Evaluating Error when Estimating the Loss Probability in a Packet Buffer

Amna Abdul Wahid

A thesis submitted to the University of London in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Electronic Engineering and Computer Science

Queen Mary University of London

United Kingdom

September 2016

Dedicated to Mum, Aliya and Sabri

Statement of Originality

I, Amna Abdul Wahid, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third partys copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature:

Amna Abdul Wahid

Date:

Acknowledgements

Firstly, I want to thank the Almighty for helping me get this far in my life against all odds. Thank You for giving me the determination, courage and wisdom to embark upon and complete this journey.

I want to express my deepest gratitude towards my supervisor Dr. John Schormans who has been exceptionally generous, patient and supportive throughout the course of my PhD. I am really grateful to him for giving me the opportunity to undertake this research and for all the assistance and invaluable advice he has been providing me. It would have been impossible for me to accomplish this work without his guidance. I am indebted to the rest of my research panel, Dr Raul Mondragon and Dr Felix Cuadrado for their useful suggestions and insights on this research work. Special gratitude towards Raul for giving me his valuable time for discussing my research. I am grateful to John and Tijana (as members of the interview panel) for giving me a chance to pursue this PhD.

Most of all, I am extremely grateful to my parents, especially my mother for all the sacrifices they have made for me. I will remain indebted to them for life. I would like to thank my sister for being someone I can count on, always. Thanks to my husband Sabri for being there. Thank you to Khala Shama, Khalu Saeed, Saba, Urwa, Fahad and Bilal, Nazreen and Ghazanfar for being family away from home. Special thanks to Mona for the dress she arranged for my big day during my PhD (you know what I am talking about)!!!

A special thanks to our Head of Research Group Prof. Steve Uhlig and to Dr Gareth Tyson for introducing more opportunities for us to develop our soft skills during our PhD. Big thank you to Dr Vindya Vijaratne and Dr Yue Chen for the opportunities they provided during my time at QMUL.

I would like to thank my colleagues and friends at Queen Mary University of London, Marjan Falahrastegar, Kishan Patel, Jie Deng, Shan Huang, Timm Bottger, Hamed Saljooghinejad, Dr. Rehana Kausar, Dr. Ammar Lilamwala, Dr. Ling Xu, Dr. Bo Zhong, Dr. Manmohan Sharma, Dr. Oleksandr Sushko, Dr. Khaleda Ali, Dr Melusine Pigeon, Dr Yansha Deng, Tian Cheng, Dr Ignacio Castro, Eder Fernandez, Kleomenis Katevas and many others for always being so nice and helpful towards me. I am also thankful to EECS Systems Support for their assistance and support during my time at Queen Mary University of London.

A shout out to my friends off campus for their support, many of whom came into my life because of my being here in London for PhD, namely: Sohaib Qamar Sheikh, Rayhan Naseem, Leire Egana Blanco Argibay, Marta Swierzyna, Parsa Rafiq and Fan Yang, and Adnan Ali, Gian Franco, Muhammad Adnan and Soyiba Jawed, Manmohan, Khaleda, Silvia, Sheherbano, Farida, Raji, Cristian and Balaji.

A final thanks to the Prof. for his guidance and support in life.

Abstract

In this thesis we explore precision in measurement of buffer overflow and loss probability. We see how buffer overflow probability compares with queuing delay measurements covered in the literature [1]. More specifically, we measure the overflow probability of a packet buffer for various sampling rates to see the effect of sampling rate on the estimation. There are various reasons for measurement in networks; one key context assumed here is Measurement Based Admission Control.

We conduct simulation experiments with analytically derived VoIP and bursty traffic parameters, in Matlab, while treating the buffer under consideration as a two-state Markov Chain. We note that estimation error decreases with increase in sampling gap (or in other words precision improves/variance decreases with decrease in sampling rate). We then perform experiments for VoIP and bursty data using NS-2 simulator and record the buffer states generated therein. We see a similar trend of increase in precision with increase in sampling gap. In our simulations, we have mainly considered static traffic passing through the buffer, and we use elastic traffic (TCP) for comparison.

We see from our results that that the sampling error becomes constant beyond certain asymptotic level. We thus look into asymptotic error in estimation, for the lowest sampling gap, to establish a lower bound on estimation error for buffer loss probability measurement. We use formulae given in recent literature [2] for computing the experimental and theoretic asymptotic variance of the buffer state traces in our scenarios. We find that the theoretical and experimental asymptotic variance of overflow probability match when sampling a trace of buffer states modelled as a two-state Markov Chain in Matlab. We claim that this is a new approach to computing the lower bound on the measurement of buffer overflow probability, when the buffer states are modelled as a Markov process.

Using Markov Chain modelling for buffer overflow we further explore the relationship between sampling rate and accuracy. We find that there is no relationship between sampling gap and bias of estimation. Crucially we go on to show that a more realistic simulation of a packet buffer reveals that the distribution of buffer overflow periods is not always such as to allow simple Markov modelling of the buffer states; while the sojourn periods are exponential for the smaller burst periods, the tail of the distribution does not fit to the same exponential fitting. While our work validates the use of a two-state Markov model for a useful approximation modelling the overflow of a buffer, we have established that earlier work which relies on simple Markovian assumptions will thereby underestimate the error in the measured overflow probabilities.

Contents

List of Figures

List of Tables

Glossary

C Service Rate of the Queue

CI Confidence Interval

FI Fisher Information

FIFO First In First Out

h Packet generation rate of source

HTTP Hypertext Transfer Protocol

k Number of unit gap(s) between samples

LCI Lower Confidence Interval

MBAC Measurement Based Admission Control

OF Overflow state

OF' Non-Overflow state

p Probability of going into an Overflow state

PBAC Probe based Admission Control

PDF Probability Density Function

PLP Packet Loss Probability

P(OF) Probability of Overflow

QoE Quality of Experience

r Probability of coming out of Overflow State

R_{On} Data Rate in on state

TCP Transmission Control Protocol

Ton Mean time of a Source in On State (sending packets at rate h)

Toff Mean time of a Source in Off State (not sending any packets)

UCI Upper Confidence Interval

UDP User Datagram Protocol

δ we have denoted Confidence Interval by δ

Chapter 1

Introduction

Network measurement is used to evaluate performance of a packet network. The parameters under consideration are usually delay, jitter and loss probability of packets. In this research, our aim is to study precision and accuracy of loss probability estimation. Two main methods for performance estimation of packet networks are active measurement and passive measurement. In case of active measurement, probe packets are injected into the network traffic and then delay, loss and/or delay jitter of probe packets are measured so as to infer the network performance [5] [6] [7] [8]. Active measurement has the disadvantage that introducing probes in the network interferes with the data and may result in skewing the measurements (for example see [9]). Especially, if we need to measure the performance of a particular network in worst-case conditions (i.e., when the network is pushed to its limits), then doing this is not practical in a live network. In passive measurement [10] [11], the network is passively sampled at intervals so as to estimate the parameter under consideration[1]. Employing passive measurement does not introduce any interference in the network so it seems the most desirable of the choices. However, passive measurement may incur additional processing cost of the sampled data, or cost of communicating the sampled data to the processing hub, etc [12]. Also, passive sampling only gives measurements from one router at a time.

[1] Note that in passive measurement, we take measurements from interface of a physical router.

Whether probing or sampling the network, packet losses in a network occur rarely, and these occur in bursts [13]. Measuring the probability of a phenomenon like Packet Loss Probability (PLP) is difficult because of its rarity of occurrence. Not only are we dealing with small fractions (because of rarity of the event), the events (of bursts) are far apart and clumped together. There is indication in literature of the difficulties with classical statistics and the associated problems of parameter estimation in the presence of very small numbers [14] [15](See especially [15] Section 4.3.2). There have been various studies [3] [16] [17] [18] and [14] aimed at determining the sampling rate and sample size for measuring packet loss and queuing delay. Parker et al [3] [16] use the Fisher Information (FI) metric for giving the upper bound for sampling rate, for various network states. The authors of [3] [16] use FI to find exact sampling points that are not equi-spaced (that is, not at a fixed rate). However we don't use the FI approach as it is computationally far too intensive (as explained in chapter 2 section 2.3). It is to note that Parker et al are only able to use the FI approach for short intervals between samples due to its computational complexity. We discuss the work [3] further in section 2.3.2.

Roughan [1] shows theoretically that sampling or probing at a lower rate over a longer interval provides more accuracy than sampling faster in a shorter interval. Note that the work [1] gives bounds for measurement regardless of the measurement technique of sampling or probing. So henceforth we use probing for referring to both probing or sampling while discussing [1]. The work [1] observes that inaccuracies result in estimations using probing close together due to correlation between successive probes; therefore the probes need to be a certain distance apart to avoid interference. The first motivation question we address is how fast we can sample for a given fixed number of samples. Roughan [1] gives analytical upper bounds on probing rate when estimating mean delay or mean queue fill. In this thesis we evaluate the upper bound on sampling rate when targeting buffer overflow probability (which affects packet loss probability). We use an analytical approach to find the upper bound on the accuracy of the estimate, in terms of the sample size and gap between samples. In our work [19] we have evaluated the accuracy of the loss estimate and its dependency on the sampling gap when constrained by a fixed number of samples, while modelling the buffer at a network node as a two-state Markov Chain (MC). We have argued that beyond a certain asymptotic level, the variance of the estimate stays virtually unchanged with respect to the sampling gap. In this thesis, we have explored asymptotic variance of an MC to obtain the least variance to expect in our estimates, for various network parameters. In order to obtain the least variance in the estimator of buffer overflow probability, we have used an analytical approach [2].

Taking this research further, we have explored the buffer overflow probability when modelling our network scenario in the NS2 simulator. We notice that the accuracy in estimate when measuring loss probability of a buffer modelled in NS2 indeed increases with the increase in sampling gap up to a certain point. This is consistent with our earlier results [19]. However, we see from the simulation results that the sojourn time distribution (in overflow and non-overflow states) of a buffer is actually not geometric (one would expect probability distribution of states in a two-state MC to be geometric). This means that while a two-state MC is a good approximation for theoretical study of queuing behaviour in packet networks, it may be a too simple as a model in practice. Estimations of buffer overflow probability when assuming the buffer model to be accurately represented by a two-state MC may underestimate the parameter under consideration. We discuss this in our findings in chapters 5 and 6.

1.1 Motivation of this Research

Optimal sampling of a network for measurement of parameters like mean delay and loss probability is an active area of research [1] [14] [12] [20] [21] [22] [16] [18] [8] [23] [24] [25] etc. Academic research has been exploring measurement patterns including packet train models [13] and provably optimal patterns of sampling that are built on a mathematical model like Poisson probing [14] and periodic non-uniform probing scheme NIMASTA [26]. Sometimes these approaches suggest a deterministic (fixed) sampling rate, while other studies conclude that non-deterministic sampling is optimal (see [27] for comparison of a few of these). We seek an optimal sampling rate solution that takes into consideration the following:

- If we sample at a large gap between samples, for a large enough sample size the accuracy of the estimated PLP will improve. However taking a large number of samples may be computationally expensive (in terms of processing time and the amount of data being used). Also, in case of an application scenario where we are constrained with time, fitting many probes/samples close together may result in correlation of the probes/samples, as argued by Roughan [1].
- If we take fewer samples and keep the samples far apart from each other (low sampling rate), then for larger gaps between samples, we could be taking too long to collect all the samples and then

estimate the parameter. This could render the estimation to be outdated and thus of no use.

An example application scenario for estimation within time constraints is Measurement Based Admission Control (MBAC) [28] [29] [30], where making decision within a certain time frame is vital. In other words a good estimation of PLP through sampling is challenging because we need accurate predictions in timely and low cost manner. The process for finding the best sampling rate will provide an in-depth knowledge on packet data network and at the same time this sampling rate distribution can be a useful tool in Traffic Management. MBAC is a mechanism for traffic management where a new user (source) is only allowed in if there is enough space for admitting another user (source) while adhering to the guaranteed requirements of existing users in the system. MBAC algorithms are required to take quick decisions; therefore, sampling the network at a high rate so as to measure the network state and take a decision in optimal time is desired. Examples for this include [31] [6]. This type of MBAC algorithms use a time window for measuring the network parameter and monitor the average value of the parameter being measured over each measurement window. We discuss these further in Chapter 2 section 2.5.

1.2 Methodology and Computational Methods

The results included in this thesis are either analytical or through simulation experiments. It is to note that we have not conducted real network experiments by setting up a test bed. This is because, on one and, there are a lot of factors that might impact the loss rate in a real network, and this would make this research much more complex. On the other hand, a test network could not reflect a real network traffic, and may limit applicability of this work to real network, anyway. NS-2 simulator has been used for simulating the network scenario which emulates conditions in a real network. This thesis discusses results from simulations that we have performed using Matlab and NS2 simulator. We generate buffer state evolution in Matlab using analytically derived parameters and then perform sampling. The results from Matlab simulations validate theory and give a relationship between sampling rate and error in estimate of buffer overflow probability. To bring our simulations closer to real network conditions, we have performed queuing simulations in NS2, considering queue overflows at a bottleneck. We have altered the source code of NS2 to record the data we require. The buffer states ae noted after fixed time intervals. This buffer state data is then processed using Matlab code. Entropy encoding is used in order to compress the data generated in NS2 for ease of porting it into Matlab.

1.3 Contributions

Main contributions of this thesis include the following:

- We evaluate for certain target traffic scenarios, when sampling the buffer states with a fixed number of samples, the relationship between gap between samples and precision of measurement. The precision increases with increase in sampling gap.
- We present a new approach to computing the lower bound on the measurement of buffer overflow probability, when the buffer state is modelled as a semi-Markov process.
- Comparing results from modelling in MATLAB to results from NS-2 which is closer to real networks, we show that theory overestimates precision of estimate of loss probability. That is, when considering that the buffer states represent a semi-Markov process, the results from NS2 show the estimation error to be much larger than that predicted by theoretical results in MATLAB. Hence we need more samples than suggested by theory, for precise estimation.
- We review the limitation of a two-state MC as a model for representing states of a buffer subjected to bursty traffic.
- We show from our experimental results that, while there is a relationship between precision of estimate and the sampling gap, the same is not true for sampling gap and accuracy of the estimate.

1.4 Organization of the Thesis

This thesis is organized into six additional chapters as follows. **Chapter 2** introduces literature related to this thesis. We focus on literature that uses MC modelling. We also discuss previous literature on sampling rate and pattern for measurement of network parameters. **Chapter 3** describes some more background theory and terminology relevant to this thesis. We describe how buffer at a network node is modelled as a two-state MC. We include how to analytically parametrize the transition probabilities when modelling buffer as a network node and subjecting it to an aggregation of on-off traffic. We also include the simulation parameters used in this thesis and how we evaluate our results. **Chapter 4** enumerates our experiments conducted in Matlab and their results. We also compare our results (for estimation of buffer overflow probability) with results in literature for estimation of delay. In **Chapter 5** we cover computation of asymptotic variance of the estimated loss probability for our experiments. We describe the method and then show results both from theory and from experimental results. **Chapter 6** discusses our experimental results from NS2 simulations. **Chapter 7** concludes this thesis and discusses future directions.

1.5 Publications

A. A. Wahid and J. Schormans, "Bounding the maximum sampling rate when measuring plp in a packet buffer," in *Computer Science and Electronic Engineering Conference (CEEC), 2013 5th*, pp. 115–118, Sept 2013.

A. A. Wahid and J. A. Schormans, "Bounds on accuracy when estimating the loss probability in a packet buffer," in *Computer Science and Electronic Engineering Conference (CEEC), 2015 7th*, pp. 24–28, Sept 2015.

Chapter 2

Related Work

This chapter covers literature relevant to te work presented in this thesis. This research explores the relationship between sampling gap and precision in estimate of buffer overflow probability, when constrained with a time window and number of samples. Hence this chapter first looks at a pertaining to sample size for estimation of network parameters in section 2.1. Next, literature pertaining to modelling buffer at a network node as a two-state Markov Chain is covered in section 2.2 (note that we have assumed the two-state MC model for representing the states of a buffer at an access node). Section 2.3 covers the gap between samples and sampling rate. The two subsections in this section (on work by Roughan [1] and on use of FI for computing optimal sampling gaps) are of significance to this research. In Chapter 4 we compare our work with that of Roughan. And in general, the sample sizes that we have used through out this research are comparable with that used by Parker et. al. (section 2.3.2). Section 2.4 covers literature on challenges involved with computation of error in estimation, when small fractions are involved, as is the case in this research. Finally, section 2.5 covers literature on an application area of this research, that is, measurement based admission control.

2.1 On Choice of Sample Size

This section covers literature on choosing sample sizes for network performance measurement. Albin [32] indicates the need for changing the sample length as the traffic intensity increases, thus increasing the variability in service. They use samples sizes from 60,000 to 1000,0000 samples for load of 30% to 90% for estimating equilibrium queue length. Despite larger sample sizes for higher load, their results show that lower load values give lower error in estimation. Whitt [33] proposes to use $1/(1-\rho)^2 \times arrivals$ as the sample size for getting a uniform standard error. On the other hand we are not looking to keep the error constant for all kinds of load values, rather, we are looking into choosing sampling rates for obtaining least error in estimation for different load values. That is, the error does not have to be uniform, but it should be low, for the particular load. Also it is clear that these numbers of samples quoted in literature [32] are too large[1] for computing loss probability in a network, say, during a busy hour or when making decision in a short time span is desirable [34] [35]. Parker et al [3] choose smaller sample sizes for evaluating loss probability in a network. Parker et al look into design of experiments for obtaining optimal sampling gaps (which translates to sampling rate) when treating the buffer at a network node as a Markov Chain. We discuss their work in further detail in section 2.3.2. We choose sample sizes slightly larger than that chosen by Parker et al (we find that sample sizes used by Parker et al are too small for our needs).

2.2 Related Work - Use of Markov Chains in Network Modelling

Traditionally Markov Chain (MC) models have been used for modelling the packet delivery process, when doing performance evaluation of network applications. The Gilbert Model [36] has been used to describe burst error model for a communication channel. The Gilbert Model represents a two-state Markov Chain, and is sometimes referred to as Simple Gilbert Model (SGM). SGM has been used in previous studies for error control in packet audio and video [37] [38], for forward error correction [39], and for evaluating Video Streaming applications [40]. Yajnik et al [41] [42] discuss use of a two

[1] Consider a network scenario where an access node is sending out packets at 1Mbps. With a VoIP packet size of 47 bytes, the service time for a packet in this case is 0.000376 seconds. If we were to take 1,000,000 successive samples as suggested by Albin [32] for 90% load, it will take 376 seconds (more than 6 minutes). If we were to follow Whitt [33], we would have to take 3,000,000 for 90% load in a time duration of 19 minutes. Taking into consideration correlation between successive samples, as pointed out by Roughan [1], we would have to keep a gap between these samples. Even a gap of 10 units will increase the sampling window to 60 minutes(or 190 minutes in case of sampling according to Whitt)

state Markov Chain and higher order MC for describing packet loss correlation in Multicast backbone network. *Yajnik et al [41] [42] also discuss that in case of less bursty traffic, the losses are less correlated.* An extension of SGM was proposed by Elliott [43], referred to as Gilbert Eliott model. In the Gilbert Eliott model, both states (good and bad state) produce losses with different probabilities, with the good state producing loss with a low probability. The SGM has been popular for its simplicity and ease of implementation and understanding. A two-state Markov Chain has also been employed for describing the states of a buffer at an access node in a network by Hasib et. al. [44] (and references therein), Wijaratne et. al. [45]. More recently by Parker et. al. [3] [16] have assumed the simplified model of two states of a buffer (being in overflow or not being in overflow) to follow a two state Markov Chain model. They use this assumption for buffer states, and then explore design of experiments for suggesting optimal sampling gap in order to measure loss probability in a network node. Similar to [44] [45] [3] [16], we model the states of a buffer at a network node as a two-state Markov Chain. We discuss the work of [3] in more detail in section 2.3.2. Also, because of the significance to our research, we elaborate on the use of Markov Chain for modelling the states of a buffer at a network node, in section 3.2. Barbu et al. [2] express that a discrete time semi-Markov process is applicable to many disciplines since the distribution of sojourn times in different states (in a semi MC) can be any distribution. They give a formula for computing the asymptotic variance of a semi-Markov process, sampled over a long amount of time. We use this formula to derive the theoretical asymptotic variance of the estimated buffer overflow probability P(OF), for our network scenario. We then show that our numerical results for asymptotic variance conform to the theory in [2], when modelling the buffer as a two-state MC in Matlab.

2.3 Probing/Sampling Rate and Pattern

Often network is probed using a uniform pattern, i.e., having a fixed time between samples. We also employ uniform sampling in this research. In the academic literature, several studies have evaluated non uniform probing patterns, including Poisson probing (PASTA) to avoid bias presented by Wolff [14] and employed, among others, by Strauss et al [18]. The PASTA approach suggests introducing probe packets using a Poisson sampling. It is argued that when using this form of probing, the mean of inter-arrival time between probes as they are received at the receiving end, can accurately estimate the waiting times for the rest of the packets on the network path being measured. Baccelli et al [26] argue that Poisson probing is not unique in avoiding sampling bias. They show that PASTA does not minimise estimation variance in the measurement. They state that lack of a minimum distance between probes in Poisson probing is a significant contributor to the estimation variance. They present an alternative periodic non-uniform probing scheme in which the inter-arrival time of the probes is correlated, and a minimum distance between probes is maintained so as to ensure non-intrusive probing (probing with minimal intrusion). They call the scheme NIMASTA (Non-mixing Arrivals See Time Averages). Baccelli et al explore train probes as a better measure for delay and loss estimation in [13], while developing further on NIMASTA. Also in their most recent work [21], they go on to show that loss probability is better measured with isolated probes than probe trains or chains. The work [46] also points out the inapplicability of PASTA to the present internet traffic, owing to the fact that internet traffic is bursty in different time scales and not all types of traffic can be modelled by Poisson process. Another non-uniform probing scheme mentioned in literature is Geometric probing [12]. The authors conduct experiments to show inaccuracy of Poisson-modulated probing and introduce geometric probing as a solution. They send closely spaced probes and repeat the process according to set rules so as to measure the length of loss period. In our work, we use uniform sampling/probing for estimating overflow probability in a buffer, owing to its simplicity and popularity in recent literature [24] [3] [16] etc.

Hasib et al [44] give a formula for the number of samples required to calculate loss probability in the buffer, taking into consideration the traffic characteristics, the acceptable packet loss probability (PLP) and desired accuracy. Also more recently Parker et al [3] [16] in their EPSRC sponsored DoENET project explore probing/sampling for network measurement as a numerical experiment and use experimental design principles from statistics for finding optimal sampling gaps for different network load scenarios.

2.3.1 Bounds on Sampling Rate as expressed by Roughan [1]

Roughan [1] shows theoretically that sampling with a lower rate, over a longer interval provides more accuracy than sampling faster in a shorter interval, when finding mean values in M/G/1, M/M/1 or M/D/1 queuing models. The work [1] observes that inaccuracies result in estimations using probing due to correlation between successive samples; therefore the samples need to be a certain distance apart to avoid bias due to these correlations. Roughan [1] gives analytical upper bounds on sampling rate when estimating mean delay or mean queue fill. In this research we evaluate the upper bound on sampling rate when targeting overflow probability, and we do this using simulation as well as analysis.

2.3.2 Use of Fisher Information Matrix, by Parker et al, for Optimizing PLP Estimation by Sampling

Fisher Information (FI) was proposed by R A Fisher [47] for designing experiments that involve dynamic results, i.e., experiments that are not exactly repeatable. It has been widely used by biologists in sampling experiments. When several sets of samples are taken from a population, FI is computed on each of the sets of samples, and the set that gives the highest FI value, is considered to represent the population most closely.

As stated earlier, Parker et al [3] [16] explore how to optimize the period between samples (or probes) when sampling the states of a network (by concentrating on the buffer at a bottleneck node). They use Fisher Information metric for determining the optimal sampling rate, for various network states. The bottleneck node can be diagrammatically represented as in figure 2.1.

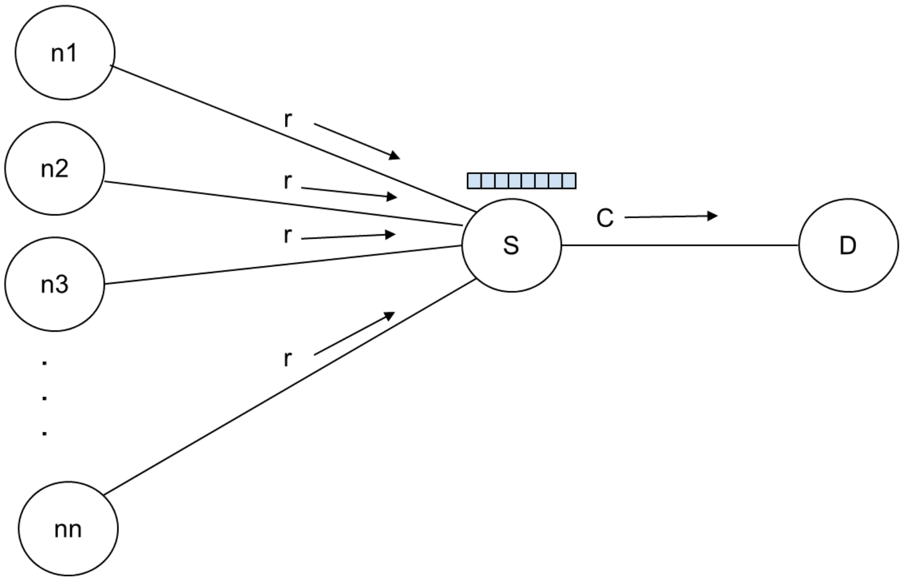


Figure 2.1: Network setup showing buffer at a network bottleneck node S, receiving aggregate traffic of *n* sources

They represent the states of a buffer at a bottleneck node as a two state Markov chain (figure 2.2) having the transition matrix given by equation 2.1. Their work [3] provides guidelines as to how far apart samples can be taken (i.e. the sampling rate) for given transition probabilities of the buffer states, when using a fixed number of samples.

$$\begin{pmatrix} 1-p & p \\ r & 1-r \end{pmatrix} \tag{2.1}$$

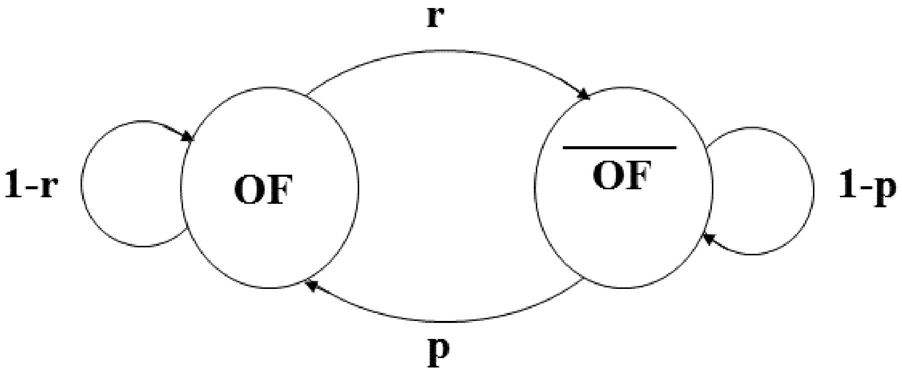


Figure 2.2: A two-state Markov Chain

Here p is the probability for the Markov Chain transition into overflow (OF) state, and r is the probability of the Markov chain to transition into non-overflow (OF') state (Explained in detail in Chapter 3 section 3.2). The queue is sampled at time instant 0 and further T number of samples are taken at intervals $k, 2k, 3k, ..., Tk$ so that the total number of samples is $T+1$. This means that $k = 1$ is sampling every interval, $k = 2$ means sampling every other interval, and so on. When the traffic load on a buffer in a network node increases and decreases with time, this results in the buffer filling up and having no more capacity and thus overflowing, or going up in capacity and not overflowing any more. They treat p, and r as the unknown parameters to be estimated. Given p and r, PLP is computed follows as:

$$PLP = \frac{p}{p+r} \tag{2.2}$$

An observation (sample) y of the Markov Chain is a vector of length $T+1$. In order to find the Fisher Information matrix for the observation y obtained when using a sampling period of $k = 1$ (or in other words, taking T samples with the smallest possible period (this may be set by some aspect of the technology). The likelihood function $L(\theta|y)$ is derived for each of the transition probabilities.

Eg, the likelihood function for k =0 is:

$$L(\theta|y) = P(Y_0 = y_0, Y_1 = y_1, Y_2 = y_2, ...Y_T = y_T) \tag{2.3}$$

$$L(\theta|y) = P(Y_0 = y_0)P(Y_1 = y_1|Y_0 = y_0)P(Y_2 = y_2|Y1 = y_1, Y_0 = y_0)...P(Y_T = y_T|Y_{(T-1)} = y_{(T-1)},, Y1 = y_1, Y_0 = y_0) \tag{2.4}$$

By Markov principle, i.e. the probability of the current state only depends on the previous state, equationA.3 simplifies to

$$L(\theta|y) = P(Y_0 = y_0)P(Y_1 = y_1|Y_0 = y_0)P(Y_2 = y_2|Y1 = y_1)...P(Y_T = y_T|Y_{(T-1)} = y_{(T-1)}) \tag{2.5}$$

$$L(\theta|y) = P(Y_0 = y_0)\prod_{j=0}^{T-1}(Y_{(j+1)} = y_{(j+1)}|Y_j = y_j) \tag{2.6}$$

$$L(\theta|y) = \frac{(y_0 p + (1-y_0)r)}{(p+r)}\prod_{j=0}^{T-1}(1-y_j)[y_{(j+1)}p + (1-y_{(j+1)})(1-p)] + y_j[y_{(j+1)}(1-r) + (1-y_{(j+1)})r] \tag{2.7}$$

Each of these likelihood functions is then differentiated twice followed by taking Expectations and mathematical simplifications. Fisher Information matrix is then formed by substituting the corresponding simplified expressions obtained for each transition probability into the corresponding positions in the transition matrix 2.1.

