Performance Evaluation of Long Range Dependent Queues
長相關隊列性能評價研究

Submitted to
Department of Electronic Engineering
電子工程系
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
哲學博士學位

by

Chen Jiongze
陳炯澤

September 2015
二零一五年九月
Abstract

With the increasing demand of Internet services, significant efforts have been made to evaluate grade of service for network design and dimensioning. Since the Long Range Dependent (LRD) characteristics of Internet traffic has been discovered, queueing performance analysis and capacity assignment for queues fed by LRD inputs, so-called LRD queues, has attracted significant attention. Compared to the traditional Poisson models inherited from the telephony world, the performance analysis of LRD queues is much more involved and faces many challenges. Despite the considerable efforts made in the past twenty years, exact results are only available for the case with the Hurst parameter equal to 0.5. In addition, simulating systems with LRD traffic inputs usually requires unrealistically long simulation times. Therefore, it is important to obtain simple and accurate analytical results, as well as time-efficient simulation methods, for LRD queues.

In this thesis, we evaluate the performance of the queues fed by LRD processes. In particular, we focus on two LRD input processes: the Poisson Lomax Burst Process (PLBP) and the fractional Brownian motion (fBm). PLBP is a variant of the popular M/G/∞ traffic process, the Poisson Pareto Burst Process (PPBP), which consists of bursts with Pareto distributed length that arrive according to a Poisson process. Since PPBP exhibits the LRD phenomenon and also captures the behaviour of Internet traffic, there have been significant research efforts to analyze the performance of queues fed by PPBP input. Here, we replace the Pareto burst distribution of PPBP with a Lomax distribution so that small traffic flows can be
taken into account. The resulting input process is named PLBP. We illustrate its advantage in modelling Internet traffic flow sizes, particularly, in its ability to capture a large number of small flows. We provide two approximations to the overflow probability of a single queue fed by PLBP based on analytical and fast simulation methods, and illustrate their accuracy by discrete-event simulations.

Fractional Brownian motion (fBm) is another important model because it captures the LRD characteristics of Internet traffic, accurately represents traffic generated as an aggregate of many sources, a prevalent characteristic of many Internet traffic streams, and is amenable to analysis. We introduce a new, simple, closed-form approximation for the stationary workload distribution (virtual waiting time) of a single server queue fed by an fBm input, the so-called fBm queue. Next, an efficient approach for producing a sequence of simulations with finer and finer details of the fBm process is introduced and applied to demonstrate good agreement between the new formula and the simulation results. This method is necessary in order to ensure that the discrete-time simulation accurately models the continuous-time fBm queueing process. Based on the closed-form formula, we provide the approximations for the mean, variance, third central moment and skewness of the occupancy of an fBm queue. Then we study the limitations of the fBm process as a traffic model with the help of a benchmark model – a truncated version of the fBm. We determine by numerical experiments the region where the fBm can serve as an accurate traffic model. These experiments show that when the level of multiplexing is sufficient, fBm is an accurate model for the traffic on links in the core of an internet. Using our result obtained for the workload distribution, we derive a closed-form expression for service rate provisioning when the desired blocking probability as a measure of quality of service is given. We then apply this result to a range of examples. Finally, we validate our fBm-based overflow probability and link dimensioning formulae through benchmark results based on a queue fed by real traffic traces as a benchmark, and demonstrate an advantage for the range of overflow probability over another
traffic model, named Markov modulated Poisson process. Finally, we compare the fBm model with the PPBP and the PLBP models and show the convergence between fBm and the PPBP/PLBP models for high aggregation of traffic.
Surname: CHEN
First Name: Jiongze
Degree: Doctor of Philosophy
College/Department: Department of Electronic Engineering

The Qualifying Panel of the above student is composed of:

Supervisor(s)
Prof. ZUKERMAN Moshe
Department of Electronic Engineering
City University of Hong Kong

Co-supervisor(s)
Prof. ADDIE Ronald G
Department of Mathematics and Computing
University of Southern Queensland

Qualifying Panel Member(s)
Dr. CHAN Chi Hung Sammy
Department of Electronic Engineering
City University of Hong Kong

Dr. KO King Tim
Department of Electronic Engineering
City University of Hong Kong

This thesis has been examined and approved by the following examiners:

Dr. CHAN Chi Hung Sammy
Department of Electronic Engineering
City University of Hong Kong

Dr. WONG Wing Ming Eric
Department of Electronic Engineering
City University of Hong Kong

Prof. ZUKERMAN Moshe
Department of Electronic Engineering
City University of Hong Kong

Prof. TRAN-GIA Phuoc
Institute of Computer Science
University of Wuerzburg
I would like to dedicate this thesis to my loving parents and grandparents.
Acknowledgements

Foremost, I would like to express my sincere appreciation to my supervisor, Prof. Zukerman Moshe, who was also the supervisor of my final year project during the undergraduate study. He is a great mentor to me, as well as my role model as a researcher. Without his enthusiastic support and continuous encouragement, I could not have finished my PhD study and this thesis.

I offer my heartfelt gratitude to my co-supervisor, Prof. Addie Ronald G., for his innovative inspirations and insightful comments. Besides, I would like to thank his student, Hirdyesh Bhatia, for discovering the link between the approximation and the Generalised Gamma distribution in Chapter 4.

I wish also to thank Dr. Timothy Neame for providing the source codes for simulating PPBP queues as used in chapter 3.

The work described in this thesis was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Projects CityU 124709 and CityU8/CRF/13G).

Finally, and always, I thank my parents and my girlfriend, Wang Dan, for their supports and encouragements during these years.
Table of contents

List of figures xv

List of tables xix

Nomenclature xxi

1 Introduction 1

1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

1.3 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

1.4 Publications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2 Long Range Dependent Models and Queueing Analysis 13

2.1 Self-Similarity and Long Range Dependence . . . . . . . . . . . . . 13

2.1.1 Definitions and Properties . . . . . . . . . . . . . . . . . . . . 15

2.1.2 Estimations of the Hurst parameter . . . . . . . . . . . . . . . 19

2.2 Heavy-Tailed Distributions . . . . . . . . . . . . . . . . . . . . . . 21

2.3 Pseudo-LRD Models . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2.3.1 Autoregressive Models . . . . . . . . . . . . . . . . . . . . . . 24

2.3.2 Markov Modulated Poisson Processes . . . . . . . . . . . . . . 25

2.4 Long Range Dependent Traffic Models . . . . . . . . . . . . . . . . 26

2.4.1 Heavy-Tailed On-Off Processes . . . . . . . . . . . . . . . . . 26
# Table of contents

2.4.2 M/G/\infty Processes ........................................... 28  
2.4.3 Fractional Brownian Motion .................................. 30  
2.4.4 Other LRD Models ............................................. 31  
2.5 Performance Analysis of LRD Queues ............................. 33  
  2.5.1 M/G/\infty Queue ............................................ 34  
  2.5.2 FBm Queue .................................................. 36  

3 Performance Evaluation of a PLBP Queue ........................... 39  
  3.1 The Poisson Lomax Burst Process ................................. 40  
    3.1.1 The Lomax Distribution .................................... 41  
    3.1.2 Statistics of PLBP ........................................ 44  
  3.2 Performance Evaluation of the PLBP SSQ ......................... 46  
    3.2.1 The QS Approximation ..................................... 47  
    3.2.2 A Fast Simulation Method .................................. 50  
  3.3 Numerical Results ............................................... 51  
  3.4 Summary ....................................................... 54  

4 Performance Evaluation of an FBm Queue ........................... 57  
  4.1 A New Analytical Result for an FBm Queue ....................... 59  
  4.2 A Sequence of Simulations of an FBm Queue ...................... 64  
  4.3 Validation of the Workload Distribution ......................... 66  
    4.3.1 Validation of the Simulation ............................... 66  
    4.3.2 Validation of the Analytical Formula ....................... 66  
    4.3.3 Discussion of Results ..................................... 69  
  4.4 Statistics of an FBm Queue .................................... 70  
    4.4.1 The Model and Related Work ............................... 70  
    4.4.2 The Mean, Variance, Third Central Moment and Skewness .. 72  
    4.4.3 Validation .................................................. 75  
    4.4.4 Simplifications of E[Q] and Var[Q] for Certain H Values . 76
# Table of contents

4.5 Summary .................................................. 80

5 Service rate provisioning of an FBm Queue 81
   5.1 Constraints on the Suitability of the FBm Model ............... 82
      5.1.1 The Problem of Negative Traffic .......................... 83
      5.1.2 Discussion ............................................. 84
   5.2 Service Rate Provisioning ..................................... 85
      5.2.1 Numerical Results ..................................... 88
   5.3 Validation Based on Real Traffic Data .......................... 90
      5.3.1 Dataset Description ..................................... 91
      5.3.2 Fitting FBm Parameters to the Real Measurements ........ 91
      5.3.3 Comparisons Between FBm, MMPP and Real Measurements 92
   5.4 Summary .................................................. 96

6 FBm Versus PPBP and PLBP 99
   6.1 Comparison Between FBm and PPBP ............................ 99
   6.2 Comparison Between FBm and PLBP ............................ 104

7 Conclusions .................................................. 109

References ..................................................... 111

Appendix A Simulation of a PLBP Queue 125
   A.1 Java Code for One Observation ............................... 125
   A.2 Flowcharts for the Java Code .............................. 132

