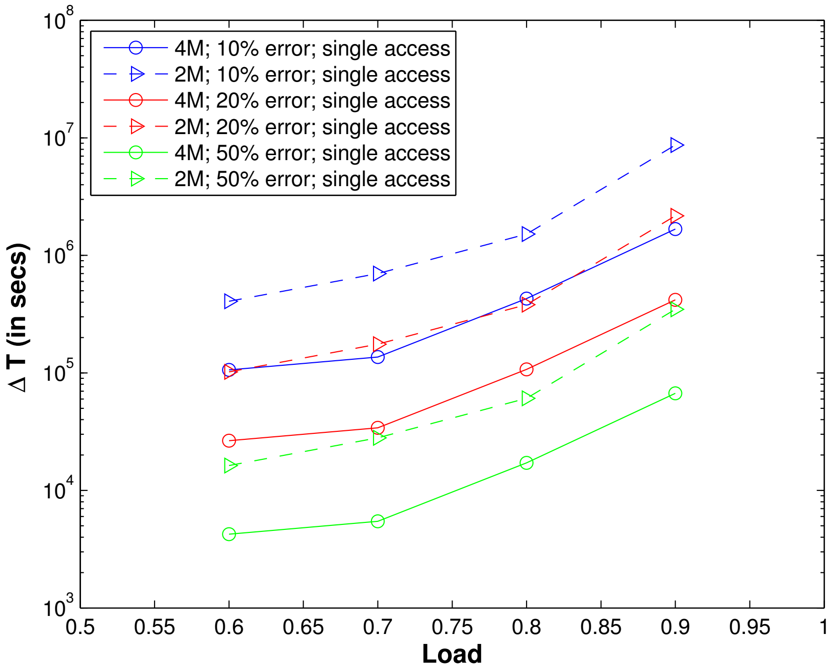


Figure 5.8: SRLASS Results - varying service rate

which requires a longer buffer to keep those waited packets. Therefore, a longer buffer takes more time to fill up, which leads to longer SRLASS.

In Figure 5.9, results are compared between single access link and an end-to-end network, when service rate is 4Mbps. Similarly to Figure 5.8, the required ΔT increases at least exponentially as the load increases. It also increases if the preset relative width reduces.

For the end-to-end network, a longer SRLASS is required to reach steady state, since the overall PLP is targeted to 10^{-4}, which is the same as the overall PLP in a single access link. However, the overall PLP in the end-to-end network is controlled by multiple buffers, where each individual PLP reduces significantly. In this case, each buffer requires longer buffer length in order to obtain less packet losses, where it takes more time to fill up each buffer than that of the single access link. Therefore, a longer ΔT is required

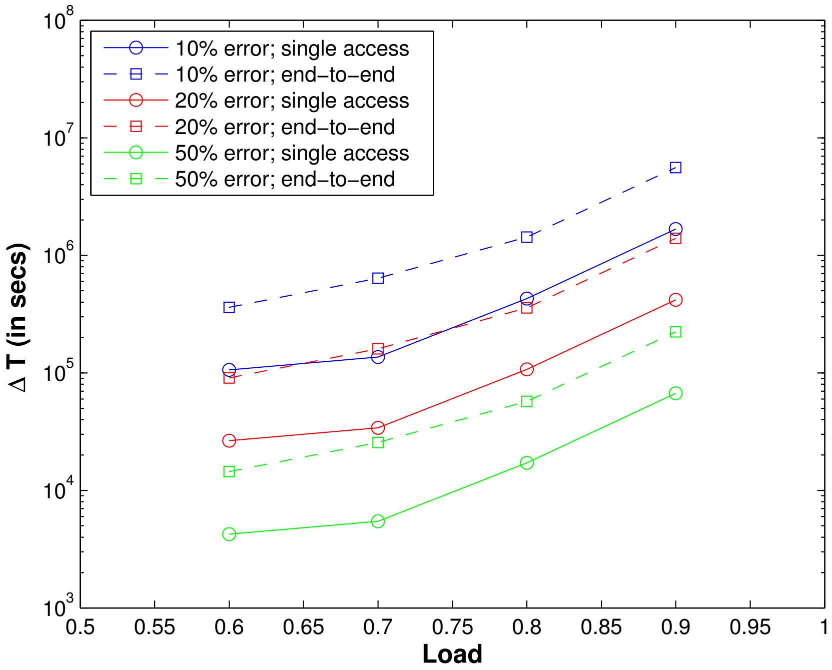


Figure 5.9: SRLASS Results - comparison between single access and end-to-end network (C=4Mbps)

for an end-to-end network simulation.

5.5 Conclusion

In this section, an analytical model for SRLASS, ΔT, is developed to indicate the required time for PLP to reach steady state in a packet multiplexing model. Validation results show that the assumption of memoryless distributions is valid, and the desired precision level is achieved in the time predicted by our approach. Results illustrate that simulation studies may consume a lot of time to reach steady state, i.e. long SRLASS, especially for high load, small target relative width in the end-to-end network.

Chapter 6

Simulation Planning of PLP in the Pareto Source PMM

This chapter examines simulation planning of PLP when the multiplexed sources follow a Pareto distribution. Pareto sources have been introduced already in Section 3.2.2. The Pareto traffic source is much more bursty, and better captures the burstiness of data traffic on networks than the Markovian source model.

However, the analysis of the Pareto distribution is difficult in simulation planning because the variance of PLP is crucial to finding the required SRLASS, and with Pareto traffic this is usually quite large, or even infinite.

In this chapter, we endeavour to fit the Overflow analysis (also used in Chapter 5) into the Pareto traffic source, and therefore find a way to plan simulation of the PLP for the Pareto source model.

6.1 Overview of Previous Pareto Traffic Source Research

Since the 1990s, there has been no lack of research [67] claiming that the feature of self similarity exists in data traffic, including ATM networks [62], Ethernet traffic [49], World Wide Web traffic [18] [19] and video traffic[30], etc.

Self similarity is commonly described using the Pareto distribution [52] [12] [66]. PLP, as one main QoS parameters, is also the focus of Pareto source research [65] [94]. There are also some papers looking at the aggregation/multiplexing of Pareto sources [54] [98]. Some research studies Pareto sources in the PMM: [55] gives the buffer overflow probability of a single Pareto source model, and [56] expands it into a Pareto source PMM. [80] also gives an analytical model for buffer overflow probability in the scenario of PMM, which can be a good approximation for the PLP.

Therefore, for consistency with Chapter 5, this chapter also examines the PLP as the metric of interest, and plans simulation of PLP for Pareto traffic source in the PMM[1].

6.2 Fitting the Pareto Source Model into the Overflow Analysis

6.2.1 Review of Planning Simulation for PLP

Recall that in Chapter 5, we proposed to plan simulation of the PLP by finding how many cycles it requires for the PLP to reach steady state. SRLASS can be calculated from the number of cycles by multiplying by the cycle time T_{cycle}.

The required number of cycles can be determined using statistical analysis introduced in Section 2.3. Equation 2.14 shows that by targeting a specific precision requirement, the number of cycles can be calculated using the analytical model of SCV for PLP. However, the SCV for PLP is very difficult to find. Therefore, an alternative method is proposed in Chapter 5 by finding the SCV of expected packets lost per OvFl period, $c^2(\mu_p)$ and the SCV of

[1] Pareto is the traffic model of non-real time, for which delay is less relevant than loss.

the cycle time, $c^2(T_{cycle})$. In conclusion, planning simulation for PLP can be done by using the analytical model of cycle time, as well as the analytical model of SCV for μ_p and T_{cycle}.

The SCV model for μ_p and T_{cycle} is found by validating that μ_p follows a Geometric distribution and T_{cycle} follows an Exponential distribution in Chapter 5.

For a Pareto source model, T_{cycle} will be much larger than that for Markovian model, because Pareto source traffic has the feature of heavy tails, which will lead to longer overflow periods and cycle times. The distribution for μ_p and T_{cycle} will also be different, which leads us to different results.

6.2.2 Model of Cycle Time for Pareto Source Model

Recall that the cycle time model is given for the Markovian source model in Equation (5.8) by

$$T_{cycle} = \frac{(R_{on} - C) \cdot T(on)}{A_p \cdot PLP} \tag{6.1}$$

For Pareto sources, in order to achieve the same PLP, the buffer must be set much larger. In this case, the time for filling up the buffer can not be neglected, and contributes a large value to the aggregate ON period, $T(on)$. Denote the new aggregated 'ON' period for Pareto source as $T_{prt}(on)$, given by

$$T_{prt}(on) = T(on) + \frac{BS}{R_{on} - C} \tag{6.2}$$

where BS is the buffer size, and $R_{on} - C$ is the excess rate[2], given by Equation (5.2) [72].

In this case, the model for cycle time is given by

$$T_{cycle} = \frac{(R_{on} - C)}{A_p \cdot PLP} \cdot (T(on) + \frac{BS}{R_{on} - C}) \tag{6.3}$$

[2] R_{on} is aggregated ON rate, and the part it exceeds to the service rate C is the rate for the buffer filling up, which is called excess rate.

6.2.3 Discussion of SCV Model for μ_p and T_{cycle}

For a Markovian source model, the expected packets lost per OvFl period has been validated to follow the Geometric distribution, while the cycle time follows the Exponential distribution, with results shown in Section 5.4.2.

However, for Pareto traffic source, this might not be the same. Figure 6.1 to Figure 6.4 show the validation results from numerical examples in Section 6.3.2, that μ_p follows approximately the Geometric distribution, while the cycle time T_{cycle} follows the Pareto distribution (approximately straight line in log-log plot).

Finding the SCV model for Pareto distributed random variables is very difficult, because the variance of such variables are infinite when $\alpha \in (1, 2]$, as shown in Equation (3.7).

Therefore, this chapter proposes to plan simulation of PLP for Pareto source just using the SCV model for μ_p, which still follows Geometric distribution, and will be validated in Section 6.3.2 by Figure 6.1 and Figure 6.2, as was done for the Markovian source model in Chapter 5.

Recall that for the expected packets lost per OvFl period, μ_p, the required sample size for μ_p to reach the steady state is given by (also see Section 5.3.1)

$$N_{\mu_p} = 4 \cdot \frac{z^2_{1-\beta/2}}{\varepsilon^2} \cdot c^2(\mu_p) \tag{6.4}$$

and $c^2(\mu_p)$ is given by

$$c^2(\mu_p) = Var(\mu_p)/\mu_p^2 = (\mu_p^2 - \mu_p)/\mu_p^2 = 1 - 1/\mu_p \tag{6.5}$$

since μ_p follows approximately Geometric distribution.

Therefore, SRLASS for Pareto sources multiplexing model is given by

$$\Delta T = N_{\mu_p} \cdot T_{cycle} = \frac{4z^2_{\beta/2}}{\varepsilon^2} \cdot (1 - \frac{1}{\mu_p}) \cdot \frac{(R_{on} - C)}{A_p \cdot PLP} \cdot (T(on) + \frac{BS}{R_{on} - C}) \tag{6.6}$$

6.2.4 Extending the Analysis to an End-to-End Network

It is very important to consider an end-to-end network for simulation planning research. This section keeps the consistency with Section 5.3.3 to use the same network model, the FG/BG network model.

Recall that Foreground Traffic (FT) traverses n identical buffer throughout the network, and each individual buffer is denoted as $IPLP$ while the overall PLP for the end-to-end network is denoted as $TPLP$.