Equation **??** is the FI matrix for realizations of the Markov chain when sampling period k is 1.

$$I(p,r) = \frac{1}{p+r}\begin{pmatrix} \frac{1}{p} - \frac{1}{p+r} + \frac{Tr}{p} + \frac{Tr}{1-p} & \frac{-1}{p+r} \\ \frac{-1}{p+r} & \frac{1}{r} - \frac{1}{p+r} + \frac{Tp}{r} + \frac{Tp}{1-r} \end{pmatrix} \tag{2.8}$$

Similarly matrices are constructed for $k = 2$, $k = 3$ and so on. Parker et al [3] then use the D-criterion of optimality for choosing which k is better. D-Criterion requires finding determinant of the matrices for different k values and the maximum determinant identifies the best k for a given transition probability pair (p, r).

In earlier work on network measurement, FI has been used to estimate flow size [15]. Among the most recent literature on using FI metric for estimation is the work by Hua et al [48] where optimal error-estimation codes for bit-error rates is explored. The generalized Error Estimation Code (gEEC) generation technique by [49] uses the FI metric to encode as much information as possible in the EEC so as to make it more efficient. In short, they use FI to fine tune the encoding aspects of an older technique (EEC) for error estimating code generation, and also combine the decoding techniques for two older techniques (EEC [49] and EToW [50]). The details of these pieces of work are beyond the scope of this thesis, but this work is a good example of using FI based analysis to choose between techniques in Computer Networks.

Riberio et al [15] use FI for finding the minimum number of samples needed for accurate flow size estimation from the output of a packet generation process. More work on flow size estimation using FI was followed by Tune et al [8] where they use flow-sampling instead of packet sampling, which, although more computationally expensive, yields to be a better estimator of flow size estimation. They [8] make this deduction using Fisher Information analysis, and propose a hybrid sampling technique combining packet sampling and flow sampling. Further details can be seen in the paper [8]. Although analytically FI approach is promising, it is computationally expensive (as obvious from the derivation above) to find the FI for arbitrary values of sampling rate. As we note in Chapter 6, the assumption of representing the buffer as a two sate Markov Chain might not be applicable in real network scenario. We focus on simulations for searching for asymptotic level of rate that gives accurate estimate of the parameter under consideration, in our case buffer overflow probability.

We express the error in our estimation from sampling as relative mean square error, and confidence interval (CI). Section 3 describes CI.

2.4 Evaluating Error

Estimation of rare phenomenon (like PLP or overflow probability) is hard and may lead to errors in estimation because of very small probability of occurrence of such rare event [51] [52]. One way of evaluating the accuracy of an estimate is using RMSE (root mean square error, which is the square-root of expected value of the squared difference between estimator and the parameter - See Chapter 3 section 3.1.4 for further details) [51] [53] [54]. When estimating a parameter by sampling from a population, the estimate is often expressed with a confidence interval.

The work [55] by Sukasih et al covers the problem with confidence intervals for estimating small proportions. They discuss that when dealing with small proportions the CI value is either too large or too small, and thus the usual CI method that uses the normality assumption results in inaccuracy. They perform experiments with real data to compute CI using the various alternative approaches and compare their performance. We find that none of these approaches are applicable for us with the CI error results. This is because, (as we explain further in Chapter 6) due to limitations on our sample size and the fact that the distribution of bursts in our NS2 experiments is different from what we expect from our theoretical assumption, we rarely obtain an overflow among our samples. We use repeated experiments and we have used CI relative to the mean estimate for evaluating error in our estimates. Computing error relative to an estimate has been favoured by Whitt [33].

2.5 MBAC

An application area of timely estimation with reasonable accuracy is measurement based admission control (MBAC) [28] [29] [30] [31] [6], [34] [56] [57] where making decision within a certain time frame is vital. MBAC is a mechanism for traffic management where a new user (source) is only allowed in if there is enough space for admitting another user (source) while adhering to the guaranteed requirements of existing users in the system. Of special interest to us are systems where loss probability is the criterion (or one of the criteria) for admission decision. Examples for this include [34] and [4] [35]. Mas et al [4] present a scheme for admission control based on probes for multicast scenario (MPBAC). The decision to admit a new source is based on the loss ratio of probes. The host sends probe packets before starting a session and the root node of the multicast tree makes decision based on loss ratio of probes. The receiver makes the accept/reject decision for session after computing a confidence interval on the loss estimated session loss probability (assuming a normal distribution on the probe loss ratio). Sathik et al [35]) enhance the MPBAC to make the scheme more efficient. Probe packets are sent by a sender for 1 second to 10 seconds and loss ratio is computed. Sakellari and Gelenbe [34] propose an MBAC scheme to improve performance of a self aware network. They propose a centralised algorithm that evaluates the admission requirements of one user at a time, and maintains a FIFO queue for subsequent user requests, until a decision on the current user is made. Admission criteria involves a decision on whether the delay, jitter and loss requirements of the user can be met. They mention the need of finding optimal probe rate and probing time to improve their algorithm. In our research we look into optimal sampling/probing rate within a sampling/probing time window.

Chapter 3

Background Theory and Concepts

Consider a hypothetical scenario of a small network with UDP traffic (as an example), where measurement of loss probability through sampling is required. Further consider that there is a limited budget of the number of samples to use, and there's a constraint on the time window within which the measurement needs to be completed. Then comes the question of how to space out the samples so as to perform a precise measurement. If all the samples are stacked together consecutively, then a quick estimation can be made. However, this measurement will have less precision as remaining traffic during the allowed time window is not sampled. On the other hand, samples could be spaced out so as to cover the entire allowed time window. It needs to be seen that whether spacing the samples out to cover the entire time window, or having less gap is better. This research looks at the effect of different sampling gaps on precision in estimation of the loss probability at a network node. In terms of traffic types that are considered in this thesis, we look at UDP traffic, including VoIP traffic and bursty traffic. By bursty traffic, we mean traffic that is a multiple of VoIP traffic of the order of 10. That is, bursty traffic that we have considered is ten times more bursty than VoIP, as considered in previous literature. The parameters used for UDP traffic are given in section 3.4 table 3.1. Later in the thesis, we have considered TCP traffic, just for completeness. Parameters for TCP traffic under consideration are given in chapter 6 section 6.9.

3.1 Some Concepts and Background Theory

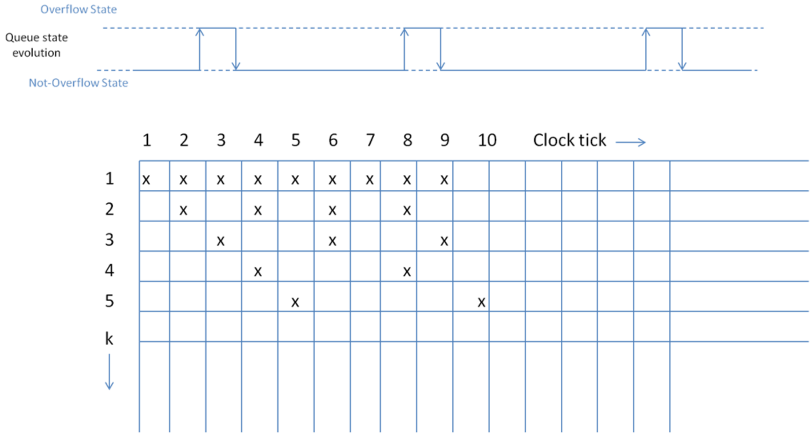
Having described the context of this work in the previous chapter, this chapter covers some more background theory. Terms that are essential to understand the background are also covered here.

3.1.1 Clock Tick

We use a discrete time base in our simulations, for simplicity. We have assumed the clock tick to be the unit of time during which one packet can be served. Every time one packet is lost, it happens because the queue is in overflow state. Also we have approximated *PLP* as the probability of buffer being in overflow state. That is, $P(OF) = PLP$ as each overflow state results in the loss of one packet (since our clock tick is service time for one packet). Therefore the variance in $P(OF)$ is also the variance in *PLP* in our results.

3.1.2 Sampling Gap *k*

We use the term *sampling gap* and *sampling rate* extensively throughout this report. By *sampling gap* we refer to how many clock ticks elapse between consecutive samples. The sampling gap is referred to as k. Thus $k = 1$ means sampling every clock tick starting from the instant of first sample. If our sample size is represented by T then the sampling window w (or the duration during which we perform sampling) would be $w = k \ \times T \ \times clock\ tick$. Thus for example of a clock tick is $4ms$ long and if our sample size T is same as the total duration (in clock ticks) of the simulation of the buffer, say n, i.e. $T = n$, then the sampling window duration would be $4ms \times T$, and we would be sampling the whole set of outcomes. In this case our estimate will be perfect, i.e. estimated metric = actual metric.



	1	2	3	4	5	6	7	8	9	10
1	x	x	x	x	x	x	x	x	x	
2		x		x		x		x		
3			x			x			x	
4				x				x		
5					x					x

Figure 3.1: Schematic representation of sampling gap k. The cross x in the diagram represents a sampling instant. When taking sample size T as 10, the sampling window is $T \times k$ duration. The sampling window varies with k

3.1.3 Sampling Rate

Sampling rate in our work is to be calculated as the reciprocal of the sampling window duration in time. If the desired sampling gap is $k = 10$, number of samples $T = 5$ and clock tick is $4ms$, then sampling window w = 200ms or $w = 0.2sec$. And thus the sampling rate would be $0.04samples/sec$. If for the same clock tick and sample size, the desirable $k = 20$ then $w = 400ms$ and thus the sampling rate will be 0.02 samples per second. Thus higher values of k mean lower sampling rate.

3.1.4 How we quantify error in estimation

3.1.4.1 Relative Error

Initially we have used Mean Square Error [53] [54] relative to the actual value to be observed (Relative MSE, in short Relative Error) in some of our results. By Relative MSE we mean

$$RelativeError = \frac{MeanSquaredError(estimate)}{P(OF)_{actual}} \tag{3.1}$$

Here $P(OF)_{actual}$ is the actual known overflow probability of the buffer, and $MeanSquaredError_{estimate}$ is given by

$$MeanSquaredError(estimate) = \frac{\sum_{i=1}^{n}(P(OF)_{estimate_i} - P(OF)_{actual})^2}{n} \tag{3.2}$$

$P(OF)_{estimate_i}$ is the estimated P(OF) for the ith iteration of the experiment We have also used Confidence Interval Error explained below.

3.1.4.2 *Confidence Interval Error*

It is an interval estimate for a sample of population that is used to indicate the reliability of an estimate. The confidence interval is a range between the upper confidence interval (UCI) and the lower confidence interval (LCI), such that

$$Pr(LCI \leq estimat \leq UCI) = 1 - \beta \tag{3.3}$$

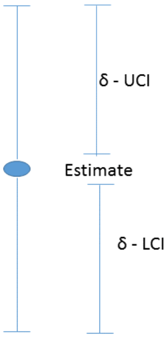


Figure 3.2: Confidence Interval of Estimate, UCI is the upper confidence interval and LCI is the lower confidence interval

Here $1 - \beta$ is the confidence level. A *Confidence Level* indicates how frequently the confidence interval contains the parameter in a series of experiments. The confidence level is often represented as a percentage. That is, as $100 * (1 - \beta)\%$. For example a confidence level value of 90% means that there is a 90% chance that the parameter of interest lies within the interval. Typically the confidence levels chosen are between 90% and 98%. e.g., in this research we use the Confidence Level of 95% that the estimate lies within the interval. We have used CI Error and Relative CI Error in some of our results. CI Error referred to as δ is the Confidence Interval (upper or lower) and is computed as as follows

$$\delta = z \times \sqrt{\frac{MSE(estimate)}{T}} \tag{3.4}$$

Here the coefficient z follows from the cumulative normal distribution function and the value for z for 95 confidence level is 1.96. We have used $z = 1.96$ throughout this thesis. Relative CI Error means CI relative to the estimate of $P(OF)$ i.e.

$$RelativeCIError = \frac{\delta}{P(OF)_{estimate}} \tag{3.5}$$

3.2 Modelling the Buffer as a Two State Markov Chain

Buffer states can also be simplified into a two-state MC model. In a two-state MC, the probability of the next state depends only on the present state. A simple two state MC model can be diagrammatically represented as in Fig 2.2 in Chapter 2. The traffic offered at a buffer in a network node comes from several sources and varies in volume, with time. During peak times when more sources are active, the packet queue in a buffer fills up and then the queue declines in size during periods when the number of active sources goes down (Figure 3.3). Thus the buffer at a network node has been represented as a two state MC in previous literature [44] [3] [16] [58], where the state when the buffer is filled up and starts overflowing is the *overflow state* (OF) (no capacity to hold more packets) while the state when the buffer is not completely filled, is the *non-overflow* state $(OF)'$ (when the buffer is not full, and therefore theres capacity to hold more packets). In this research we have modelled the buffer as a two-state MC. Parker et al [3] [16] also adopted this as a simple model to start with. The transition probabilities for the MC are: the probability of moving from non-overflow state to overflow state p and the probability of moving from overflow state to non-overflow state r. The probability of staying in an overflow state is $1-r$ and that of staying in non-overflow state is $1-p$. Thus the one step transition matrix of the system is given by equation 2.1. Fig. 3.4 shows the visual representation of the queue state evolution against the two-state MC.

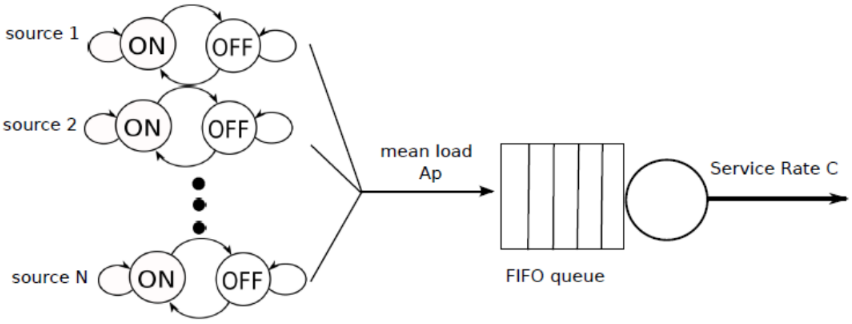


Figure 3.3: N-on-off/D/1 Model of Queue

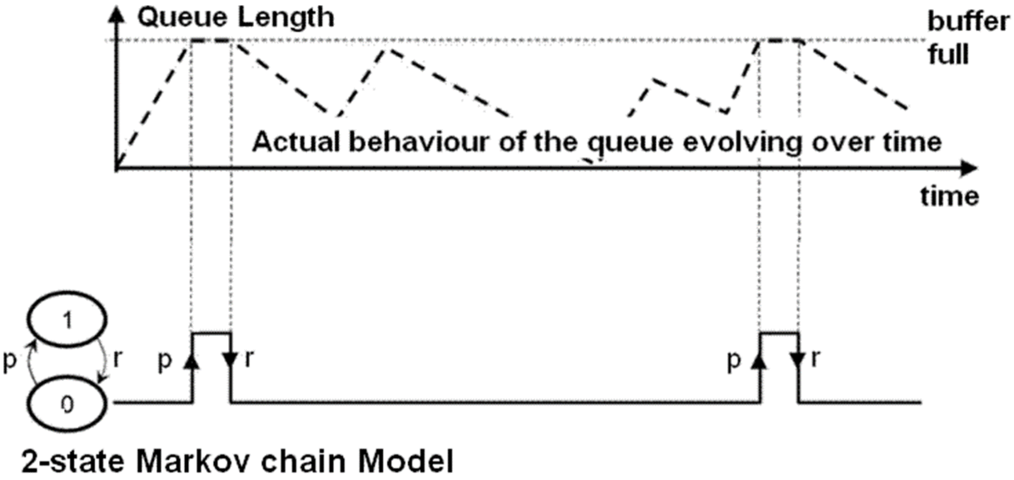

Queue Length
buffer
full
Actual behaviour of the queue evolving over time
time
1
0
p
r
p
r
p
r
2-state Markov chain Model

Figure 3.4: Mapping two-state MC against queue state evolution over time

When the overall number of active flows exceeds the service rate for an amount of time required to fill the buffer, the queue starts overflowing. In the two state MC model, the probability with which this overflow state is entered is p. As the number of active flows goes down, after some time, the queue goes back from overflow period to not being in overflow, and the probability of transitioning into this state is r. After every clock tick, the queue either stays in its current state, say overflow state with probability

$(1-r)$, or it goes into the other state, say non overflow state, with transition probability r.

3.3 Multiplexing N On-Off Sources as a single On-Off Source

Previous literature, including [44] [58] [45] [59] [20] [60] have used aggregate of multiple on/off sources for modelling traffic arriving at a buffer on a network node. That is, a number of homogeneous on-off sources are aggregated (multiplexed) together into a single on-off traffic source feeding a FIFO buffer. A diagrammatic representation of multiplexing several on-off sources this is given in Figure 3.5

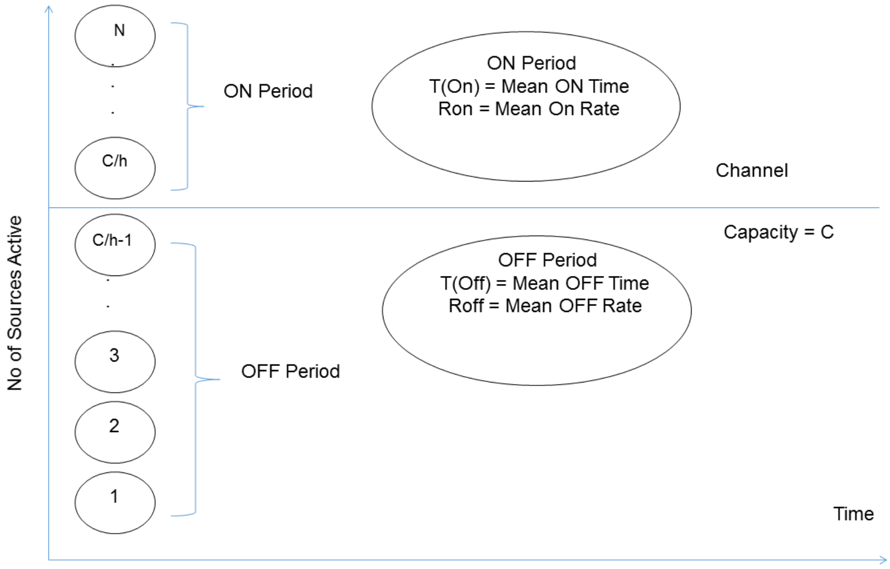


Figure 3.5: Two State Model of Aggregation of N on-off Sources

Elaborating further on the simplified model, N homogeneous sources are multiplexed into a $FIFO$ queue that serves with rate C (in packets per second). Each on-off source generates packets with a rate h (in packets per second) when active, and sends no packets when it is idle (i.e. in the OFF state). The durations in the ON state and OFF state for each source are denoted as T_on and T_off respectively. It is to note that the sources are not synchronised, and go into their ON and OFF periods randomly (while staying on or off at an average around the fixed ON and OFF period specified). The distribution of the resultant traffic from an on-off source is exponential. The sojourn times in the states are usually modelled as an exponential distribution for Markovian traffic source. We have used this model of On/Off traffic sources in this research, for both our VoIP and bursty data traffic sources.

3.4 Parameterizing p and r

Having described both modelling buffer states as a two-state MC, and modelling of aggregation of N sources as a single on-off source, we now describe some parameters that are used in this report. When each of the N sources is sending packets at rate h, the mean arrival rate at the buffer during on period T_{on} is R_{on}, and that in off state is R_{off} (which is zero here). Thus in the on state, the buffer is filled up at a rate of $R_{on}C$. R_{on} is given by equation 3.7 [44] [24].

$$R_{on} = \left(C + \frac{h \times A}{N_0 - A} \right) \tag{3.6}$$

Here

1. A is the total offered traffic (of all sources), and is defined as:

$$A = Ap/h \tag{3.7}$$

2. A_p is the mean packet arrival rate at the buffer in packets per second

3. N_0 is the maximum number of flows that can be served at a time

The rate in the On state is given by equation 3.7 [44] [58]

$$R_{on} = \left(C + \frac{h \times A_p}{C - A_p} \right) \tag{3.8}$$

The mean time spent in overflow period is $T(On)$ (note that it's different from T_{On})and mean time duration of a non overflow period is $T(Off)$. As described in [44] [24], $T(On)$ can be used as estimator for the mean duration of an overflow period, denoted by $E(OF)$ and given in equation 3.9.

$$E(OF) \triangleq T(On) = \left(\frac{h \times T_{On}}{C - A_p} \right) \tag{3.9}$$

Note that $T(On)$ is different from T_{on} which as described earlier, is the mean on state duration of an individual on-off source. Further explanation of $T(On)$ being same as $E(OF)$ is given in [24]. The mean number of excess rate packets μ_p in an overflow period (or the packet lost during overflow period) is a product of the duration in On state $T(On)$ and the excess rate $(R_{on} - C)$, given by

$$\mu_p = T(On) \times (R_{on} - C) \tag{3.10}$$

Replacing the values of $T(On)$ and R_{on} (from equations 3.9 and 3.8) in equation 3.10 yields equation 3.11.

$$\mu_p = \left(\frac{h \times T_{On} \times A_p}{(C - A_p)^2} \right) \tag{3.11}$$

To conclude this section we include our analytical method for computation of the p and r values that we have used in our experimentation.

From [44] we know that the expected number of packets lost per overflow period, μ_p, for a single buffered link is given by equation 3.11. A_p as given in [44] [24] can be defined as in equation 3.12

$$A_p = N \times \left(\frac{T_{On}}{T_{On} + T_{Off}} \right) \times h \tag{3.12}$$

We regard the inverse of equation (3.6), i.e., the expected number of packets that are not lost, as r which is the probability of transitioning into non-overflow period. Thus

$$r = \frac{1}{\mu_p} \tag{3.13}$$

As defined in [44] we denote the mean number of packets in the non-overflow period as N_{Nof}. We derive p which is the transition probability for going into overflow, as the inverse of N_{Nof} (i.e. $p = 1/N_{Nof}$). Also from [44] $N_{Nof} = \frac{\mu_p}{PLP}$ where PLP stands for packet loss probability, and this leads us to equation 3.14

$$p = \frac{PLP}{\mu_p} \tag{3.14}$$

Following this aggregate On-Off model, we perform simulations in order to determine the best or optimal sampling rate that accurately estimates the buffer states over time. The initial experimentation was simulating a simple model of a two state MC by providing specific p and r values. As an example of how the p and r values reflect specific network scenario, we have Table 3.1 that shows the parameter values for 70%, 80% and 90% load that give us p and r values. We see that the values for p and r are well under 0.1 for these parameters.

Table 3.1: Parameter Chart: The following parameters yield specific *p* and *r* values when following the analytical equations 3.13 and 3.14. Note that the buffer length varies for each load value in order to keep PLP at the samelevel of 10e-3

Load	Parameter	Value (Bursty Data)	Value (VOIP Data)
70%	C	2Mbps (250 pps)	2Mbps(5.3191e3 pps)
	N	61	61
	h	64kbps = 8pps	64kbps = 170pps
	Queue Threshold	x packets	70 packets
	Source Type	Bursty Data	VOIP Data
	Packet Length	1000 bytes	47 bytes
	Service Time for 1 packet	4ms	0.000188s
	Ton	9.6s	0.96s
	Toff	16.9s	1.69s
	Ap	176.7849	3.76E+03
	p	4.94E-05	2.32E-05
	r	0.0494	0.0232
80%	C	2Mbps (250 pps)	2Mbps(5.3191e3 pps)
	N	70	70
	h	64kbps = 8pps	64kbps = 170pps
	Queue Threshold	xx packets	1100 packets
	Source Type	Bursty Data	VOIP Data
	Packet Length	1000 bytes	47 bytes
	Service Time for 1 packet	4ms	0.000188s
	Ton	9.6s	0.96s
	Toff	16.9s	1.69s
	Ap	202	4.32E+03
	p	1.78E-05	8.38E-06
	r	0.0178	0.0084
90%	C	2Mbps (250 pps)	2Mbps(5.3191e3 pps)
	N	78	70
	h	64kbps = 8pps	64kbps = 170pps
	Queue Threshold	10 packets	3300 packets
	Source Type	Bursty Data	VOIP Data
	Packet Length	1000 bytes	47 bytes
	Service Time for 1 packet	4ms	0.000188s
	Ton	9.6s	0.96s
	Toff	16.9s	1.69s
	Ap	242	4.81E+03
	p	4.13E-06	1.94E-06
	r	0.0041	0.0019406

3.5 Computing the queue length for a given PLP

In the experiments conducted on VOIP traffic, we have used equations from [58] to derive the queue length required for a given *PLP* requirements, when using *VoIP* traffic parameters (On-Off source). From [58] [45], we know the relationship between the queue length and the queue overflow probability as in equation 3.15.

$$PLP = \frac{h \times D}{C - A_p} \times \left(\frac{1 - \frac{1}{T(On) \times (R_{on} - C)}}{1 - \frac{1}{T(Off) \times (C - R_{off})}} \right)^{x+1} \tag{3.15}$$

Where D is the probability of a call (packet flow) delayed, which in terms of Erlangs loss probability D is given by equation 3.16

$$D = \left(\frac{N_o \times B}{N_o - A + A \times B} \right) \tag{3.16}$$

B is given by 3.17

$$B = \frac{\frac{A^{N_o}}{N_o!} \times N_o \times B}{\sum_{r=0}^{N_o} \frac{A^r}{r!}} \tag{3.17}$$

We already know how R_{on} and $T(On)$ from equations 3.8 and 3.9. To find R_{off}, we know from [58] that the mean load A_p in packets per second can be expressed as the weighted sum of the rates in *On* and Off states, i.e.

$$A_p = D \times R_{on} + (1 - D) \times R_{off} \tag{3.18}$$

Therefore we can express R_{off} by rearranging 3.18

$$R_{off} = \frac{A_p - D \times R_{on}}{1 - D} \tag{3.19}$$

In order to parameterize the off state $T(Off)$, we again use D as the probability that a packet flow is delayed. This is the probability that the aggregate on-off process is in *ON* state

$$D = \frac{T(On)}{T(On) + T(Off)} \tag{3.20}$$

Rearranging 3.21, we get

$$T(Off) = T(On) \times \frac{1 - D}{D} \tag{3.21}$$

Thus we can plug in the relevant values in equation 3.15. We initially used 3.15 to find the queue length for our desirable *PLP* of $10e^{-3}$

3.6 Summary

In this chapter we have covered the theory and terminology relevant to this thesis. We have described how buffer at a network node is modelled as a two-state MC. We include how to analytically parametrize the transition probabilities when modelling buffer as a network node and subjecting it to an aggregation of on-off traffic. We have also included the simulation parameters used in this thesis and described how we evaluate our results in subsequent chapters. In the next three chapters, we include simulation results and discussion that forms the core of this thesis.

Chapter 4

Effect of Sampling Gap on Error in Estimate of Buffer Overflow Probability

4.1 Introduction

In this chapter we look into the relationship between error in estimation and the sampling gap. We also look at how the error in estimate is affected by having a fixed time window for taking samples, when sampling at various gaps and with different sample sizes. We find that when using a fixed number of samples (which is a constraint in this thesis, because of the application areas discussed in Chapter 2) with varying gaps between them and a flexible time window (the time window increases with the increase in sampling gap), sampling with the larger gaps will bring more precision. Previous literature, by Roughan [1] also confirms this, where it is argued that samples taken closer together give more error in estimation because of correlation between samples. On the other hand, we find that if we have a fixed time window within which we need to sample, then sampling with the least gap and using maximum possible samples gives more accuracy. For results included in this chapter, we have conducted two different sets of experiments, as explained later in section 4.2.1 and section 4.2.2. As explained in Chapter 3, we have modelled the buffer being sampled, as a two-state MC. In light of our results presented in this Chapter, we also discuss that modelling the buffer at a network node as a two-state MC may be too naive.

4.2 Sampling Gap vs Error in Estimate with Fixed Sample Size and varying Sampling Window

4.2.1 Experiments in Matlab

As stated earlier (Chapter 3 section 3.2), we have assumed the states of a buffer transitioning between overflow and not-overflow to follow a two state MC process. In order to have results closer to theory, we first perform experiments in Matlab. We generate a two state MC in Matlab, with transition probability values derived from the analytical method described in section 3.2. The transitions of the MC in each experiment are different from each subsequent experiment as there is an element of randomness with regards to the transitions of the MC between states. We sample the MC with gaps of $k = 1$ up to $k = 1000$. As explained earlier, the sampling gap is in terms of time, that we refer to as a clock tick. We consider a clock tick as the time needed to serve one packet (as noted earlier, this would be set in practice by some irreducible aspect of the network technology). Following the procedure suggested by Parker et al [3], sampling is done at various rates, by changing the k values. That is, if we are taking T samples (referred to as a sample size) in one go, $k = 1$ would mean sampling every clock tick for these T samples (that is, taking consecutive samples), $k = 2$ means sampling every other clock tick and so on. The experiment is repeated 200,000 times for each T and p, r value. We have chosen 200'000 as the number of repeats of an experiment for an experiment with fixed number of samples and fixed network burstiness (p and r) as this is the number of repeats that gave us results that were not noisy even for T as small as $T = 5$. The outcome of the repeated experiments is an estimation of the probability of being in overflow state $P(OF)$, form the various sampling rates used. The relative error in the estimation of $P(OF)$, after repeated experiments, is plotted along the y-axis in figures 4.1, 4.2 and 4.3. All three results are for load of 70%, with the difference being in the number of samples taken. Note that we are considering load of 70% and higher because the buffer needs to be subjected bursty traffic. When load is lower than 60%, the aggregated traffic is not sufficiently bursty. [33] [61] [62].