Appendix B Simulations of an FBm Queue 137
   B.1 C Code for Generating FBm Sequences ......................... 137
   B.2 C Code for Obtaining Confidence Intervals Based on the Student’s T-Distribution ................................. 139
   B.3 C Code for Obtaining Overflow Probabilities ................ 141
# List of figures

1.1 Global IP Traffic, 2013-2018. ................................................. 2

1.2 A real traffic trace vs. a sequence generated from a compound Poisson model on different time scales. ......................... 4

2.1 Graphical “proof” of self-similarity in Ethernet traffic by [9]. .... 14

2.2 Heavy-tailed distribution vs. Exponential distribution. .......... 21

2.3 Two-state MMPP. .................................................. 25

2.4 Simple on-off process with transition rate $P_{on}$ and $P_{off}$. ...... 27

3.1 Formation of the Poisson Lomax Burst Process. ................. 41

3.2 The complementary cumulative distribution, $P(d > x)$, of Pareto (with $\gamma = 1.108$) and Lomax (with $\gamma = 1.054$) vs. real traces. ... 43

3.3 PLBP queue. .................................................. 47

3.4 Separation of the bursts. .......................................... 48

3.5 Conventional simulation vs. fast simulation. ...................... 50

3.6 Overflow probabilities based on QS approximation vs. simulation results for Case 1 to 6, with $\lambda = 10$ and $\delta = 1$ for all cases. ... 53

3.7 Overflow probabilities for PPBP and PLBP SSQs. ............... 54

4.1 Fractional Brownian motion queue. .................................. 59

4.2 Overflow probabilities based on exact formula vs. simulation results for $H = 0.5, \sigma^2 = 1, t = -0.5$ with $\Delta t = 1, 0.1$ and 0.01. .... 67
List of figures

4.3 Overflow probabilities based on the asymptotics of Norros [100] and
Hustler-Piterbarg [34], and our approximation vs. simulation results
for $H = 0.3, \sigma^2 = 1, t = -0.5$. ................................. 67

4.4 Overflow probabilities based on the asymptotics of Norros [100] and
Hustler-Piterbarg [34], and our approximation vs. simulation results
for $H = 0.4, \sigma^2 = 1, t = -0.5$. ................................. 68

4.5 Overflow probabilities based on the asymptotics of Norros [100] and
Hustler-Piterbarg [34], and our approximation vs. simulation results
for $H = 0.6, \sigma^2 = 1, t = -0.5$. ................................. 68

4.6 Overflow probabilities based on the asymptotics of Norros [100] and
Hustler-Piterbarg [34], and our approximation vs. simulation results
for $H = 0.7, \sigma^2 = 1, t = -0.5$. ................................. 69

4.7 Overflow probabilities based on the asymptotics of Norros [100] and
Hustler-Piterbarg [34], and our approximation vs. simulation results
for $H = 0.8, \sigma^2 = 1, t = -0.5$. ................................. 70

4.8 Approximations vs. simulation results for $\mu_q$ with $\sigma = 1$........ 76

4.9 Approximations vs. simulation results for $\text{Var}[Q]$ with $\sigma = 1$.... 77

4.10 Approximations vs. simulation results for $\text{E}[(Q - \text{E}[Q])^3]$ with $\sigma = 1$. 78

4.11 Approximations vs. simulation results for $\text{Skewness}[Q]$ with $\sigma = 1$. 79

5.1 Regions where fBm is a realistic model or unrealistic model due to
the presence of negative traffic. ..................................... 84

5.2 Capacity vs. $\epsilon$ by (5.2) for $H = 0.8, \sigma = 0.05, m = 1$ and $q = 0.1$. 89

5.3 Capacity vs. $H$ by (5.2) for $\sigma = 0.05, m = 1, q = 0.1$ and $\epsilon = 0.001$. 89

5.4 Capacity vs. $\sigma$ by (5.2) for $H = 0.8, m = 1, q = 0.1$ and $\epsilon = 0.001$. 90

5.5 Capacity vs. $q$ by (5.2) for $H = 0.8, \sigma = 0.05, m = 1$ and $\epsilon = 0.001$. 91

5.6 Fitting the variance-time curve. .................................. 93
5.7 Comparison of overflow probabilities for fBm and MMPP models, and real measurements for a link with capacity $\mu^* = 320$ [MByte/sec.]. 94

5.8 Comparison of the least link capacities for fBm and MMPP models, and real measurements for the overflow probability $\varepsilon \leq 0.001$. 96

6.1 Variance-time curve of fBm vs. PPBP for $\gamma = 1.5$ and $\delta = 1$. 102

6.2 Overflow probabilities for PPBP model with various arrival rates $\hat{\lambda}$, estimated by the QS algorithm vs. the fBm model by our approximation $\delta = 0.1$. 104

6.3 Overflow probabilities for the PPBP model with various arrival rates $\hat{\lambda}$, estimated by the QS algorithm vs. the fBm model by our approximation for $\delta = 0.2$. 105

6.4 Overflow probabilities for the PPBP model with various arrival rates $\hat{\lambda}$, estimated by the QS algorithm vs. the fBm model by our approximation $\delta = 0.5$. 105

6.5 PLBP vs. fBm for $\gamma = 1.6$ ($H = 0.7$) and $t = -0.5$. 107

A.1 One observation of the simulation. 133

A.2 Observe the queue. 134

A.3 Add a burst to the queue. 135
List of tables

2.1 Pareto Distributions. .................................................. 23
2.2 Classification of results on queues with $M/G/\infty$ traffic by [99] ............................... 36
3.1 Total times for obtaining $P(Q > x)$ for all selected $x$ values. ............................... 52
3.2 Parameters of PPBP and PLBP. ............................................. 54
5.1 Mean and confidence intervals of the overflow probabilities. ............................... 95
Nomenclature

Roman Symbols

$e$  | Euler’s number
$H$  | Hurst parameter
$m$  | mean of the input traffic in a time interval of length 1
$Q$  | steady state queue size
$r$  | the rate at which a burst generates work

Greek Symbols

$\delta$ | scale parameter of Pareto distribution
$\gamma$ | tail index of Pareto/Lomax distribution
$t$ | mean net input that is equal to $m - \mu$
$\lambda$ | burst arrival rate
$\mu$ | service rate of the queue (link capacity)
$\mu^*$ | provisioned link capacity
$\sigma^2$ | $\sigma^2(1)$, variance of the input traffic in a time interval of length 1
$\varepsilon$ | overflow probability of the queue
Nomenclature

Subscripts

$L$ long burst process

$S$ short burst process

Other Symbols

$\exp(.)$ exponential function

$G(.)$ regularized incomplete Gamma function

$\Gamma(.)$ Gamma function

$\gamma(.)$ lower incomplete Gamma function

$m(t)$ mean of the input traffic during the time interval, $t$

$\sigma^2(t)$ variance of the input traffic during the time interval, $t$

Acronyms / Abbreviations

ACF autocorrelation function

approx. approximation

CCDF complementary cumulative distribution function

CDF cumulative distribution function

fBm fractional Brownian motion

FIFO first in, first out

LHS left hand side

LRD long range dependence

PDF probability density function
Nomenclature

PLBP  Poisson Lomax Burst Process

PPBP  Poisson Pareto Burst Process

QoS   quality of service

RHS   right hand side

SRD   short range dependence

SSQ   single server queue

VBR   variable bit rate
Chapter 1

Introduction

1.1 Background

Developed rapidly in last two decades, the Internet is becoming more and more important for our life; it is foreseeable to be the ubiquitous global communication infrastructure [1]. The blooming of the Internet leads to various kinds of applications, billions of users, as well as large amounts of traffic. In 2014, there were around three billion Internet users in the world, and the number is still growing [2]. Millions of Internet applications have been developed that keep transmitting massive volume of data over the Internet. According to Cisco white paper in 2014 [3], “annual global IP traffic will surpass the zettabyte (10^{12} \text{ gigabytes}) threshold in 2016.” Figure 1.1 shows the global IP traffic trends from 2013 to 2018 forecasted by Cisco. The traffic will almost triple by 2018 compared with that in 2013. On the other hand, the demands for quality of service (QoS) from end users are growing simultaneously. How to handle the increasing traffic efficiently while guaranteeing service quality has become a vital issue for Internet service providers (ISPs).