In this case, similar to Equation 5.22, the relationship between $TPLP$ and $IPLP$ is:

$$TPLP \approx n \cdot IPLP \tag{6.7}$$

And the cycle time of the Pareto source in an end-to-end network is:

$$T_{cycle} = n \cdot \frac{(R_{on} - C)}{A_p \cdot TPLP} \cdot (T(on) + \frac{BS}{R_{on} - C}) \tag{6.8}$$

Therefore, the SRLASS, ΔT_{e2e} for Pareto traffic source in an end-to-end network is given by

$$\Delta T_{e2e} = n \cdot \frac{4z^2_{\beta/2}}{\varepsilon^2} \cdot (1 - \frac{1}{\mu_p}) \cdot \frac{(R_{on} - C)}{A_p \cdot TPLP} \cdot (T(on) + \frac{BS}{R_{on} - C}) \tag{6.9}$$

6.3 Numerical Examples

6.3.1 Simulation Set-up Parameters

Recent literature has no lack of Pareto source research, which we followed to set up the parameters used in this research. The ON sojourn time, T_{on}, is ranging from the magnitude of 10^2 ms [12] [66], 10^1s [57] [65] and 10s [94] [37], where the sending rate h is around 10 packets/s [66] [63].

In this research, T_{on} and T_{off} are used following ref.[57], while T_{on} is ranging from 3s to 10s, and T_{off} is 10s. And the sending rate for each Pareto source is set to be 10 packets/s, which is reasonable and supported by recent literature [66][63].

In conclusion, numerical examples are given using parameters as follows:

- T_{on} set to be 3s, 5s, 8s and 10s,
- T_{off}=10s,
- ON rate h=10packets/s,
- packet size=1000 bytes,
- utilisation ranging from 0.6 to 0.9 [3].
- Buffer Size is changed to make the PLP keep 10^{-4}.

And these parameters are listed in Table 6.1 and Table 6.2.

N=10			N=20		
Load	T_{on}	**Buffer Size**	**Load**	T_{on}	**Buffer Size**
0.6	3s	1770	0.6	3s	240
0.7	5s	3000	0.7	5s	550
0.8	8s	8500	0.8	8s	1800
0.9	10s	1E+05	0.9	10s	2.2E+04

Table 6.1: Set-up Parameters for Pareto Source - Single Access Link

[3] Use of utilisation in the range [0.6, 0.9] is because: 1) This is a typical load range on an access node; 2) If utilisation is under 0.6, the arrival process will approximately tend to a Poisson process, i.e. it won't exhibit burst scale queueing, and standard classical simulation run length planning techniques [89] can be used instead.

N=10			N=20		
Load	T_{on}	**Buffer Size**	**Load**	T_{on}	**Buffer Size**
0.6	3s	3800	0.6	3s	450
0.7	5s	6800	0.7	5s	950
0.8	8s	18000	0.8	8s	3200
0.9	10s	2.5E+05	0.9	10s	5E+04

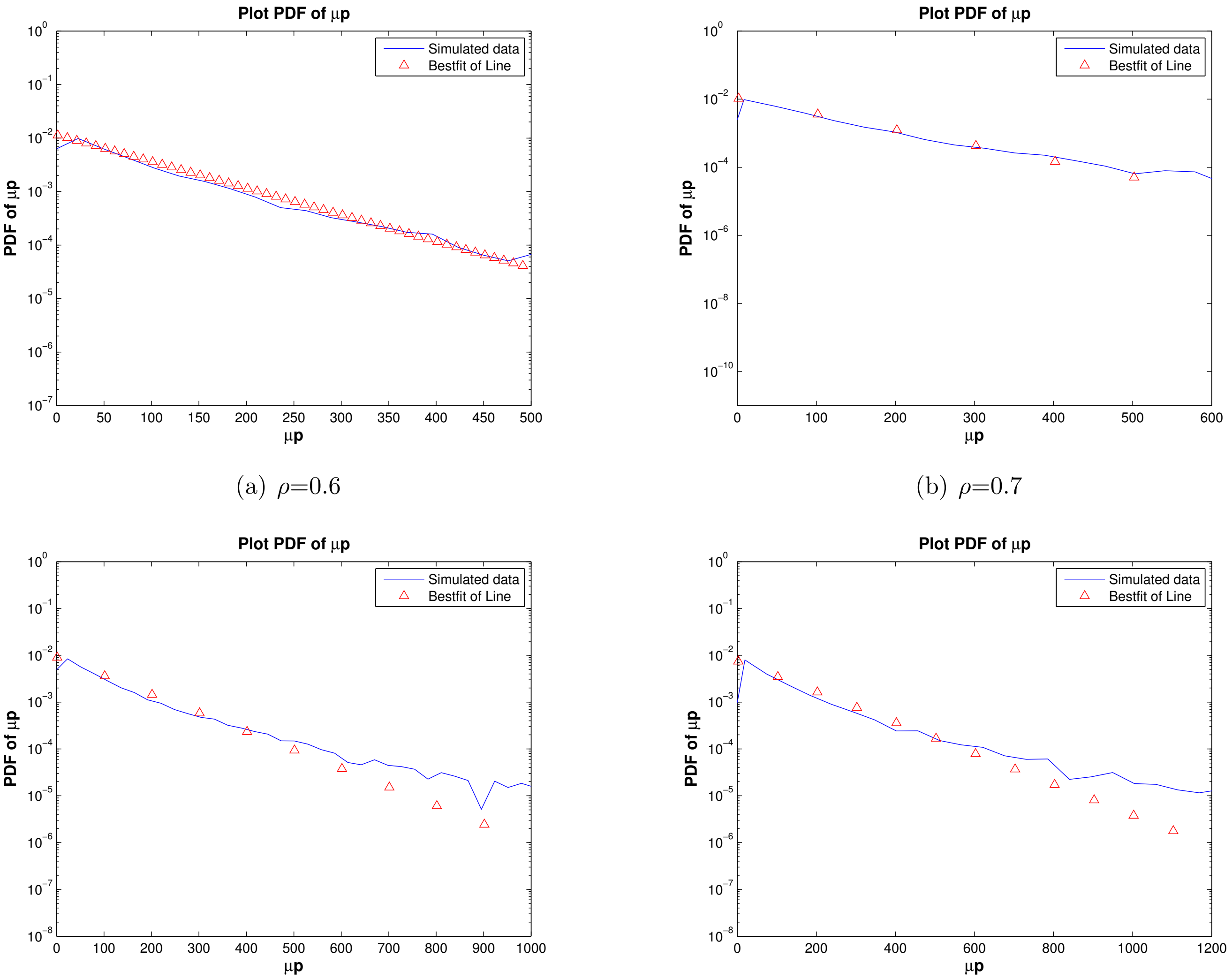
Table 6.2: Set-up Parameters for Pareto Source - End-to-end (n=3)

6.3.2 Validation of Results

Validation Results for μ_p

This section aims to validate the assumption of Geometric distribution for the expected packets lost per OvFl period, μ_p. Validation is done against Table 6.1.

Figure 6.1 and Figure 6.2 are all plotted log-linear, where a straight line indicates the Geometric distribution. Figure 6.1 shows the distribution of μ_p with the load ranging from 0.6 to 0.9, when the number of sources $N = 10$ by applying bestfit technique to its pdf. The same applies to μ_p when $N = 20$ in Figure 6.2.



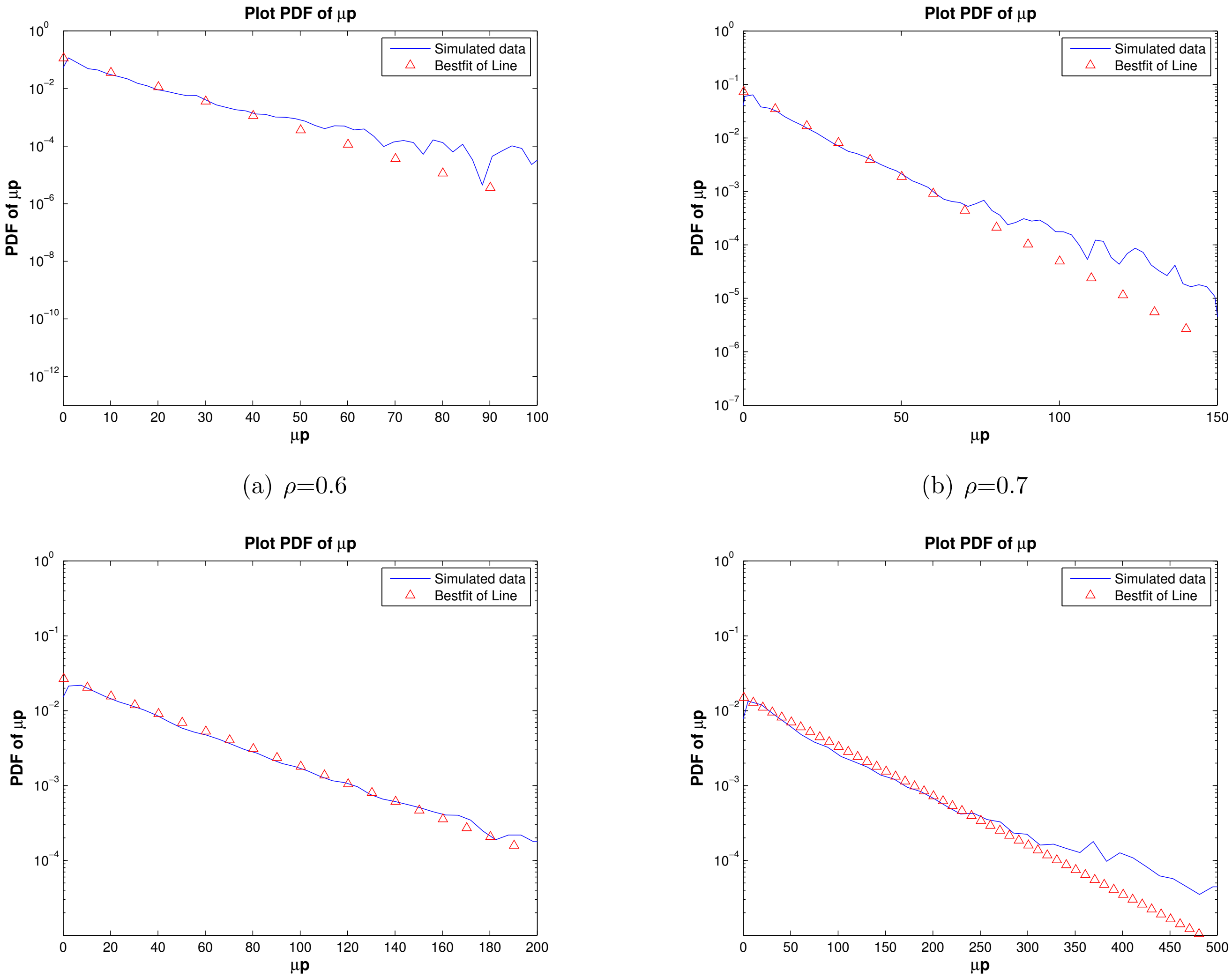
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 6.1: Validate Distribution of expected packets lost per OvFl period, μ_p with N=10