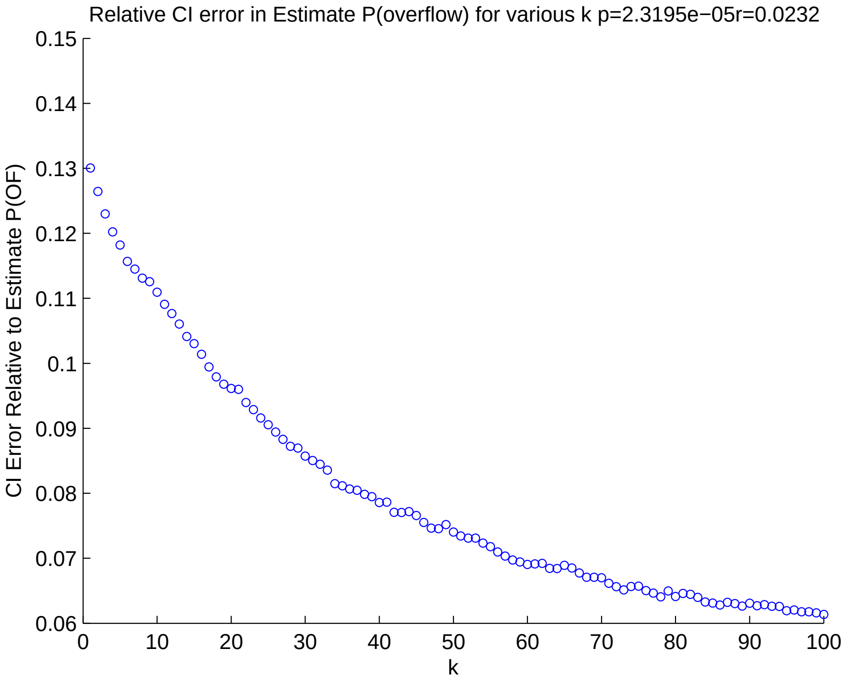


Figure 4.1: Relative Error for the estimated P(OF) for various k values, obtained for sample $T = 5$. The error in estimate decreases with increase in sampling gap. This is because we have a fixed number of samples (fixed samples is one of the constraints for the scenarios under consideration) so as we space the samples out, we capture more precise information

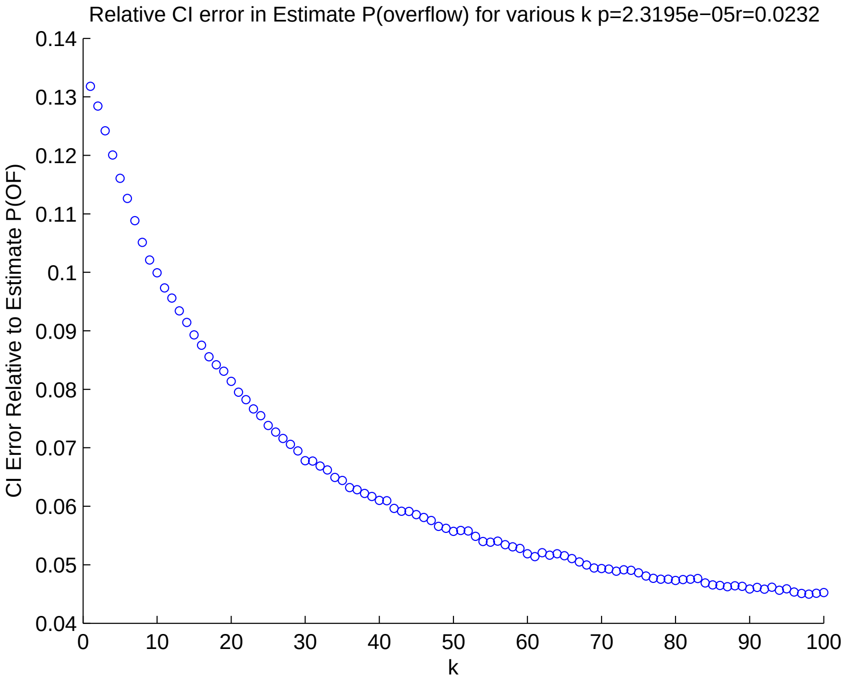


Figure 4.2: Relative CI Error for the estimated P(OF) for various k values, obtained for sample $T = 10$

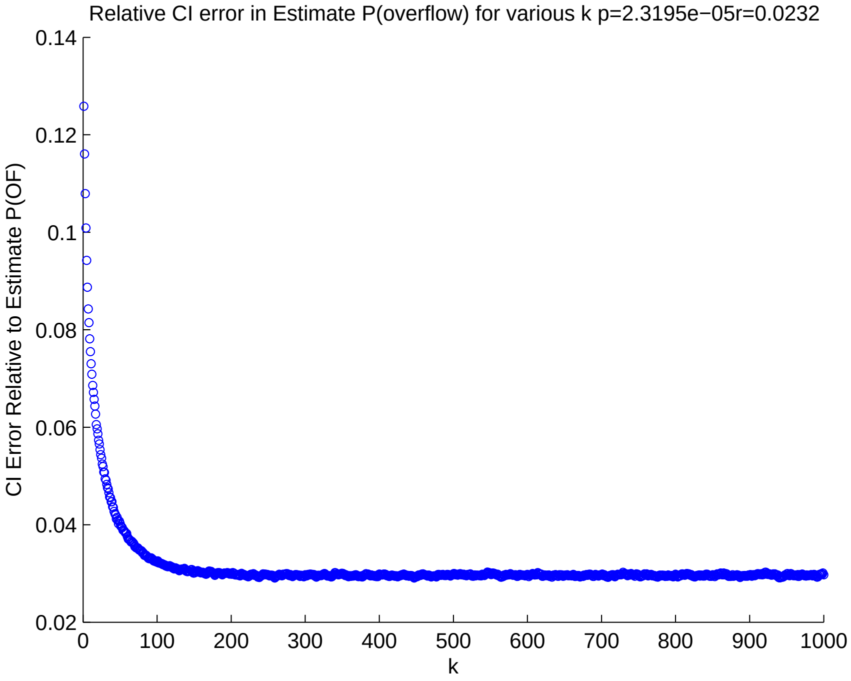


Figure 4.3: Relative CI Error for the estimated P(OF) for various k values, obtained for sample $T = 20$

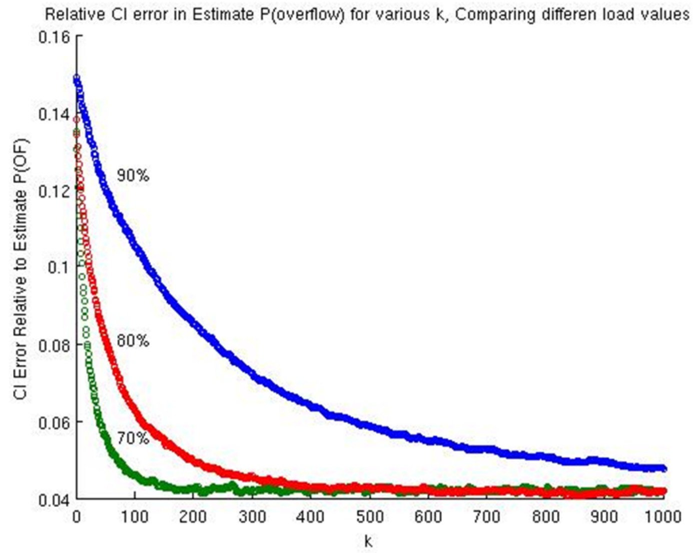


Figure 4.4: Error for the estimated P(OF) for various k values, obtained for sample size $T = 20$

We can see from figues 4.1 through 4.3 that, the error in estimate decreases with increase in sampling gap. This is intuitively also true, since we have a fixed number of samples, and hence having the samples farther apart captures more information about the buffer states of overflow and non-overflow. In this thesis, we have chosen to include results for sample sizes of $T = 5$ and $T = 10$ as these are the sample sizes chosen by Parker et al [3, 16]. We have also included $T = 20$ just for comparison. A gap of $k = 1$ means a gap of 0.183 milliseconds, since the link rate is assumed to be $2Mbps$ and it takes $0.183ms$ to serve a 47 byte packet (standard VoIP packet length as used by [24]). The results show a relationship between sampling gap and accuracy of parameter estimation. From figure 4.1, we may say that for accuracy in sampling, $16.47ms$ (0.183ms*90) is the least gap (which translates to highest sampling rate) at which we can sample for a sample size of $T = 5$.If we extend the experiment to account for larger sampling gaps for up to $k = 1000$, the results look as in figure 4.3. We can see that there is an asymptotic level for sampling gap beyond which error does not seem to change much. So we may say that we need not sample with a bigger gap than this for particular p and r values. Figure 4.4 shows the change of error in estimate with respect to sampling gap for various load values. The p and r used for these load values are derived from parameters in chapter 3.

4.2.2 Experiments in NS-2

Taking our simulation work one step closer to a more realistic network scenario, we perform simulations in NS-2. We use parameters shown in chapter 3 for simulating the states of a buffer at an access node. We focus on access node because, as noted in previous literature (see DoE Parker 2011 and Adams et al 2005 [5]), these network nodes generally have links with lower bandwidths and higher loads. These nodes, therefore, act as bottlenecks which contribute significantly to network losses and delays.

The network topology of the scenario is given in figure 4.5. The buffer under consideration is at node *S*.

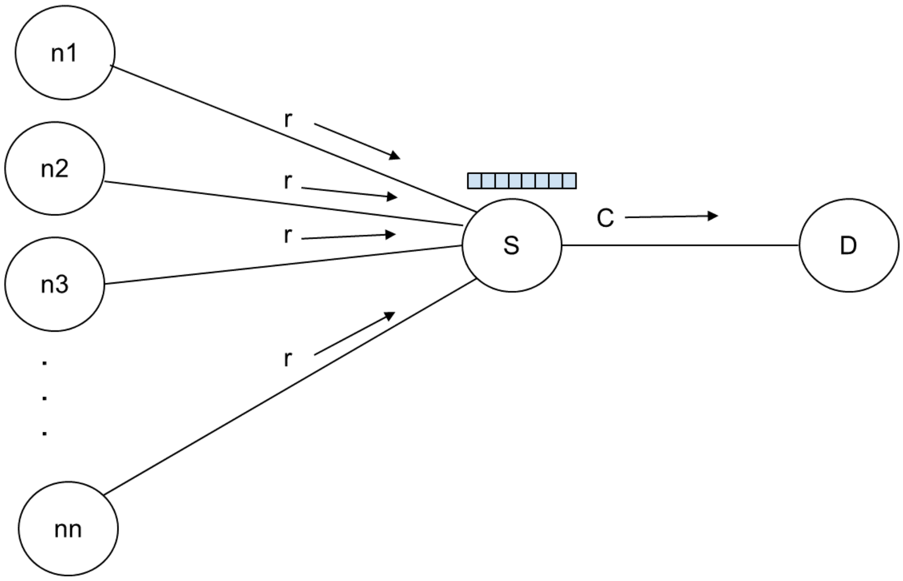


Figure 4.5: Network setup for probing buffer at a network node S, receiving aggregate traffic of n sources. This can be considered an N-on-off/D/1 queuing model since we are using on-off sources for simulating VoIP traffic

The buffer is sampled after each packet being served out of the queue (after each service-time unit) in order to see whether the queue is in overflow state or not. A sequence of 0s and 1s is thus generated where 0 indicates queue being in overflow while 1 means not being in overflow state. These outcomes in form of a sequence of 0s and 1s are then sampled similar to the process in section 4.2.1 From Figure 4.6 we can see that the standard error for sampling NS2 data keeps going down (changing) and does not become constant when considering k up to 100. In contrast, in the Matlab curve in Figure 4.3 the error in estimate seems to settle down after $k = 100$.

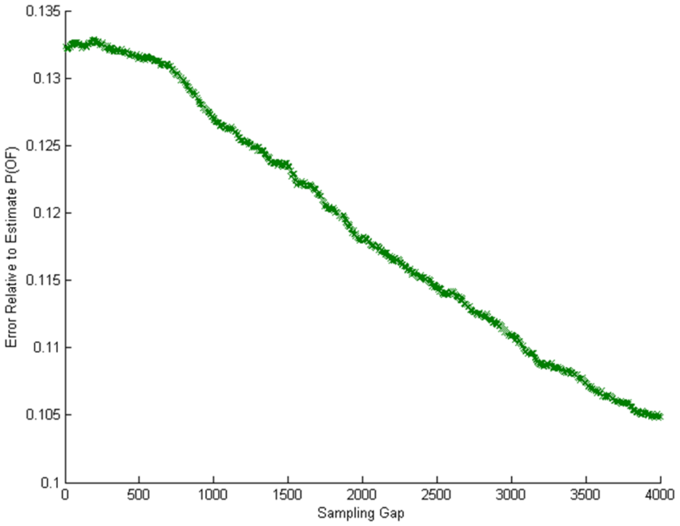


Figure 4.6: Error for the estimated P(OF) for various k values, obtained for sample $T = 100$ using NS2 simulations, for load of 80%

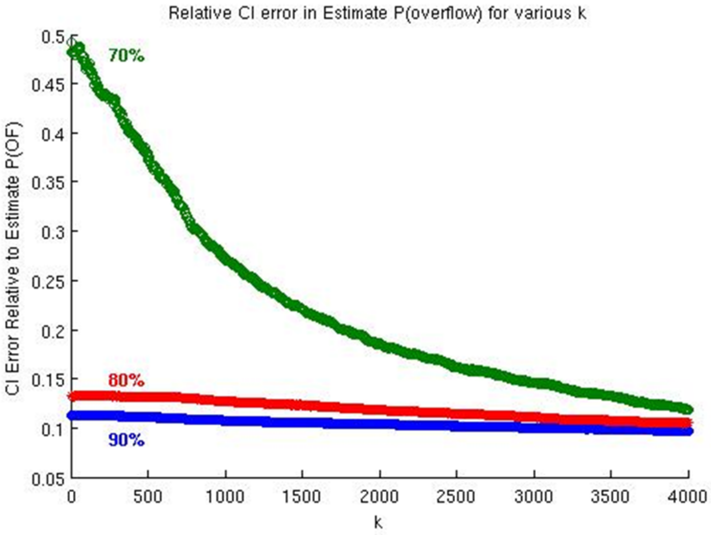


Figure 4.7: Error for the estimated $P(OF)$ for various k values, obtained for VoIP data, simulated in NS2. Note that here the precision of estimate is higher for higher load. This is a result opposite to what we have seen in figure 4.4This is because the scenario in practice (in NS simulation), there are several other parameters that come into play, which are not there in analytical formula. We can see in chapter 6 section 6.5, that for larger load, we need to keep a larger buffer. And precision increases with larger buffer sizes.

.

Figure 4.7 shows a comparison graph for error in estimation of P(OF) for various sampling interval values, and load values from 70% to 90%, for VOIP sources generated in NS2. Contrary to the earlier case of bursty data, the sampling error converges quicker for higher load than for lower load values. We notice that the error in estimation in these results is the least for highest load. This is because we have higher PLP for loads of 80% and 90%. Hence we have more rare events caught in the samples for load of 90% than 80% or 70% load values for the same sampling gaps, and thus more accuracy in estimation.

4.3 Sampling Gap vs Error in Estimate when Keeping fixed Sampling Window

In section 4.2.1 and 4.2.2 we have discussed results when varying the sampling window size and keeping a fixed number of samples. That is, in case of sampling at k=2, the time window within the sampling process finishes would be twice as long as that in case of k=1. In this section we briefly cover our experiments performed to see the trend when we remove the restriction of keeping a fixed number of samples. That is, we look at the effect of varying sample size and sampling gap while keeping a fixed time window within which we sample. To achieve this, we run a simulation in matlab using different combinations of sample size and sampling gap, such that the product of the two stays within a specified window size. Results included in this section consider a time window of 250 clock ticks. For VOIP traffic with parameters for 70% load, the sampling window is 0.04525 seconds long. Figures 4.8 plots the sampling gap with respect to CI Error in estimation of P(OF) when we vary both the sample size T and the sampling gap k, for VOIP data traffic. The figure shows that the relative error is higher for larger k values. that is, error in estimate is lower for larger sample size. This is intuitively true also, since if we are allowed a large number of samples, taking the maximum number of samples within the time constraints (sampling window) would give a more accurate estimate. This contradicts Roughan's theoretical lower bounds for least gap in estimate, where he states that error in estimate increases if samples are too close together.

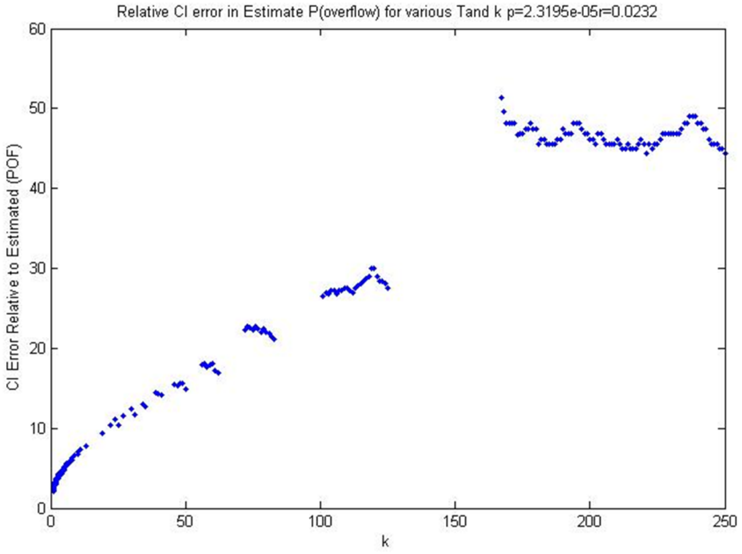


Figure 4.8: Error for the estimated P(OF) for various k and T values within a time window of 1000 clock ticks. Here, the time window is 1000 clock ticks, which means that for $k = 250$, T cannot be more than 4. Similarly for $k = 2$ T will be 500 or less to remain within a window of 1000 clock ticks.

4.4 Comparison with Roughan's Work

Roughan [1] observes that inaccuracies result in estimations using probing or sampling due to correlation between successive samples; therefore the samples need to be a certain distance apart to avoid bias due to these correlations. Roughan [1] gives analytical upper bounds on sampling rate when estimating mean delay or mean queue fill. In this research we evaluate the upper bound on sampling rate when targeting PLP, and we do this using simulation.Roughan uses Gaussian Confidence Interval for estimating the error in parameter estimation. He gives equations for error in estimating mean number of packets in the queue and mean waiting time in the queue. In order to compare our *sampling gap* with that of Roughan's, we use equations for mean number in the queue, as follows.

$$B^{+}_{-}z_{b/2} * \sqrt{\frac{\delta_B^2(p',\lambda,\mu)}{T'}} \tag{4.1}$$

Here $\delta_B^2(p',\lambda,\mu)$ is the variance of the parameter being estimated (in this case, mean number in the queue). Thus the error in estimation is the term:

$$\Delta B = z_{b/2} * \sqrt{\frac{\delta_B^2(p',\lambda,\mu)}{T'}} \tag{4.2}$$

The variance $\delta_B^2(p',\lambda,\mu)$ is given as

$$\delta_B^2(p',\lambda,\mu) \triangleq \frac{\rho}{(1-\rho)^2} + p'\frac{4\rho^3}{(1-\rho)^4} \tag{4.3}$$

ρ is the load, $\rho = \frac{\lambda}{\mu}$. Note that we have placed a prime on some of the variables in formulas from Roughan, so as to differentiate them from variable names used in our model. p' here determines the probing or sampling rate. That is, $samplingrate = arrival_rate * p'$. $p' = 1$ will mean sampling every interval, $p' = 0.5$ will mean sampling every other interval and so forth. Roughan states that in order to obtain the variance with respect to the observation interval T', we need to divide the equation 4.3 with $p' \times \lambda$. dividing equation 4.3 and simplifying, we get the equation 4.5.

$$\delta_B^2(p',\lambda,\mu) \triangleq \frac{\frac{\rho}{(1-\rho)^2} + p'\frac{4\rho^3}{(1-\rho)^4}}{p' \times \lambda} \tag{4.4}$$

Simplifying 4.4 and dividing by 8, we get 4.5

$$\delta_B^2(p',\lambda,\mu) \triangleq \frac{1}{p' \times \mu} \times \frac{\rho}{(1-\rho)^2} + \frac{4 \times \rho^2}{\mu \times (1-\rho)^4} \tag{4.5}$$

It is to note that the equation above (4.5) is for M/D/1 queue. As in case of Roughan's equation (24), we can obtain an equation of asymptotic variance parameter estimation for M/D/1, which is $1/8th$ of the asymptotic variance for M/M/1 (see [1] (24)). Considering the simulation in section 4.2.1, with a sample size of $T = 20$, if we keep $C = 2Mbps$, the time for serving each 47 byte packet would be $0.188ms$. Thus when taking 20 samples at a gap of $0.188ms$ and $k = 1$, would give us T' (duration of sampling) as $T' = 0.188ms * 20 * 1 = 0.0038\ s$. Replacing the values in equation 4.5 and then solving it gives variance of 0.0059. And replacing values in 4.3 we get an error of 2.45 packets.

$$\delta_B^2(p',\lambda,\mu) \triangleq \frac{1}{1 \times 5319.1} \times \frac{0.7}{(1-0.7)^2} + \frac{4 \times 0.7^2}{5319.1 \times (1-0.7)^4}$$

$$\delta_B^2(p',\lambda,\mu) \triangleq 0.0059$$

Solving for Error ΔW, with 95th percentile Confidence Interval, when $z = 1.96$

$$\Delta B = 1.96 * \sqrt{\frac{0.0059}{0.0038}}$$

$$\Delta B = 2.4488$$

Table 4.1: Probing Gap vs Error in estimate

k	p'	ΔB	B	Relative ΔB
1	1	2.4488	1.633	1.499
2	0.5	1.7583	1.633	1.0765
3	0.33	1.4572	1.6333	0.8922
4	0.25	1.2803	1.6333	0.7839
5	0.2	1.1613	1.6333	0.7110
150	0.0067	0.4748	1.6333	0.2907
500	0.002	0.4454	1.6333	0.2727
1000	0.001	0.4388	1.6333	0.2687
2000	0.0005	0.4355	1.6333	0.2666
10000	0.00001	0.4328	1.6333	0.2650

If we consider sampling gap of $k = 2$, then $p' = 0.5$ and the error in estimate becomes 1.76 (and so forth). Some sampling gap values and corresponding error values are given in Table 4.1. We can see from theoretical results according to Roughan's formula (in table 4.1), that for the parameters in our experiment, when taking 20 samples at a gap of 1000, the relative error in estimate of mean number in the queue is 0.2687. In comparison, we get a relative error of about 0.03 from our simulation results, when estimating overflow probability of a buffer. Similarly, if we consider the numerical experiment parameters for M/D/1 queue worked out by Roughan (see Section 6 [1]). That is, a link rate of 2.48 Gbps with packet transmission rate of 200,000 pps and service time of 4.8μs. Then according to Roughan, the asymptotic variance with N=10000 will be 1.5610e-04, error will be 0.1118 and the relative error will be 0.0684. If we increase 'k' to 5, then relative error becomes 0.0325. From a simulation in Matlab, with p and r derived for the same parameters, we get a relative error value of 0.0014. For getting a comparable relative error for Roughan's scenario, we need to increase N to 100,000 and have a gap of 1000 between probes. We can see from the numerical examples above that, although Roughan suggests to have probes far apart for certain bounds on error in estimation of mean number in queue, our sampling gap is much smaller for a similar error in estimate.

4.5 Summary

From the results included in this chapter, we can see that sampling the buffer for overflow probability gives more precision if the samples are far apart. This is true for the theoretical bounds presented by Roughan [1] and Whitt [33] for estimation of mean waiting time and mean number in the queue as well. We have shown from a numerical experiment that although having samples far apart brings more precision in estimation, it is still a matter of how far apart the samples can be while staying within reasonable time bounds. For example, having 1000 samples with a gap o 1000 time ticks will mean that the time window itself is 10^6 time units. In the example presented here, this means a time window of $10^6 \times 2e - 4$ = 200 seconds, or roughly 3 minutes. This time window seems reasonable for estimation during a busy hour.

We have noticed in our results that beyond a certain (large) sampling gap, the precision of the estimate does not increase. That is we do not need to sample at a rate lower than this. To point out this 'asymptotic' value of sampling gap, beyond which the estimation does not improve, we could approach this problem from the opposite direction. That is, to identify the lowest possible variance in the estimation. In the next chapter, we address this by computing the asymptotic variance of the estimate, from a theoretical formula found in literature.

Chapter 5

Bounding Error in Estimate - Asymptotic Variance

Results in Chapter 4 show a relationship between error in estimate of buffer overflow probability and the sampling gap. We have also seen that when limited by a fixed number of samples, keeping them (samples) far apart gives more accuracy in estimating the overflow probability. We have observed that there appears to be an asymptotic level (of sampling gap) beyond which the error in estimate does not significantly decrease further. Keeping this in mind, we set out to compute asymptotic variance of the estimate of overflow probability of a buffer. This chapter covers our work on computation of asymptotic variance of a buffer overflow probability.

5.1 Asymptotic Variance of Estimate of Overflow Probability of a Semi-Markov Chain

If we are able to find a formula for asymptotic variance of the estimate, when taking large enough number of samples, then we can use it to determine the theoretical lower bound on variance. From here we can try to look for an optimal combination of sampling gap and number of samples, within the constraints of a sampling window. This thought led us to the research by Barbu et al [2], who give a formula for estimating the stationary distribution of a semi-Markov Chain:

$$\sqrt{M} * [\pi_i(M) - \pi_i] \overset{\Delta}{\rightarrow} N(0, \sigma^2_{\pi_i}),\ as\ M \overset{\Delta}{\rightarrow} \infty \tag{5.1}$$

The asymptotic variance of the mean of the distribution, according to Barbu et al [2] is given by:

$$\sigma_{\pi_i}^2 = \frac{1}{\mu_{ii}} * \frac{\frac{\sigma_i^2}{m_i^2} + \frac{\rho_{ii}^2 - \sigma_i^2}{\mu_{ii} - m_i}}{\left(\frac{1}{m_i^2} + \frac{1}{\mu_{ii} - m_i}\right)^2} \tag{5.2}$$

Here m_i is the sojourn time in the state of interest i. In our two-state Markov Chain model, the state i is the overflow state 1. The sojourn time in state i is given by the reciprocal of the probability r of going out of state 1 (since we are assuming exponential sojourn times). That is,

$$m_i = \frac{1}{r} \tag{5.3}$$

μ_{ii} is the mean recurrence time in state i. Thus for state 1, the mean recurrence time for state 1 is the mean sojourn time in state 0. Therefore it is expressed as the reciprocal of probability of moving into state 1. Thus,

$$\mu_{ii} = \frac{1}{p} \tag{5.4}$$

Here ρ_{ii}^2 is the variance of the mean recurrence time of state i. The mean recurrence time is the time it takes to come back to state i once the chain has moved out of state i. In our two-state Markov Chain model, this is the time spent in the other state 0. Since p is the probability of moving into state 1, the mean time spent in state 0 is given by $1/p$. Thus mean recurrence time of state 1 is $1/p$.

In case of our two state model, the variance of the recurrence time of state 1 can be approximated as the square of the mean time in the other state 0 (the square of μ_{ii}). This is because the sojourn times are exponentially distributed. Thus, for our two-state Markov Chain model

$$\rho_{ii}^2 = \left(\frac{1}{p}\right)^2 \tag{5.5}$$

Similarly,σ_i^2 is the variance of the sojourn time in state i. For our two-state MC model, the mean sojourn time in state 1 is $1/r$. We can approximate the variance of the mean sojourn time in state 1 as the square of the mean sojourn time in state 1.

Thus,

$$\sigma_i^2 = \left(\frac{1}{r}\right)^2 \tag{5.6}$$

Replacing *p* and *r* parameters for different load values, as in chapter 3 (note that the calculation of *p* and *r* is described in chapter 3 section 3.4), we find the asymptotic variance that the formula suggests for different scenarios (traffic parameters that result in different Markov Chain probability values). It is to note that since the parameter *p* in our experiments is much smaller than the parameter *r* (see parameter chart table 3.1 in chapter 3 section 3.4), the formula in equation 5.2 can be very closely approximated to:

$$\sigma_{\pi_i}^2 = 2 * \frac{p}{r^2} \tag{5.7}$$

Table 5.1: Theoretical Asymptotic Variance computed for load of 70%-90%. The theoretical asymptotic variance is computed using equation 5.2.

load %	p	r	Theoretical Asymptotic Variance eq. 5.2
70	2.3195e-05	0.0232	0.0862
80	8.3766e-06	0.0084	0.2388
90	1.9406e-06	0.0019	1.0306

5.2 Numerical Asymptotic Variance of a Semi-Markov Chain

In this section, we show results obtained from Markov Chain simulation experiments performed in Matlab. As before, we used the Markov Chain transition probability parameters (*p* and *r*) as described in Chapter 3. Each of our experiment of simulating Markov Chain was run for a million cycles, We discarded the first 5000 cycles. Each experiment was repeated 200000 times. Figure 5.1 shows the numerical asymptotic variance, computed from simulation results.

The numerical asymptotic variance from the outcomes of buffer states (of overflow and non-overflow) from the experiments is computed, according to $\sqrt{M} * [\pi_i(M) - \pi_i]$. That is, we have computed $\sqrt{M} * [P(OF)_i - P(OF)_{actual}]$ for each of the $200,000$ experiments (here i represents the experiment index), and then taken a variance of these $200,000$ values. M in our experiments is 100000, which is the number of samples.

We can see from figure 5.1 (and Tables 6.2-6.4) that the numerical asymptotic variance is quite close to the theoretical values, when we have a large sample size (100000 samples).

In figure 5.1 , we also notice that the asymptotic variance decreases for larger sampling gaps (for $k = 2$ to $k = 5$). In order to compute the theoretical asymptotic variance, we need to revisit our Markov Chain model. When $k = 2$, then the Markov Chain would have transitioned twice. Thus we need to consider transition probabilities for each state change for $k = 2$ (or in the terminology of Markov Chains, we are looking for a two-step transition matrix, for our sampling gap of $k = 2$). Here's how we work out the k-step transition probabilities for the Markov Chain. If the Markov Chain is in state 0 after two steps, then we have the following four cases:

1. Either the system started in state 0, then stayed in state 0 twice, which means:$(1-p)^2$.
2. Or previously the system transitioned into state 1 and then came back to state 0. Which means the probability is pr.
3. Or previously the system started and stayed in state 1, stayed in the state 1 once and then transitioned into state 0. Which gives the product $(1-r)r$.
4. Or lastly, the previously the system transitioned from state 1 to state 0 and then stayed in state 0. Which means the probability $r(1-p)$

Thus the probability of transitioning from state 0 to state 0 (or staying in state 0) is the sum of the four probabilities computed above:

$$(1-p)^2 + pr + (1-r)r + r(1-p)$$

Similarly, we find the other three transition probabilities. Actually, for our two state Markov Chain with the transition matrix

$$P = \begin{pmatrix} 1-p & p \\ r & 1-r \end{pmatrix}$$

then the k-step transition matrix is simply the matrix raised to power k (Also explained in [3]).