Naturally, the Internet exists for transmitting data from source to destination via links. A link can be either physical or virtual, and represents/constitutes a shared transmission resource. To support the Internet links, adequate bandwidth needs
to be allocated. In general, increasing the bandwidth tends to decrease the loss, and improves the QoS; but it raises costs. Consequently, ISPs face a simple but fundamental question: What is the minimum bandwidth (capacity) of a network link to ensure a particular grade of service? This problem is known as link dimensioning and its solution is indispensable for the Internet providing QoS. Adequate link dimensioning requires a thorough understanding of the interrelationships amongst: 1) the offered traffic, 2) the expected QoS level, and 3) the allocated bandwidth (capacity) [4]. Since replicating the whole Internet link is unrealistic, modeling the link by a corresponding mathematical queueing system is regarded as a fundamental approach. Queueing performance evaluation, dealing with the relationships amongst 1), 2) and 3), is considered as an important tool for link dimensioning. A prerequisite for queueing performance evaluation is a suitable model for the offered traffic.
1.1 Background

Traffic Model

Researchers have been searching for an accurate stochastic process to model Internet traffic for decades. The most important criterion of a good model is that it can capture the nature of the traffic. Network traffic models were developed based on short range dependent (SRD) stochastic processes in the 1980s [5–8]. These models are mostly related to Markovian models - which are commonly used to model traditional telephony traffic - such as the Markov-Modulated Poisson Process (MMPP) [7, 8]. The correlation functions of these SRD models decay exponentially fast over time. Researches were focused on traditional Poisson-based models until Leland and his colleagues revealed the self-similarity and long range dependence (LRD) characteristics in LAN traffic [9]. The analysis of Ethernet traffic measurements shows that the traffic variability is invariant to the observed time scale. Such behaviour could not be imitated by SRD models, making the researchers realize that SRD models were obsolete. Subsequently, ample evidences of the self-similarity and LRD in Internet traffic were provided with the Internet traffic measurements obtained from a wide ranges of sources [9–23]. After two decades, we can still find the evidence of the LRD phenomenon in the present Internet traffic. Figure 1.2 shows the plots of the packet counts (i.e. number of packets time unit) for three different choices of time units. The data of the plots on the LHS are obtained from an IP traffic trace provided by CAIDA’s equinix-sanjose monitors on high-speed Internet backbone links [24] in 2013. We can still observe the variability for different time units; such behaviour can not be found in the plots on the RHS, the traffic of which is generated by a compound Poisson model. Compared with SRD models, the correlations of LRD models decay more slowly, like a power law. One common explanation of LRD and self-similarity is that the transmitted files (flows) are heavy-tailed distributed [18, 19, 25].

On the other hand, the analysis results of the Internet traffic in backbone by Cao et al. [26] are inconsistent with the perceived ubiquity of the LRD phenomenon in
Introduction

the Internet. It is reported that the distribution of the inter-arrival times converges to exponential as the load increases. Thus the arrival process tends to be Poisson. According to [27], the long-term correlations of the arrival process are weakened as the multiplexing level increases, when the traffic is transmitted from the access network to the core network. Also, it has been shown in [28] that the behaviour of some Internet backbone traffic is nearly uncorrelated for certain time scales. However, the smoothing of the traffic burstiness only happens when the traffic is extremely heavily-multiplexed. Even for backbone or core network links, this is not always the case. For example, we can still observe the burstiness of the Internet backbone traffic in Figure 1.2. In Chapter 4, we will show that an LRD process can accurately model this traffic. Therefore, at this stage, LRD processes are appropriate to model the behaviour of the Internet traffic.

Fig. 1.2 A real traffic trace vs. a sequence generated from a compound Poisson model on different time scales.
1.1 Background

Another criterion is that the model can be defined by a small number of parameters; these parameters can be fitted by the measurable statistics of real traces. Mean, variance, and Hurst parameter are three crucial statistics that can be measured from real traces. A sequence of numbers can be obtained by summing up the amount of work arrived within each time unit. The mean is simply the average number of these numbers. The variance is a measurement of the spread between the numbers. The Hurst parameter, or Hurst exponent, is referred to as the index of LRD. It is a commonly used measure which describes the degree of correlation of a process. More details and the methods of estimating the Hurst parameter are provided in the next chapter. Of course, we do not want a model too complex to analyze. Thus, an LRD model fulfilling these criteria is considered as a model of choice.

QoS Measures

Four of the most important queueing performance measures are now listed:

- Delay – how long does it take for a bit of data from the moment it arrives until it is completely served;

- Jitter – standard error, or variance of delay;

- Blocking probability – also known as dropping probability, it is the probability (fraction of time) an arrived packet is blocked as the buffer is full;

- Overflow probability – the probability (fraction of time) that the occupied buffers exceed a certain level, which is closely related to the concepts of unfinished work distribution and virtual waiting time.

Although they do not always represent the QoS perceived by users, these four measures are commonly used in practice since they are easy to measure, monitor, and accommodate.
Introduction

Amongst these four measures, the overflow probability is the most amenable to analysis. Although it relies on the unrealistic assumption that the buffer size is $\infty$, the overflow probability is still a useful measure for network links with large buffer size. Moreover, in many publications, the overflow probability is used to approximate another more practical measure, the blocking probability. The blocking probability and the overflow probability can be very different for some extreme cases. However, the two are closely related in more realistic situations [29]. For this reason the overflow probability is chosen as the QoS measure in this thesis.

In addition, overflow probability is generally related to quality of experience (QoE). Often related to QoS, QoE is a subjective measure of a customer’s experiences with a service. It is user-centric while QoS is network-centric [30]. Kim et al. [31] proposed the QoS-QoE correlation model, and studied the QoE evaluation method using QoS measures in the converged network environment. Also, Fiedler et al. [32] have presented and evaluated a natural and generic exponential relationship between QoE and QoS for streaming service. High overflow probability represents high level of congestion and therefore it can be viewed as represents QoS perceived by users or QoE. However, it is demonstrated in [33] that low overflow probability does not always guarantee low packet delay or good QoE, e.g., when the traffic is very bursty.

Queueing Modeling and Objective

The simplest queueing model with a single server has been intensively studied in recent years. A single server queue (SSQ) is a suitable model for links where packets are multiplexed before they get served (i.e. transmitted). This queueing system is also the most fundamental component in store-and-forward networks. In this thesis, we consider an SSQ with infinite buffer, under the first in first out (FIFO) service discipline. The main objective of this thesis is to evaluate the overflow probability of a single server queue fed by an LRD input.
There are two methods to evaluate LRD queues: simulation and analytical formula. Simulation is straightforward and is simple to manipulate the parameters of the system. However, it is time-consuming, especially for LRD queues since LRD behavior is for large time scales which requires a very long simulation time to reach a consistent state. Moreover, simulation results are only for particular cases.

An LRD queue is not amenable to analysis by known methods and its behaviour is different from that of a corresponding queue fed by a Markovian input. Accordingly, obtaining analytical results for LRD queues has been considered difficult. Despite the significant efforts made regarding the performance analysis of LRD queues for a quarter of century, there are still no exact results and the existing studies have mainly focused on obtaining asymptotic results. Asymptotic results are accurate only for limiting cases. Thus, it is important to obtain accurate analytical results for LRD queueing performance over the full range of parameters, and this is the objective of this thesis.

1.2 Overview

This thesis deals with the performance evaluations of single server queues fed by two LRD traffic processes, namely the Poisson Lomax Burst process (PLBP) and the fractional Brownian motion (fBm) process. Both processes are described by a small number of parameters, and seem to have the potential to accurately model the Internet traffic. The organization of this thesis is presented as the following.

Chapter 2: Long Range Dependent Models and Queueing Analysis provides definitions and properties of self-similarity and LRD, along with the relationship between the two. Then, two related concepts: the Hurst parameter and the heavy-tailed distribution, are explained. Several pseudo-LRD models are reviewed. Although they are not purely LRD, these models can capture LRD behavior over a limited range of time scales. Next, we review the LRD models proposed for Internet traffic.
modeling, including heavy-tailed on-off processes, M/G/∞ processes, fBm, fractional autoregressive integrated moving average processes, stable non-Gaussian fractional processes, and multi-fractal processes. We focus on the M/G/∞ process and fBm as they are the models used in this thesis. In the last section, the existing performance evaluation results for LRD queues – in particular, the queues fed by M/G/∞ and fBm inputs – are presented.

Chapter 3: Performance Evaluation of a PLBP Queue introduces a novel LRD model, PLBP, which consists of bursts with Lomax distributed length that arrive according to a Poisson process. PLBP is a variant of the popular M/G/∞ traffic model, the Poisson Pareto Burst Process (PPBP). It inherits the LRD characteristics from PPBP while eliminating its limitation on modelling small traffic flows. In this chapter, we clarify the definition of PLBP and provide its key statistical parameters, the mean and the variance. The advantage of Lomax over Pareto on modelling small flows is presented through real traces. Two methodologies for evaluating the overflow probability of a PLBP queue are provided. One is the approximation based on the Quasi-Stationary (QS) algorithm. The other is a simulation method that can save much simulation time compared to the conventional simulation method. Both methods are validated by the results obtained via the conventional simulation method.

Chapter 4: Performance Evaluation of an FBM Queue provides a neat and accurate closed-form approximation for the stationary workload distribution of an SSQ fed by an fBm input. The approximation arises from a new interpretation of an asymptotically accurate formula for the tail of the distribution in [34]. A new simulation approach that re-uses a single fBm sequence is introduced, thereby saving time for generating fBm sequences. The consistency between the simulation and analytical results is discussed. Finally, we provide a link between the fBm queue size distribution and the Generalized Gamma distribution which leads to very accurate
1.3 Contributions

closed-form approximations for the important queueing statistics of an fBm queue, namely the mean, the variance, the second central moment, and the skewness.

Chapter 5: Service rate provisioning of an FBm Queue provides an elegant service provisioning formula obtained by deriving the inverse of the performance formula of an fBm queue. We discuss one weakness of the fBm model, i.e. it contains negative traffic. The effect of negative traffic weakness on the queueing performance of an fBm queue is investigated. A region, where fBm overcomes its negative traffic weakness, is obtained by means of comparing its queueing performance with simulations. At last, we demonstrate the accuracy of our fBm-based closed-form formulae by numerically comparing the overflow probability and link dimensioning results with equivalent results obtained by simulating an SSQ fed by a real traffic trace and fitted MMPP input traffic.

Chapter 6: FBm Versus PPBP and PLBP compares the queueing performance of the fBm model with that of the PPBP/PLBP model. The convergences between the fBm model and the other two models are shown for high aggregation of traffic. The pros and cons of the PLBP model and the fBm model are discussed.