(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

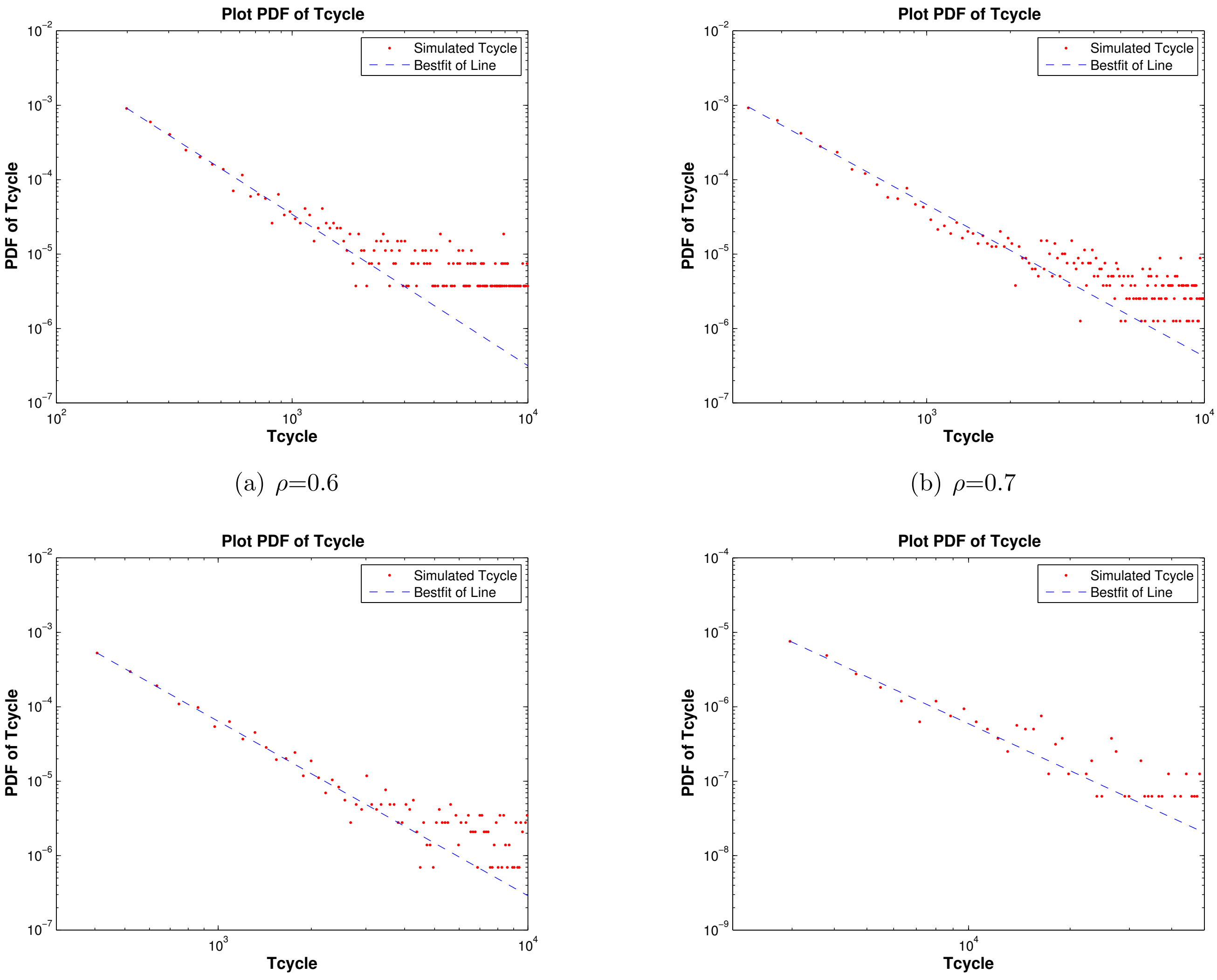
Figure 6.2: Validate Distribution of expected packets lost per OvFl period, μ_p with N=20

From Figure 6.1 and Figure 6.2, results indicate an approximately straight lines for the majority of cases, from which it can be concluded that the expected packets lost per OvFl period, μ_p, is well modelled as being Geometrically distributed.

For some cases, there are distorted tails in the distribution, which is mainly caused by rare occurrence of events, which is usual. This distorted tails can be found more accurately by running longer simulations.

Validation Results for T_{cycle}

Different from Markovian source multiplexing model, the cycle time, T_{cycle} of Pareto source multiplexing model follows Pareto distribution, rather than Exponential distribution. Figure 6.3 and Figure 6.4 are all plotted in log-log scale, where straight line indicates a Pareto distribution. Figure 6.3 shows the distribution of T_{cycle} with the load ranging from 0.6 to 0.9, when the number of sources $N = 10$ by applying bestfit technique to its pdf. The same applies to T_{cycle} when $N = 20$ in Figure 6.4. Simulated T_{cycle} distribution is plotted in red dots, while the bestfit is plotted in blue dotted lines. From Figure 6.3 and Figure 6.4, cycle time T_{cycle} is well modelled as Pareto distribution. Again, similar to μ_p, the distorted tails in those distributions are all caused by rare occurrence of events (pretty long cycle time), which can be removed by running much longer simulation runs.



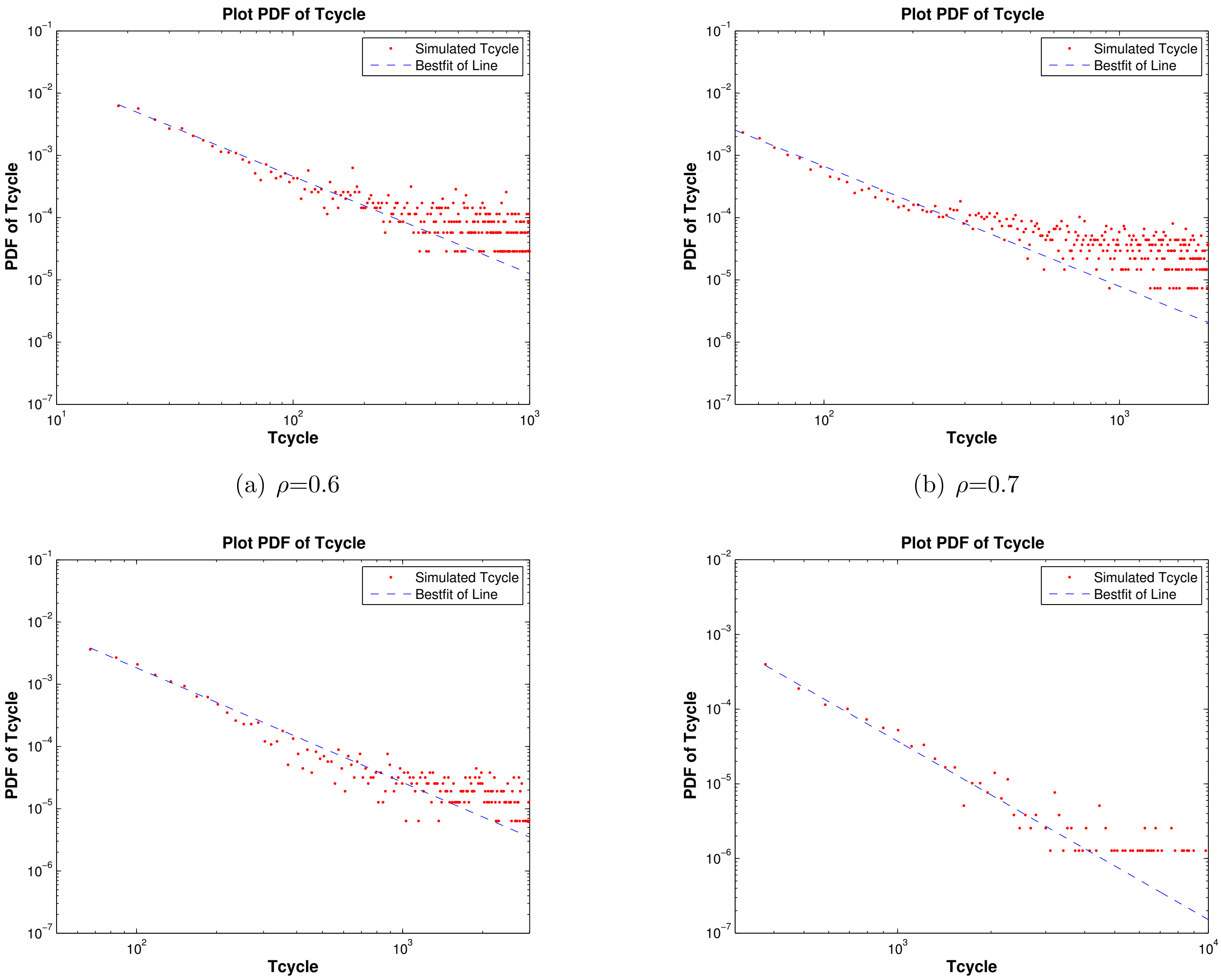
(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 6.3: Validate Distribution of Cycle Time, T_{cycle} with N=10



(a) ρ=0.6

(b) ρ=0.7

(c) ρ=0.8

(d) ρ=0.9

Figure 6.4: Validate Distribution of Cycle Time, T_{cycle} with N=20

6.3.3 Results

This section gives the results, required SRLASS, ΔT, for the parameters in Table 6.1 and Table 6.2. It will be shown in two groups: 1) single access scenario, with different number of sources N=10 and N=20 (see Figure 6.5); 2) comparison between single access and end-to-end network (see Figure 6.6).

Results are plotted log-linear for different target relative width, 10%, 20% and 50%. The utilisation ρ ranges between 0.6 to 0.9 since they are typical loads on an access node. All plotted points in both Figure 6.5 and Figure 6.6 have all been validated as falling within the relative error C.I.'s, as defined in this thesis.

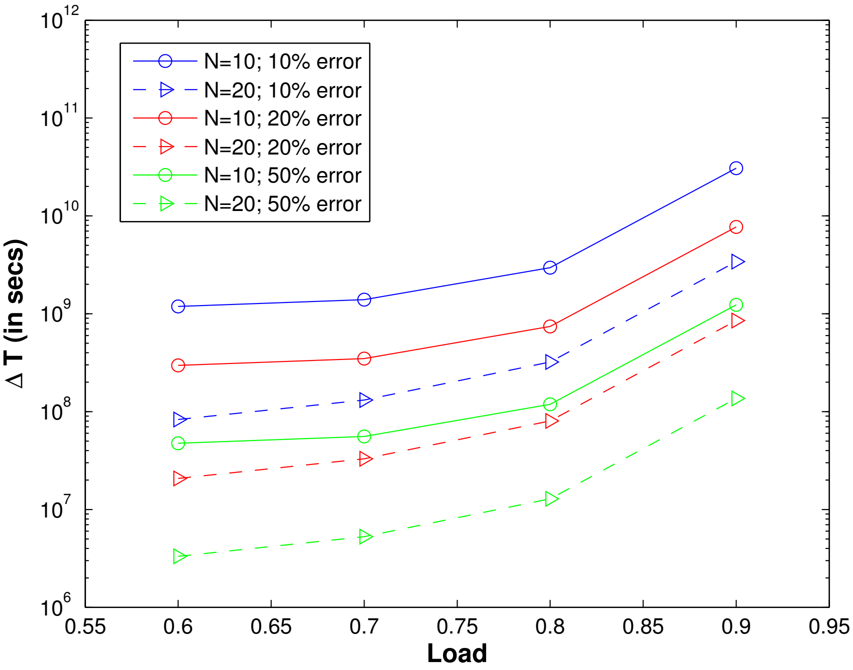


Figure 6.5: SRLASS Results for Single Access

As shown in Figure 6.5, ΔT increases exponentially as the load increases. It requires longer SRLASS for high load scenario to reach the steady state than that of low load.