Hence, using the transition values obtained from the k-step transition matrix for each k (sampling gap), we find the theoretical transition probabilities for $k = 2$ to $k = 5$ and comparison of the theoretical values with the numerical ones is given in Tables 5.2 - 5.4

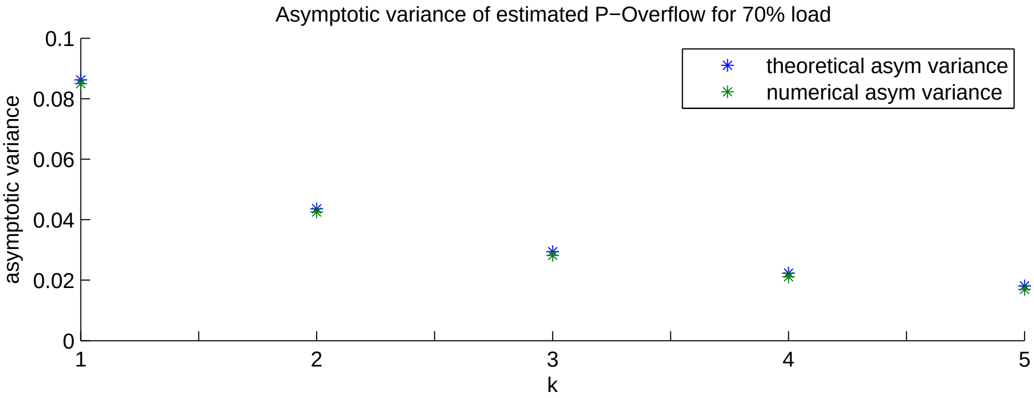


Figure 5.1: Theoretical vs Numerical Asymptotic Variance computed for load of 70% for $k = 1$ to $k = 5$. Note that this figure compares theoretical asymptotic variance computed from equation 5.2 for $k = 1$ and then following the procedure explained in this section, for subsequent values of k. The numerical Asymptotic Variance has been obtained by computing variance of sampled estimate using $\sqrt{M} * [P(OF)_i - P(OF)_{actual}]$ from simulations in Matlab. We can see that results here confirm prior findings, that precision of estimate increases with increasing sampling gap when modelling buffer as a two state MC

We note that the theoretical asymptotic variance indeed decreases when we increase k. And the numerical results match the theoretical values, thus validating our simulation results. Thus for our VOIP traffic scenario for load of 70% up to 90%, when we keep the number of samples fixed, the variance decreases, or in other words, precision increases, as we increase the sampling gap.

In the results shown in this chapter, we have kept a sample size large enough to accurately compute the asymptotic variance. On the other hand, if we decrease the sample size to 10, then the numerical variance does not match the theoretical results, as the sample size is too small. This can be seen from results enumerated in Table 5.5.

Tables 5.2 through 5.5 show asymptotic variance as calculated for different load and sampling gap k values. Note that table 5.5 shows numerical variance computed for a sample size of $T = 10$. Hence this cannot be called asymptotic variance. We have included it to show that for smaller sample size, the variance does not match theoretical asymptotic variance.

Table 5.2: Theoretical vs Numerical Asymptotic Variance computed for load of 70% for $k = 1$ to $k = 5$ for VoIP Data

load %	k	p	r	Theoretical Asymptotic Variance	Numerical (Matlab) Asymptotic Variance eq 5.2
70	1	2.3195e-05	0.0232	0.08622	0.084969
70	2	0.458514e-4	0.0458514	0.04362	0.04252
70	3	0.67982e-4	0.067982	0.02942	0.02821
70	4	0.895984e-4	0.089598	0.02232	0.02115
70	5	0.1107130e-3	0.11071313	0.018065	0.01695

Table 5.3: Theoretical vs Numerical Asymptotic Variance computed for load of 80% for $k = 1$ to $k = 5$ for VoIP Data

load %	k	p	r	Theoretical Asymptotic Variance	Numerical (Matlab) Asymptotic Variance 5.2
80	1	8.3766e-06	0.0083766	0.23876	0.23581
80	2	0.16683e-4	0.016683	0.11988	0.118107
80	3	0.249197e-4	0.0249197	0.08026	0.078809
80	4	0.330873e-4	0.0330873	0.06045	0.059319
80	5	0.411865e-4	0.04118649	0.04856	0.047719

Table 5.4: Theoretical vs Numerical Asymptotic Variance computed for load of 90% for $k = 1$ to $k = 5$ for VoIP Data

load %	k	p	r	Theoretical Asymptotic Variance	Numerical (Matlab) Asymptotic Variance 5.2
90	1	1.9406e-06	0.0019406	1.0306091	1.0312499
90	2	0.387743e-5	0.0038774	0.5158055	0.509574
90	3	0.5810498e-5	0.005810498	0.3442045	0.341229
90	4	0.7739811e-5	0.00773981	0.258404	0.255352
90	5	0.9665376e-5	0.009665	0.206924	0.204756

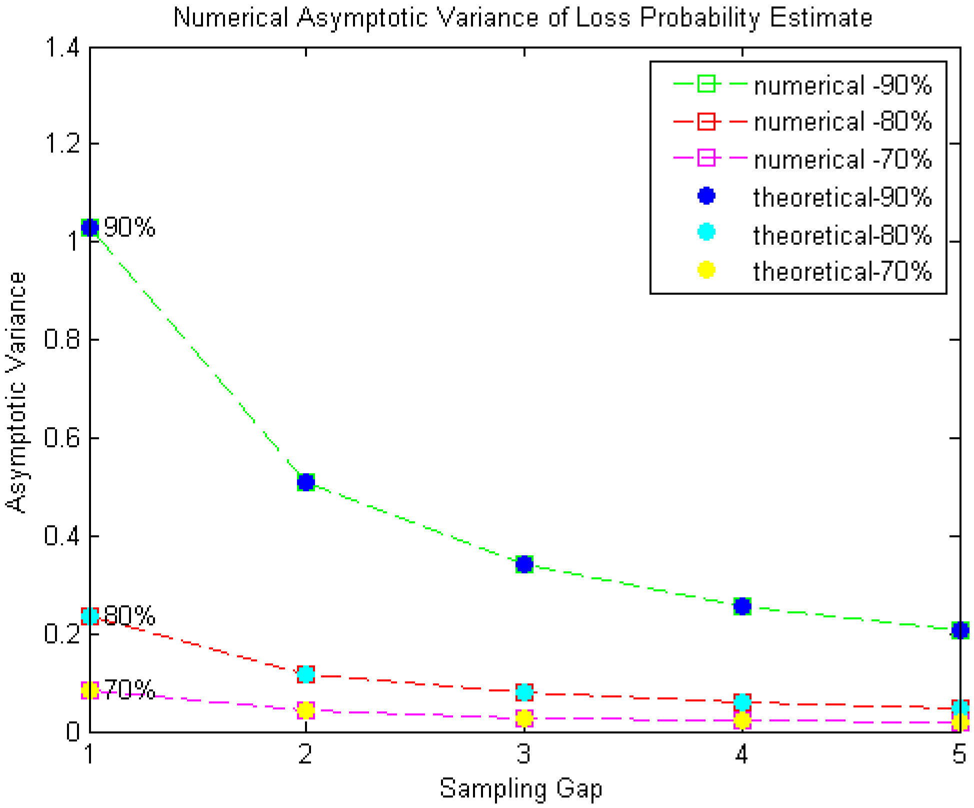


Figure 5.2: Theoretical vs Numerical Asymptotic Variance computed for load of 70% to 90% for $k = 1$ to $k = 5$. Note that this figure compares theoretical asymptotic variance computed from equation 5.2 for $k = 1$ and then following the procedure explained in this section, for subsequent values of k. The numerical Asymptotic Variance has been obtained by computing variance of sampled estimate using $\sqrt{M} * [P(OF)_i - P(OF)_{actual}]$ from simulations in Matlab. We can see that results here confirm prior findings, that precision of estimate increases with increasing sampling gap when modelling buffer as a two state MC. (And this acts as a validation of our simulation results)

Table 5.5: Theoretical vs Numerical Asymptotic Variance computed for load of 80% for $k = 1$ to $k = 5$ with smaller T ($T = 10$)

load %	k	p	r	Theoretical Asymptotic Variance	Numerical (Matlab) Asymptotic Variance 5.2
70	1	2.3195e-05	0.0232	0.08622	0.010405
70	2	0.458514e-4	0.0458514	0.04362	0.009667
70	3	0.67982e-4	0.067982	0.02942	0.008894
70	4	0.895984e-4	0.089598	0.02232	0.008137
70	5	0.1107130e-3	0.11071313	0.018065	0.00751
80	1	8.3766e-06	0.0083766	0.23876	0.0099
80	2	0.16683e-4	0.016683	0.11988	0.0096
80	3	0.249197e-4	0.0249197	0.08026	0.0094
80	4	0.330873e-4	0.0330873	0.06045	0.0091
80	5	0.411865e-4	0.04118649	0.04856	0.0088
90	1	1.9406e-06	0.0019406	1.0306091	0.009758
90	2	0.387743e-5	0.0038774	0.5158055	0.009595
90	3	0.5810498e-5	0.005810498	0.3442045	0.009441
90	4	0.7739811e-5	0.00773981	0.258404	0.009360
90	5	0.9665376e-5	0.009665	0.206924	0.009294

5.3 Numerical Asymptotic Variance of the Distribution of Overflow Periods obtained from NS-2 Experiments

In the previous section, we show that we've obtained a match with theoretical asymptotic variance, when sampling a two-state MC generated in Matlab. In this section, we look at the asymptotic variance of overflow probability computed from samples taken from buffer state evolution in NS-2. Similar to experiments in Matlab, we generate several experiments in NS-2 using VOIP parameters enumerated in Chapter 3. We record the overflow and not-overflow states of the buffer after each cycle (service time of a packet). We then sample this recorded data of buffer states with various sampling gaps and compute Overflow Probability in each experiment. We then use $\sqrt{M} * [P(OF)_i - P(OF)_{actual}]$ to compute asymptotic variance of the Overflow Probability of the buffer, as explained in section 5.2. M is the total number of samples. We have enumerated the results in Table 5.6 We can see from the results in Table 5.6 that the theoretical asymptotic variance does not match numerical asymptotic variance in this case. We show in the next chapter that the distribution of over-flow and not-overflow states of the buffer recorded from NS-2 is not exponential. We cover the discrepancies with NS-2 results in Chapter 6 in further detail.

Table 5.6: Theoretical vs. Numerical (NS-2 Experiments) Asymptotic Variance computed for load of 70%-90%

load %	p	r	Numerical-NS2- Asymptotic Variance	Theoretical Asymptotic Variance (eq 5.2)
70	0.036635e-6	0.015319	0.41981	0.0312148
80	6.8075e-6	0.026931	0.56075	0.0188
90	6.5963e-6	0.023239	0.00011219	0.024421

5.4 Summary

In this chapter, we have shown the upper bound on the precision of the estimate of buffer overflow probability. It is to note that this is the most precision (or least variance) we can expect when estimating $P(OF)$ for a certain network scenario. We have computed the upper bound on variance of the buffer overflow estimate using a formula provided in literature [2]. We have first shown how this formula for asymptotic variance applies to our scenarios under consideration, when modelling the buffer as a two-state MC. Then we have computed theoretical asymptotic variance for our different load scenarios, as well as numerical asymptotic variance from the simulation results. We ave seen that theoretical and numerical asymptotic variances match when modelling in Matlab simulation. However, theoretical asymptotic variance does not match numerical asymptotic variance for NS2 results. Hence we take a detailed look at our NS2 experiments in the next chapter.

Chapter 6

Sampling Buffer States in NS-2 Simulations

In order to bring our simulation results closer to real word, we perform experiments in *NS-2* and sample the buffer state outcomes at various sampling gaps similar to our sampling in Matlab based experiments (Chapter 4). We have included some NS-2 results in Chapters 4 and 5. In this Chapter, we include discussion on our NS-2 experiments and perform a further analysis of the results. We first describe our experiments and challenges involved therein. We then discuss the effect of buffer length on estimation of overflow probability and its error. Acknowledging that experimental results from NS2 experiments do not match theory, we discuss that the process is not fully modelled by a Gilbert Model and try to fit the distribution of outcomes from NS-2 experiments.

6.1 NS2 Experiments for Estimating Overflow Probability

Using the parameters included in Chapter 3 we repeat simulation experiments several times (same as in Matlab experiments) to compute the variance of the estimate. In each experiment we record the buffer state after every $clocktick$ (Recall from Chapter 3 that a $clocktick$ is equal to a packet service time). If the buffer is in overflow we record a 1 otherwise we record a 0. Thus we generate a stream of $0s$ and $1s$ and then process the output as we did the outcome of MC Process in Matlab. We sample the outcome of each experiment for various sample sizes ($T = 10$, $T = 20$ and $T = 100$). We notice that when $T = 10$, our samples do not capture any overflows $(1s)$ for various k values in any of the 200000 experiments. This is evident from the curve in figure 6.2 that has missing values. From our results, we notice that $T = 20$ is also too few samples for sampling outcomes of *overflow* states of the buffer. After trying several sample sizes, we decide to use $T = 100$ as our sample size since this gives us enough overflows for each k to be able to perform a meaningful comparison with Matlab our results in section 4.2.1.In addition to sample size, computing error in estimate is also a challenge in NS2 experiments. We could use Confidence Interval on MSE as our measure of error in estimate (see Chapter 3) in our Matlab results quoted in Chapter 4 section 4.2.1. However, we notice when computing error in estimate (as MSE) in NS-2, that the error in estimate increases with increase in sampling gap. This seems a result opposite to that in our Matlab experiments. A closer inspection of raw data reveals that in many experiments we do not capture any overflows in the samples (no occurrence of rare state in the samples). That is, we only capture $0s$ for different k values thus getting $P(OF) = 0$. This occurrence of $0s$ skews the resultant error in estimate as the difference of estimated P(OF)=0 for a certain experiment for k under consideration from the mean Estimate P(OF) for that k becomes large (and thus we get a large error for larger k values, as shown in figure 6.1). The reason for absence of overflows in many samples is because, as we discuss in this ans subsequent sections in this chapter, the distribution of bursts that result in overflows in NS-2 simulation is different from what we expect from a MC model. It is to note that we have used 10e-3 as our desired PLP in choosing the parameters for all of our experiments, since this is a typical loss probability value chosen in Service Level Agreements between network service providers and customers.

To rectify the problem, we compute CI error relative to the mean estimate (that is what we refer to as 'relative error in estimation', or simply error in estimate) for each sampling gap k. For consistency, we have used this way of computing error in both our Matlab results in Chapter 4 and NS-2 results. When $T = 10$, the figure 6.2 shows a relationship between sampling gap and relative error in estimate. Figure 6.3 shows the same for T=100. We use $T = 100$ for computing error in estimate in the sampling results quoted in this section.

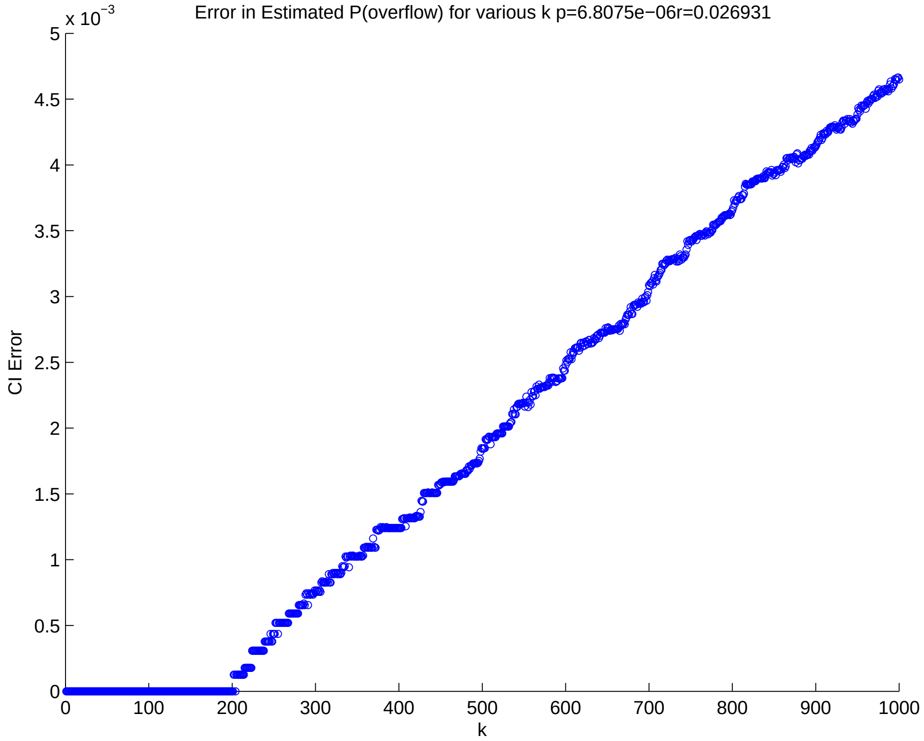


Figure 6.1: CI Error for the estimated $P(OF)$ for various k values, obtained for sample $T = 10$. Note that the samples did not capture any overflows for k=1 to k=200 in most of the experiments. This means that for upto 200*10=2000 slots, there were mostly no overflows.

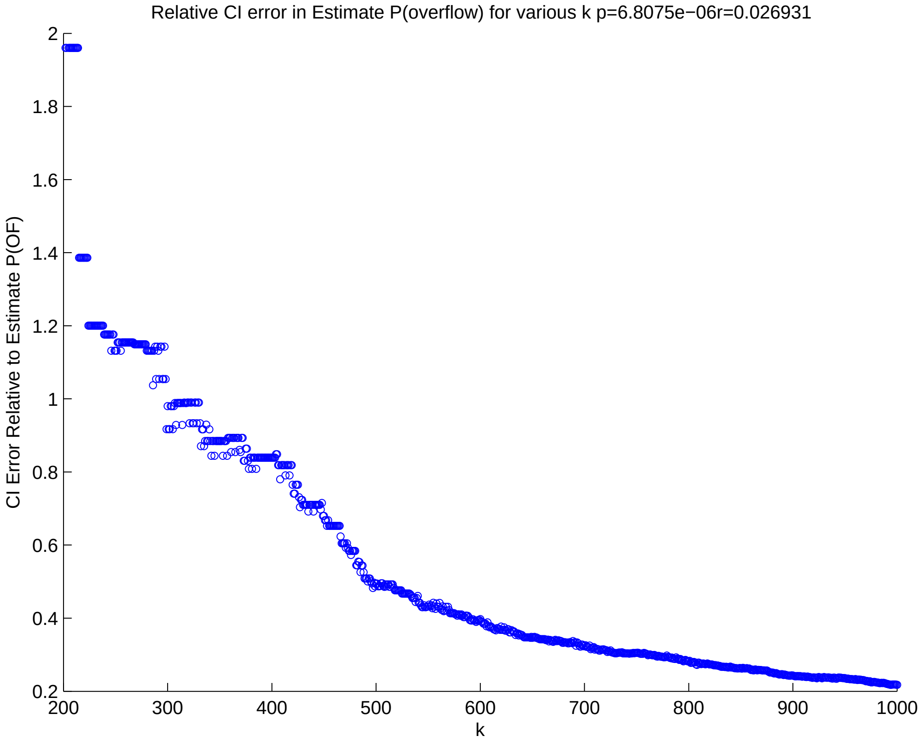


Figure 6.2: Relative CI Error for the estimated $P(OF)$ for various k values, obtained for sample $T = 10$. Here we do see that there were some overflows captured by the samples, but their presence is more obvious in this figure than in figure 6.1. This is because for computing the relative CI error we take CI error relative to the mean estimate for each k.

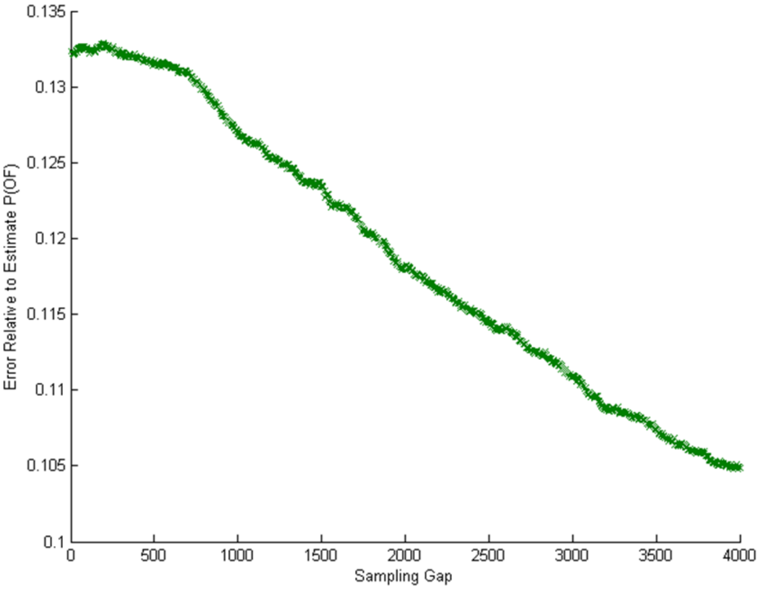


Figure 6.3: Relative Error for the estimated $P(OF)$ for various k values, obtained for sample $T = 100$ from VoIP simulation in NS-2

Note that in our Matlab experiments in Chapter 4 we capture enough occurrences of overflows (on average) within a sample size of $T = 20$, for each k. Taking a larger sample size does not change the results significantly (see figure B.2 in appendix B for comparison). However, we need to use a larger sample size of $T = 100$ for estimation from our NS-2 results. This is because the pattern of $1s$ and $0s$ is different in NS-2 outcomes, than the MC generated in Matlab with corresponding p and r.

Owing to a different distribution of *bursts* (and subsequently overflows) in our NS-2 simulation results, we compute p and r from NS-2 outcome (of $0s$ and $1s$). We notice that, for load of 80%, we get $p = 6.8075e-6$ and $r = 0.026931$ which are different from what we predict from analytical formulae in Chapter 3. Henceforth, we call these p and r values computed from buffer state distribution in NS-2 outcomes as $Experimental\_p$ and $Experimental\_r$ respectively. We then try these new $Experimental\_p$ and $Experimental\_r$ values to generate a MC in Matlab and sample that to see what the curve for Sampling Gap vs Relative CI Error looks like (figure [?] appendix C). We find that the sampling results for these p and r values are still different from those obtained from sampling NS-2 outcome. We thus argue that a two-state MC does not fully model the process of being in overflow or not, for a buffer that is fed data by several VoIP sources. We look at this further in section 6.3.

In order to look at some other type of data, we also simulated bursty data (ten times more bursty than VoIP, as in [45]). The simulation parameters used are given in Chapter 3 section 3.4. Error in estimation of P(OF) when sampling buffer states subjected to this bursty traffic is given in figure 6.4. The figure 6.4 shows relative error in estimate for various sampling gap values for load of 70% to 90%. Similar to the case of VoIP data, the relative error in estimate is larger for smaller sampling gap vales. That is, having samples farther apart brings more precision in estimation. However, similar to the case of VoIP traffic passing through a buffer in an NS-2 simulation, the relative error in estimate is lower for larger load. This seems counter intuitive (and opposite to what we see from Matlab results of modelling buffer states as a MC in section 4.2.1), but we see similar results in a different set of NS-2 experiments in section 6.6. Hence we can say that since the buffer states of overflow and not-overflow as recorded in NS-2 simulations may not follow MC process, so the burstiness in these results for larger load is such that it brings more precision in estimation.

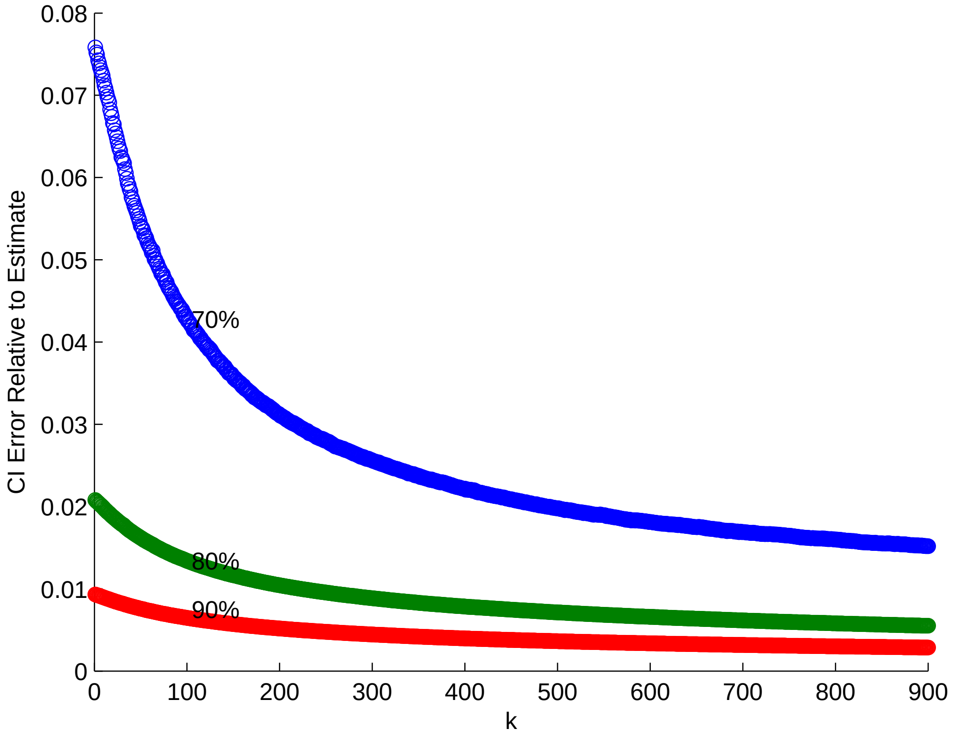


Figure 6.4: Error for the estimated $P(OF)$ for various k values, obtained for sample $T = 20$ Bursty data. Note that here the precision of estimate is higher for higher load. This is a result opposite to what we have seen in figure 4.4. This is because the scenario in practice (in NS simulation), there are several other parameters that come into play, which are not there in analytical formula. We can see in section 6.5, that for larger load, we need to keep a larger buffer. And precision seems to increase with larger buffer sizes.

6.2 Computing Experimental Asymptotic Variance of Estimate for NS2 Experiments

We have included results of experimental asymptotic variance in Chapter 5 section 5.3. We discuss how we computed the variance in this section. We also cover the experimental asymptotic variance for k larger than 1 (which we have not included in 5.3) in this section. In order to compute experimental asymptotic variance of the estimated overflow probability, we choose to use $T = 100000$. This is because $P(OF) = p/(p+r) = 2.571e-4$ for 80% load. Therefore in order to obtain at least 10 instances of overflow, we need a sample of $100,000$. The theoretical asymptotic variance for this experiment for 80% load is 0.24. The experimental asymptotic variance turns out as 0.56075. We can say that theory (for asymptotic variance of a semi-MC) over-estimates the precision of estimation. This means that we need to take more samples in practice for precision in estimation. This is an important finding of our research.

Table 6.1: Theoretical vs Experimental (NS-2) Asymptotic Variance computed for load of 70% for $k = 1$ to $k = 5$ for VoIP Data

load %	k	p	r	Theoretical vs Experimental Asymptotic Variance	Experimental (NS-2) Asymptotic Variance eq 5.2
70	1	0.036635e-4	0.015319	0.0312148	0.41489
70	2	0.072708e-4	0.030403	0.015728	0.9789
70	3	0.108229e-4	0.045256	0.010566	1.0331
70	4	0.143206e-4	0.059882	0.007985	0.8743
70	5	0.177647e-4	0.074283	0.006437	0.7693

Table 6.2: Theoretical vs Experimental (NS-2) Asymptotic Variance computed for load of 80% for $k = 1$ to $k = 5$ for VoIP Data

load %	k	p	r	Theoretical Asymptotic Variance	Experimental (NS-2) Asymptotic Variance
80	1	6.8075e-6	0.026931	0.0188	0.5608
80	2	0.134316e-4	0.053136	0.0095	1.4189
80	3	0.198773e-4	0.078636	0.0064	1.6108
80	4	0.261493e-4	0.103449	0.00488	1.5401
80	5	0.32252e-4	0.127593	0.00396	1.3692

Table 6.3: Theoretical vs Experimental (NS-2) Asymptotic Variance computed for load of 90% for $k = 1$ to $k = 5$ for VoIP Data

load %	k	p	r	Theoretical Asymptotic Variance	Experimental (NS-2) Asymptotic Variance
90	1	6.5963e-6	0.023239	0.024421	0.0001219
90	2	0.130392e-4	0.045938	0.0123543	0.0774
90	3	0.193324e-4	0.068109	0.008332	0.2888
90	4	0.254794e-4	0.089765	0.006322	0.6283
90	5	0.314834e-4	0.110917	0.005117	0.9536

We can see from the experimental *p* and *r* values in the tables 6.1 to 6.3 (especially for load of 80% and 90%) that the burstiness is similar in both cases (*p* values are similar) which means that the outcomes of 0s and 1s in these cases is quite different from what we expect analytically. We also see that theoretical asymptotic variance decreases for increasing *k* for corresponding p and r values for different *k*, when considering these (p and r values) as transition probabilities of a MC. On the other hand, from the NS-2 results we see that the asymptotic variance increases for larger k.