Chapter 7: Conclusions provides a summary of the whole thesis as well as the direction of the future research work.

1.3 Contributions

The contributions of this thesis are mainly threefold. First, a novel LRD model, namely PLBP, is proposed. Compared to the PPBP model, the advantage of PLBP is its ability of capturing a large number of small flows. An approximation for the overflow probability of an SSQ fed by a PLBP input is developed based on the QS algorithm. This approximation is expected to help traffic engineers evaluate the performances of the Internet links.
Introduction

Second, a closed-form approximation for the overflow probabilities of an SSQ fed by an fBm input is provided. It is demonstrated, by the results of the simulations with fBm sequences and real traces, that this approximation is accurate (potentially exact) over full range of parameters. The approximation can be used as a tool to evaluate the performances of the Internet links with highly aggregated traffic. Based on the approximation, important queueing statistics and a service provisioning formula for an fBm queue are obtained, which can be used to solve the problem of bandwidth allocation and link dimensioning for network engineering and network planning.

Third, two simulation methods are introduced, one for PLBP queues and one for fBm queues. A fast simulation method is used for PLBP queues, which omits the time of training the queue and reduces the time to reach a consistent state, via manipulating the number of existing bursts at the initial stage of the simulation. For simulations of fBm queues, a new approach for re-using a single fBm sequence is developed to save time for generating fBm sequences. Both methods are able to save simulation time, and can be used to evaluate the analytical results.

The detail of the contributions of this thesis are listed in the following in the order of appearance, with corresponding chapter numbers.

• Proposal of a novel LRD model, the Poisson Lomax Burst process, which consists of bursts with Lomax distributed length that arrive according to a Poisson process. (Chapter 3)

• A new simulation method for PLBP queues that can save a lot of time compared with the conventional simulation method. (Chapter 3)

• An analytical approximation for the overflow probability of a PLBP queue based the Quasi-Stationary algorithm. (Chapter 3)
1.4 Publications

• A neat and accurate closed-form approximation for the stationary workload distribution of an fBm queue. (Chapter 4)

• A novel simulation approach for an fBm queue that re-uses a single fBm sequence. (Chapter 4)

• Closed-form approximations for the important queueing statistics of an fBm queue: the mean, the variance, the second central moment, and the skewness. (Chapter 4)

• A region, where fBm conquers its negative traffic weakness, is provided. (Chapter 5)

• A concise service provisioning formula for an fBm queue. (Chapter 5)

• Demonstration of the accuracy of the fBm-based closed-form formulae by numerically comparing the overflow probability and link dimensioning results with simulation results. (Chapter 5)

• Comparisons between the fBm model and the PPBP and the PLBP model. (Chapter 6)

1.4 Publications

The publications based on the work presented in this thesis are listed as follows.


Introduction


Chapter 2

Long Range Dependent Models and Queueing Analysis

Self-similar and long range dependent processes have become common place for Internet traffic modeling, since the two phenomena were revealed as the characteristics of network traffic in 1994 [9]. In this chapter, we attempt to clarify the definitions and the properties of self-similarity and long range dependence, review the commonly used long range dependent models, and summarize the achieved queueing analysis results for queues fed by two specific long range dependent inputs, i.e. the M/G/∞ process and the fractional Brownian motion process.

2.1 Self-Similarity and Long Range Dependence

Self-similarity and fractals are notions describing phenomena where an object is similar to parts of itself. An object is statistically self-similar if parts of it at different scales share the same statistical properties [35, 36]. This phenomenon exist with many objects in the real world, like a fern or a Romanesco broccoli in nature [37], Shepard tones in music [38], and some stock market movements in economics [39]. A phenomenon that is often related to self-similarity in telecommunications is long
range dependence (LRD). The correlation of a model involving LRD decays like a power law, much more slowly than the exponential decay exhibited by the classical Markovian models. Although LRD does not necessarily imply self-similarity and vice versa [40, 41], the two phenomena often appear together in characterizing Internet traffic since the two are interchangeable under certain restrictions. Further discussion of the relationship between the two is provided in Section 2.1.1. The empirical evidence of self-similarity and long range dependence phenomena of data network traffic was first provided by Leland and his colleagues in 1994 [9]. A graphic “proof” was provided as shown in Figure 2.1. The measurements from Ethernet traffic are presented in packets per time unit for different choices of time units as shown on the left-hand side (Figure (a) - (e)). On the right hand side, the sequences used in Figure (a’) - (e’) are from some compound Poisson
2.1 Self-Similarity and Long Range Dependence

processes. This figure clearly shows that the Ethernet traffic looks similar to itself for different time scales, which is very different from the Poisson model. Following this discovery, much work was done to provide evidence of the LRD phenomenon in the Internet traffic measurements obtained from a wide range of sources in variable bit rate (VBR) video [10–14], asynchronous transfer mode (ATM) cell traffic [15], Ethernet local area network (LAN) traffic [9, 16], metropolitan area network (MAN) traffic [17], general Internet wide area network (WAN) traffic [18–20], and between on-chip modules in typical area traffic [21].

2.1.1 Definitions and Properties

Besides the intuitive explanations of self-similarity and LRD, it is necessary to understand the mathematical definitions and properties of these two notions. The definitions given below are commonly used provided in [42] and [43].

Strictly Self-Similarity

Consider a real-valued stochastic process \( \{Y(t), t \in \mathbb{R}\} \). It is said to be strictly self-similar with self-similarity or Hurst parameter \( H \) (\( 0 < H < 1 \)), denoted \( H \)-ss, if

\[
Y(t) \overset{d}{=} a^{-H}Y(at), \ t \geq 0, \ a > 0,
\]

where \( \overset{d}{=} \) denotes equality of the finite-dimensional distributions of the two process, meaning that \( Y(t) \) follows the same distribution as \( a^{-H}Y(at) \). In addition, the stochastic process \( \{Y(t), t \in \mathbb{R}\} \) is \( H \)-ss with stationary increments, denoted \( H \)-ssi, if it is \( H \)-ss and

\[
Y(b+t) - Y(b) \overset{d}{=} Y(t) - Y(0), \ t \geq 0, \ b > 0.
\]
It can be observed that a $H$-ss process can never be stationary, because if the process $\{Y(t), \ t \in \mathbb{R}\}$ is stationary, we have $Y(t) \overset{d}{=} Y(at)$, which contradicts (2.1).

**Second-Order Self-Similarity**

For network traffic modeling, a process with second-order self-similar has drawn much attention. Let $X = \{X_t, t \in \mathbb{Z}_+\}$ be a discrete-time series, which can be interpreted as a sample of Internet traffic with a sequence of fixed-length intervals, and $X_t$ represents the aggregated or averaged values of the traffic within $t$th interval, which can be measured in units of flows, packets, bytes or bits.

The process $X$ is called strictly stationary if the processes $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ and $(X_{t_1+k}, X_{t_2+k}, \ldots, X_{t_n+k})$ follow the same joint distribution for all $n \in \mathbb{Z}_+, t_1, t_2, \ldots, t_n, k \in \mathbb{Z}$. Thus the joint distribution of $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ does not vary with time shifting. As this strict stationary turns out to be too restrictive, a weaker form of stationary, second-order stationary, is considered. Let $m = \mathbb{E}[X_t]$ and $\sigma^2 = \mathbb{E}[(X_t - m)^2]$ be the mean and variance of the process $X$. The autocorrelation function (ACF) of $X$ can be expressed as

$$
\rho(k) = \frac{\mathbb{E}[(X_t - m)(X_{t+k} - m)]}{\sigma^2}, \ k \in \mathbb{Z}. \quad (2.3)
$$

The process $X$ is called second-order stationary (or covariance stationary) if it has constant $m$, finite $\sigma^2$ and its ACF, $\rho(k)$, depends only on $k$. Henceforth, all processes considered are second-order stationary.

Now consider a second-order stationary stochastic process, $\{X_t, t \in \mathbb{Z}_+\}$. Let $X^{(j)} = \{X^{(j)}_i, i = 1, 2, \ldots\}$ be its aggregated process at level $j$, that is

$$
X^{(j)}_i = \frac{1}{j} \sum_{t=j(i-1)+1}^{ji} X_t, \ i, j = 1, 2, \ldots.
$$
2.1 Self-Similarity and Long Range Dependence

Intuitively, the process $X$ is partitioned into blocks of size $m$ indexed by $i$, and $X_i^{(j)}$ represents the average value of $X$ for the $i$th block. Let $\rho^{(j)}(k)$ be the ACF of $X^{(j)}$.

The process $X$ is exactly second-order self-similar with Hurst parameter $H$ ($0.5 < H < 1$) if

$$\rho(k) = \frac{1}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}], \ k = 1, 2, \ldots.$$  \ (2.4)

Notice that the equation above actually implies $\rho(k) = \rho^{(j)}(k)$ for all $j \geq 1$ [43], which represents the intuitive description of exact second-order self-similarity – $X$ and its aggregated process $X^{(j)}$ have the same correlation structure. Another interpretation is that $X$ and $X^{(j)}$ are indistinguishable when it comes to their second-order statistical properties [9]. A typical example of an exactly self-similar process with parameter $H$ is fractional Gaussian noise (fGn), which is the increment process of fractional Brownian motion (fBm) [44]. This process will be explained in Section 2.4.3.