Moreover, when the number of sources increases, the capacity of the buffer also increases (service rate is larger). In this case, traffic (when N=20) is less bursty than fewer number of sources (when N=10). This results is consistent with the results in Section 4.6.3.

What's more, the required SRLASS for Pareto sources is much longer than that of Markovian source. This is intuitive since Pareto traffic source is more bursty and requires a longer time to reach steady state.

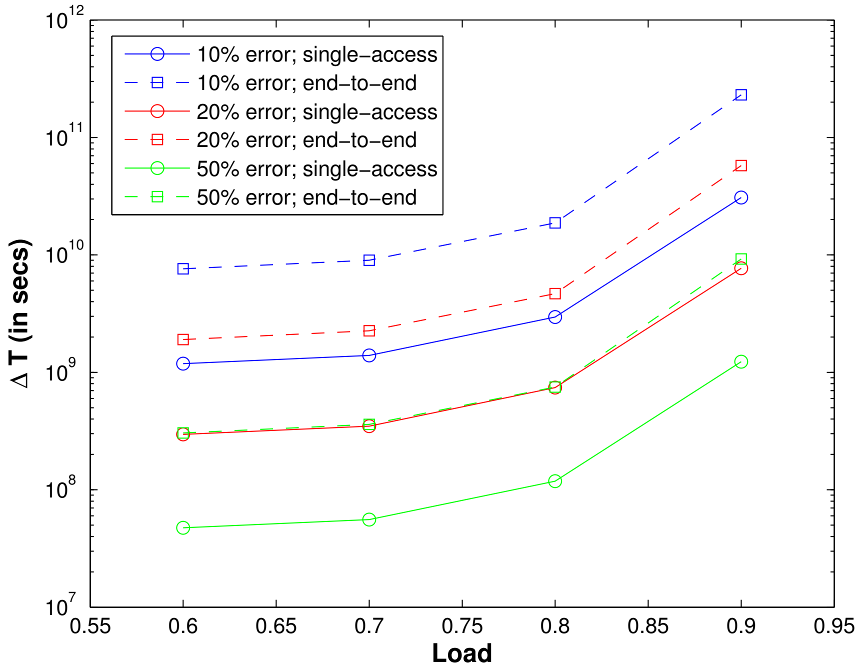


Figure 6.6: SRLASS Results when N=10

Figure 6.6 gives results, comparing the single access scenario and end-to-end network when N=10. Similarly, the required ΔT increases at least exponentially as the load increases. It also increases if the preset relative width reduces, because higher precision level requires longer run.

From the developed formula, end-to-end network requires longer time to reach the steady state than single access. This is consistent with results in Section 5.4.3 for Markovian sources.

6.4 Conclusion

In this section, an analytical model for the SRLASS is developed for the Pareto source PMM. Pareto sources are more variable than Markovian sources. Models are designed for Pareto traffic source PMM by modifying the mathematical model, T_{cycle}. Validation results show that the assumption of Geometric distribution of μ_p is valid and the proposed run length planning technique is accurate.

Chapter 7

Mapping SRLASS into Wall Clock Time

7.1 Overview of Wall Clock Time Taken by Simulations

As the main objective for this research, the SRLASS ΔT is the key parameter to be found. In the previous chapters, SRLASS ΔT is given in the units of simulation time, or the number of packet arrivals, which gives a good indication of how to set the run length of the simulation. However, the wall clock time (real time used by the computer processor) consumed for such simulations can not be predicted using our previous research in Chapter 4 to Chapter 6.

Therefore, in this chapter, the aim is to find a method to map the SRLASS ΔT into corresponding wall clock time, which gives an idea before a long replication is run. If the time consumed for the simulation is so long, a redesign and recode of the simulator might be required.

7.1.1 Analysing Factors Affecting Wall Clock Time

In this research, NS2 is used as the simulation tool. NS2, as already introduced in section 3.1.5. NS2 is a discrete-event simulator, which uses C++ as the core language. C++ is an object oriented language.

There are several factors affecting the wall clock time, including

- The load, ρ
- Number of flows
- Network Topology, e.g. number of links/number of nodes
- Traffic type, e.g. Poisson, or Markovian/Pareto ON/OFF
- Hardware configuration, e.g. CPU processor, memory usage

7.2 Finding Indicator - Packet Arrivals

Instead of finding how those factors affect the wall clock time, this thesis investigates how the wall clock time increases in proportion to the packet arrivals. Because, in NS2, all the events in the simulation process of a queuing system are related to the object of packet. Packets are generated, packets are sent, queued to be served, either drop or received. Through this analysis, it is intuitive to believe that number of packets processed are an essential parameter affecting the wall clock time consumed by the computer processors. In this chapter, a standard indicator for predicting the wall clock time is proposed - packet arrivals.

Also, another essential is the hardware parameters, i.e. the speed of the computer processors. A powerful computer is very important to the simulation. A more powerful computer will make the simulation run faster. Hardware parameters are usually represented using the computer processor,

processor speed, and the memory allocated to the program, i.e. NS2. However, as this research is not focused on the design of computer processor, the results will show the differences between different computer hardware.

Therefore, the number of packets processed is regarded as the standard indicator for the wall clock time.

7.3 Numerical Examples

This section shows the numerical examples. The parameter settings are all obtained from the previous section, i.e. different types of traffic source, different utilisation, different numbers of sources, and different numbers of links. And the results are run using two different machines, one is using 8GB RAM,2.4GHz Intel Core 2 Duo while the other is using 48GB RAM,1333MHz CPU.

The results comprise of four groups, all plotted as a cross on the figures. The blue crosses represent the short runs, with the packet arrivals only in the magnitude of 10^5 - 10^7, for which simulations usually consume only tens of seconds to finish. Using these data, a fitted line is plotted as a solid blue line. This line is trying to predict the wall clock time consumed by the long run replications. As shown in the Figure 7.1 and Figure 7.2, the red crosses represent the simulation runs with packet arrivals in the magnitude of 10^7 - 10^8, and green crosses for 10^8 - 10^9. As the simulation runs longer, the wall clock time consumed to finish the simulation increases linearly with the number of packets processed.

As shown in Figure 7.1 and Figure 7.2, for the longer runs (represented by the red, green, black crosses), wall clock time lies within the prediction range using the solid blue line. So, from the results from a short run, the wall clock time of a much longer run can be predicted.

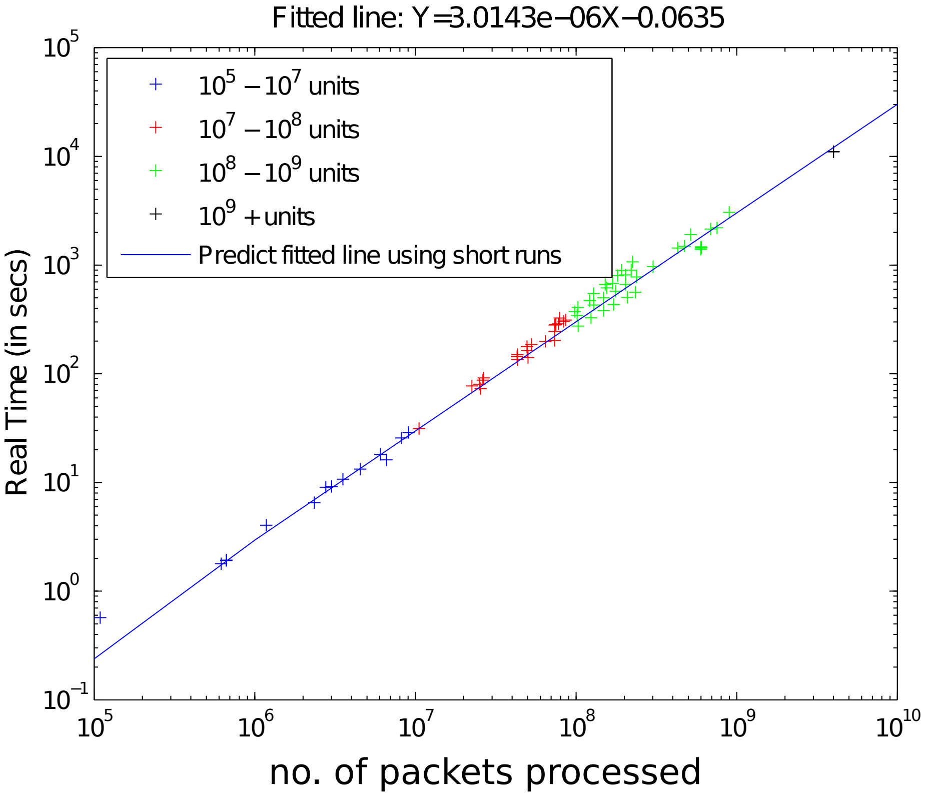


Figure 7.1: Mapping packet arrivals into Wall Clock Time - 8GB RAM,2.4GHz Intel Core 2 Duo

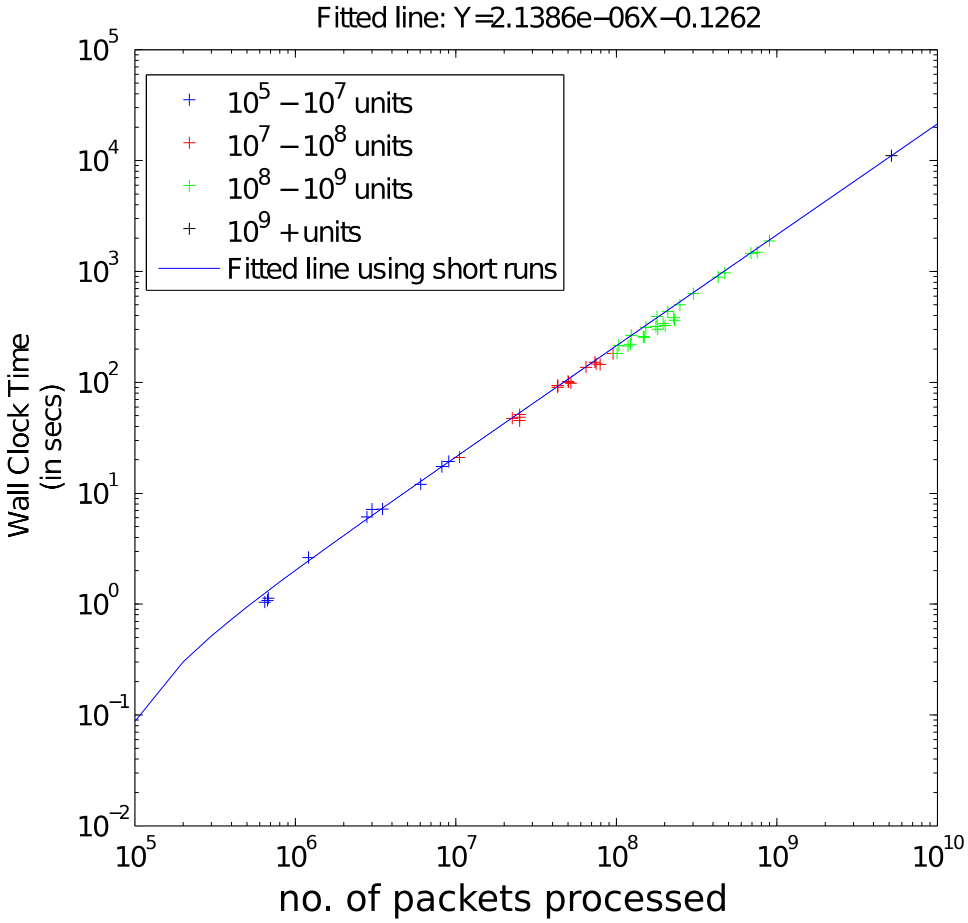


Figure 7.2: Mapping packet arrivals into Wall Clock Time - 48GB RAM,1333MHz

Also, these two figures use two different sets of hardware, running the same simulation. Results show that a more powerful computer will run the same simulation quicker, as would be expected.