Owing to this difference in the theoretical and experimental asymptotic variances, in the next section, we look at the distribution of duration of buffer overflow states to see whether it fits an exponential distribution (for verifying whether a two-state MC is a good approximation of modelling the buffer overflow state)

Table 6.4: Theoretical vs Experimental (NS-2) Asymptotic Variance computed for load of 90% for $k = 1$ to $k = 5$ for Bursty Data

load %	k	p	r	Theoretical Asymptotic Variance	Experimental (NS-2) Asymptotic Variance
70	1	0.04817e-3	0.04649	0.04453	11
80	1	0.0003050	0.0083403	8.44	202
90	1	0.0014449	0.002248	204	737

6.3 Fitting Distribution of Sojourn times in Overflow

In this section we discuss fitting the distribution of sojourn times in overflow and non-overflow states to determine whether it indeed fits to exponential distribution (Note that in case of a two-state MC, the distribution of sojourn times in each of the two states fits exponential distribution). If we look at the distribution of sojourn times in overflow and non-overflow states from an example scenario of 80% load in NS-2, the distribution seems to follow Power Law (figure 6.5), to the naked eye. However if we investigate further, we can see that the distribution is exponential ($f(x) = ae^{bx}$ with $a = 54$ $b = -0.5$) for say up to 10^2 consecutive overflow states. This is depicted in figure 6.6 for 80% load. Since the Distribution fits to exponential distribution with R-squared fit statistic of 0.99 for up to 10^2 consecutive overflow states, however, we can see that long the tail of the distribution of sojourn times in overflow state, weakly fits a power law distribution ($f(x) = ax^b$ with $b = -0.65$). This can be seen in Figure 6.8. To elaborate further that the overflow distribution fits a power law we show the distribution of sojourn times in overflow state for load of 60% to 90%, for up to 10^2 consecutive overflow states. In figure 6.8 we have plotted the Probability Density Function (PDF) of the tail of the distributions of consecutive buffer overflow states for 60% to 90% load, and have fitted the tail of 70% load to a power law with power -0.65. Thus we can say that our distribution fitting is inconclusive, but we can say that the distribution of the lengths of buffer overflows and non-overflows, when seen in an NS-2 simulation, do not follow a MC process. Thus we re-iterate our result that buffer states as captured from an NS-2 experiment, when the buffer is subjected to bursty traffic, do not follow a MC process.

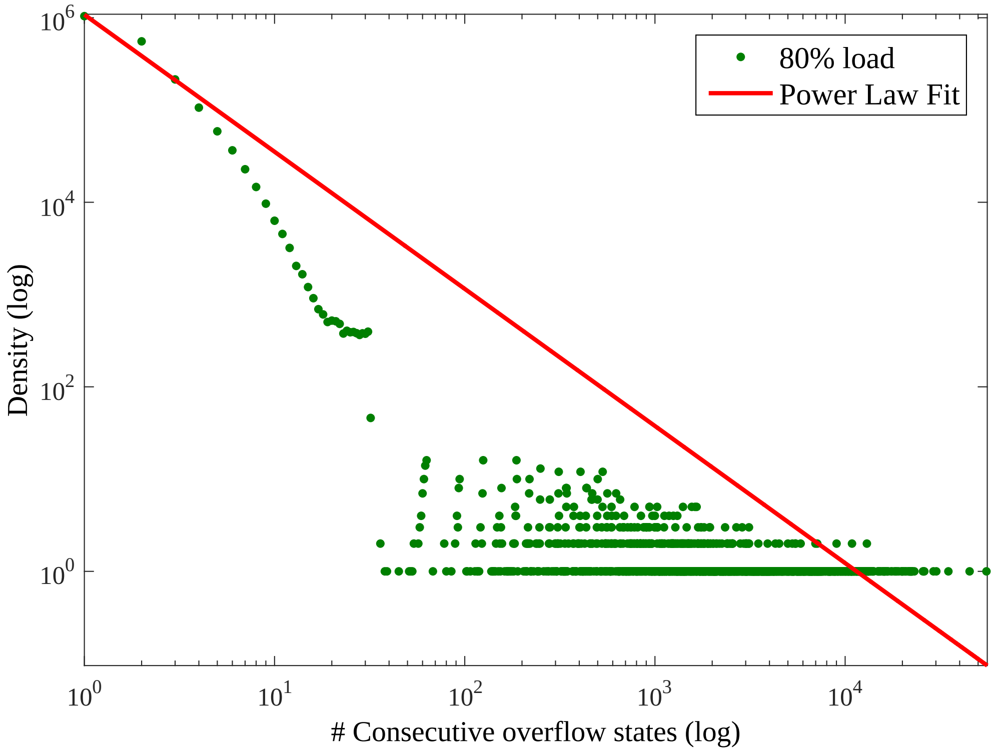


Figure 6.5: Distribution Fit on Sojourn Time of the buffer in Overflow state. GoF: When plotting frequency distribution of different sojourn times in overflow state, the curve fitted to a Power law Distribution ax^b

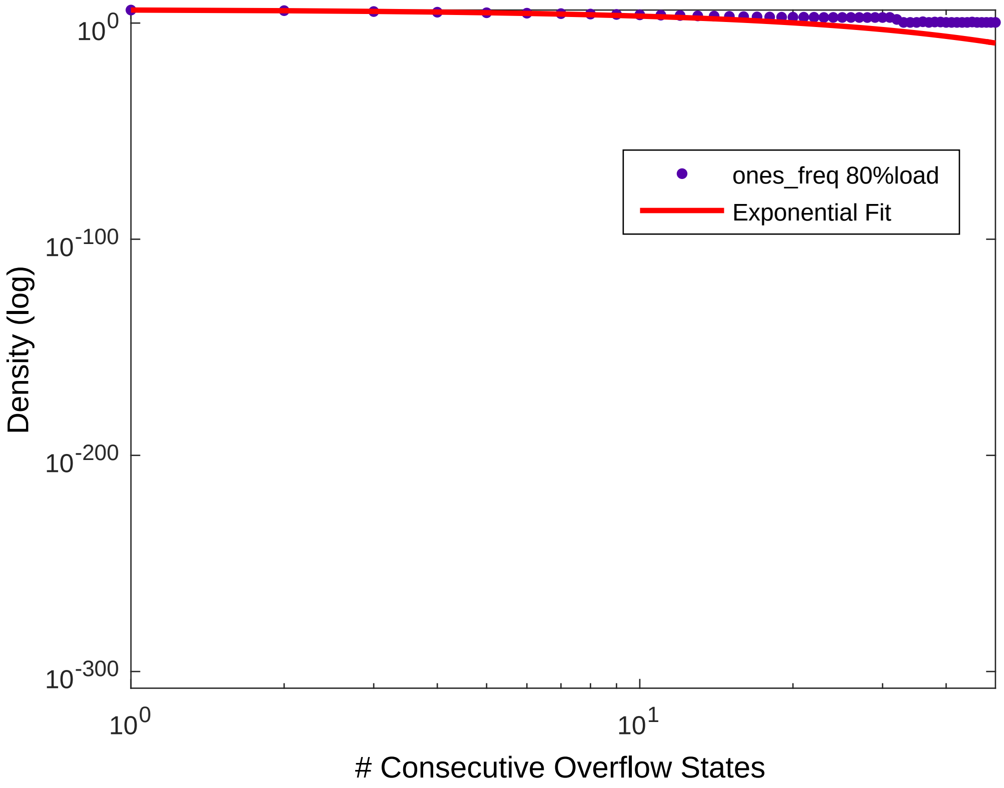


Figure 6.6: Exponential Distribution Fit ($f(x) = ae^{bx}$) on Sojourn Time of the buffer in Overflow state, on the body of the Distribution, that is, the smaller sojourn times in overflow state.

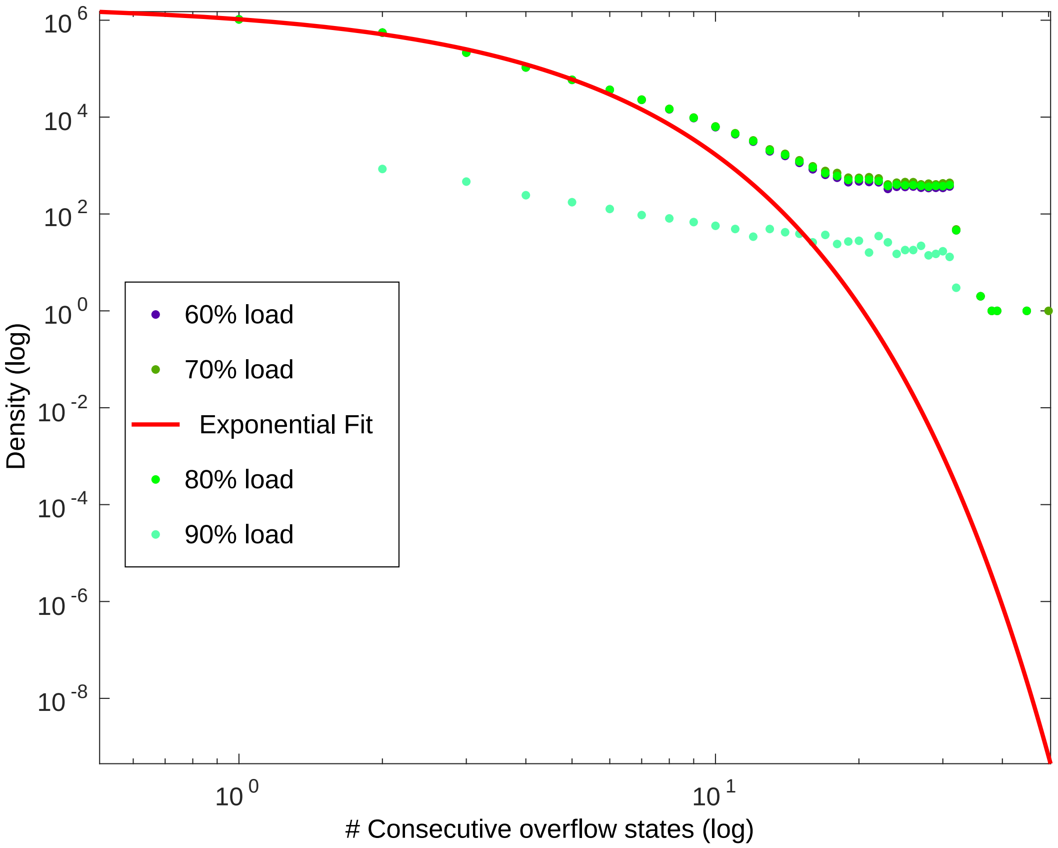


Figure 6.7: Distribution Fit on Sojourn Time of the buffer in Overflow state, on the body of the Distribution (smaller sojourn times of upto 10^2). Showing 60% to 90% load. ($f(x) = ae^{bx}$) $a = 2.14e6$, $b = -0.715$

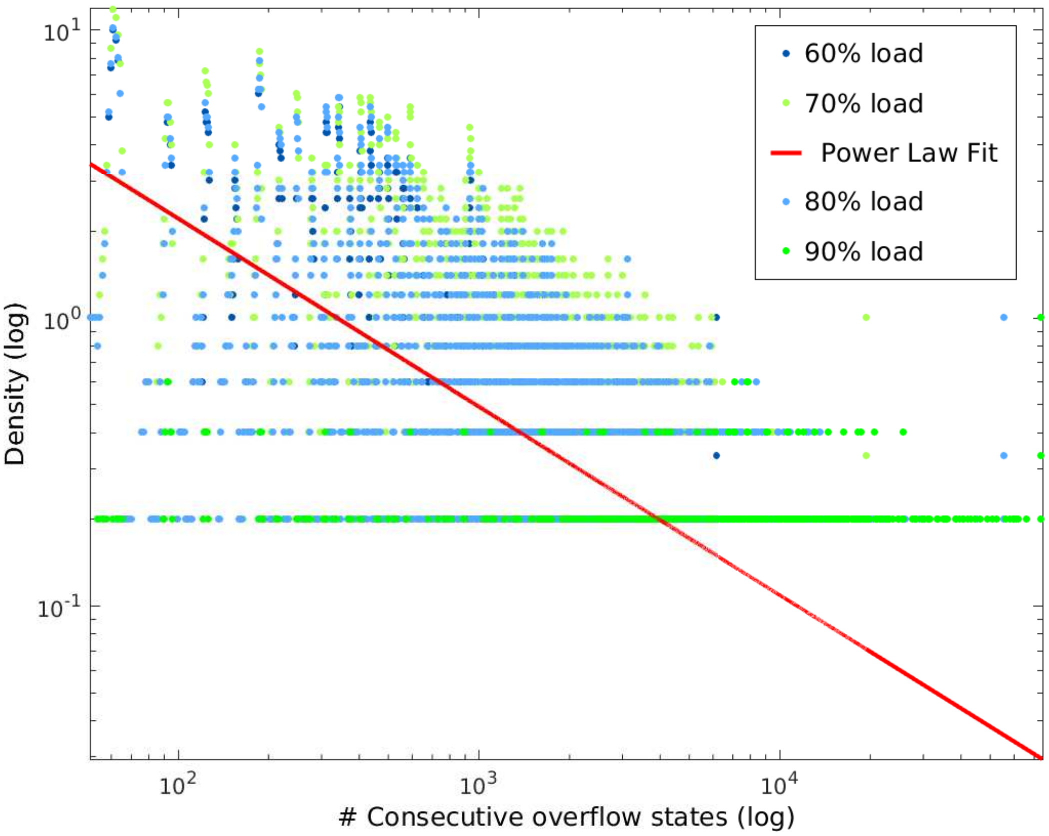


Figure 6.8: Distribution Fit on Sojourn Time of the buffer in Overflow state, The tail of the distribution for 80% load fits Power Law distribution (showing 60% to 90% load) $f(x) = ae^b$ and here a=44, b =-0.65

6.4 Cleaning the Data for Distribution Fitting

Since the distribution of 0*s* and 1*s* from NS-2 results looks heavy-tailed, we resort to techniques for analysis of such distributions. (Note that in heavy-tailed distributions, the distribution has a long range of values where the long tail has small but non-negligible probability). We can see the Curve fitting included in section 6.3 is not convincing, which may be because of statistical noise introduced due to finite sampling. We feel that applying a cleaning technique to the data would result in better fitting. We this use two different data cleaning techniques taken from [63]:

1. Data Cleaning by Integration (Cumulative degree distribution)

2. Binning the Data

When cleaning the data by binning, we bin the data into exponentially wider bins and normalise by the width of bins. Figure 6.9 shows the binned degree distribution of sojourn times of the buffer in *OF* state. The bin width in figure 6.9 is 500. We can see that the distribution can still be seen as a two-part, with some more noise introduced. We later try cleaning the data by integration to see any improvement.

In case of Data Cleaning by Integration, the cumulative degree distribution is computed according to:

$$P^c(k) = \sum_{k'=k}^{\infty} P(k') \tag{6.1}$$

The function $P^c(k)$ is, simply put, integration over k of the distribution. According to [63], this cumulative distribution makes it possible to identify a cut-off of the distribution as it provides a smoother tail. Figure 6.10 shows the cumulative degree distribution of the frequency of sojourn times in overflow state, for 80% load. We can see that the distribution indeed looks like two separate distributions, with a cut-off point at k=31. We try fitting these two curves and find that they both fit to two different exponential distributions, meaning a two state MC is too naive an assumption.

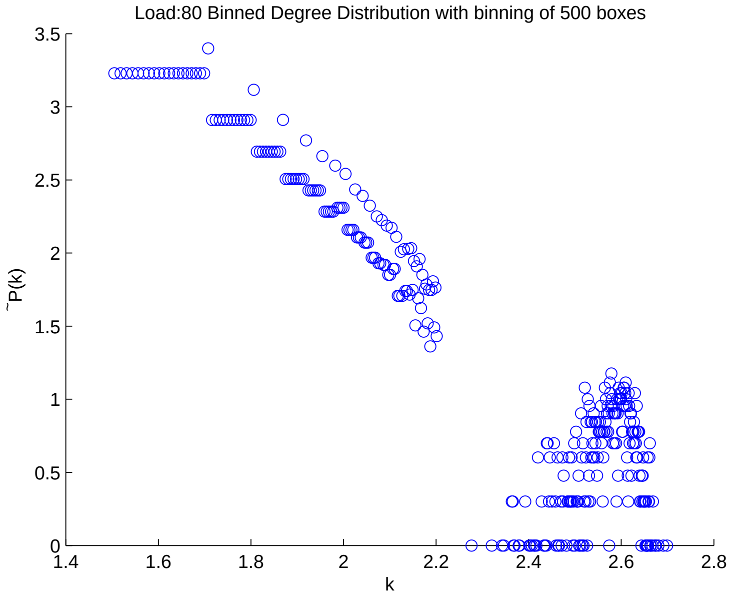


Figure 6.9: Binned Degree Distribution of Frequency of Sojourn Times of the buffer in Overflow state with binning of 500 boxes

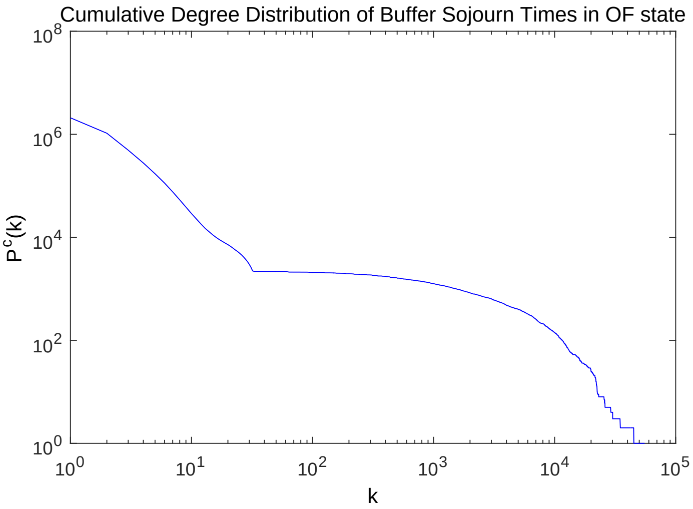


Figure 6.10: Cumulative Degree Distribution of Frequency of Sojourn Times of the buffer in Overflow state

6.5 Relationship between Buffer Length and Accuracy of Estimate when Probing a Network

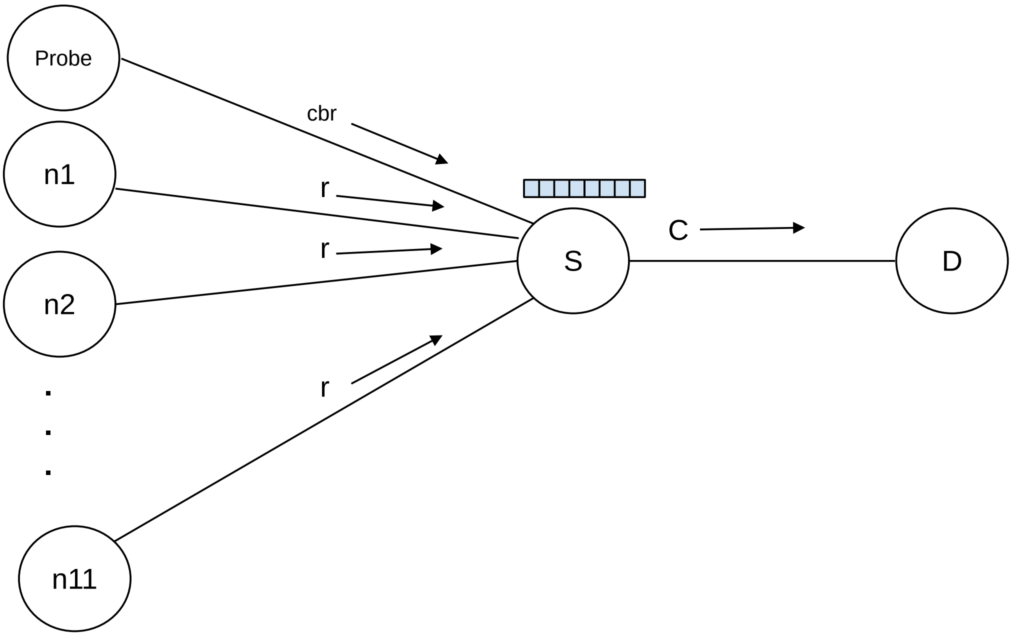


Figure 6.11: Topology for Probing aggregate VoIP traffic passing through a bottlenect node during a busy hour

In Chapter 3 section 3.5 we have covered an analytical formula that relates simulation parameters with PLP and buffer length. We have also mentioned how the formula did not work for our scenarios as it is based on approximations. Therefore we tried different buffer length values for our load under consideration (70% to 90%) for obtaining buffer length values that give the same PLP of 10^{-3}. These lengths were: $x = 310$ for 70% load, $x = 1100$ for 80% load and $x = 3000$ for 90% load for VoIP sources. We elaborate further how buffer length effects the loss probability measurement by the following figure 6.12. Here we use probing to measure the *PLP* in a buffer and measured how accuracy of measured *PLP* decreases as we increase the buffer length (that is, accuracy increases with increased buffer length). The results are from a simulation of a one-hour busy period. The parameters used in each experiment are given in table 6.5. Table 6.6 shows the probe set-up for probing the traffic generated from scenario 6.5.

Table 6.5: Simulation Parameters for Probing 1 Busy Hour

Parameter	Value	Unit	Unit2
Packet Size	L	424 bits	53 bytes
Number of Sources	N	10	
Packet Rate of Src	rate R	167pps	
Capacity of agg link	C (adjusted for retaining PLP for different queue lengths)	270000bps	636.7925pps
Mean Arrival Rate into buffer	Ap	604.9811	
Buffer Load	ρ	95%	
Burst Time	T_on	0.96sec	
Idle Time	T_off	1.69sec	
Time unit	$(C/(L * bytesize))^{-1}$	0.0016 sec	

Table 6.6: Probe setup for Probing 1 hour of busy period

Simulation Duration	4800 sec
probe start time	1000 sec
probe finish time	4600 sec
probe packet length	53 bytes
probe rate	1 pps

The figure 6.12 plots the buffer state probability for for different available buffer lengths. In each experiment, the buffer length was recorded after every service-time unit. (The queue length is recorded after every 0.0016 seconds, which is the service time for one packet in this scenario).The figure 6.12 also plots the actual PLP in each experiment, and PLP as seen by a probe. In the results included in figure 6.12, a probe is sent once per second. The actual *PLP* from the experiment is represented by asterisk in the figure, while the *PLP* as measured by a probe is shown by a circle. We can see that the estimate and actual *PLP* overlap for smaller queue length and the gap between the two increases as we increase the queue length. This is because, when we increase the buffer length, the chances of overflow are reduced. Since the chances of occurrence of an overflow reduce, this means that *OF* becomes an even rarer event, and capturing a rare event by probing (as in this case) or sampling becomes even more difficult (Note that probes have captured no loss for queue length of 10000 in figure 6.12). Hence error arises in estimation of occurrence of this rare event. Thus we can say that the accuracy of the estimate increases with the increase in buffer length (evident from the increasing gap between the circle and asterisk for each buffer length).

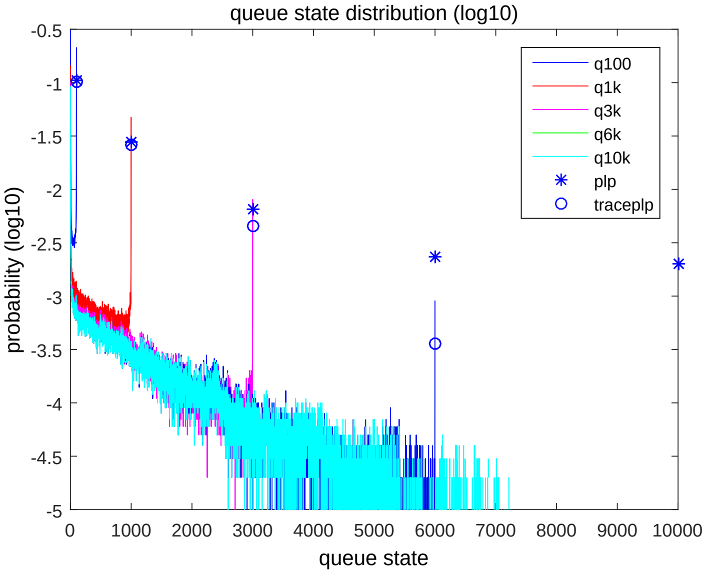


Figure 6.12: Buffer state probability with different available buffer sizes. Also includes *PLP* as seen by a probe vs actual PLP for different buffer lengths

6.6 Relationship between Probing Rate, Probe Window and Error in Estimation

In Appendix C we have included a table for different probing rates and number of probes. We can see that the estimate is be more accurate if probe window covers longer duration with larger number of samples. We have computed Wilson's Confidence Interval showing that the interval is narrower (more precision) with longer probing period for same probing rate. In this section we have included one for a probing rate of 1.5 probes per second, for different buffer lengths, with same PLP of 0.0045. We can see from the results that the estimation is most precise for probing over the whole hour rather than shorter probe windows. A more important result here is that even with larger buffers, although accuracy of estimate reduces, precision increases.

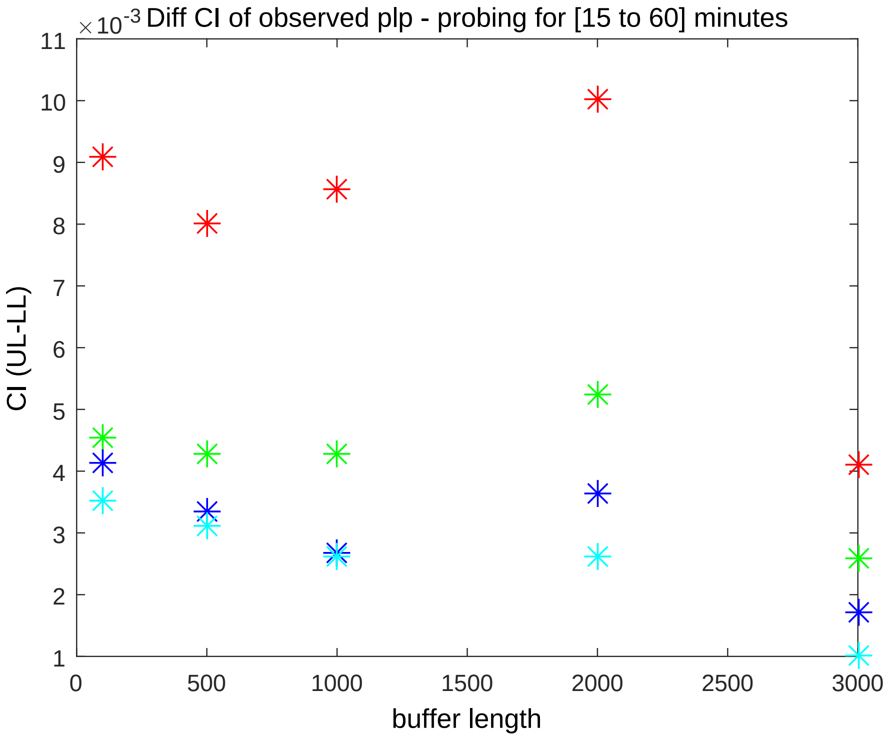


Figure 6.13: CI width (UCI limit - LCI limit) of *PLP* estimates when probing for different durations, for different buffer lengths. CI has been computed using Clopper-Pearson Interval. Here red: probed for 15 minutes, green: probed for 30 minutes, blue: probed for 45 minutes, cyan:probed for 60 minutes.

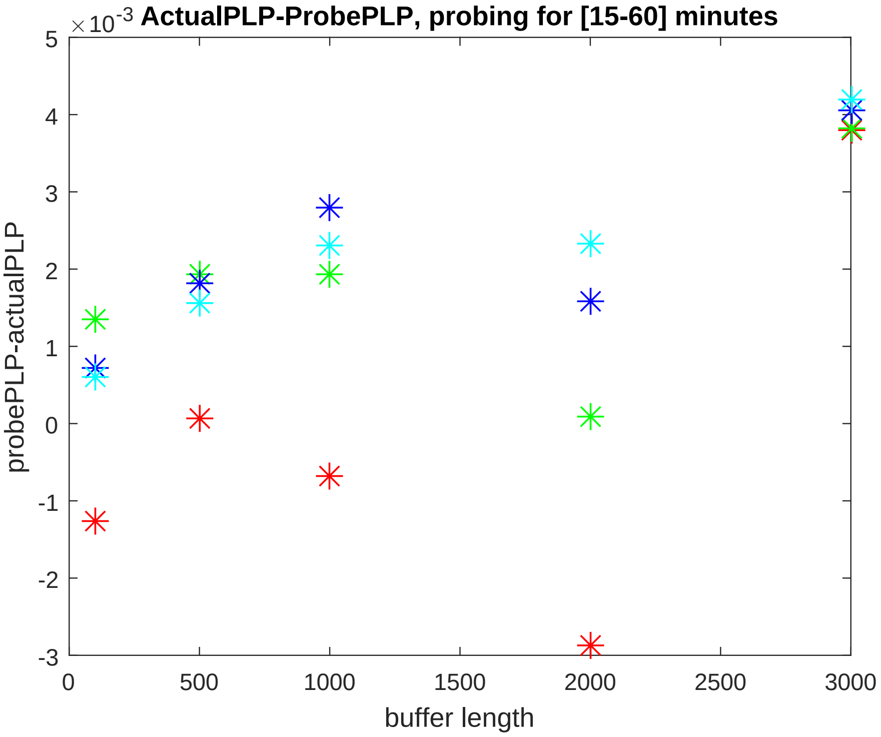


Figure 6.14: Difference in *PLP* (Actual PLP - Probe PLP) when probing for different durations, for different buffer lengths. Here red: probed for 15 minutes, green: probed for 30 minutes, blue: probed for 45 minutes, cyan:probed for 60 minutes.

Note that, for keeping the PLP same in experiments with different buffer sizes, we have increased the load on the network node under consideration. We an see that the accuracy of estimate decreases with increase in the buffer length (evident from the increase in difference between actual PLP and estimated PLP in figure 6.14). On the other hand, with increase in buffer length, precision of the estimate increases (figure 6.13). This agrees with our earlier NS-2 results where the precision is higher for 90% load than 70% load in figure 5.6

6.7 Accuracy in Estimation

It is interesting to look at accuracy of the estimate (that is, how close is the estimated P(OF) to the actual P(OF)) when precision seems to increase with increase in sampling gap. We can see from the results included in this section (figure 6.15, figure 6.16) and figures C.3 to C.8 in Appendix C, that for results from NS-2 experiments, the accuracy does not necessarily decrease when precision improves. That is, even though we get higher precision for larger sampling gap k values, the accuracy (lower accuracy) does not necessarily improve for larger sampling gaps. It is to note that the shape of the curve is not significant in the figures included in this section. However, we are looking at whether the estimated P(OF) gets close to the actual P(OF) shown by the straight horizontal lines in these figures.

We have included figures C.9 to C.11 in Appendix C (and figure 6.15 in this section) that show accuracy versus sampling gap when sampling a MC generated in Matlab for VoIP parameters (p and r obtained from NS-2 experiments for corresponding load values). We can see that when the overflow estimation is for buffer states modelled as a MC, then accuracy is not affected by sampling gap.