The process $X$ is called asymptotically second-order self-similar with Hurst parameter $H$ ($0.5 < H < 1$) if

$$\lim_{j \to \infty} \rho^{(j)}(k) = \frac{1}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}], \ k = 1, 2, \ldots.$$  \ (2.5)

In other words, $\rho^{(j)}(k)$ approximates $\rho(k)$ for large $j$. Asymptotic second-order self-similarity is closely related to long range dependency, the property and definition of which are shown in the next section.

There is an equivalent definition for exact second-order self-similarity. The process $X$ is called exactly second-order self-similar with Hurst parameter $H$ ($0.5 < H < 1$) if

$$X \equiv m^{1-H}X^{(j)}, \ j = 1, 2, \ldots.$$  \ (2.6)
which means that $X$ follows the same distribution as $m^{1-H}X^{(j)}$ for all $j = 1, 2, \ldots$. By (2.6) and the definition of the variance of a random variable, we have the variance of the process $X^{(j)}$:

$$\text{Var}[X_i^{(j)}] = \sigma^2 j^{2(H-1)}, \text{ for all } i = 1, 2, \ldots$$  \hspace{1cm} (2.7)

Also, $X$ is called asymptotically second-order self-similar with Hurst parameter $H$ ($0.5 < H < 1$) if (2.6) holds for $j \rightarrow \infty$. Similarly, we have

$$\text{Var}[X_i^{(j)}] \sim cj^{2(H-1)}, \text{ } j \rightarrow \infty, \text{ for all } i = 1, 2, \ldots,$$  \hspace{1cm} (2.8)

where $0 < c < \infty$. This equation provides a method to identify self-similarity of real measurements, which will be discussed later.

Either exactly or asymptotically second-order self-similar processes have the feature that their aggregated processes hold a non-degenerated correlation structure. This feature was used by Leland in 1994 [9] to reveal the self-similarity nature of Ethernet traffic, by a graphical “proof” as shown in Figure 2.1. The measurements from Ethernet traffic are presented in packets per time unit for different choices of time units as shown on the left-hand side of the figure (plot (a) - (e)). From up to down, the time unit is reduced by 10 times. Let $X$ be the time series used in plot (e), the time series of plot (a) to (d) can be regarded as the aggregated processes of $X$: $X^{(100000)}$, $X^{(1000)}$, $X^{(100)}$, and $X^{(10)}$. The four plots looks similar, implying that the ACFs of $X$ and its aggregated processes are similar.

**Long Range Dependence**

A (second-order) stationary process $X$ is called long range dependent (LRD) if its ACF decays slowly, that is

$$\sum_{k=-\infty}^{\infty} |\rho(k)| = \infty.$$  \hspace{1cm} (2.9)
2.1 Self-Similarity and Long Range Dependence

On the other hand, \( X \) is called short range dependent (SRD) if its autocorrelation sum is finite.

From (2.4), we can obtain the ACF of a second-order self-similar process for \( k \to \infty \) as

\[
\rho(k) \sim ck^{2(H-1)}, \quad k \to \infty,
\]

(2.10)

where \( c \) is a finite constant. Thus the sum of \( \rho(k) \) for an asymptotically second-order self-similar process holds (2.9) for \( 0.5 < H < 1 \), which imply \( X \) is LRD. A self-similar process is not necessarily LRD. For example, fractional Brownian motion is a self-similar process, but it does not exhibit LRD when \( H \leq 0.5 \). An LRD process may not be self-similar. In [40], it has been shown that several Gaussian processes with LRD features are not self-similar. However, in sense of asymptotic second-order self-similarity, with \( 0.5 < H < 1 \), self-similarity implies LRD and vice versa [43]. This is the reason why the two phenomena are used interchangeably for Internet traffic modeling.

It is difficult to obtain the correlation function from the real measurements of Internet traffic. In practice, the Hurst parameter \( H \) is a commonly used measure for describing the degree of correlation of a process. An LRD traffic stream has \( H > 0.5 \). The larger the value of \( H \) is, the more bursty the traffic is, which usually indicates worse performance for a queue fed by an LRD input. In the following section, we will introduce the methods developed by other researchers to estimate \( H \).

2.1.2 Estimations of the Hurst parameter

Various methods have been developed to estimate the Hurst parameter. One commonly used technique is called aggregated variance or variance-time plots. Taking logarithm of both sides of (2.8), we have

\[
\log(\text{Var}[X^{(j)}]) \sim 2(H - 1) \log(j) + 2H - 1 \log(c), \quad j \to \infty.
\]

(2.11)
Long Range Dependent Models and Queueing Analysis

First, we need to obtain a log-log plot of \( \text{Var}[X^{(j)}] \) versus \( j \). Then, according to (2.11), we can draw a line through the resulting points for large \( j \) and estimate \( H \), since the slope of the line is equal to \( 2(H - 1) \). For real Internet traffic traces, we usually partition the traffic process with fix-length time interval, \( t \). We can obtain a process for the total traffic volume within each interval \( t \). The concept of this process is quite similar to the aggregated process \( X^{(j)} \) without the division by \( j \). It can be actually considered as \( tX^{(t)} \). Thus, by (2.8), the variance of this process can be expressed as

\[
\sigma^2(t) = \text{Var}[tX^{(t)}] \sim t^2c_2t^{2(H-1)} = ct^{2H}, \ t \to \infty, \quad (2.12)
\]

and

\[
\log(\sigma^2(t)) \sim 2H \log(t) + 2H \log(c), \ t \to \infty, \quad (2.13)
\]

where \( \sigma^2(t) \) denotes the variance of \( tX^{(t)} \). With the same process as we described above, we can get a slope of a line for large \( t \) from the log-log plot of \( \sigma^2(t) \) versus \( t \). The only difference is that the slope is now given by \( 2H \). In this thesis, we estimate \( H \) by this method. Also, by (2.13), we can identify LRD phenomenon of a process if \( \sigma^2(t) \sim ct^{2H} \) for large \( t \).

In addition to this method, R/S statistics [45, 46] and periodogram-based analysis [47, 48] are two well-known approaches to estimate \( H \). For example, Whittle’s approximation maximum likelihood (MLE) [47] is a periodogram-based method that also provides a confidence interval for the estimate of \( H \). These three methods mentioned above were implemented in [49] and yielded similar \( H \) values (\( > 0.8 \)) for Wireless LAN (WLAN) traffic. Wavelet analysis [50] is another important and popular approach to analyze LRD, estimate \( H \) and simulate data [51, 52]. These methods are summarized in [53–56]. In spite of these various approaches, researcher are still seeking more accurate and faster method to estimate \( H \) [57, 58].
2.2 Heavy-Tailed Distributions

The heavy-detailed distribution of object or file sizes has been widely considered as the cause of LRD of Internet traffic as suggested by [18, 25, 59]. Several studies [59–64] have provided evidence that file, flow, and burst sizes of Internet traffic are heavy-tailed, which arouse the interest in modeling Internet traffic with heavy-tailed distributions. The use of heavy-tailed distributions to model Internet traffic has been discussed in [62, 65, 66].

![Image of distribution comparison]

Fig. 2.2 Heavy-tailed distribution vs. Exponential distribution.

Literally, a distribution with a “tail” heavier than Exponential distribution is called heavy-tailed, which is illustrated in Figure 2.2. The complementary cumulative distribution function (CCDF) of an Exponential random variable, \( X \), with parameter, \( \lambda > 0 \), is

\[
P(X > x) = e^{-\lambda x}, \quad x \geq 0.
\]  (2.14)
Long Range Dependent Models and Queueing Analysis

The Poisson process with its inter-arrival times as $X$, is a very important model for telecommunications.

Figure 2.2 shows the CCDF of the heavy-tailed and exponential distributions. Obviously, the tail part (large $x$) of the heavy-tailed distribution, $P(X > x)$ is larger than that Exponential distribution. To put it differently, the probability of the occurrence of large values in a heavy-tailed distribution is much larger than under the Exponential distribution. We say the distribution of a random variable $X$ is heavy-tailed if

$$\lim_{\lambda \to \infty} e^{\lambda x} P(X > x) = \infty, \text{ for all } \lambda > 0.$$  \hspace{1cm} (2.15)

$$P(X > x) \sim L(x)x^{-\alpha}, \text{ } x \to \infty,$$ \hspace{1cm} (2.16)

where $X$ is the random variable of the distribution, $L(x)$ is a slowly varying function [67], and $0 < \alpha < 2$ is called the shape parameter or the tail index.

An canonical heavy-tailed distribution is the Pareto Distribution. The distribution was created by Pareto to describe the allocation of wealth and the distribution of income [68]. Now it has been applied to variety of fields including describing file sizes distribution of Internet traffic. The CCDF of a Pareto random variable, $X$, is

$$P(X > x) = \begin{cases} \left(\frac{x}{\delta}\right)^{-\gamma}, & x > \delta, \\ 1, & \text{otherwise}, \end{cases}$$ \hspace{1cm} (2.17)

where $\gamma > 0$ is the shape parameter, and $\delta > 0$ is the scale (or location) parameter. When $1 < \gamma < 2$, Pareto has infinite variance and finite mean $\frac{\delta \gamma}{(\gamma-1)}$. There are several variants of Pareto. Table 2.1 shows the CCDF and the mean $E[X]$ of three variants of classical Pareto – Pareto distribution Type II, III and IV – as introduced by [69]. The original Pareto distribution is called Pareto distribution Type I. For Pareto Type II - IV, $\delta$ is the scale parameter, $\mu$ is the location parameter, $\gamma$ is the scale parameter,
and $\alpha$ is the inequality parameter. According to the CCDFs, $\mu$ is the minimum value of $X$, in other words, the curves of the distributions start from the location $\mu$. The classical Pareto is a special case of Pareto Type II for $\mu = \delta$, thus $\delta$ is both the scale and location parameters. The Lomax distribution is another spacial case for Pareto Type II for $\mu = 0$. The further discussion of the Lomax and Pareto distributions is provided in Section 3.1. Furthermore, Pareto Type I - III can be considered as special cases of Pareto Type IV.