7.4 Conclusion

In this chapter, all factors affecting the wall clock time consumed to run a simulation are considered, and the number of packet arrivals are proposed as a standard indicator. Results show that the wall clock time increases linearly with the number of packet processed, and this is illustrated also using different sets of hardware parameters.

From the figures, for the number of packets processed which is larger than 10^9 magnitude, it can take many hours to finish such simulation runs. Using the method introduced in this chapter, this long wall clock time can be predicted using short runs, i.e. just tens of seconds. Since it is not in the scope to analyse the computer processors, it is hard to give a back-of-envelope formula to calculate the wall clock time. It is still useful to use short runs to predict long run simulation time.

Chapter 8

Conclusions and Further Work

8.1 Conclusions

Simulation plays a nontrivial role in networks research, and provides an alternative approach to implementing a real environment, owing to its features of scalability, flexibility and ease of setup. Simulating large-scale networks can be expensive and this research proposes to plan simulations by providing mathematical and logical expressions for SRLASS, ΔT, which is the time consumed for the metric of interest to reach steady state.

Analytical models are developed for the SRLASS, ΔT, in a packet multiplexing model. Results show that simulation planning is very model dependent, and it can be solved using statistical analysis. Before the simulation is run, SRLASS can be found by analysing the SCV model of the metric of interest.

In Chapter 4, simulation is planned for the delay in a multiplex of Markovian sources. We used the packet-scale/burst-scale characteristics of PMM to develop the corresponding analytical model of SRLASS, for time taken for the delay to reach the steady state. A direct analytical model of SCV of delay in the PMM is developed, from which the SRLASS can be calculated. Sample interval is obtained by finding the minimum time duration to get two consecutive samples from different regeneration cycle, which can be used to reduce correlation. This used the c.d.f. of the busy period. Numerical results are provided and further compared with previous results in [89], showing that our results are much closer to simulated ones, while Whitt's formula overestimates the SRLASS significantly.

Chapter 5 provide simulation planning for PLP of a multiplex of Markovian sources. Simulation planning has never been done for the PLP before. The direct analytical model of SCV for PLP is difficult to find, therefore, we provide an alternative method. We find the SCV model for packet losses during Overflow period, and cycle time, and using this we plan the simulation for PLP. Also, instead of viewing queue behaviour in a packet basis, we view the queue behaviour in a cycle basis. We propose to see how many cycles it requires for PLP to reach steady state. Cycle analysis will not only provide a new methodology/technique to plan simulation, but also remove correlation naturally. Results show that simulation requires a long time to reach steady state, especially for high load with small target error. Our approach provides an accurate prediction of the time this will require.

Chapter 6 extends the research to explore the simulation planning for Pareto traffic sources. A Pareto traffic source is a heavy-tailed distribution, which makes the variance very large and it may be infinite. A new analytical model of the cycle time is developed for Pareto sources in the PMM. Results show that this required much longer SRLASS for the PLP to reach steady state than for the Markovian source model.

As SRLASS is in the units of simulation time, it also is meaningful to show how much wall clock time (actual time used by the computer) is required for the simulation to reach steady state. We use the number of packets processed to map the SRLASS to wall clock time. Different computers will have different processing speed, therefore, with small test runs on each computer, the mapping relationship between SRLASS and wall clock time can be found. This can be used to predict wall clock time for very long runs, using SRLASS required.

8.2 Further Work

Simulation planning is a generic topic, and it is an important step when simulating networks. It arouses the awareness of ensuring the accuracy of the simulation results, as well as knowing useful information before the simulation is run. There are many ways to further extend this research.

This thesis examines simulation planning in an end-to-end network for every metric of interest, based on assuming that the nodes along the routes are independent and identically distributed. However, in real networks, when the nodes are not identical, the results might change significantly. In that case, the network might be controlled by the bottleneck node, which can be used as a proper point to plan simulation for a non-identical node end-to-end network.

Moreover, the methodology used in Chapter 6 is suitable for Pareto traffic source, while the methodology used in Chapter 5 is for Markovian traffic source. It would be interesting to explore which technique should be used when the traffic is the combination of Pareto and Markovian traffic source, or even unknown traffic type.

Furthermore, this thesis all deals with UDP packets in PMM. However, there are also TCP packets existing in network being researched. TCP protocol needs feedback from the receiver ends, which will make the scenario more complicated to analyse.

Another valuable aspect to look at is wireless networks. Since in the wireless network, there are more parameters affecting/controlling the simulator, it is much more variable simulating a wireless network than a wired network.

The complicated topology and many parameters might lead to longer times for the simulation to reach steady state. Therefore, simulation planning for wireless networks is also important.

Finally, it is meaningful to find a general methodology to plan simulations. Because simulation planning is model dependent, the methodology for planning different network scenarios may be different. Ward Whitt intended to use general methodology to plan simulations by approximating metric of interest into a statistical processes. This works well for classical queue models, as shown in his work [89][92], but it overestimates for very specific scenarios, as shown in Section 4.6.3. Therefore, a new general methodology is required.

Appendix A

Buffer size dimensioning for packet multiplexing model

Evidence reports that the distribution of state probability[1] of a Markovian queuing system is in the form shown in Figure A.1 [72].

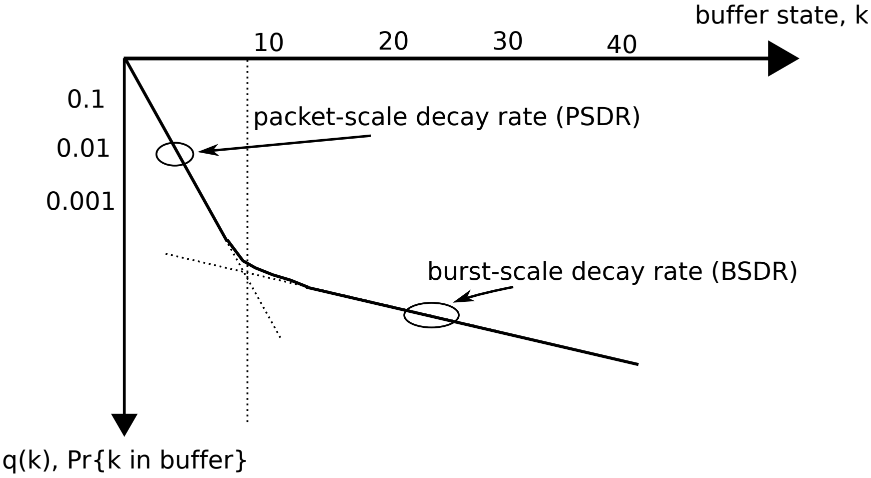


Figure A.1: Packet and burst-scale queuing

Figure A.1 shows the relationship between state probability, denoted as $q(k)$, and buffer state[2] k. A buffer overflow probability, $Q(X)$, in an infinite buffer is usually used as a reasonable approximation for the PLP in a finite

[1] State probability: the probability of a buffer state or queue state. State corresponds to the number of packets in the buffer.

[2] Buffer state: the number of packets in a buffer at some instant.

buffer with X as the buffer size[3], where

$$Q(X) = 1 - q(0) - q(1) - \cdots - q(X). \tag{A.1}$$

It is known that the both packet scale and burst scale follow separate Geometric distribution which can be written as

$$q(k) = \begin{cases} (\frac{a}{s})^k \cdot q(0), & \text{for } 0 < k < X; \quad \text{(A.2)} \\ (\frac{s}{1-a}) \cdot (\frac{a}{s})^k \cdot q(0), & \text{for } k = X. \quad \text{(A.3)} \end{cases}$$

where a and s are the parameters [72].

Since the summation of the state probability must be 1 as:

$$\sum_{k=0}^{X} q(k) = 1 \tag{A.4}$$

After some rearrangement, $q(0)$ is given by

$$q(0) = \frac{1 - \frac{a}{s}}{1 - (\frac{1-s}{1-a}) \cdot (\frac{a}{s})^X} \tag{A.5}$$

Assume $X \to \infty$, and thus $[1 - (\frac{1-s}{1-a}) \cdot (\frac{a}{s})^X] \to 1$, the state probability can be written as

$$q(k) = (1 - \frac{a}{s}) \cdot (\frac{a}{s})^k \tag{A.6}$$

[3] Buffer size: the capacity of the buffer, maximum number of packets a buffer can contain.

Similarly, the probability of queue exceeds k packets, $Q(k)$ is

$$Q(k) = (\frac{a}{s})^{k+1} \tag{A.7}$$

$\frac{a}{s}$ is the decay rate [72], where $\frac{a}{s}$ is often denoted as η. The queue overflow probability $Q(X)$ is the probability the queue exceeds k packets conditioned on the probability of experiencing burst-scale queuing, denoted as P_B. Therefore, $Q(X)$ is given by

$$Q(X) = P_B \eta^{X+1} \tag{A.8}$$

In this research, a more accurate burst-scale decay rate is employed [3]

$$\eta \rightarrow \frac{1 - [ln(h/C)/ln(\rho) + (h^2 T_{on} \rho)/(C(1-\rho)^2)]^{-1}}{1 - [\rho(1-\rho)^2/(h/C) \cdot T_{on} \cdot [(1-\rho)C + h \cdot \rho]]} \tag{A.9}$$

P_B can be obtained as [76]

$$P_B \approx \frac{1}{(1-\rho)^2 \cdot (C/h)} \cdot \frac{(\rho \cdot (C/h))^{\lfloor (C/h) \rfloor}}{\lfloor (C/h) \rfloor !} \cdot e^{-\rho(C/h)} \tag{A.10}$$

Based on Equation (A.8), (A.9) and (A.10), the PLP can be calculated accordingly.

Appendix B

Bestfit Algorithm for Exponential Distribution

In this research, the exponential distribution is important, so the validation of such distributions is important too. This chapter gives a brief overview about the exponential distribution first, followed by the bestfit algorithm for fitting raw data into exponential curves.