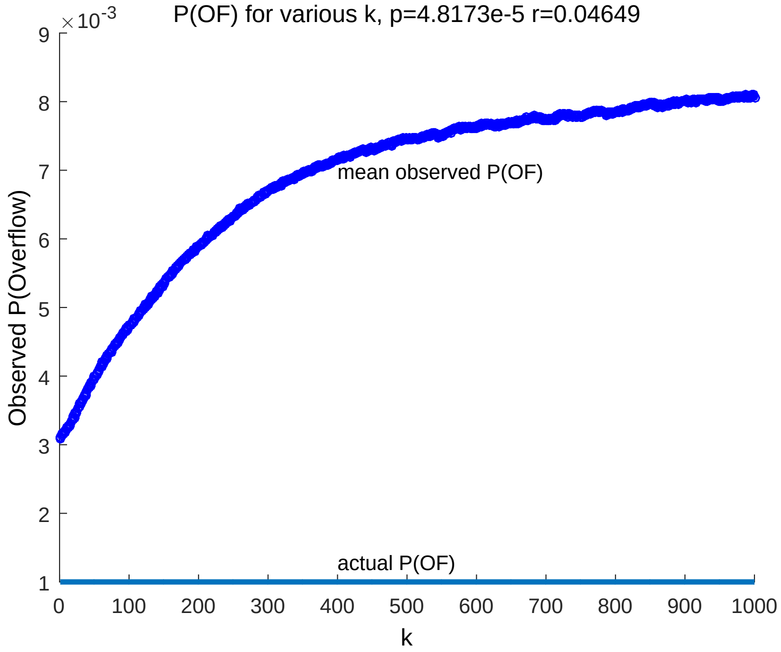


Figure 6.15: Accuracy of estimated $P(OF)$ for load of 70% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with bursty data sources

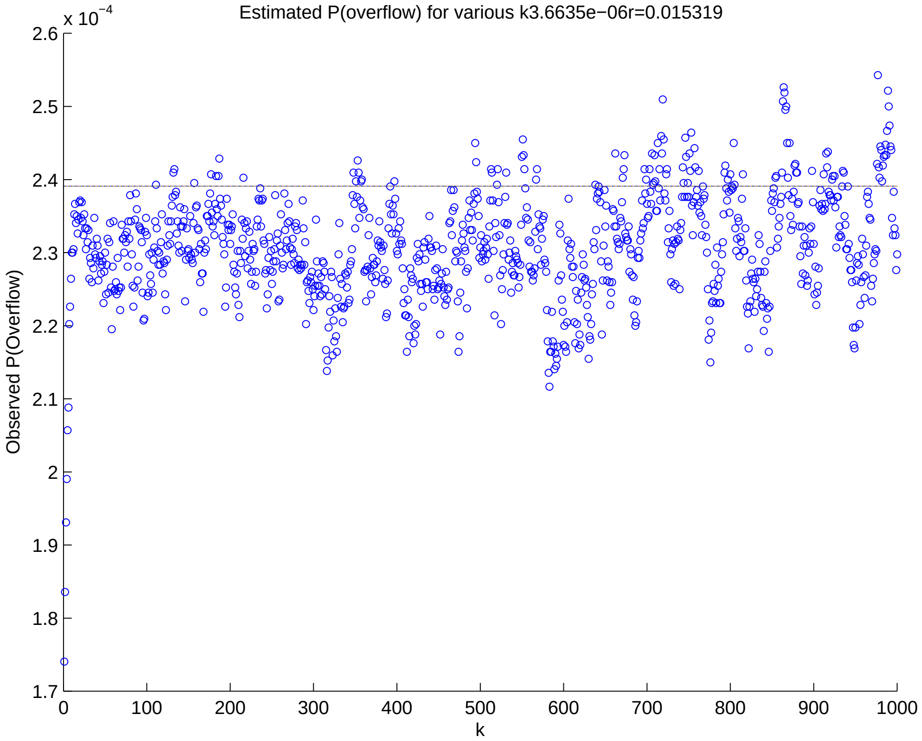


Figure 6.16: Accuracy of estimated $P(OF)$ for load of 70% when using sample size $T = 100$ for sampling buffer states as seen in a Matlab experiment with parameters for VoIP data sources. The horizontal line in the figure shows the actual known $P(OF)$ and the circles represent individual estimated mean $P(OF)$ for each k

6.8 A Practical Example

We have discussed PBAC scheme by Mas et al [4] (later enhanced by Sathik et al [35]) in Chapter 2 where probes are sent for a period of between 1 second and 10 seconds for making a connection admission control decision. A simple network topology that Mas et al [4] take into consideration is given in figure 6.17. The admission control decision is made by the root node. This scenario fits well with our topology under consideration represented by figure 4.5.

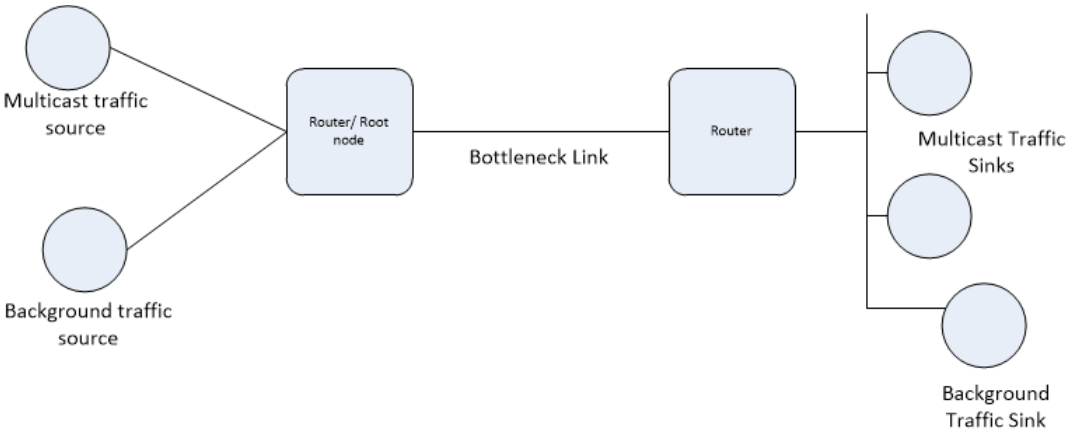


Figure 6.17: A simple Topoloy as considered by Mas et. al. [4]

Let us consider our sampling scenario for VoIP traffic with a load of 80%, results for which are shown in figure 6.3. (And theoretical results in figure 4.3). If we were to sample at a gap of $k = 250$ with a sample size of $T = 100$, considering the time unit is 0.000188 seconds (for service rate 2Mbps and packet size of 47 bytes, service time for one packet= $2Mbps/8 \times 47$=0.000188), we would be able to measure the overflow probability within 5 seconds. For the same sample length, if we choose a small gap of, say $k = 1$, then the estimation will be finished within $1/50^{th}$ of a second.

Roughan [1] suggests bounds on accuracy of queueing delay measurement for M/G/1, M/D/1 and M/M/1 queues. As discussed in chapter 4 section 4.4, Roughan shows that having samples close together results in higher error in estimation of mean queuing delay because of correlation between samples. We can see from error values which we have computed according to Roughan's equation for error in mean number in queue for M/D/1 queue in table 4.1 that the relative error in estimation decreases by a third to 1.07 for $k = 2$ compared with 1.5 for $k = 1$. At $k = 150$ (having samples 150 time units apart), the relative error becomes 0.3. We can see from our results in figure 4.7 (for N-on-off/D/1 queue) that the difference in relative error for sampling gap of $k = 1$ and that of $k > 1$ is not too large for higher load of 80%-90%(for example the relative error is 0.132 for $k = 1$ and it is 0.13 for $k = 150$ for 80% load). Thus we can say that contrary to Roughan, taking samples close together when estimating buffer overflow probability does not result in loss of precision and we can have samples (or probes) close together in order to estimate within a short time span. Thus as in the example above, having 100 samples at a gap of $k = 1$ will enable us to estimate within 0.02 seconds ($1/50^{th}$ of a second), which is ideal for the MBAC scenario in Mas et al [4] and Sathik et al [35]. This has great potential significance for network operators as they need not worry about correlation between samples when sampling for loss probability, in order to obtain an estimate in a timely manner.

6.9 Sampling TCP Traffic

We have covered UDP traffic up until now in this thesis. In this section, we look at sampling TCP traffic. We create *N* TCP sources sending FTP traffic. This traffic passes through a bottleneck node on it's way to the receiver. The buffer under consideration is at the bottleneck node. Simulation parameters are given in table 6.7. We use TCP Reno agents in the simulation and keep default values for the TCP window size in NS-2.

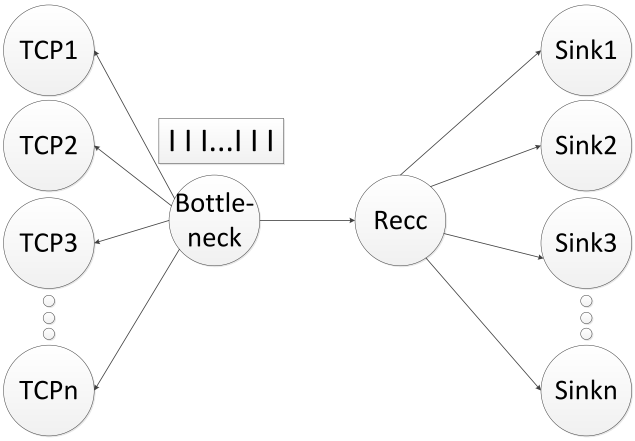


Figure 6.18: Topology for TCP traffic simulation scenario. Each TCP source is sending FTP traffic.

Table 6.7: Simulation Parameters for TCP traffic scenario

Parameter	Value	Unit
Packet Size	L	1040 bytes
Number of Sources	N	70
Rate of Src link	rate R	499200bps
Capacity of agg link	C	20.8Mbps
Time unit	$(C/(L*bytesize))^{-1}$	0.0004 sec

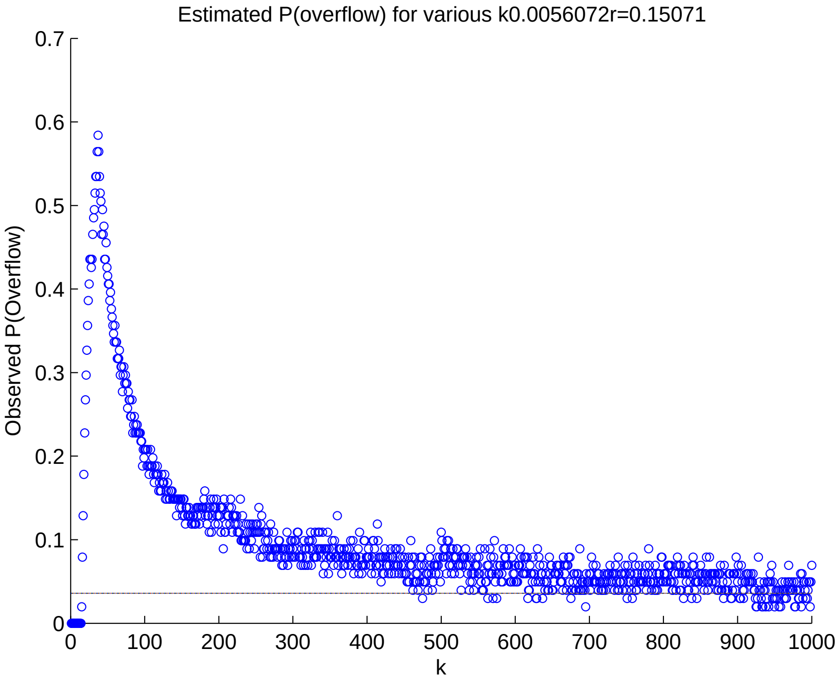


Figure 6.19: Accuracy of estimated $P(OF)$ when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with FTP sources sending data over TCP

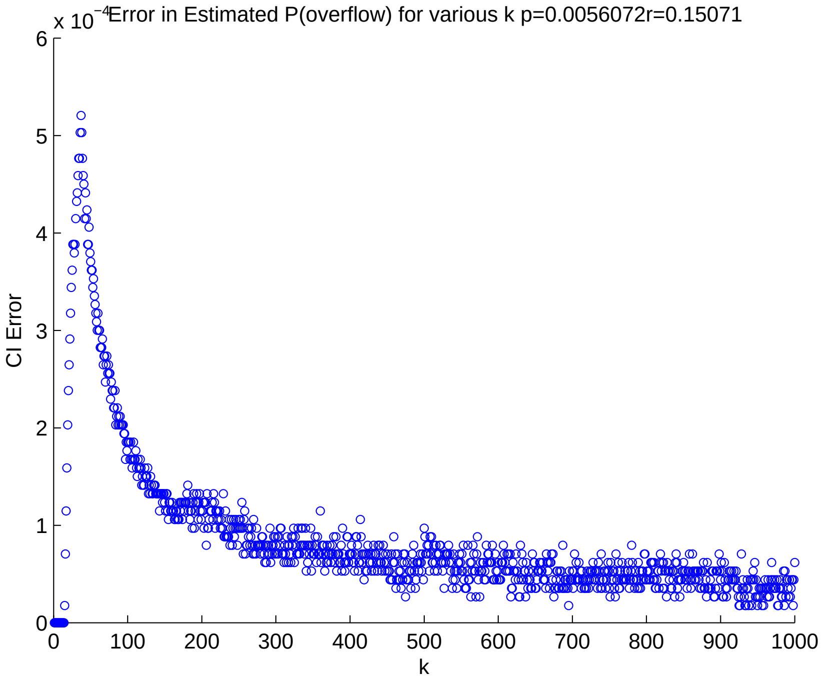


Figure 6.20: Error in estimated $P(OF)$ when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with FTP sources sending data over TCP

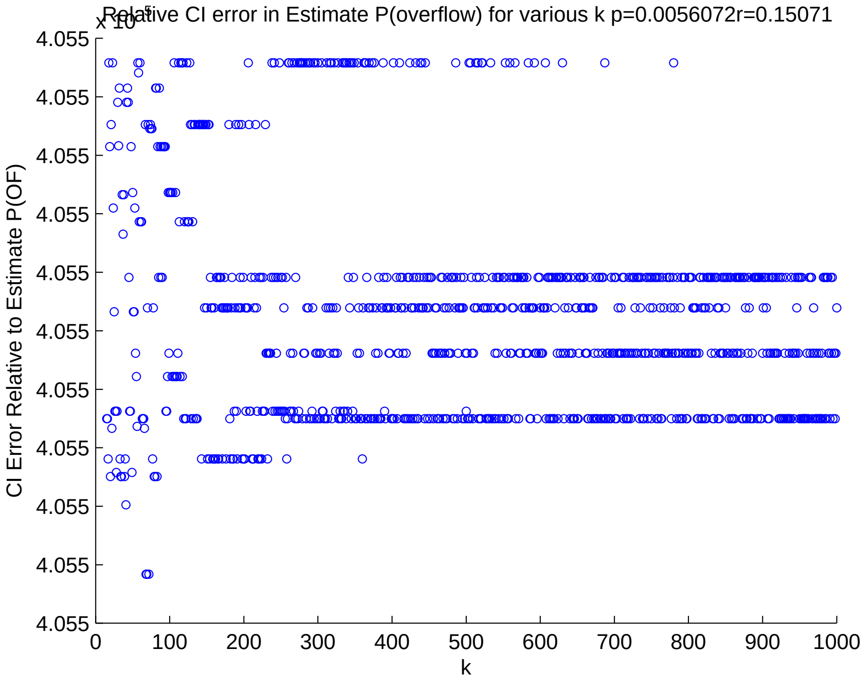


Figure 6.21: Error relative to estimated $P(OF)$ when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with FTP sources sending data over TCP

We see from simulation results with TCP traffic that we cannot draw similar relationship between error in estimate of overflow and the sampling gap, as in the previous scenarios of static traffic. The error in estimate does not show a trend with respect of sampling rate. This is due to the elastic nature of TCP and the fact that because the repeated experiments have similar distribution of overflows (since TCP reacts to overflow and tunes the traffic accordingly). We see from figure 6.19 that the accuracy in estimate of $P(OF)$ decreases with increase in sampling gap in this scenario. However, in figure 6.21 we see that the relative error in estimate of $P(OF)$ does not necessarily decrease with increasing sampling gap. Note that in figure 6.21 we plot the error in estimate of $P(OF)$ for each sampling gap value k relative to the mean estimated $P(OF)$ for that k.

6.10 Summary

In this chapter we have looked at error in estimation of buffer overflow probability when sampling a buffer simulated in NS2. We have seen that the trend for precision of error in estimation is similar to the results from Matlab experiments. That is, precision of the estimate increases as we lower the sampling rate (larger sampling gap). However we have also noticed that the distribution of bursts that the buffer under consideration is subjected to, is different from what we expect based on analytical results for MC parameters. Thus the distribution of overflow and non-overflow states in this case is not as that of a MC (with the expected parameters). We have also explored the possibility that the overflow distribution could be from a MC with different parameters, but have found that even if we derive MC parameters from the outcomes in NS-2 simulation, we cannot reproduce a similar trend of error in estimation vs sampling gap for these parameters, in Matlab. This confirms our belief that the buffer overflow distribution is not a simple two state Markov Chain, and hence an MC model may be too naive in practice. Results in this chapter also show that precision of estimate increases with increase in load (if we keep the same PLP when tuning simulation parameters), even though accuracy of the estimate decreases with increase in load since we have increased the buffer length.

Additionally we have looked into fitting distribution of overflow (and non-overflow) to an exponential curve, assuming that this should be a good fit if the outcome were from a MC distribution. However, we notice that the long tail of distribution does not fit to an exponential distribution. Further, we also show that although there is a relationship between sampling gap and precision of estimation of overflow, we find no relationship between sampling gap and accuracy in estimation. We look at a practical example of esstimating $P(OF)$ and show how estimation can be performed within time bounds of MBAC schemes in literature. And ascertain that despite the improvement in precision with larger sampling gap, for a case of low $P(OF)$ of 10^{-3} in our scenarios, the error at $k = -1$ is also acceptable and is enough to make timely $P(OF)$ estimation. Finally, we also look into estimation of P(OF) for elastic traffic (TCP) and we do not see a relationship between estiation of $P(OF)$ and sampling gap.

Chapter 7

Conclusions

In this research, we have looked at the relationship between sampling rate and error in estimation of buffer overflow probability in the light of earlier work [1] which suggests that samples may have to be very widely spaced, especialy at higher loads. We find that the precision in estimation of buffer overflow probability through sampling is a function of sampling rate. In other words, the variance of estimate of buffer overflow is strongly dependent on the sampling rate. For VoIP and bursty traffic sources that we have simulated in NS-2 (as well as Matlab), the error in estimation decreases with decrease in sampling rate (when keeping the number of samples fixed), until an asymptotic level beyond which the error does not change much.

A major contribution of this work is devising a new way of validation with theory, using formulas in recent literature [2] for asymptotic variance of our buffer overflow estimate. We have been able to compute theoretical values of asymptotic variance of the buffer overflow estimate for our VoIP scenarios under consideration, from the general formula for semi-Markov processes given in [2]. We have seen that the overflow probability of a buffer at a network node, when modelled as a two-state Markov Chain (MC), indeed conforms to theoretical results obtained according to [2]. Further, we have shown analytically how to use the formula in [2] for calculating theoretical buffer overflow probability for sampling gap larger than 1 time unit, and have shown that for increasing gap between samples (or decreasing sampling rate), the asymptotic variance estimate indeed decreases. This validates our findings that error in estimate decreases (or precision increases) when increasing gap between samples.

We have also found that representing the buffer at a network node as a simplified model of a two-state MC, as done in previous literature including [3] [16] is not realistic. In order to verify this, we set up an NS-2 simulation of N on-off/ D/ 1 queue, where N on-off sources multiplexed together feed a buffer at a network node. The N on-off sources represent VoIP like traffic flows. Then we observe the buffer states of being in overflow or not being in overflow, and sample them. The sampling results thus obtained are different from that of an MC simulation. When we use the transition probabilities from NS2 simulations and substitute them into a system to generate an MC, the resulting MC gives different results than what we get when sampling the buffer overflow in NS2 simulation. This is because the overflow states are either too close due to burstiness of traffic, or too sparse. Thus the MC model with exponentially distributed sojourn times is not realized and we conclude that a two-state MC does not accurately represent the states of a real buffer at a network node.

We have briefly looked at sampling the buffer in case of elastic traffic (TCP) scenario for buffer overflow measurement. From our simulation with TCP traffic we have found that the relationship between the error in estimate and sampling gap is not similar to that in case of UDP-only traffic scenarios explored in this thesis. We suggest that this is because TCP traffic exhibits synchronisation over time that is completely absent from static (e.g. VoIP traffic).

Another contribution of this work is that we have compared sampling for measuring buffer overflow probability with Roughans work [1] on measuring queuing delay. We have seen that in comparison to sampling for mean queuing delay, where having samples farther apart reduces error in estimation, we can achieve a higher precision with a small sampling gap (higher sampling rate) when sampling for measuring buffer overflow probability. This is a significant finding since, contrary to Roughan, this means that having samples close together for speedy estimation of network conditions is possible without significant loss of precision due to correlations between samples.

7.1 Future Work

In this section we propose some future directions for this work. In our experiments, we have mainly considered UDP-only traffic (and TCP-only traffic) for simplification. Similar work can be done for estimating loss probability when dealing with a mix of TCP and UDP traffic. TCP protocol requires feedback for receiver and adjusts the traffic according to the network conditions. This makes it more complex to measure loss probability with uniform probing.

A closely related scenario, when estimating buffer overflow probability is that of video play-out buffer in case of video streaming. The play-out buffer at a video streaming recipient makes sure not to playout the stream at a rate too high or too low for its needs. The rate is adjusted not to be too high so that it does not underflow, that is, it has enough buffered data to stream while network conditions are changing. The rate is adjusted so that it is too low, such that the video quality is below the user requirements. Therefore predicting the buffer overflow and underflow probability for play-out buffer at the recipient is important for QoE guarantees. Considering that mostly video streaming is now done on top of TCP protocol (HTTP over TCP/IP), this complicates overflow estimation further because TCP forces hosts to adjust traffic according to network conditions.

We have used homogeneous traffic in our simulations, for simplicity. Another aspect to consider in future is to estimate buffer overflow probability at the bottleneck node, in case of heterogeneous traffic arrivals. However, this will need careful planning as the packet sizes will also vary, depending on the type of traffic. Different packet sizes will mean that we need to reconsider *clock tick* which is the unit of time for serving a packet.

We have seen from our experimental results from NS-2 simulator that the distribution of overflows in a buffer at a bottleneck node does not follow an Exponential distribution, and thus cannot be accurately modelled by a two-state MC. It will be interesting to see what model can fit this distribution. We have pondered over a four-state MC or an n-state MC but these models also seem simple for fitting the distribution shown in Chapter 6.

Another important task for consideration in future is to sample or probe a test-bed network, with a topology similar the one we ave considered where a bottleneck node is fed traffic by several sources. It will be useful to see how the precision of the estimate of loss probability in such a test-bed network varies for various sampling gaps.

Appendix A

Use of Fisher Information for Optimal Sampling Gap

A.1 Use of Fisher Information Matrix, by Parker et al, for Optimizing PLP Estimation by Sampling

As stated earlier, Parker et al [3] use the Fisher Information metric for giving the upper bound for sampling rate, for various network states. They further use it to find exact sampling points that are not equi-spaced (that is, not at a fixed rate). They represent the network states by modelling the states of a buffer at a network node, as a simplified two-state Markov chain. When the traffic load on a buffer in a network node increases and decreases with time, this results in the buffer filling up and having no more capacity and thus overflowing, or going up in capacity and not overflowing any more. They treat p, and r as the unknown parameters to be estimated. Given p and r PLP follows as:

$$PLP = \frac{p}{p+r} \tag{A.1}$$

An observation (sample) y of the Markov chain is a vector of length T+1. In order to find the Fisher Information matrix for the observation y obtained when using a sampling period of k = 1 (or in other words, taking T samples with the smallest possible gap (this may be set by some aspect of the technology). The likelihood function $L(\theta y)$ is derived for each of the transition probabilities. Eg, the likelihood function for k =0 is:

$$L(\theta|y) = P(Y_0 = y_0, Y_1 = y_1, Y_2 = y_2, ...Y_T = y_T) \tag{A.2}$$

$$L(\theta|y) = P(Y_0 = y_0)P(Y_1 = y_1|Y_0 = y_0)P(Y_2 = y_2|Y1 = y_1, Y_0 = y_0)...P(Y_T = y_T|Y_{(T-1)} = y_{(T-1)},, Y1 = y_1, Y_0 = y_0) \tag{A.3}$$

By Markov principle, i.e. the probability of the current state only depends on the previous state, equationA.3 simplifies to

$$L(\theta|y) = P(Y_0 = y_0)P(Y_1 = y_1|Y_0 = y_0)P(Y_2 = y_2|Y1 = y_1)...P(Y_T = y_T|Y_{(T-1)} = y_{(T-1)}) \tag{A.4}$$

$$L(\theta|y) = P(Y_0 = y_0)\prod_{j=0}^{T-1}(Y_{(j+1)} = y_{(j+1)}|Y_j = y_j) \tag{A.5}$$

$$L(\theta|y) = \frac{(y_0 p + (1-y_0)r)}{(p+r)}\prod_{j=0}^{T-1}(1-y_j)[y_{(j+1)}p + (1-y_{(j+1)})(1-p)] + y_j[y_{(j+1)}(1-r) + (1-y_{(j+1)})r] \tag{A.6}$$

Each of these likelihood functions is then differentiated twice followed by taking Expectations and mathematical simplifications. Fisher Information matrix is then formed by plugging in the corresponding simplified expressions obtained for each transition probability into the corresponding positions in the transition matrix (equation 1). Equation A.7 is FI matrix for realizations of the Markov chain when sampling gap k is 1.

$$I(p,r) = \frac{1}{p+r}\begin{pmatrix} \frac{1}{p} - \frac{1}{p+r} + \frac{Tr}{p} + \frac{Tr}{1-p} & \frac{-1}{p+r} \\ \frac{-1}{p+r} & \frac{1}{r} - \frac{1}{p+r} + \frac{Tp}{r} + \frac{Tp}{1-r} \end{pmatrix} \tag{A.7}$$

Similarly matrices are constructed for k=2, k =3 and so on. Parker et al [3] then use the D-criterion of optimality for choosing which k is better. D-Criterion requires finding determinant of the matrices for different k values and the maximum determinant identifies the best k for a given transition probability pair (p, r).

A.2 Simulations to Determine 'Experimental' Optimal Sampling Rate

This section covers the initial simulations carried out to determine the experimental 'best *k*' to compare with the results of Parker et. al. [3]. Here *k* determines the sampling rate as explained earlier.

A.2.1 Simulation Description

Our simulation involves having the on-off buffer model described in Chapter 3 and letting the simulation run for about a million (or more) *clockticks*. Sampling is performed on this generated data in the following fashion. Following the procedure suggested by Parker [3], sampling has been done using various k values. That is, if we are taking T number of samples (referred to as a sample sub-set) in one go, $k = 1$ would mean sampling every clock tick for these T number of samples, k=2 means sampling every other clock tick and so on. After taking T samples, a gap or offset is left before repeating T sampling again. The offset is chosen to be sufficiently large so that there is no auto-correlation between sample sub-sets. Thus a complete sample set comprises several sub-samples of length T. The total length of the complete sample-set, no matter what the k is given by: Total Sample length = T * Total Simulation Ticks or outcomes / offset It must be noted that the offset should be greater than highest k*T. This is to ensure that we are sampling only once for each k and there is no overlap, within the same range of outcomes. That is to say, if offset were less than T*highest k, say T*highest k/2, in that case samples would be overlapping. Having taken samples for various k values, the next step is to choose a metric for comparison of the actual (theoretical) and sampled results. Finally best k is determined by calculating the standard deviation of the results for the metric for each k, from the theoretical (actual) value of the same metric. The metric used in order to determine optimal sampling is: the standard deviation of probability of the buffer staying in non-overflow state (P(0)or P((OF)) A typical Simulation Experiment comprises twenty sub-experiments using the same p and r value. The reason for choosing to do 20 experiments is simply to have multiple/ repeated experiments in order to choose optimal k results. Sampling is done at various k rates.

A.2.2 Calculating the P(OF) metric over a sample

P(OF) is the probability of being in a non-overflow state, is then computed over the whole sample set.For the sake of validation, this computation of the overall P(OF) was tried in two ways.

In the first procedure, which shall be referred to as the mean of means, the mean of zeros, P(OF), is calculated over each of the sub-samples of length T, and then a mean of all these sub-sample-means is taken for computing the final P(OF) value. The second procedure simply takes the mean of zeros over the whole of the sample rather than taking mean of zeros in each T sized sub-sample. Both ways gave similar results for the experiments which helped validate the sampling procedure.

A.2.3 Comparison of Experimental Results with those of Parker et al [3]

In order to compare with the work of Parker et al, P(OF) is calculated for each of the sampling rates being compared. Having computed the experimental values for the metric, in each of the various experiments (twenty experiments in the results being shown in this report), the Standard Deviation is taken for these computed metrics with respect to the theoretical values. Based on the Standard Deviation results, the k value with the least standard deviation is declared as the best. The results of the experiments were then compared with the theoretical results in the work of B.M Parker [3]. The results and comparison follow in the next section.

A.2.3.1 Experimental Results and Discussion

Each experiment consists of 20 sub-experiments, in each of which, sampling using k=1 to k=6 is done. Duration of Each sub-experiment 2 was 1 million clock-ticks for Experiment Set 1, and 10,000000 (ten million) cycles/ clock ticks. The sample size, T, is 5. The offset between each of the T samples (within each sub-experiment) is 1000 clock ticks. A clock tick, here, is defined as a unit of time after which the system may change state from overflow to non-overflow. In each experiment, the observed P(OF')for each experiment for different ks is computed as the mean of P(OF')s computed in each of the T samples. The Standard Deviation of each P((OF)) calculated for each of the k values is then taken across the 20 sub-experiments. The actual P(OF') is taken as mean, when calculating the Standard Deviations. For ease of referencing, Table 2.1 from [3] is being reproduced here (labelled Table A.1) in order to compare results.

A.2.3.1.1 Result Tables from Experiment Set 1 For comparison with Table 2.1, the Table A.2 shows results for p and r between 0.05 and 0.55.