Besides Pareto-based distributions, there are various heavy-tailed distributions, e.g. the log-normal distribution [70], the Lévy distribution [71] and the Weibull distribution [72]. A heavy-tailed distribution is called fat-tailed if its tail is even “heavier”. The CCDF of such distribution follows

$$P(X > x) \sim x^{-\alpha}, \ x \to \infty,$$

with $\alpha > 0$. For example, Pareto is fat-tailed, however, log-normal is not although it is heavy-tailed.

## 2.3 Pseudo-LRD Models

The discovery of the LRD phenomenon of Internet traffic made the researchers realize that traditional SRD models are no more suitable. This motivated the development
Long Range Dependent Models and Queueing Analysis

of approaches to generate processes that could capture the LRD behavior based on SRD models. Although LRD sequences cannot be directly generated by SRD models, several ways have been found to generate pseudo-LRD process that are able to capture LRD behavior over a limited range of time scales. One common method is the superposition of SRD processes. These models are of some value since only a finite range of time scales may be considered in some practical situations. Furthermore, pseudo-LRD processes have an advantage of modeling some Internet traffic which contains the short range correlation closed to SRD. Next, two commonly used pseudo-LRD models are reviewed.

2.3.1 Autoregressive Models

The autoregressive (AR) model is the most commonly used model for time series data. An AR model of order $p$, denoted as $AR(p)$, is defined as

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + c \varepsilon_t, \quad t = 1, 2, \ldots, \quad (2.18)$$

where $X_t$ is the output of the process, $X_0 = 0$, $\varepsilon_t$ is white noise, $\phi_1, \ldots, \phi_p$ are the parameters of the model and $c$ is a constant. As can be observed from (2.18), the output $X_t$ depends linearly on its $p$ previous outputs values, $X_{t-p}, \ldots, X_{t-1}$. If $B$ denotes the backshift operator, namely $X_{t-1} = BX_t$, then (2.18) can be expressed as

$$X_t = c + \sum_{i=1}^{p} \phi_i B^i X_t + c \varepsilon_t. \quad (2.19)$$

With the polynomial notation, $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$, we have

$$\Phi(B)X(t) = b \varepsilon_t. \quad (2.20)$$
2.3 Pseudo-LRD Models

The ACF of an AR model decays exponentially, so the model is SRD. An approach of generating a pseudo-LRD process which consists in superposing two AR(1) processes, was provided in [17]. It has been shown in [73] that the performance of a single server queue (SSQ) fed by such a process is equal to that of the corresponding queue with a purely LRD process. Another approach of mixing AR processes has been provided in [74]; such process was applied to model Internet traffic in [75].

2.3.2 Markov Modulated Poisson Processes

The Markov arrival process (MAP) [76], one of the earliest models for data network traffic, is widely used for modeling communication network traffic. The simplest MAP is the Poisson process, used as a dominant model for telephony networks. Although the Poisson process itself is SRD, many pseudo or purely LRD models are built based on it, for instance, the Markov modulated Poisson process (MMPP) [7, 8].

![Fig. 2.3 Two-state MMPP.](image)

The MMPP is another commonly used MAP, consisting of $m$ Poisson processes, the switching between them being controlled by an embedded continuous-time Markov chain. Figure 2.3 illustrates an MMPP with two states. There is a different Poisson arrival process for each state with rates $\lambda_1$ and $\lambda_2$, and the transitions between the two states occur at rates $P_{12}$ and $P_{21}$. Several pseudo-LRD processes have been
constructed by superposition of MMPPs [77–81]. In [77], the superposition of \( m \) two-state MMPPs was proposed, and shown to mimic LRD behavior within a range of timescales. It has been shown in [78] that such process converges to a popular LRD process, the fractional Brownian motion process when \( m \to \infty \). This model was extended by [79], where \( M \)-state MMPPs were considered. A new hierarchical MMPP model has been proposed in [81], based on the notion of sessions and flows of the real traces. Algorithms to estimate the parameters of the above models were provided in [78–81].

2.4 Long Range Dependent Traffic Models

Despite the merits of pseudo-LRD models, purely LRD models still provide more flexibility since pseudo-LRD models are unable to capture the LRD behavior for full range of time. In the following, the existing LRD models are reviewed. We mainly focus on the M/G/\( \infty \) process and the fBm process, as they are closely related to the work provided by this thesis.

2.4.1 Heavy-Tailed On-Off Processes

A simple on-off process contains only two states, namely ON and OFF. It is usually used to describe the traffic generated by a single traffic source. During the ON period, the source generates the traffic at a constant rate \( r \), and no traffic is generated during the OFF period. Figure 2.4 illustrates the formation of the process, \( \{X_t, t \geq 0\} \). In steady state, the lengths of ON and OFF periods are generally distributed with means \( \mu_{on} \) and \( \mu_{off} \), respectively. The stationary probability of ON state is

\[
P_{on} = \frac{\mu_{on}}{\mu_{on} + \mu_{off}}.
\]

(2.21)
2.4 Long Range Dependent Traffic Models

Here, $X_t$ represents the state at time $t \geq 0$ with values 1 and 0 representing the ON and OFF states, respectively. In other words, 1 means that the source is generating traffic at rate $r$. For each $t \geq 0$, we see that $X_t$ is Bernoulli random variable with parameter $P_{on}$. Thus the mean and variance of $X_t$ are given by $E[X_t] = P_{on}$ and $Var[X_t] = P_{on}$, respectively. Let $R_t$ as the rate of the source generating traffic at $t$, i.e., $R_t = rX_t$. We can obtain the mean and variance of $R_t$ as $E[R_t] = rP_{on}$ and $Var[R_t] = r^2P_{on}(1 - P_{off})$.

![Fig. 2.4 Simple on-off process with transition rate $P_{on}$ and $P_{off}$.](image)

When the distributions of the ON and OFF periods are both exponential, the process is SRD and can be modeled as a two-state Markov chain with transition rates $P_{on}$ and $P_{off}$. Its state transition diagram is shown in Figure 2.4. When the distribution of ON and/or OFF periods is heavy-tailed, the resulting on-off process is LRD [82]. Notice that the distribution of either ON or OFF state is heavy-tailed, the on-off process with multiple sources can be LRD [83–85].

In most cases, multiple on-off sources are used and the aggregation of their work forms a new process which is used to model the traffic. It has been proved in [25, 86, 87] that the aggregate process is also LRD. As said in Section 2.2, the packets (files) sizes of Internet traffic are heavy-tailed distributed. Under the assumption that the sources contribute work at a constant rate, the distribution of durations of files staying in the queue can be considered as heavy-tailed. Since the sources contains
heavy-tailed distributed periods and the aggregate process is LRD, we believe that such process captures both the cause and effect of LRD in Internet traffic.

2.4.2 M/G/∞ Processes

The M/G/∞ process is popular for modeling Internet traffic; it is also called the Poisson burst process by Roberts et al. in [88]. A continuous-time version of this process was proposed by [89] in 1974, while the idea of its discrete-time version first appeared in [90]. A discrete-time M/G/∞ process describes the number of busy servers of a discrete-time infinite server system fed by an input with Poisson arrivals, the service times $d$ of which are generic distributed. Let $\{B_{t+1} : B_{t+1} \in \mathbb{Z}^+, t = 0, 1, \ldots\}$ be an i.i.d. Poisson process with rate $\lambda$, where $B_{t+1}$ denotes the number of the sessions arrive within the time interval $[t, t + 1)$. Let $d_{t+1,i}, i = 1, \ldots, B_{t+1}$, be the service time of the session arrived within the interval $[t, t + 1)$, where $i$ is the index of the session. The random variables $\{d_{t+1,i}, t, i = 1, 2 \ldots\}$ are i.i.d. with common probability mass function (PMF) $G$. This PMF $G$ is assumed to have a finite first moment. The busy server process denoted $\{b_t, t = 0, 1, \ldots\}$ with $b_t$ indicating the number of busy servers at time $t$, and is referred as the M/G/∞ process. Thus, a stationary M/G/∞ process can be expressed as

$$b_t = \sum_{s=1}^{t} \sum_{i=1}^{B_s} \mathbb{1}_{\{s+d_{s,i}>t\}} + \hat{B} \sum_{i=1}^{\hat{B}} \mathbb{1}_{\{\hat{d}_i > t\}},$$

(2.22)

where

$$\mathbb{1}_X = \begin{cases} 1 & \text{if } X \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

Here, $\hat{B}$ is the number of initial sessions that existed already in the system at $t = 0$, and $\{\hat{d}_1, \ldots, \hat{d}_{\hat{B}}\}$ are the remaining service times of these sessions. The random variable
Long Range Dependent Traffic Models

$\hat{B}$ is Poisson distributed with rate $\lambda E[d]$, and the random variables $\{\hat{d}_1, \ldots, \hat{d}_{\hat{B}}\}$ are i.i.d. and distributed according to the forward recurrence time of $d$.