B.1 Exponential Distribution

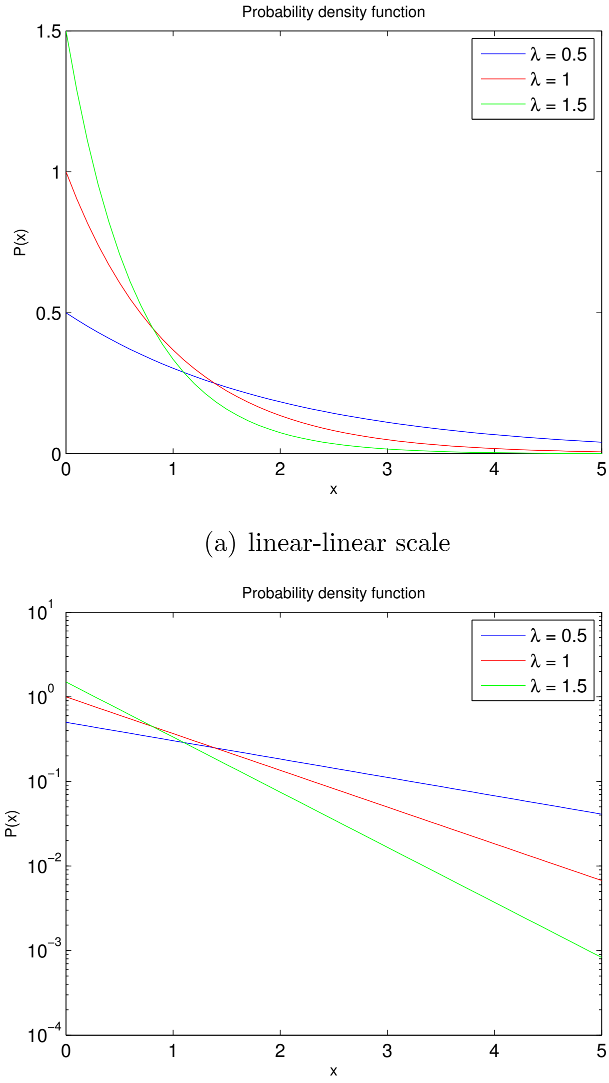
The probability density function (pdf) of an exponential distribution is

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{for } x \geq 0, \quad \text{(B.1)} \\ 0, & \text{for } x < 0. \quad \text{(B.2)} \end{cases}$$

where the mean is $1/\lambda$ and the variance is $1/\lambda^2$.

In Figure B.1, the exponential distribution is plotted in linear-linear scale and log-linear scale, respectively, with parameter λ set to be 0.5, 1, and 1.5. The Exponential distribution shows a curve when plotting in linear-linear scale, as shown in Figure B.1(a), and shows a straight line in a log-linear scale, as shown in Figure B.1(b). Therefore, a straight line when plotting the pdf of raw data in log-linear shows an approximate exponential distribution.

In order to ensure this, a bestfit of line is used to fit the exponential curve and is introduced in section B.2.



(a) linear-linear scale

(b) log-linear sclae

Figure B.1: pdf of exponential distribution in different scale

B.2 Bestfit algorithm

In this research, raw data is obtained from simulation results and fitted into an Exponential curve using the bestfit algorithm.

Algorithm 1 Bestfit Algorithm

```
Input: xdata, ydata
Output: $\lambda$
  $\lambda \Leftarrow 0$
  FittedCurve=$\lambda \cdot e^{-\lambda \cdot xdata}$
  sse=sum[(FittedCurve-ydata)$^2$]
  while sse is not minimum do
    $\lambda \Leftarrow \lambda + 0.01$
    FittedCurve=$\lambda \cdot e^{-\lambda \cdot xdata}$
    sse=sum[(FittedCurve-ydata)$^2$]
  end while
  return $\lambda$
```

As shown in Algorithm 1, there are two inputs, xdata and ydata, which are the raw data. Initialize λ to be 0, and calculate FittedCurve using pdf of exponential distribution, as shown in Equation (B.1). And then the standard error is calculated through summing up the square of the difference between the FittedCurve and ydata. If this standard error is the minimum, then this is the required parameter λ, else we continue to increase λ by 0.01 until we find the λ which makes the standard error minimum.

Example

Figure B.2 shows an example of a bestfit for raw data. Raw data is generated using an Exponential random generator. The pdf of the raw data is plotted in the blue line as shown in Figure B.2. The fitted curve is generated using Algorithm 1, and plotted in red stars. As shown in Figure B.2, the fitted line bestfits the raw data.

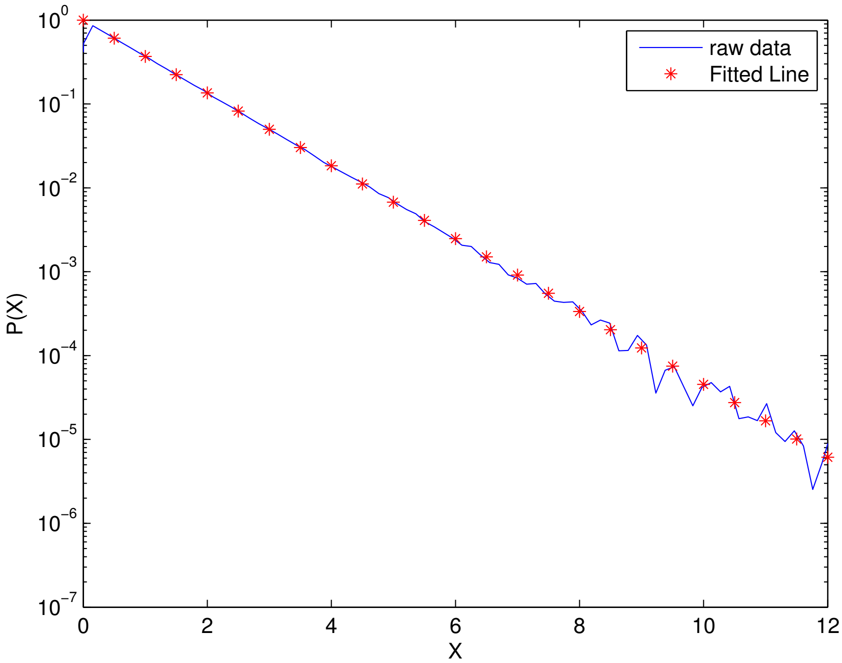


Figure B.2: A example of a bestfit for raw data

Appendix C

Measurement Correlation in Queuing Systems

It is reported that all measurements are correlated in queuing systems, which leads to inaccurate measurements, even for quite simple queue models [77]. It is intuitive that measurements are correlated in some pattern. Take the waiting time in M/M/1 queue for example, if packet i suffers a long waiting time, it is more probable that also packet $i+1$ will experience a long waiting time.

C.1 Background

C.1.1 Uncorrelated measurements

Define $\{X_i\}$ to be a set of i.i.d. measurements, with true mean $\bar{X}$ and variance $\sigma_X^2 < \infty$. An estimator is calculated using the sample mean, defined as

$$\hat{X}_N = \frac{1}{N}\sum_{i=1}^{N} X_i. \tag{C.1}$$

Based on CLT, when the measurements are uncorrelated, the sample mean converges as

$$\sqrt{N}(\hat{X}_N - \bar{X}) \sim N(0, \sigma_x^2), \tag{C.2}$$

where σ_x^2 represents the variance of uncorrelated samples.

C.1.2 Correlated measurements

What happens if the samples are correlated in some pattern? In this case, a new version of the CLT applies [77] as

$$\sqrt{N}(\hat{X}_N - \bar{X}) \sim N(0, s_N^2), \tag{C.3}$$

where s_N^2 is the asymptotic variance[1][89] of the correlated measurements, defined as

$$s_N^2 \equiv \lim_{N\to\infty} N \; Var(\hat{X}_N) \tag{C.4}$$

s_N^2 can be calculated by the following relationship [29]

$$s_N^2 = \sigma_x^2 + 2\sum_{i=1}^{\infty} R(i), \tag{C.5}$$

where $R(i)$ is the auto-covariance, defined as

$$R(i) = E[X_j X_{j+1}] - E[X_j]^2. \tag{C.6}$$

C.1.3 Discussion of measurement rate

The degree of correlation can be defined by the measurement rate, λ_s, since faster measurement will lead to a higher degree of correlation. In order to discuss the correlation, a continuous version of s_t^2 is defined as

$$s_t^2 = \lim_{t\to\infty} t \; Var(\hat{X}_t) = 2\int_o^{\infty} R(u)du. \tag{C.7}$$

[1] Asymptotic describes limiting behavior, thus asymptotic variance gives variance with a sufficiently large sample size.

[77] proved that

$$s_N^2 = \lim_{N \to \infty} N\ Var(\hat{X}_N) = \sigma_X^2 + 2\lambda_s \int_o^\infty R(u)du \qquad \text{(C.8)}$$

where the integral is finite.

Now, consider two extreme cases: very high measurement rate ($\lambda_s \to \infty$), and very low measurements rate($\lambda_s \to 0$).

When $\lambda_s \to \infty$, since t and N is related by $N = \lambda_s t$, Equation(C.8) can be written as

$$s_N^2 = \lim_{t \to \infty} t\ Var(\hat{X}_N) = \frac{\sigma_X^2}{\lambda_s} + 2 \int_o^\infty R(u)du \qquad \text{(C.9)}$$

where s_N^2 tends to the continuous version as shown in Equation(C.7).

When $\lambda_s \to 0$, from Equation(C.8), s_N^2 can be obtained by

$$s_N^2 = \lim_{N \to \infty} \lim_{\lambda_s \to 0} N\ Var(\hat{X}_N) \to \sigma_X^2, \qquad \text{(C.10)}$$

which is the uncorrelated variance. This makes sense since the measurement rate is so low that it will be at least several multiples of the correlation scale apart, where the correlations will be negligible.

C.2 Discussion for the correlation of the PLP measurements in packet multiplexing model

Figure C.1 shows how queue length changes over time in the packet multiplexing model in continuous time scale. It is intuitive that this process is correlated. For example, point A measures a relatively large queue state. Point B, which is close to point A, also experiences a relatively large queue state.

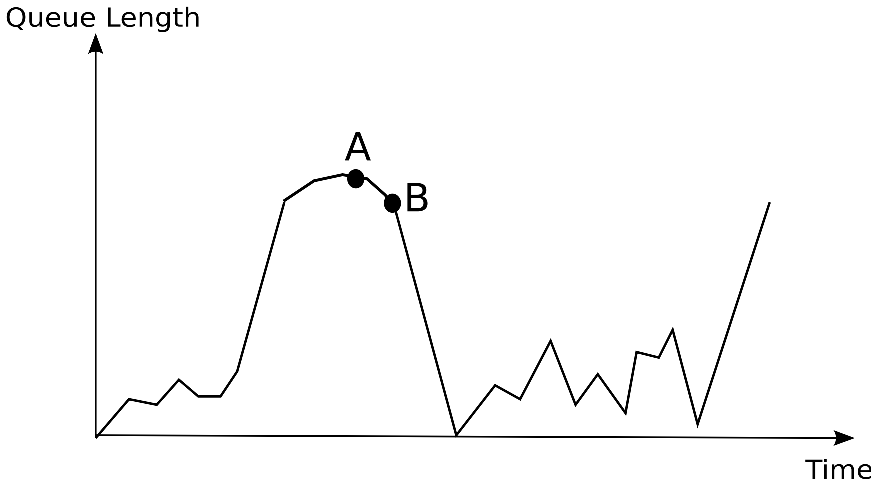


Figure C.1: Correlation Illustration of Queue Length

However, on one hand, the metric of interest in this research is the PLP, which is calculated by accumulated packet losses and accumulated packet arrivals. Therefore, the measurement of PLP always focuses on the overall queue behaviour.