Looking at the table A.2, it appears that there is not any sort of a pattern in the optimal k table. An explanation could be that there is very little to choose from, if we look at the standard deviation values for each k. A typical standard deviation comparison for the ten k values is given in Figure A.1 Figure A.2 gives pictorial representation of how the various calculated P(OF') values look like for each k, in each of the 20 sub-experiments.

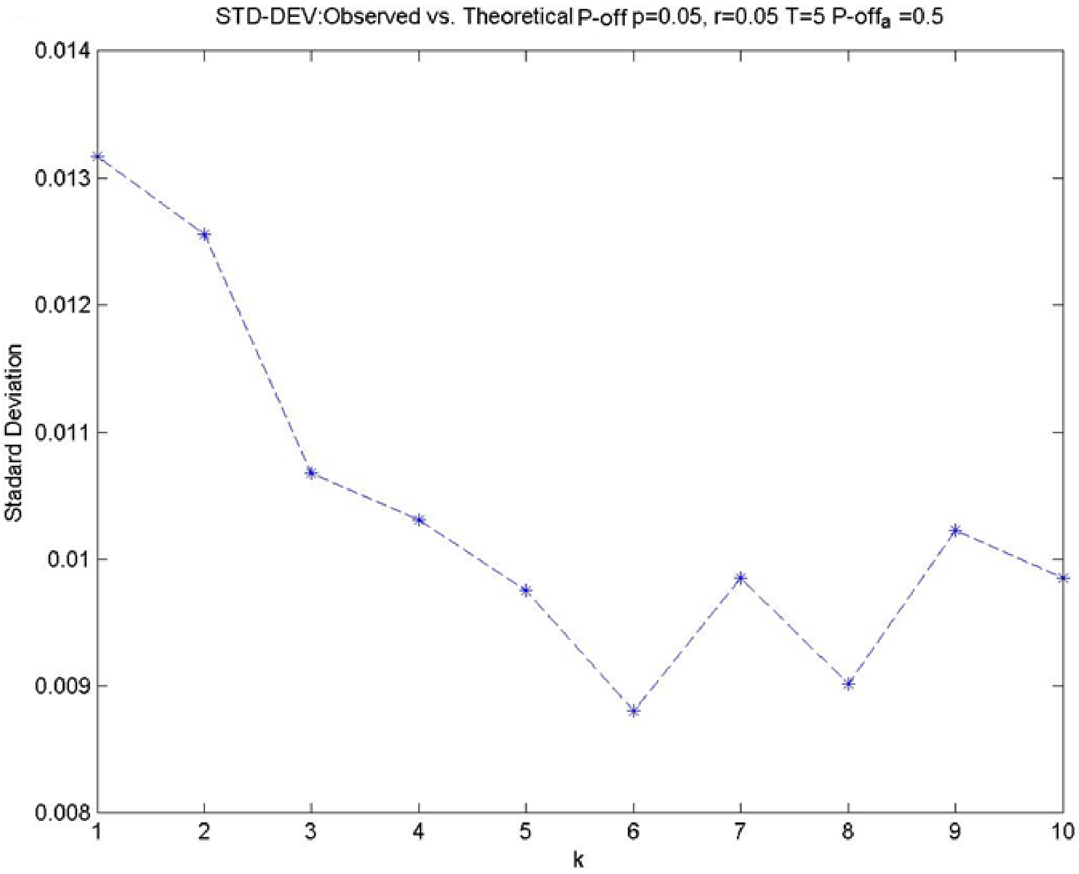


Figure A.1: Graph showing Standard Deviation for Various values of k, p = 0.05, r = 0.05

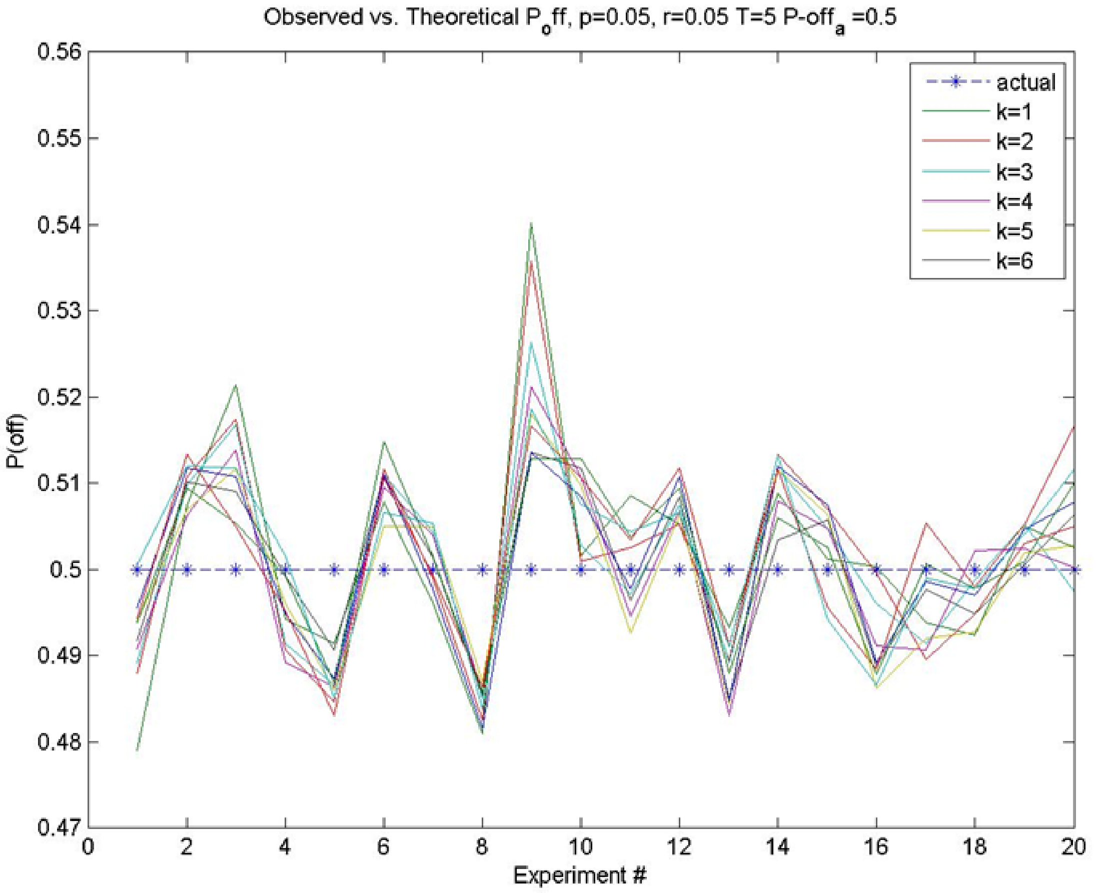


Figure A.2: Graph showing overall P(OF’) for various values of k, p = 0.05, r = 0.05

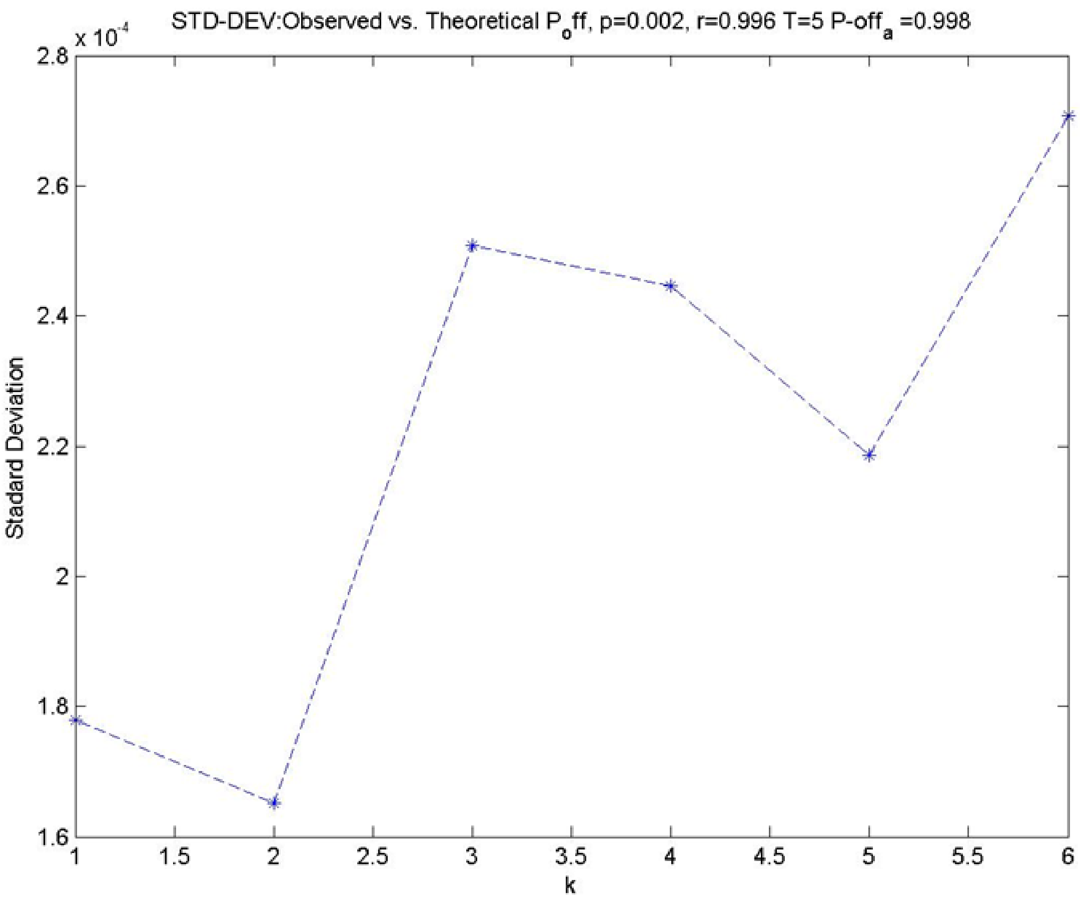


Figure A.3: Graph showing Standard Deviation for Various values of k, p = 0.002, r = 0.996

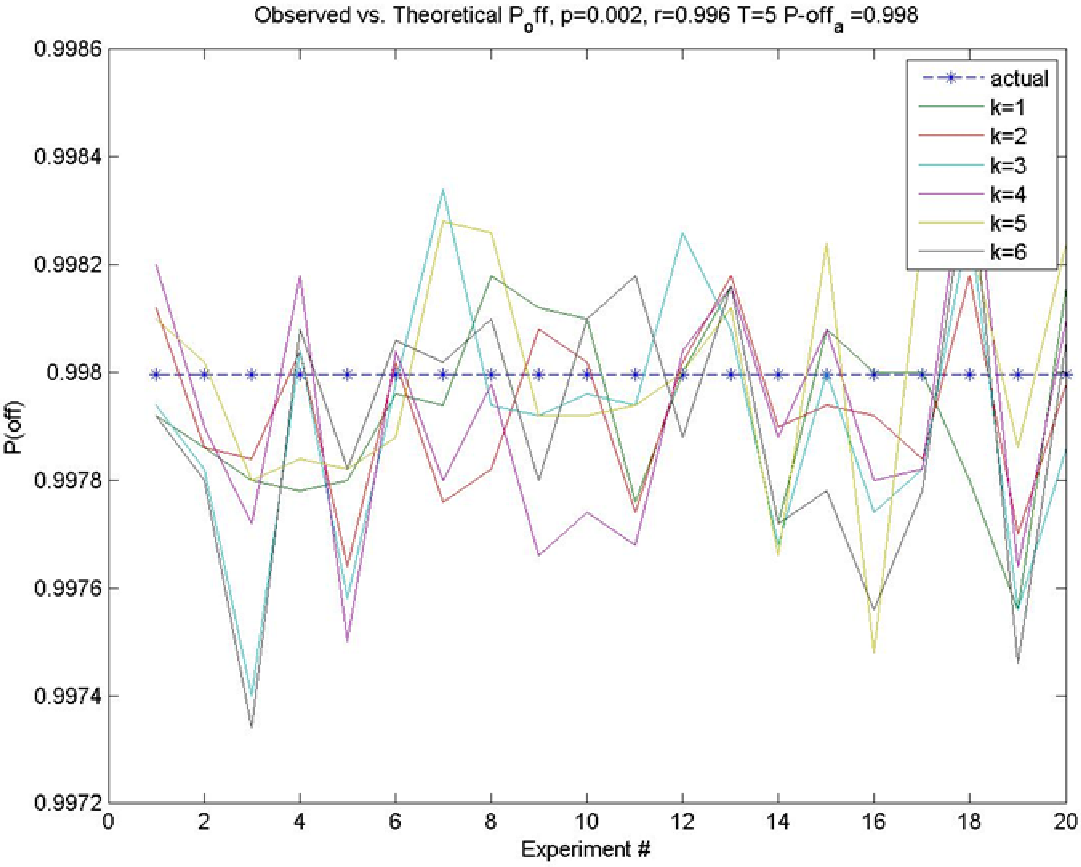


Figure A.4: Graph showing overall P(OF’) for various values of k, p = 0.002, r = 0.996

Table A.1: Table 2.1 from [3] Theoretical best values for k, T=5 by Parker et al [3]

0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
0.05	10	7	6	4	4	3	3	2	2	2	2
0.1	7	5	4	3	3	2	2	2	2	1	1
0.15	6	4	3	3	2	2	2	2	1	1	1
0.2	4	3	3	2	2	2	2	1	1	1	1
0.25	4	3	2	2	2	2	1	1	1	1	1
0.3	3	2	2	2	2	1	1	1	1	1	1
0.35	3	2	2	2	1	1	1	1	1	1	1
0.4	2	2	2	1	1	1	1	1	1	1	1
0.45	2	2	1	1	1	1	1	1	1	1	1
0.5	2	1	1	1	1	1	1	1	1	1	1
0.55	1	1	1	1	1	1	1	1	1	1	1

It can be seen from the figure A.1 that the difference in standard deviation values for various k is very small. This brings forth an argument that, whether the difference in F.I. values that originally resulted in

Table A.2: Table showing best value of k for T=5, based on experimental results, taking k=1 to k=10

0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
0.05	6	7	9	7	7	4	9	8	9	4	10
0.1	10	10	8	6	7	6	7	10	7	9	7
0.15	9	10	5	10	6	10	9	10	6	7	10
0.2	8	10	9	9	8	7	5	3	9	10	2
0.25	6	6	8	10	4	9	4	9	8	7	10
0.3	10	5	4	10	10	5	9	9	9	8	3
0.35	10	9	9	6	8	5	6	7	5	1	8
0.4	7	9	6	2	5	6	4	9	5	9	3
0.45	10	8	4	7	5	6	2	4	8	8	3
0.5	10	6	8	3	9	5	6	10	3	6	2
0.55	7	6	7	3	7	8	9	8	2	3	6

choosing certain k values over the others (in the theoretical results as represented in Table 2.1) actually matters. And how much difference in the F.I. actually is significant or sufficiently large enough to be able to take a decision. This is a future direction that will be explored during the course of this research.

A.2.3.2 Table from Experiment Set 2, Smaller p and r

Having conducted experiments for p and r values same as considered by Parker, more experiments were conducted with fairly small values of p and large values of r, in order to picture a more realistic scenario. That is, in the real world scenario, probability of the rare state is quite low. Table A.3 gives the resulting best k values.

Table A.3: Table showing best value of k for T=5, based on experimental results, taking k=1 up to 6

0	0.9998	0.9996	0.9994	0.9992	0.999	0.998	0.996	0.994	0.992	0.99
0.0002	2	6	3	6	1	1	4	3	6	6
0.0004	5	2	2	2	1	2	1	4	6	3
0.0006	3	5	5	3	6	4	4	3	3	5
0.0008	1	3	6	6	3	5	2	4	3	5
0.001	1	5	4	1	4	5	5	6	1	5
0.002	3	3	2	1	1	4	2	5	5	4
0.004	5	5	6	2	3	2	5	4	5	5
0.006	6	1	3	2	5	5	4	5	3	3
0.008	1	1	5	1	2	4	1	2	3	2
0.01	5	2	6	3	5	2	1	4	2	4

Figure A.3 shows the standard deviation of the calculated P(OF') for each of the k values. It can be seen that the standard deviation for the computed P(OF')s is further reduced for lower values of p and r. As stated earlier, this means that choosing any k would give us similar results and that theres not much evidence that one k is better than another.

A.3 Numerical Fisher Information

We later tried computing the Fisher Information numerically and generated various results. However, similar to the results above, the results with numerical computation of Fisher Information were inconclusive. One such set of results is given in figure A.5.

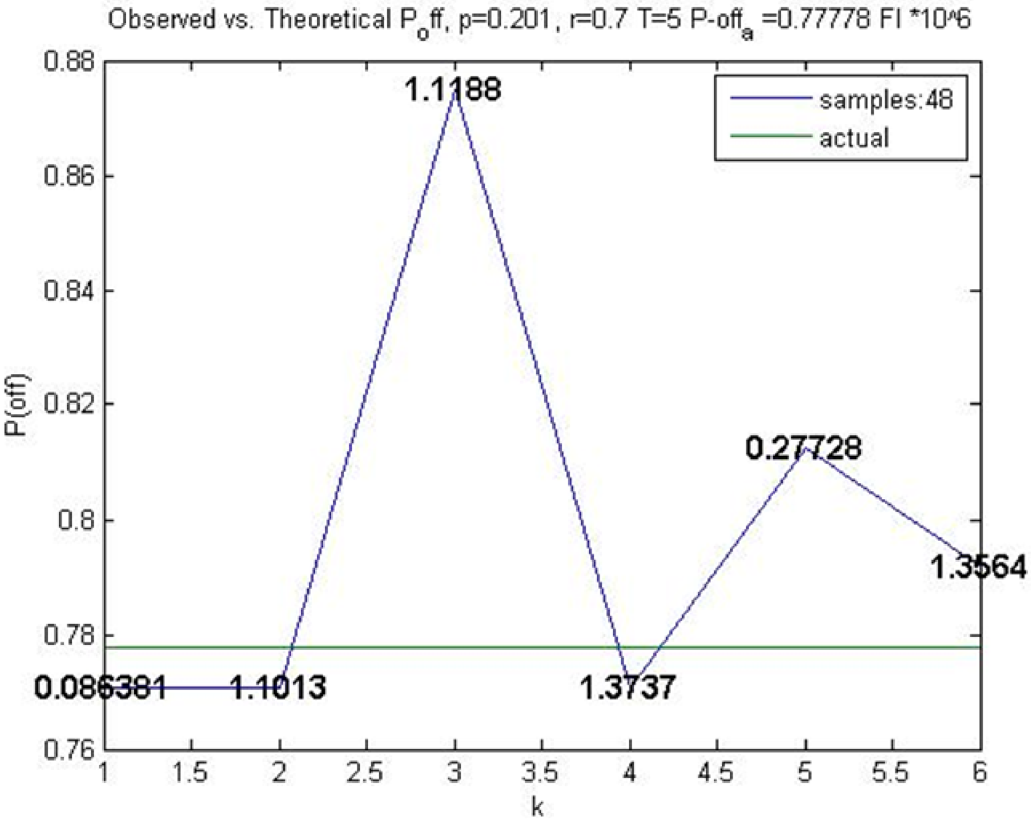


Figure A.5: Graph showing overall P(OF') for various values of k, p = 0.002, r = 0.996

Appendix B

Appendix for Chapter 4

Figures B.1 and B.2 show how similar the error in estimate is when T=20 or T=100 for Matlab based experiments since the positioning of bursts, although rare in case of $P(OF) = 10^{-3}$ are similar.

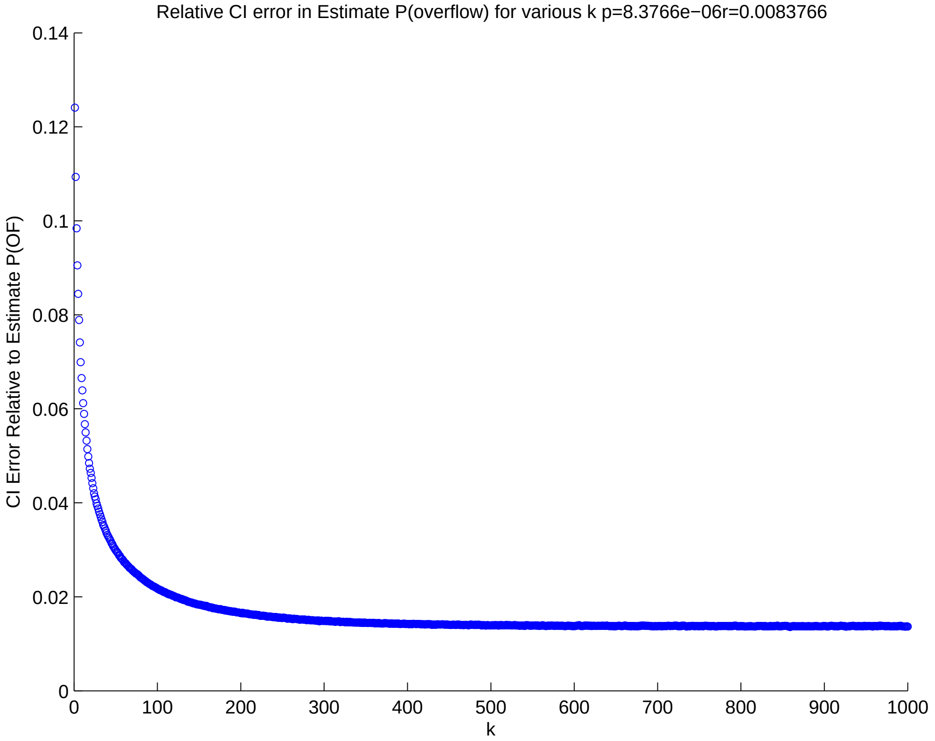


Figure B.1: Relative CI Error for the estimated P(OF) for various k values, obtained for sample $T = 100$

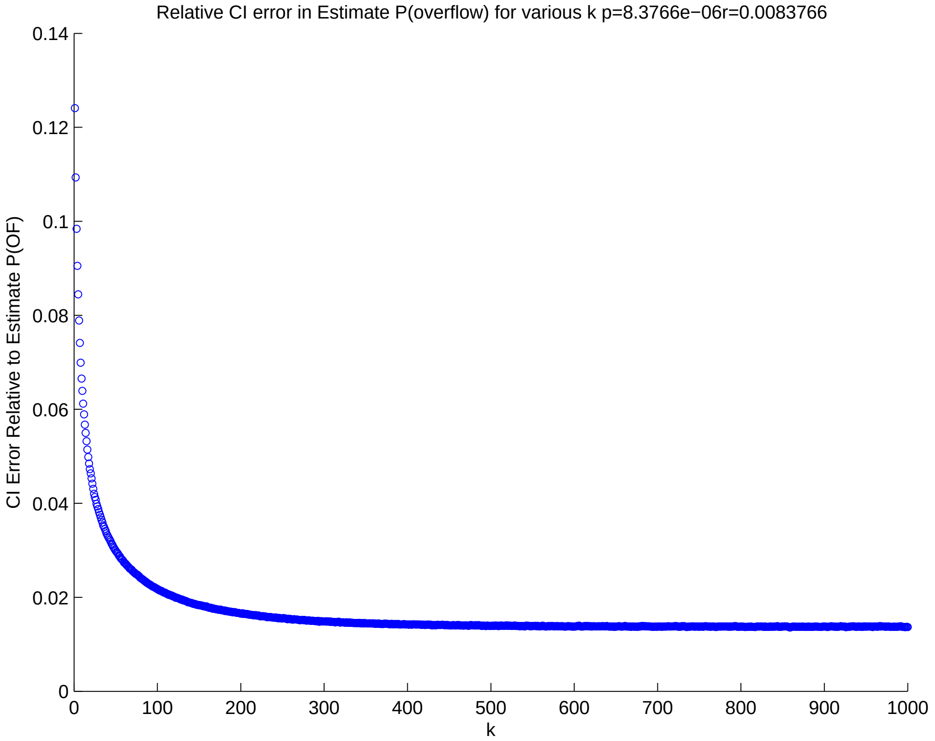


Figure B.2: Relative CI Error for the estimated P(OF) for various k values, obtained for sample $T = 20$

Appendix C

Appendix for Chapter 6

Steps of the Experimental Setup

1. Simulate the network Scenario

2. After Every 4ms check the queue length

3. IF queue length ¿ threshold output 1

4. Else output 0

5. Save data as a row in text file

6. Repeat the experiment noExp times

7. Take the output of 0s and 1s as MC into Matlab

8. Sample the MC data for k=1 to k=noK

9. Generate Output Figure

Since the text file thus generated is too large to be able to read into Matlab without going out of memory. Each row generated as a result of step 5 becomes larger than the vector size acceptable in the default Matlab settings. In order to overcome this, we used Run Length Encoding (RLE) to encode each row. A flow chart for the RLE process is given in Figure 7.x.

RLE Encode Block

1. Repeat for each repetition of the experiment and save data in a row in the file

a. Repeat until end of this simulation experiment.

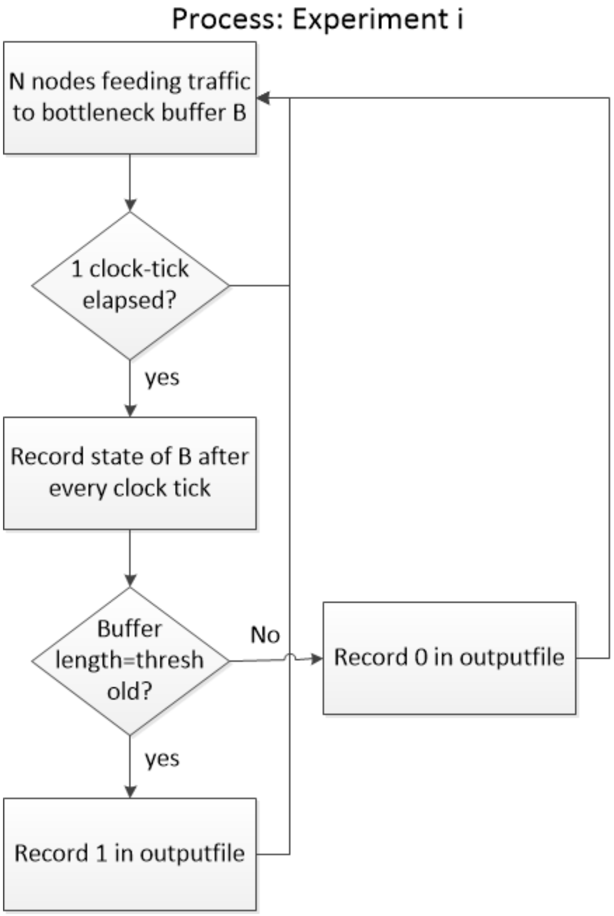


Figure C.1

i. Zero Encountered? Count the number of consecutive 0s and record this after appending a 0 at the end.

ii. Record as the next element in file

iii. 1 encountered? Count the number of consecutive 1s and record this after appending a 1 at the end

iv. Record as the next element in file

RLE Decoder

The Matlab code to decode and then sample the data works as explained in Figure 7.x

1. Repeat for each repetition of the experiment

a. Repeat until end of this simulation experiment (row).

i. Extract an element from the row/ experiment result, and generate an empty vector

1. If this element is an odd-number

a. Remove the last 1 from the number

b. Expand this number as a series of 1s of size same as this number. Append this series to the end of the vector

2. If this element is an even number

a. Remove the last 0 from the number

b. Expand this number as a series of 0s of size same as this number. Append this to the end of the vector

Sample this resultant vector for k=1 up to k=noK.

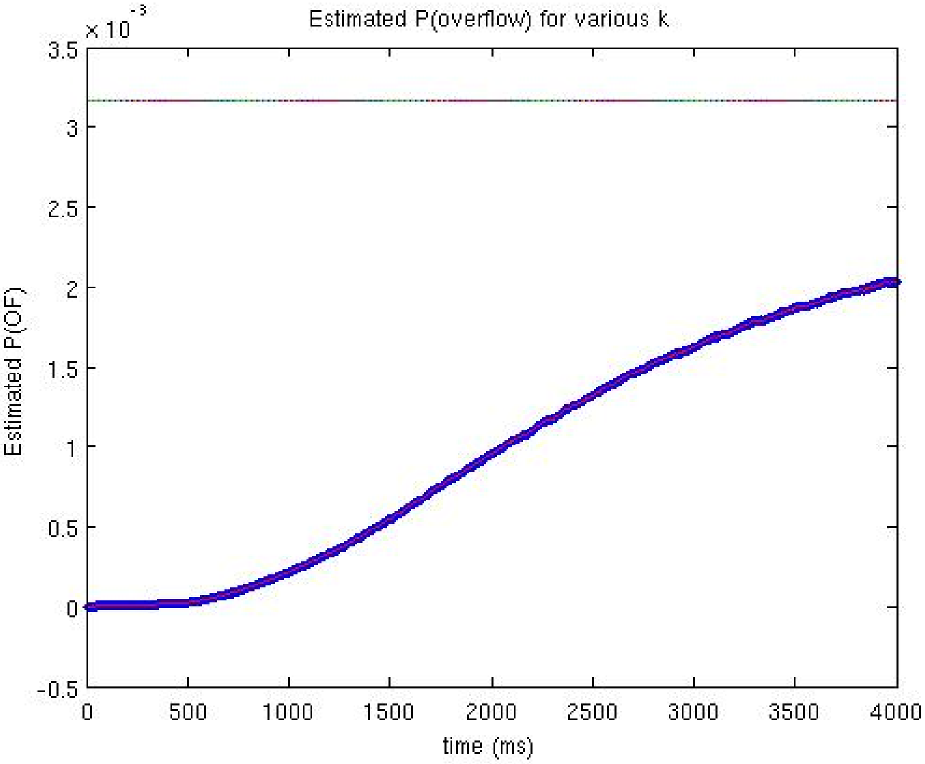


Figure C.2: Estimated vs Actual $P(OF)$ for various k values, obtained for sample size $T = 20$ and load 70%.

(figures C.3 to C.8), We have included figures C.9 to C.11 that show accuracy versus sampling gap when sampling a MC generated for VoIP parameters (p and r obtained from NS-2 experiments for corresponding load values). We can see that when the overflow estimation is for buffer states modelled as a MC, then accuracy is not affected by sampling gap.