The structure of the $M/G/\infty$ process can be considered as a natural abstraction of a layered network. After a session or a connection has been established at the upper layer, packets are transmitted continuously for a certain duration at the lower layer. The relevance of this input model to Internet traffic modeling has been explained by Likhanov, Tsybakov and Georganas [86], by showing that the aggregation of multiple i.i.d. on-off sources with Pareto distributed activity periods behaves like the $M/G/\infty$ process as the number of sources increases. Another advantage of the $M/G/\infty$ is that it is stable under multiplexing. Its positive dependencies over a wide range of time scales can be captured easily by manipulating the tail behavior of the service time. Owing to its flexibility and tractability, the $M/G/\infty$ process was argued as a viable model for network traffic by Parulekar and Makowski [91]. Several studies have shown that the $M/G/\infty$ can be used to model the LRD traffic streams [86, 92–94].

The Poisson Pareto Burst Process (PPBP) [88, 95], also known as the $M/Pareto$ process [73, 96], is a special case of the $M/G/\infty$ process. Rather than the number of sessions existing in the server, PPBP describes the workloads contributed by the sessions within the consecutive time intervals. PPBP consists of bursts, arriving according to a Poisson process with rate $\lambda$. Each burst establishes a session with a duration $d$. The random variable $d$ is Pareto distributed with scale parameter $\delta$ and shape parameter $\gamma$. It is assumed that each burst generates work at a constant rate, $r$. Let $t$ be the length of the time interval. From [95], the mean of the process is finite as given by $\frac{\lambda \delta^\gamma}{(\gamma-1)}$ for $1 < \gamma < 2$. Its variance function is proportional to $t^{2H}$ with $H = \frac{3-\gamma}{2}$ for $t \to \infty$, which implies that PPBP is LRD with $H > 0.5$ according to (2.13). It was proved in [96, 97] that PPBP is capable of modeling real Internet traffic. Zukerman et al. [97] provided a method to fit the four parameters of PPBP to the real measurements. One minor weakness of PPBP is that it omits the small
flows of real traffic, the durations of which can be much smaller than the threshold $\delta$. Although Pareto is chosen for its “heavy tail” and the loss of those small flows may not always influence the queueing behavior, it is still desirable to find a distribution that can take care of those small bursts while retaining the same tail-behavior as Pareto.

Considering the $M/G/\infty$ model as a limiting case of the aggregation of many on-off processes, it is natural to wonder as to the limiting case of PPBP for $\lambda \to \infty$. Based on the central limit theorem (CLT), it has been suggested in [98] that the input process formed by the aggregation of many i.i.d. sources tends to be Gaussian, as the number of sources increases. Furthermore, it has been shown in [29, 99] that PPBP tends towards a Gaussian process as $\lambda \to \infty$.

### 2.4.3 Fractional Brownian Motion

The fractional Brownian motion (fBm), as a generalization of the Brownian motion, is a continuous-time Gaussian process. The term, “fractional Brownian motion”, was introduced by Mandelbrot and Van Ness in 1968 [44]. The process reduces to a standard Brownian motion when its Hurst parameter $H = 0.5$. FBM is strictly self-similar, and exhibits LRD for $H > 0.5$.

A normalized fBm process, $B = \{B(t), t \geq 0\}$, with $0.5 < H < 1$ is characterized by the following properties [100]:

- $B$ has stationary increments;
- $B(0) = 0$ and $E[B(t)] = 0$ for all $t \geq 0$;
- $\text{Var}[B(t)] = t^{2H}$ for all $t \geq 0$;
- $B$ has continuous sample paths;
- $B$ is a Gaussian process, i.e. all of its finite-dimensional marginal distributions are Gaussian.
2.4 Long Range Dependent Traffic Models

The stationary increment process of fBm \( \{Z_k, k = 0, 1, \ldots\} \), is known as the fractional Gaussian noise (fGn), and is defined by

\[
Z_k = B(k+1) - B(k), \quad k = 0, 1, \ldots
\]

fBm is the only Gaussian process that is self-similar and has stationary increments [90]. Since it is Gaussian and LRD for \( H > 0.5 \), the fBm process is often considered a limiting case for many other LRD traffic models. For example, it has been shown in [101] that the superposition of \( N \) identical on-off sources, with heavy-tailed distributed ON and/or OFF periods, converges to a Gaussian process with LRD as \( N \to \infty \). The work also compared the queueing performance of this limiting, LRD, Gaussian process with that of fBm from [100], providing a further motivation for the fBm model. Another example, as discussed in the previous section, is that PPBP tends towards a Gaussian process as \( \lambda \to \infty \) shown in [29, 99]. Thus, fBm has been widely used for modeling the traffic on the links that share a large number of users, e.g., core and metropolitan Internet links. The fBm process has been validated as an accurate model for such traffic using real measurements, in a number of publications [9, 25, 102–104]. Several exact/approximate methods have been developed to generate fBm sequences, for example, the Hosking method [105], the Cholesky method [106], the Davies and Harte method [107], the stochastic representation method [44], and the Paxson method [108].

2.4.4 Other LRD Models

FARIMA Processes

The fractional autoregressive integrated moving average (FARIMA) process was originally introduced by Granger & Joyeux [109], and Hosking [110], and is one of most frequently studied LRD traffic models. It is an extension to the autoregressive
integrated moving average (ARIMA) process, which is an SRD model. The ARIMA model is a generalization of the autoregressive moving average (ARMA) model, and contains the autoregressive (AR) and moving average (MA) models.

A FARIMA process, denoted FARIMA($p, d, q$), is a time series model, where $p$ is the order of the autoregression, $d$ is the difference parameter, and $q$ is the order of the moving average. The only difference between the FARIMA process and the ARIMA process is that $d$ is now allowed to be a non-integer value. A FARIMA process is LRD with the Hurst parameter $H = d + 0.5$, when $0 < d < 0.5$ [110]. The simplest FARIMA model is FARIMA(0, $d$, 0), which is similar to an fGn process with $d = H − 0.5$. The ability of capturing both LRD and SRD correlation structures is one of the advantages of this model. The process was used to model VBR video traffic in [10, 111], and parameter fitting techniques were presented in [11]. However, the generation time for a FARIMA sequence is longer that that for other LRD models, which motivates the studies of faster generation methods, such as the one provided by [112].

Stable Non-Gaussian Fractional Processes

The fBm model, as a Gaussian process, does not allow for large fluctuations, thus cannot model burstiness well [100]. Stable non-Gaussian fractional processes are used to model those traffic that are both LRD and busty. Samorodnitsky and Taqqu have provided a thorough study of stable non-Gaussian random processes in [113]. It has been shown in [114] that the superposition of heavy-tailed on-off sources converges to an $\alpha$-stable Lévy motion process under certain conditions. Several other self-similar stable motions have been proposed and their ability to capture the properties of measured traffic streams have been shown, such as models based on the linear fractional stable noise [115], the linear fractional stable motion [116], and
2.5 Performance Analysis of LRD Queues

the fractional Lévy motion [117]. An $\alpha$-stable LRD process was used as the model for broadband network traffic in [118].

Multi-Fractal Processes

Since LRD behavior only reflects the correlations for large time scale, the LRD models mentioned above may not be suitable when the short term correlations also exhibit scaling behavior. For such situations, multi-fractal processes – which exhibit different behaviors for different time scales retaining LRD and burstiness properties – are considered to be appropriate models. In fact, self-similar processes, such as fBm, are a sub-class of multi-fractal processes. The need for multi-fractal processes for traffic modeling has been proved in [20]. In [119], it was shown that a multi-fractal wavelet model can capture the behavior of a TCP trace better than fBm. Maulik and Resnick [120] have proposed a family of models, exhibiting self-similarity at large time scales and multi-fractality at small time scales. On the other hand, self-similar processes were shown to be good enough for Internet traffic modeling and multi-fractal processes are not necessary, with the LAN and WAN traces from Bellcore [103].

2.5 Performance Analysis of LRD Queues

Many research papers have shown that the characteristics of the queueing system with LRD inputs differ from those of the corresponding systems with SRD traffic models. For example, consider a simple SSQ with infinite buffer, where $Q$ denotes the steady state queue size. In such a queue under Markovian input, the overflow probability (tail distribution of the queue size) decays exponentially [59], i.e.,

$$P(Q > x) \sim e^{-hx} \ (x \to \infty),$$  \hspace{1cm} (2.23)
where $x$ is the buffer threshold and $b > 0$ is the asymptotic decay rate. On the other hand, the overflow probability of the queue with LRD input (in particular, the model based on fBm) follows a Weibull type asymptotic, i.e.,

$$P(Q > x) \sim e^{-e^{-c2^{-2H}}} \quad (x \to \infty),$$  \hspace{1cm} (2.24)

where $H$ is the Hurst parameter and $c > 0$ is a constant. It is obvious that (2.23) allows more optimistic forecasts compared to (2.24). At present, no general analytical results for LRD queues have been found, with existing results being mostly asymptotic. For example, upper and lower bounds for the overflow probabilities based on (2.24) have been derived, where the buffer threshold $x$ becomes large. As they are used to describe limiting behavior, asymptotic results are not always applicable to empirical situations. Moreover, it has been demonstrated by Choudhury, Lucantoni and Whitt [121] that the tail behavior of LRD queues may not be characteristic of the entire distribution. As a result, it is desirable to seek analytical expressions which are accurate over the full range of parameter values. In the following, the existing results for $M/G/\infty$ and fBm queues are reviewed.