On the other hand, this research is based on an aggregate OvFl/NOF analysis, as introduced in Section 5.2, shown in Figure C.2.

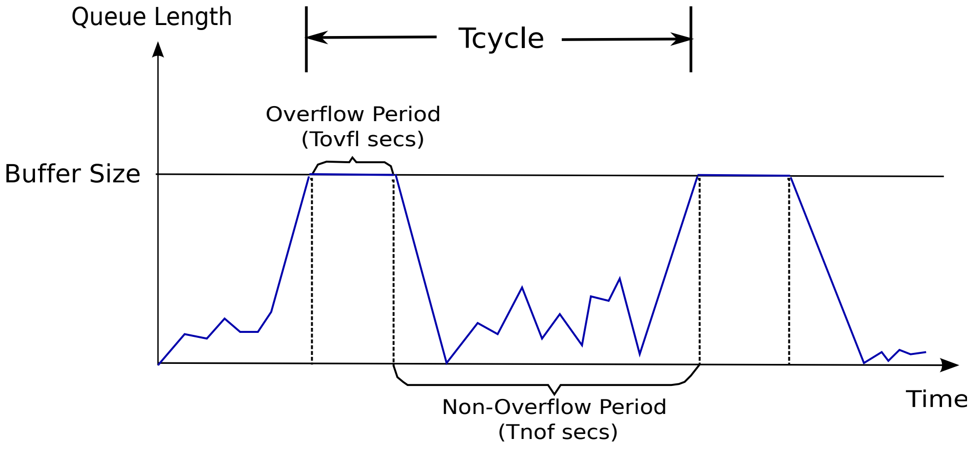


Figure C.2: OvFl/NOF analysis for packet multiplexing model

As shown in C.2, OvFl periods are separated by relatively long NOF periods, which are generally much larger than the correlation scale. Therefore, Equation (C.10) applies to our analysis.

In conclusion, the correlation in this research is negligible and we assume that the process of the PLP is uncorrelated.

Appendix D

Method of Obtaining the Simulated SRLASS from Simulation Raw Data

This Appendix will show how the simulated SRLASS is obtained from simulation raw data for waiting time.

Suppose that a simulation of the PMM is run, and the waiting time raw data is collected. Once data of waiting time is collected, the relative width

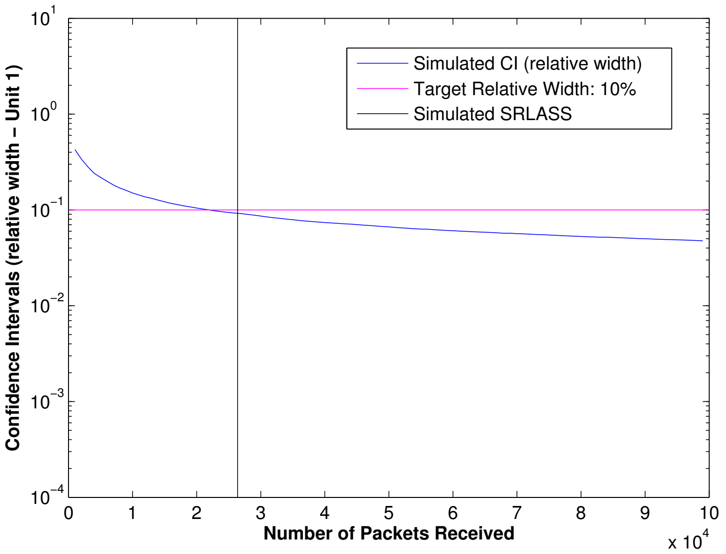


Figure D.1: The way to obtain SRLASS from simulation raw data

can be obtained as a function of number of packets received N through Equation (D.1) (which is first introduced in Section 2.3.3 in this thesis)

$$\varepsilon_r = 2z_{1-\beta/2} \cdot \frac{\sigma}{\bar{X}\sqrt{N}} \tag{D.1}$$

and this relative width ε_r can be plotted against the number of packets that arrived, as shown in Figure D.1, where the relative width is in the unit of 1.

In this case, with a targeted precision level, the required sample size can be obtained. For example, if 10% is targeted, as plotting in the pink line in the figure, then the corresponding number of packets required for the metric to reach steady state (with targeted 10% relative width) is obtained. With the number of packets required, we can easily get the SRLASS in the units of simulation time(method introduced in Section 2.4), as well as in the units of wall clock time (method introduced in Chapter 7).

Bibliography

[1] Joseph Abate and Ward Whitt. Limits and approximations for the busy-period distribution in single-server queues. *Prob. Engr. Inf. Sci. 9*, 9:581–602, 1995.

[2] M. Abusubaih, B. Rathke, and A. Wolisz. A dual distance measurement scheme for indoor ieee 802.11 wireless local area networks. In *Mobile Wireless Communications Networks, 2007 9th IFIP International Conference on*, pages 121 –125, sept. 2007.

[3] V. Amaradasa, J. Schormans, J. Pitts, and C.M. Leung. Evaluating overflow probability for voice over internet protocol buffer dimensioning. *IET Communications*, 3(4):636–643, 2009.

[4] D. Anick, D. Mitra, and M. Sondhi. Stochastic theory of a datahandling system with multiple sources. *American Telephone and Telegraph Company The Bell System Technical Journal*, 1982.

[5] H. Arsham. *Systems Simulation: Systems Simulation: The Shortest Route to Applications.*

[6] Roland Baddeley, L. F. Abbott, Michael C. A. Booth, Frank Sengpiel, Toby Freeman, Edward A. Wakeman, and Edmund T. Rolls. Responses of neurons in primary and inferior temporal visual cortices to natural scenes. In *Proc. R. Soc. Lond. B*, pages 1775–1783, 1997.

[7] P. Baran. Reliable digital communications systems using unreliable network repeater nodes. *The RAND Corporation*, pages Report P–1995, 1960.

[8] P. Baran. On distribution communications: Introduction to distributed com- munications networks. *The RAND Corporation*, pages RM–3420,, 1964.

[9] A. Bhadra and M.N.O. Sadiku. Simulation of an atm network using an on-off model. In *Proceedings of the IEEE*, pages 467 –470, April 2000.

[10] N. Bibinagar and Won jong Kim. Switched ethernet-based real-time networked control system with multiple-client-server architecture. *Mechatronics, IEEE/ASME Transactions on*, 18(1):104 –112, feb. 2013.

[11] P. Billingsley. *Probability and Measure.* John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 1986.

[12] Thomas Bohnert and Edmundo Monteiro. A comment on simulating lrd traffic with pareto on/off sources. In *Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, CoNEXT '05, pages 228–229, New York, NY, USA, 2005. ACM.

[13] E. J. Chen and W. D. Kelton. Determining simulation run length with the runs test. *Simulation Modelling Practice and Theory*, 11(3-4):237 – 250, 2003. Simulation in Air Traffic Management.

[14] L. Chiaraviglio, M. Mellia, and F. Neri. Minimizing isp network energy cost: Formulation and solutions. *Networking, IEEE/ACM Transactions on*, 20(2):463 –476, april 2012.

[15] Hyun-Ho Choi, Ohyoung Song, Yeon-Kyung Park, and Jung-Ryun Lee. Performance evaluation of opportunistic vertical handover considering

on off characteristics of voip traffic. *Vehicular Technology, IEEE Transactions on*, 59(6):3115 –3121, july 2010.

[16] Jim Clark and Gene Daigle. The importance of simulation techniques in its research and analysis. In *Proceedings of the 29th conference on Winter simulation*, WSC '97, pages 1236–1243, Washington, DC, USA, 1997. IEEE Computer Society.

[17] Computer Systems Engineering Group at Lawrence Berkeley Laboratory. *Network Simulator 2*.

[18] Mark Crovella and Azer Bestavros. Explaining world wide web traffic self-similarity, 1995.

[19] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Trans. Netw.*, 5(6):835–846, December 1997.

[20] Mark E. Crovella and Lester Lipsky. Long-lasting transient conditions in simulations with heavy-tailed workloads. In *Proceedings of the 29th conference on Winter simulation*, WSC '97, pages 1005–1012, Washington, DC, USA, 1997. IEEE Computer Society.

[21] D. W. Davies, K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson. A digital communication network for computers giving rapid response at remote terminals. In *Proceedings of the first ACM symposium on Operating System Principles*, SOSP '67, pages 2.1–2.17, New York, NY, USA, 1967. ACM.

[22] B.B. Devi. Aggregated equivalency for on-off models. In *Aerospace conference, 2009 IEEE*, pages 1 –7, march 2009.

[23] A. Di Pietro, D. Ficara, S. Giordano, F. Oppedisano, and G. Procissi. End-to-end inference of link level queueing delay distribution and vari-

ance. In *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium on*, pages 503 –510, june 2008.

[24] Allen B. Downey. *Think Stats.* O'Reilly Media, chapter 4 edition, July 2011.

[25] N.G. Duffield and F. Lo Presti. Multicast inference of packet delay variance at interior network links. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1351 –1360 vol.3, mar 2000.

[26] N.G. Duffield and F. Lo Presti. Network tomography from measured end-to-end delay covariance. *Networking, IEEE/ACM Transactions on*, 12(6):978 – 992, dec. 2004.

[27] Hans Joachim Einsiedler, Antonio J. Elizondo, María L. García, Rudolf Roth, Michael Smirnov, Balázs Varga, and Tanja Zseby. Differentiated services – network configuration and management. Technical report, EDIN 0065-1006 Project P1006, January 2001.

[28] Liu Fang-mei, Liu Shu-ru, Zhang Su-zhi, Cai Zeng-yu, and Li Na-Na. The simulation and assessment of network performance based on confidence interval. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 394 –397, may 2011.

[29] George S. Fishman. Estimating sample size in computing simulation experiments. *MANAGEMENT SCIENCE*, 18(1):21–38, 1971.

[30] José R. Gallardo, Dimitrios Makrakis, and Luis Orozco-Barbosa. Use of &agr;-stable self-similar stochastic processes for modeling traffic in broadband networks. *Perform. Eval.*, 40(1-3):71–98, March 2000.

[31] G. Gnugnoli and H. Maisel. *Simulation of Discrete Stochastic Systems.* Science Research Associates, Inc., Chicago, USA, 1972.

[32] Dawn Griffiths. *Head First Statistics.* Head first. O'Reilly, Sebastopol, CA, 2009.

[33] H. Hagirahim. Expression for the probability of packet loss owing to buffer overflow for multiplexing packetised voice. *Communications, IEE Proceedings-*, 153(2):238 – 244, april 2006.

[34] Woon Young Han, Sung Jae Lee, Chi Moon Han, and Seung Hwan Kim. Queueing analysis for an atm multiplexer loaded by cbr and on/off traffic sources. In *Singapore ICCS '94. Conference Proceedings.*, volume 2, pages 760–764, November 1994.

[35] M. Hasib. *Analysis of Packet Loss Probing in Packet Networks.* PhD thesis, Queen Mary, University of London, June 2006.

[36] M. Hasib, J. Schormans, and J. Pitts. Probing limitations for packet loss probability measurement on buffered access links. *Electronics Letters*, 40(20):1315–1316, 2004.

[37] M. Hasib, J. Schormans, and T. Timotijevic. Accuracy of packet loss monitoring over networked cpe. *Communications, IET*, 1:507–513, June 2007.