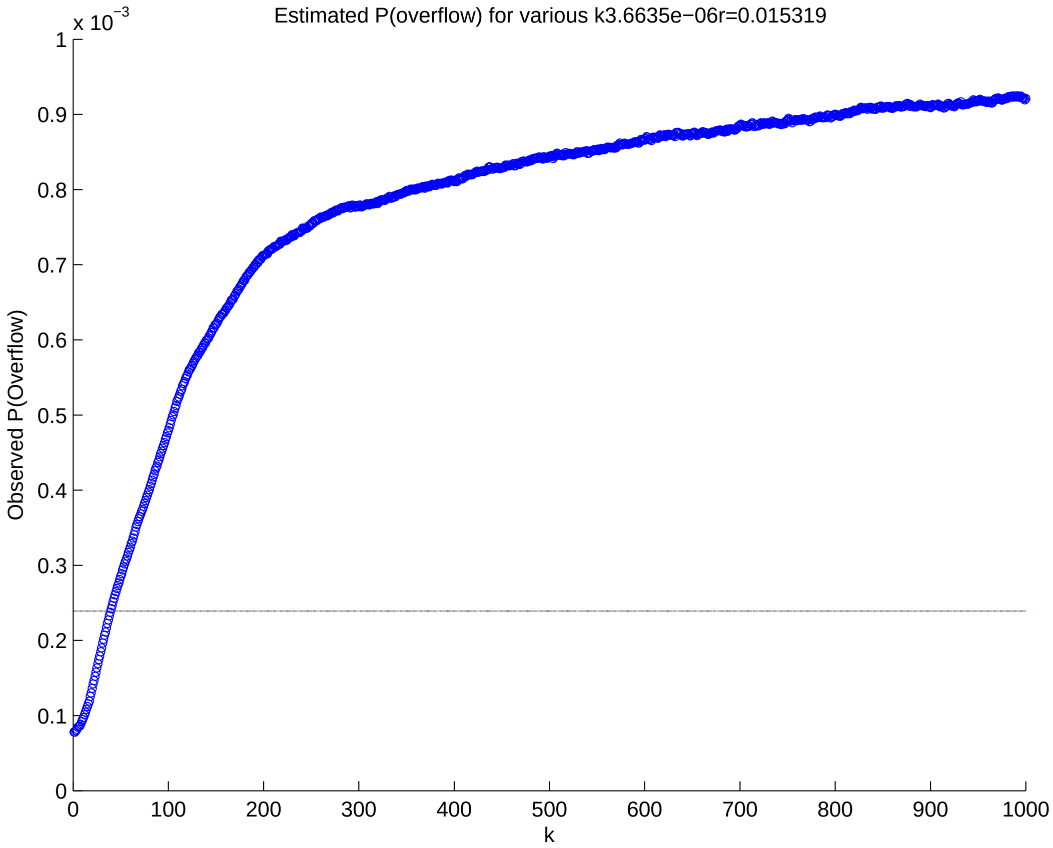


Figure C.3: Accuracy of estimated $P(OF)$ for load of 70% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with VoIP sources

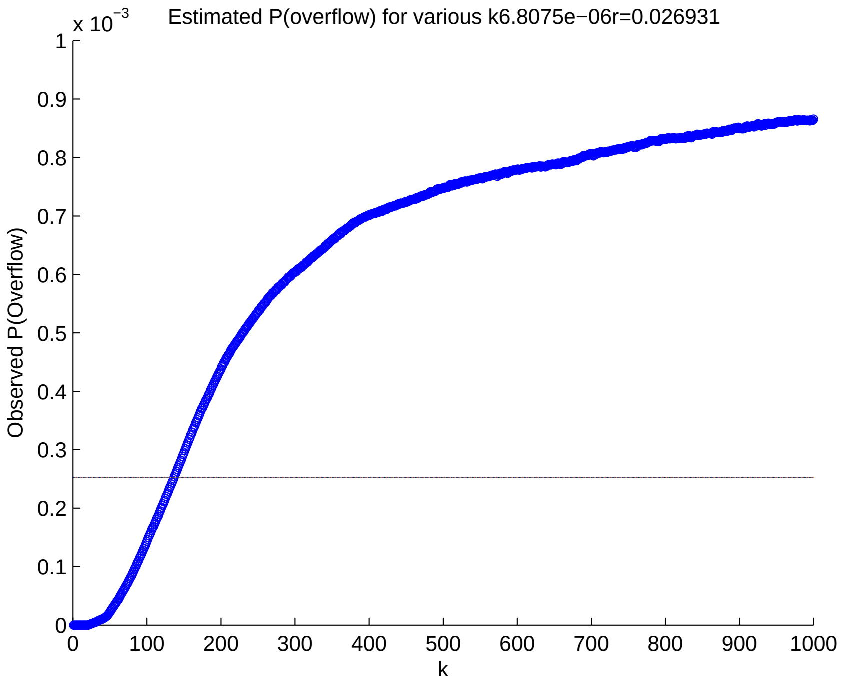


Figure C.4: Accuracy of estimated $P(OF)$ for load of 80% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with VoIP sources

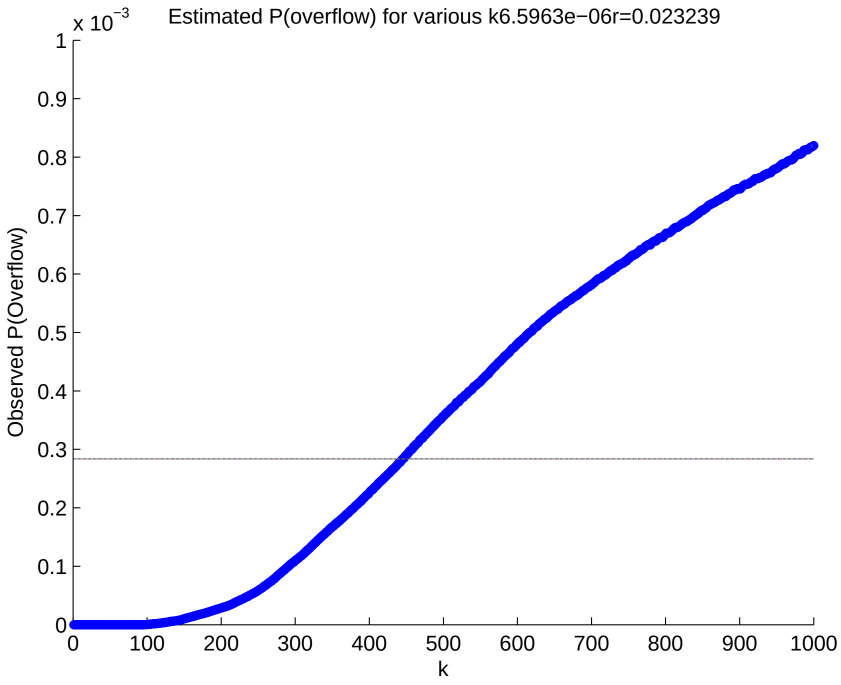


Figure C.5: Accuracy of estimated $P(OF)$ for load of 90% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with VoIP sources

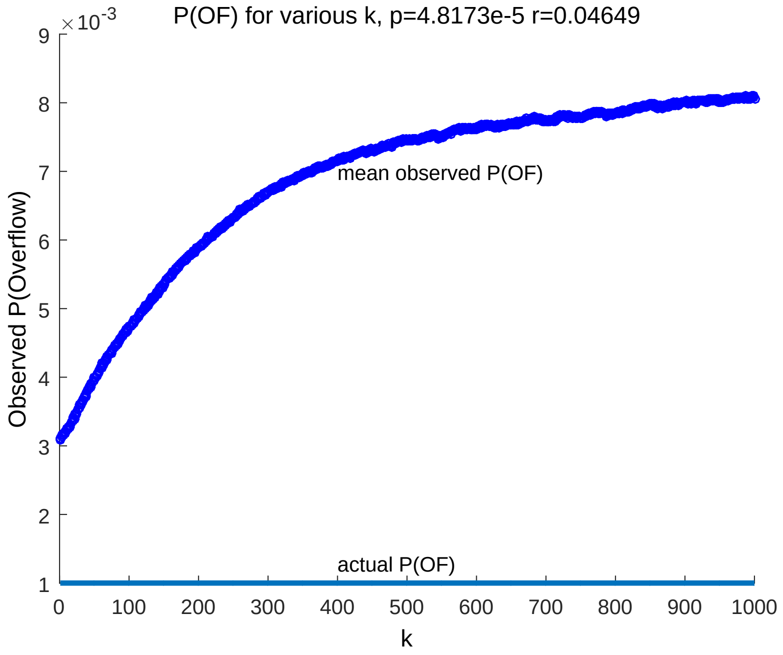


Figure C.6: Accuracy of estimated $P(OF)$ for load of 70% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with bursty data sources

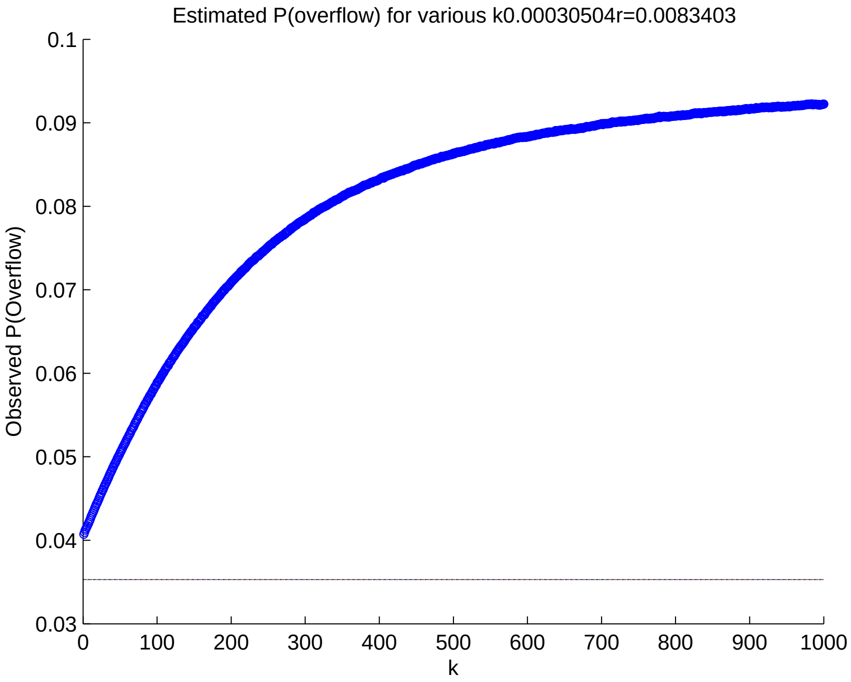


Figure C.7: Accuracy of estimated $P(OF)$ for load of 80% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with bursty data sources

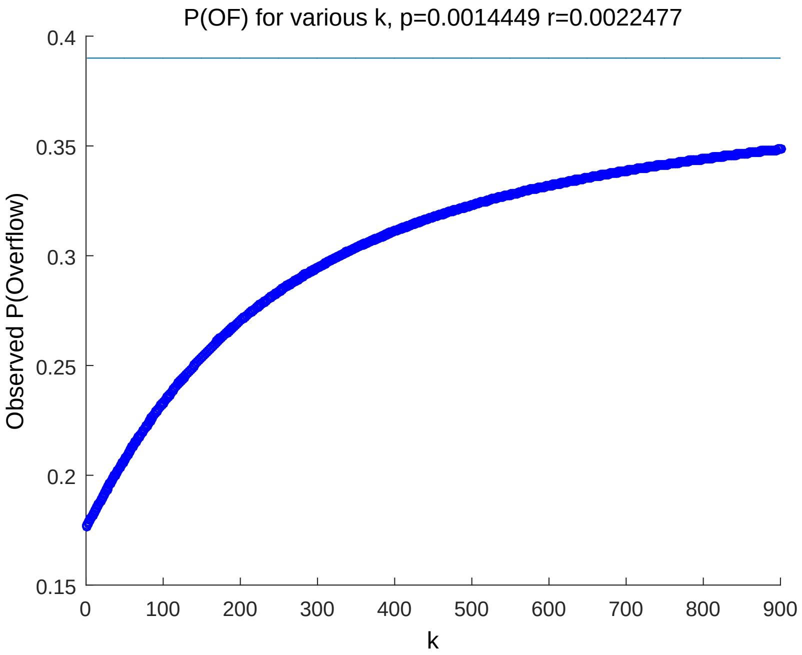


Figure C.8: Accuracy of estimated $P(OF)$ for load of 90% when using sample size $T = 100$ for sampling buffer states as seen in an NS-2 experiment with bursty data sources

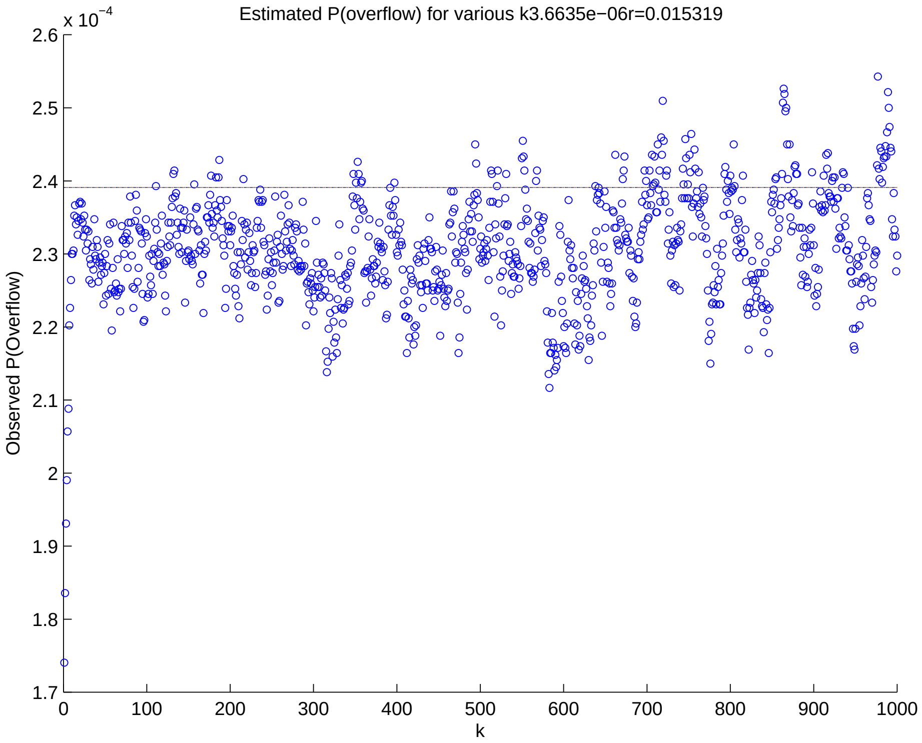


Figure C.9: Accuracy of estimated $P(OF)$ for load of 70% when using sample size $T = 100$ for sampling buffer states as seen in a Matlab experiment with parameters for VoIP data sources

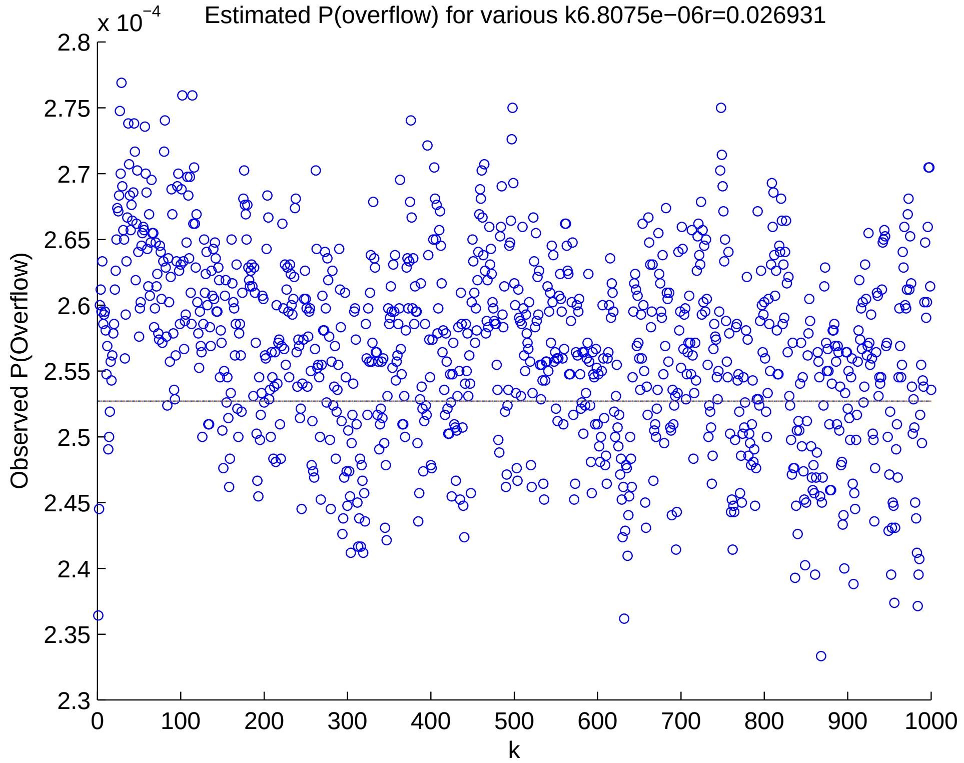


Figure C.10: Accuracy of estimated $P(OF)$ for load of 80% when using sample size $T = 100$ for sampling buffer states as seen in a Matlab experiment with parameters for VoIP data sources

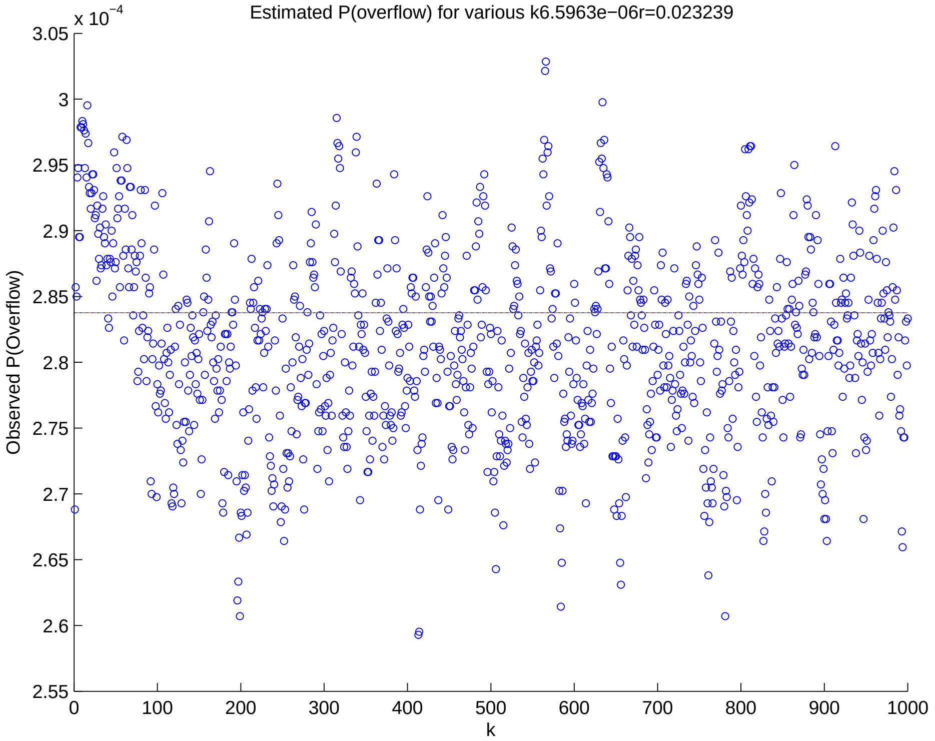


Figure C.11: Accuracy of estimated $P(OF)$ for load of 90% when using sample size $T = 100$ for sampling buffer states as seen in a Matlab experiment with parameters for VoIP data sources

References

[1] M. Roughan, "Fundamental bounds on the accuracy of network performance measurements," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, pp. 253–264, June 2005.

[2] V. Barbu, J. Bulla, and A. Maruotti, "Estimation of the stationary distribution of a semi-markov chain," *Journal of Reliability and Statistical Studies*, vol. 5, pp. 15–26, 2012.

[3] B. M. Parker, S. G. Gilmour, and J. Schormans, "Measurement of packet loss probability by optimal design of packet probing experiments," *Communications, IET*, vol. 3, no. 6, pp. 979–991, 2009.

[4] I. Mas, V. Fodor, and G. Karlsson, "Probe-based admission control for multicast," in *Quality of Service, 2002. Tenth IEEE International Workshop on*, pp. 99–105, 2002.

[5] J. L. Adams, L. G. Roberts, and A. IJsselmuiden, "Changing the internet to support real-time content supply from a large fraction of broadband residential users," *BT Technology Journal*, vol. 23, no. 2, pp. 217–231, 2005.

[6] S. Jamin, S. Shenker, and P. B. Danzig, "Comparison of measurement-based call admission control algorithms for controlled-load service," in *Proceedings IEEE INFOCOM '97, The Conference on Computer Communications, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution, Kobe, Japan, April 7-12, 1997*, pp. 973–980, 1997.

[7] J. Spall, "Resampling-based calculation of the information matrix for general identification problems," in *American Control Conference, 1998. Proceedings of the 1998*, vol. 5, pp. 3194–3198 vol.5, Jun 1998.

[8] P. Tune and D. Veitch, "Towards optimal sampling for flow size estimation," in *Proceedings of*

the 8th ACM SIGCOMM Conference on Internet Measurement, IMC '08, (New York, NY, USA), pp. 243–256, ACM, 2008.

[9] C. J. Geyer, *Stat 5102 Notes: Fisher Information and Confidence Intervals Using Maximum Likelihood.* No. Stat5102, School of Statistics, University of Minnesota, Twin Cities Campus: University of Minnesota, 2003.

[10] B. Bolker, *Stochastic Simulation, Ecological Models and Data in R.* Princeton University Press, 2008.

[11] J. A. Schormans, *Models for Packet Traffic, Lecture Notes on Network Modelling and Performance*. School of Electronics Engineering and Computer Science, Queen Mary University of London, London, 2012.

[12] J. Sommers, P. Barford, N. Duffield, and A. Ron, "A geometric approach to improving active packet loss measurement," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 307–320, Apr. 2008.

[13] F. Baccelli, S. Machiraju, D. Veitch, and J. C. Bolot, "On optimal probing for delay and loss measurement," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, (New York, NY, USA), pp. 291–302, ACM, 2007.

[14] R. W. Wolff, "Poisson Arrivals See Time Averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[15] B. Ribeiro, D. Towsley, T. Ye, and J. C. Bolot, "Fisher information of sampled packets: An application to flow size estimation," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, (New York, NY, USA), pp. 15–26, ACM, 2006.

[16] B. M. Parker, S. G. Gilmour, and J. A. Schormans, "Design of experiments for categorical repeated measurements in packet communication networks," *Technometrics*, vol. 53, pp. 339–352, November 2011.

[17] S. A. Frank, "Natural selection maximizes Fisher information," *ArXiv e-prints*, Jan. 2009.

[18] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, (New York, NY, USA), pp. 39–44, ACM, 2003.

[19] A. A. Wahid and J. Schormans, “Bounding the maximum sampling rate when measuring plp in a packet buffer,” in *Computer Science and Electronic Engineering Conference (CEEC), 2013 5th*, pp. 115–118, Sept 2013.

[20] R. Mondragon, A. Moore, J. Pitts, and J. Schormans, “Analysis, simulation and measurement in large-scale packet networks,” *IET Communications*, vol. 3, pp. 887–905, June 2009.

[21] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot, “Probing for loss: The case against probe trains,” *IEEE Communications Letters*, vol. 15, pp. 590–592, May 2011.

[22] V. Markovski, F. Xue, and L. Trajković, “Simulation and analysis of packet loss in user datagram protocol transfers,” *J. Supercomput.*, vol. 20, pp. 175–196, Sept. 2001.

[23] M. Yajnik, J. Kurose, and D. Towsley, “Packet loss correlation in the mbone multicast network,” in *Global Telecommunications Conference, 1996. GLOBECOM ’96. ’Communications: The Key to Global Prosperity*, pp. 94–99, Nov 1996.

[24] M. Hasib, J. Schormans, and J. Pitts, “Probing limitations for packet loss probability measurement on buffered access links,” *Electronics Letters*, vol. 40, pp. 1315–1316, Sept 2004.

[25] P. Barford and J. Sommers, “A comparison of probe-based and router-based methods for measuring packet loss,” *submission,(see http://www. cs. wisc. edu/pb/publications. html)*, 2003.

[26] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot, “The role of pasta in network measurement,” *Networking, IEEE/ACM Transactions on*, vol. 17, pp. 1340–1353, Aug 2009.

[27] N. Duffield, “Sampling for passive internet measurement: A review,” *Statistical Science*, vol. 19, no. 3, pp. 472–498, 2004.

[28] A. W. Moore, “An implementation-based comparison of Measurement-Based Admission Control algorithms,” *Journal of High Speed Networks*, vol. 13, no. 2, pp. 87–102, 2004.

[29] M. Grossglauser and D. N. C. Tse, “A time-scale decomposition approach to measurement-based admission control,” in *INFOCOM ’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1539–1547 vol.3, Mar 1999.

[30] S. S. Alwakeel and N. M. Alotaibi, "End-to-end measurement based admission control voip protocol with loss policy," *Journal of King Saud University - Computer and Information Sciences*, vol. 23, no. 1, pp. 37 – 43, 2011.

[31] O. T. Brewer and A. Ayyagari, "Comparison and analysis of measurement and parameter based admission control methods for quality of service (qos) provisioning," in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pp. 184–188, Oct 2010.

[32] S. Albin, "Approximating a point process by a renewal process, ii: Superposition arrival processes to queues," *Oper. Res.*, vol. 32, pp. 1133–1162, Oct. 1984.

[33] W. Whitt, "Planning queueing simulations," *Management Science*, vol. 35, no. 11, pp. 1341–1366, 1989.

[34] G. Sakellari and E. Gelenbe, "A multiple criteria, measurement-based admission control mechanism for self-aware networks," in *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pp. 1060–1064, Aug 2008.

[35] I. S. Ali and P. S. A. Khader, "A study on probe based admission control for multicast and further enhancement," in *Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on*, pp. 27–32, March 2011.

[36] E. N. Gilbert, "Capacity of a burst-noise channel," vol. 39, pp. 1253–1265, Sept. 1960.

[37] J.-C. Bolot, "The case for fec-based error control for packet audio in the internet," 1997.

[38] J. C. Bolot and T. Turletti, "Adaptive error control for packet video in the internet," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1, pp. 25–28 vol.1, Sep 1996.

[39] S. R. Kang and D. Loguinov, "Modeling best-effort and fec streaming of scalable video in lossy network channels," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 187–200, Feb 2007.

[40] P. U. Tournoux, E. Lochin, J. Lacan, A. Bouabdallah, and V. Roca, "On-the-fly erasure coding for real-time video applications," *IEEE Transactions on Multimedia*, vol. 13, pp. 797–812, Aug 2011.

[41] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the mbone multicast networ experimental measurements and markov chain models," tech. rep., Amherst, MA, USA, 1995.

[42] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 345–352 vol.1, Mar 1999.

[43] E. Elliott, *Estimates of Error Rates for Codes on Burst-noise Channels*. Bell telephone system technical publications, Bell Telephone Laboratories, 1963.

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

[45] V. Amaradasa, J. Schormans, J. Pitts, and C. M. Leung, "Evaluating overflow probability for voice over internet protocol buffer dimensioning," *IET Communications*, vol. 3, pp. 636–643, April 2009.

[46] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, Jun 1995.

[47] R. Fisher, *The design of experiments. 1935*. Edinburgh: Oliver and Boyd, 1935.

[48] N. Hua, A. Lall, B. Li, and J. Xu, "Towards optimal error-estimating codes through the lens of fisher information analysis," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, pp. 125–136, June 2012.

[49] B. Chen, Z. Zhou, Y. Zhao, and H. Yu, "Efficient error estimating coding: Feasibility and applications," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 3–14, Aug. 2010.

[50] N. Hua, A. Lall, B. Li, and J. Xu, "A simpler and better design of error estimating coding," in *INFOCOM, 2012 Proceedings IEEE*, pp. 235–243, March 2012.

[51] P. B. Stark, "Lecture notes etext : Sticigui," 21 January 2013.

[52] R. Levy, *Probabilistic models in the study of language*. San Diego, California: University of California Textbook Online, October 2012.

[53] E. Lehmann and G. Casella, *Theory of Point Estimation*. Springer Verlag, 1998.

[54] S. Zheng, *Lecture Notes Math 541: Methods of Evaluating Estimators*. Missouri State University, July 2014.

[55] A. S. D. Jang, "An application of confidence interval methods for small proportions in the health care survey of dod beneficiaries," mathematica policy research reports, Mathematica Policy Research, 2005.

[56] E. Chromy, M. Jadron, and T. Behul, "Admission control methods in ip networks," *Adv. MultiMedia*, vol. 2013, pp. 2:2–2:2, Jan. 2013.

[57] B. Meskill, A. Davy, and B. Jennings, "Server selection and admission control for ip-based video on demand using available bandwidth estimation," in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pp. 255–258, Oct 2011.

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

[59] B. B. Devi, "Aggregated equivalency for on-off models," in *Aerospace conference, 2009 IEEE*, pp. 1–7, March 2009.

[60] C. T. A. Wautier, J. Antoine, "Capacity analysis of voice over ip over geran with statistical multiplexing," in *Multiaccess, Mobility and Teletraffic for Wireless Communications, volume 6*, vol. 6, pp. 25–42, Springer US.

[61] T. Timotijevic, C. M. Leung, and J. Schormans, "Accuracy of measurement techniques supporting qos in packet-based intranet and extranet vpns," *IEE Proceedings - Communications*, vol. 151, pp. 89–94, Feb 2004.

[62] C. M. Leung, J. A. Schormans, J. M. Pitts, and A. Woolf, "Accurate decay rate prediction for burst-scale queueing in packet buffers," *Electronics Letters*, vol. 39, pp. 253–254, Jan 2003.

[63] R. Pastor-Satorras and A. Vespignani, *Evolution and Structure of the Internet*. Cambridge University Press, 2004. Cambridge Books Online.