[38] H. Heffes and D. Lucantoni. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J.Sel. A. Commun.*, 4(6):856–868, September 2006.

[39] M.K. Iqbal, M.B. Iqbal, I. Rasheed, and A. Sandhu. 4g evolution and multiplexing techniques with solution to implementation challenges. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on*, pages 485–488, 2012.

[40] Raj. Jain. *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling.* Wiley, New York :, 1991.

[41] Predrag R. Jelenković and Aurel A. Lazar. Evaluating the queue length distribution of an atm multiplexer with multiple time scale arrivals. In *Proceedings of the Fifteenth annual joint conference of the IEEE computer and communications societies conference on The conference on computer communications - Volume 2*, INFOCOM'96, pages 521–528, Washington, DC, USA, 1996. IEEE Computer Society.

[42] Dong Jin and David M. Nicol. Fast simulation of background traffic through fair queueing networks. In *Winter Simulation Conference'10*, pages 2935–2946, 2010.

[43] E. K. Lada and J. R. Wilson. A wavelet-based spectral procedure for steady-state simulation analysis. *European Journal of Operational Research*, 174(3):1769 – 1801, 2006.

[44] E. K. Lada, J. R. Wilson, N. M. Steiger, and J. A. Joines. Performance of a wavelet-based spectral procedure for steady-state simulation analysis. *INFORMS JOURNAL ON COMPUTING*, 19(2):150–160, 2007.

[45] Hai Lan, B.L. Nelson, and J. Staum. A confidence interval for tail conditional expectation via two-level simulation. In *Simulation Conference, 2007 Winter*, pages 949 –957, dec. 2007.

[46] Ben Lauwens, Jan Potemans, Bart Scheers, and Antoine Van de Capelle. Hybrid simulation of a fifo queuing system with trace-driven background traffic. In *Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, ValueTools '07, pages 40:1–40:10, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for

Computer Sciences, Social-Informatics and Telecommunications Engineering).

[47] A.M. LAW and W.D. KELTON. *Simulation modelling and analysis.* McGraw-Hill Higher Education, 3rd edition, 2000.

[48] J.-H. Lee, J.-M. Bonnin, I. You, and T.-M. Chung. Comparative handover performance analysis of ipv6 mobility management protocols. *Industrial Electronics, IEEE Transactions on*, 60(3):1077 –1088, march 2013.

[49] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2:1–15, February 1994.

[50] C.M Leung. *Non-Intrusive Measurement in Packet Networks and its Applications.* PhD thesis, Queen Mary, University of London, 2004.

[51] Gang Liang and Bin Yu. Maximum pseudo likelihood estimation in network tomography. *Signal Processing, IEEE Transactions on*, 51(8):2043 – 2053, aug. 2003.

[52] Shu-Gang Liu, Pei-Jin Wang, and Lin-Jie Qu. Modeling and simulation of self-similar data traffic. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 7, pages 3921 –3925 Vol. 7, aug. 2005.

[53] S. Lu. *A systematic multi-level abstraction approach to error constrained time-stepped accelerated simulation for MANETs.* PhD thesis, Queen Mary, University of London, June 2006.

[54] A. Ma, J. Schormans, and L. Cuthbert. Aggregation technique for networks with power law traffic and application to accelerated simulation. *Communications, IEE Proceedings-*, 150(3):177 – 183, june 2003.

[55] A.H. Ma, J.A. Schormans, J.M. Pitts, E.M. Scharf, A.J. Pearmain, and C.I. Phillips. Design rules and equivalent capacity for buffering of pareto source. *Electronics Letters*, 36(15):1274 –1275, jul 2000.

[56] A.H.I. Ma and J.A. Schormans. Hybrid technique for analysis of multiplexed power-law traffic in broadband networks. *Electronics Letters*, 38(6):295 –297, mar 2002.

[57] H.I. Ma. *Accelerated Simulation of Power-Law Traffic in Packet Networks.* PhD thesis, Queen Mary, University of London, September 2003.

[58] M.M. Meky and T.N. Saadawi. Degradation effect of cell loss on speech quality over atm networks. In *Broadband Communications, 1996. Global Infrastructure for the Information Age. Proceedings of the International IFIP-IEEE Conference on*, pages 259 –270, 1996.

[59] Michael Menth, Andreas Binzenhöfer, and Stefan Mühleck. Source models for speech traffic revisited. *IEEE/ACM Trans. Netw.*, 17(4):1042–1051, August 2009.

[60] Michael Menth and Stefan Muehleck. Packet waiting time for multiplexed periodic on/off streams in the presence of overbooking. *Int. J. Commun. Netw. Distrib. Syst.*, 4(2):207–229, January 2010.

[61] H. Michiel and K. Laevens. Teletraffic engineering in a broad-band era. *Proceedings of the IEEE*, 85(12):2007 –2033, December 1997.

[62] Sándor Molnár and Attila Vidács. On modeling and shaping self-similar atm traffic, 1997.

[63] D. Nitnaware and A. Verma. Energy evaluation of two on demand routing protocol under stochastic traffic. In *India Conference, 2008. INDICON 2008. Annual IEEE*, volume 1, pages 183 –187, dec. 2008.

[64] Antony Oodan, Keith Ward, Catherine Savolaine, Mahmoud Daneshmand, and Peter Hoath. *Telecommunications Quality of Service Management from legacy to emerging services.* 48. IET Telecommunications, 2003.

[65] M. Ozdem and Y.M. Erten. Performance of distribution networks under vbr video traffic. In *SoutheastCon, 2007. Proceedings. IEEE*, pages 639 –644, march 2007.

[66] A. Pal, J.P. Singh, P. Dutta, P. Basu, and D. Basu. A study on the effect of traffic patterns on routing protocols in ad-hoc network following rpgm mobility model. In *Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on*, pages 233 –237, july 2011.

[67] Kihong Park, Gitae Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Network Protocols, 1996. Proceedings., 1996 International Conference on*, pages 171 –180, oct-1 nov 1996.

[68] B. M. Parker. *Design of experiments for packet communication networks.* PhD thesis, Queen Mary, University of London, January 2009.

[69] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, pages 226–244, 1995.

[70] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, pages 226–244, 1995.

[71] M. Pirahandeh and Deok-Hwan Kim. Co-designing an intelligent doctors-colleagues-patients social network. In *Cloud Computing and*

Social Networking (ICCCSN), 2012 International Conference on, pages 1–4, April.

[72] J. M. Pitts and J. A. Schormans. *Introduction to IP and ATM Design and Performance: With Applications Analysis Software*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2001.

[73] San qi Li and Hong dah Sheng. Discrete queueing analysis of multimedia traffic with diversity of correlation and burstiness properties. In *Proc. of IEEE Infocom'91*, pages 368–381, 1991.

[74] Kavita Ramanan and Jin Cao. A poisson limit for buffer overflow probabilities. In *INFOCOM'02*, pages –1–1, 2002.

[75] J. Roberts. *Performance evaluation and design of multiservice networks: final report of action cost 224*. Commission of the European Communities, Brussels, 1992.

[76] J. Roberts, U. Mocci, and J. Virtamo. *Broadband network traffic: performance evaluation and design of broadband multiservice networks: final report of action cost 242*. Springer, 1996.

[77] Matthew Roughan. Fundamental bounds on the accuracy of network performance measurements. *SIGMETRICS Perform. Eval. Rev.*, 33:253–264, June 2005.

[78] Shriram Sarvotham, Rudolf Riedi, and Richard Baraniuk. Network and user driven alpha-beta on-off source model for network traffic. *Comput. Netw.*, 48(3):335–350, June 2005.

[79] J. Schormans, E. Liu, L. Cuthbert, and G. Stoneley. Analytic technique for accelerating simulation of generic network traffic. *Electronics Letters*, 35, 1999.

[80] J. Schormans, J. Pitts, R. Mondragon, E. Scharf, A. Pearmain, and C. Phillips. Design rules for buffering overlapping pareto processes in packetised networks. *Electronics Letters*, 36(12):1086 –1088, jun 2000.

[81] J. Schormans and T. Timotijevic. Timing performance sampling on packet wireless access buffers. Technical Report JAS-TT-2012, Queen Mary, University of London, 2012.

[82] J. A. Schormans. A hybrid technique for accelerated simulation of atm networks and network elements. *ACM Trans. Model. Comput. Simul.*, 11:182–205, April 2001.

[83] J.A. Schormans, E.M. Scharf, and J.M. Pitts. Waiting time probabilities in a statistical multiplexer with priorities. *Communications, Speech and Vision, IEE Proceedings I*, 140(4):301 –307, aug. 1993.

[84] R. Srikant and W. Whitt. Simulation run length planning for stochastic loss models. In *Simulation Conference Proceedings, 1995. Winter*, pages 1384 –1391, December 1995.

[85] Wim C.M. van Beers and Jack P.C. Kleijnen. Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research*, 186(3):1099 – 1113, 2008.

[86] N. Van Vorst and J. Liu. Realizing large-scale interactive network simulation via model splitting. In *Principles of Advanced and Distributed Simulation (PADS), 2012 ACM/IEEE/SCS 26th Workshop on*, pages 120–129, 2012.

[87] J.S. Vardakas, I.D. Moscholios, M.D. Logothetis, and V.G. Stylianakis. An analytical approach for dynamic wavelength allocation in wdm-tdma

pons servicing on off traffic. *Optical Communications and Networking, IEEE/OSA Journal of*, 3(4):347 –358, april 2011.

[88] A. Wautier, J. Antoine, L. Husson, J. Brouet, and C. Thirouard. Capacity analysis of voice over ip over geran with statistical multiplexing. *Multiaccess, Mobility Teletraffic Wireless Commun.*, 6:25–42, 2002.

[89] W. Whitt. Planning queueing simulations. *Manage. Sci.*, 35:1341–1366, November 1989.

[90] W. Whitt. Simulation run length planning. In *Proceedings of the 21st conference on Winter simulation*, WSC '89, pages 106–112, New York, NY, USA, 1989.

[91] W. Whitt. Asymptotic formulas for markov processes with applications to simulation. *Oper. Res.*, 40:279–291, March 1992.

[92] Ward Whitt. Analysis for the design of simulation experiments, 2005.

[93] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '95, pages 100–113, New York, NY, USA, 1995. ACM.

[94] T.C. Wong, J.W. Mark, K.C. Chua, and Y.H. Chew. Performance analysis of leaky-bucket controlled pareto-distributed on/off sources. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, volume 3, pages 1685 – 1696 vol.3, dec. 2003.

[95] Yi Xu and Wenye Wang. Scheduling partition for order optimal capacity in large-scale wireless networks. *Mobile Computing, IEEE Transactions on*, 12(4):666–679, 2013.

[96] Ke Zeng, Wenli Liu, Xiao Wang, and Songhang Chen. Traffic congestion and social media in china. *Intelligent Systems, IEEE*, 28(1):72–77, Jan.-Feb.

[97] Y. Zhang and W. Li. An integrated environment for testing mobile ad-hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 104–111, New York, NY, USA, 2002. ACM.

[98] Bert Zwart, Sem Borst, and Michel Mandjes. Exact queueing asymptotics multiple heavy-tailed on-off flows. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM 2001*, volume 1, pages 279–288, Piscataway, 2001. IEEE Computer Society Press.