### 2.5.1 $M/G/\infty$ Queue

The performance analysis of the $M/G/\infty$ processes has attracted a great deal attention. The queueing analyses are generally focused on an SSQ fed by an $M/G/\infty$ process. We consider an SSQ with service rate $\mu$. For the $M/G/\infty$ process, the arrival process of sessions are Poisson with rate $\lambda$, and the service time of each session $d$ follows the distribution $G$. Details of the $M/G/\infty$ process have been provided in Section 2.4.2. A great variety of asymptotic results for $M/G/\infty$ queues have been developed in different regimes:

1) buffer threshold tends to infinity,
2.5 Performance Analysis of LRD Queues

2) \( n \) tends to infinity, buffer threshold and \( \mu \) are linear in \( n \),

3) \( n \) tends to infinity, buffer threshold is linear in \( \sqrt{n} \), \( \mu \) increases with \( n \) such that the net mean input – as the difference between \( \lambda \) and \( \mu \) – is linear in \( \sqrt{n} \),

4) \( \rho = \lambda E(d)/\mu \) increases to 1, buffer threshold \( \sim (1 - \rho)^{1/\gamma - 1} \),

where \( \rho \) denotes the utilization of the queue, \( \gamma \) is the shape parameter of the Pareto distribution. Here, \( n \) is an increasingly large integer, and can be considered as an intensity level for arrivals, and a pooling level of resources. In particular, the parameters of the system are scaled by \( n \), i.e., the arrival rate is set to be \( n\lambda \) and the service rate is set to be \( n\mu \).

Asymptotic regimes 1) and 2) are generally known in the large buffer limit, and the many source limit, respectively. For both regimes, either the buffer threshold increases or the pooling level of resources increases, with the overflow probability decaying exponentially to 0; thus they can be referred to as as large deviations limit. Asymptotic regime 3) has been proposed in [122] as a practical way of provisioning buffers to satisfy Internet traffic growth. Since the utilization tends to 1 as \( n \) increases, the asymptotic results are termed heavy traffic approximations. In addition, the results are associated with the CLT limit since they are obtained by adopting the Gaussian assumption. Asymptotic regime 4) [64] also applies the heavy traffic limit; but, in contrast to regime 3), the service rate here remains constant. The arrival rate increases to \( c/E(d) \) leading to \( \rho \to 1 \). In addition, the buffer threshold is scaled by the heavy traffic normalizer \( (1 - \rho)^{1/(\gamma - 1)} \) where \( d \) is distributed according to the Pareto distribution with the shape parameter \( \gamma \). A summary of the asymptotic results for M/G/\( \infty \) queues has been provided in [99]. Table 2.2 lists the publications for the four cases mentioned above. Other than these asymptotic results, the Quasi-Stationary approximation for the overflow probabilities of a PPBP queue has been provided in [123], with the accuracy of this approximation shown by the comparisons with simulation results.
Table 2.2 Classification of results on queues with $M/G/\infty$ traffic by [99]

<table>
<thead>
<tr>
<th>Asymptotic Regime</th>
<th>Method</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>Large buffer large deviations limit</td>
<td>[42, 124–132]</td>
</tr>
<tr>
<td>2)</td>
<td>Many sources large deviations limit</td>
<td>[85, 130, 133–138]</td>
</tr>
<tr>
<td>3)</td>
<td>Heavy Traffic &amp; CLT limit</td>
<td>[98, 101, 114, 139]</td>
</tr>
<tr>
<td>4)</td>
<td>Heavy Traffic non CLT limit</td>
<td>[64]</td>
</tr>
</tbody>
</table>

2.5.2 FBm Queue

A single server queue fed by fBm input has been considered an important model for Internet queueing performance analysis and capacity assignment, and has attracted significant attention. A storage model with fBm was introduced by Norros in 1994 [100], where an easy lower bound of the overflow probability was observed based on the scaling laws. An consistent result was obtained by Addie, Zukerman and Neame for a single server queue fed by fractal traffic in [17]. Duffield and O’Connell [140] derived an asymptotic result for an fBm queue by applying the large deviations theory of Gaussian processes. Roberts et al. [141] attempted to apply the so-called Beneš method to obtain the queue length distribution. These three approaches to study the performance of an fBm queue – scaling relations, large deviations theory and a Beneš formula – were described in [142], which also presents another approach based on Girsanov formula [143]. Subsequently, a queue length asymptotic result based on Fourier expansion for the fBm process was developed by Narayan [144]. The extreme value theory was applied by Choe and Shroff [145] to derive asymptotic results for a general class of Gaussian processes with stationary increments. Almost simultaneously, Massoulie and Simonian [146] applied the extreme value theory of Gaussian processes to obtain the same results. Such results were extended for more general trends and more general Gaussian processes by Hüsler and Piterbarg [34]. Addie, Mannersalo and Norros [123, 147] developed estimates for overflow probability of an SSQ with a general Gaussian process; the result was extended in
Chapter 3

Performance Evaluation of a PLBP Queue

There has been extensive research on Internet traffic modelling following the discovery of its Long Range Dependence (LRD) characteristics [9]. The Poisson Pareto Burst Process (PPBP) [95, 99, 147] has been considered as a suitable model for Internet traffic because of its inherent structure and statistical characteristics. PPBP consists of bursts with Pareto distributed length that arrive according to a Poisson process. Several authors have referred to this process as an M/G/\infty input process [92, 96, 127–129, 133, 135, 150] with Pareto distributed service times. There has been significant research efforts on the performance analysis of queues fed by PPBP [86, 92, 95, 97, 99, 127, 128, 130, 133, 135, 151, 152]. Of particular interest was a discrete G/D/1 system (considered also here) fed by PPBP. Here, we replace the Pareto burst distribution of PPBP with a Lomax distribution [153–156] so that small traffic flows can be taken into account. The resulting input process is called the Poisson Lomax Burst Process (PLBP).

A Pareto distribution is characterized by two parameters – its shape parameter $\gamma > 0$, and its scale parameter $\delta > 0$. Furthermore, $\delta$ also serves as the lower bound of the Pareto random variable, with the consequence that the modeled flow sizes
have a positive minimum value. The fact that $\delta$ is both the shape parameter and the minimum flow size introduces the following difficulty. The scale parameter $\delta$ is one of the key parameters of the Pareto distribution together with its shape parameter (and possibly other traffic parameters) that are often selected to fit certain statistical characteristics of a process or distribution which are of interest in traffic modelling applications. Such fitting normally results in $\delta$ taking a value that is too large to be appropriate for the minimum flow size which can be only few bytes. To release the scale parameter $\delta$ from its second role as the minimum flow size, we adopt the Lomax distribution, retaining the Pareto heavy tail characteristics but allowing the modeling of a large number of small traffic flows that are excluded using the traditional Pareto distribution.

The remainder of this chapter is organized as follows. In Section 3.1, we clarify the definition of PLBP and provide its key statistical parameters. In Section 3.2, we describe approximation and simulation methodologies for evaluating of the overflow probability of a PLBP queue. In Section 3.3, we validate the approximation by simulations and compare PLBP with PPBP for a certain set of parameters.

### 3.1 The Poisson Lomax Burst Process

Like PPBP, PLBP is based on a stream of bursts with burst arrivals following a Poisson process with rate $\lambda$ [bursts/s]. In PLBP, burst durations are assumed to be i.i.d. Lomax distributed random variables with parameters $\gamma$ and $\delta$. Let $d$ [s] be a generic random variable representing a burst duration. Assume that each burst generates work at a constant rate, $r$ [B/s]. Because arrivals of new bursts may occur before the completion of existing bursts, multiple overlapping bursts may be transmitted simultaneously, and since the number of overlapping bursts is unbounded, this arrival process can be viewed as a special case of an $M/G/\infty$ queueing model where the burst length (or the transmission time) are Lomax distributed. Figure
3.1 The Poisson Lomax Burst Process

3.1 presents how the PLBP is formed by the Exponential and Lomax distributions. In particular, the inter-arrival times of the bursts are exponentially distributed with parameter $\lambda$ implying a Poisson arrival process, and the burst durations follow a Lomax distribution with parameters $\delta$ and $\gamma$. In the following, the key parameters of the Lomax distribution and PLBP are provided.

![Image of the Poisson Lomax Burst Process]

Fig. 3.1 Formation of the Poisson Lomax Burst Process.

3.1.1 The Lomax Distribution

As a variant of Pareto, the Lomax distribution permits values arbitrarily close to 0 with non-zero probability. The complementary cumulative distribution function (CCDF) of a Lomax random variable $d$ is

$$P(d > x) = \left(1 + \frac{x}{\delta}\right)^{-\gamma}, \quad x \geq 0 \quad (3.1)$$

where $\delta$ is the scale parameter, and $\gamma$ is the shape parameter that controls the tail behavior of the distribution. The CCDF of Pareto has been provided in Section 2.2 as

$$P(d > x) = \begin{cases} 
\left(\frac{x}{\delta}\right)^{-\gamma}, & x > \delta, \\
1, & \text{otherwise}.
\end{cases}$$
Performance Evaluation of a PLBP Queue

We see that the tail behaviors of Lomax and Pareto shall be similar. The difference between the two is that the minimum value of $d$ for Lomax is 0 while the minimum value of $d$ for Pareto is equal to its scale parameter $\delta$.

The mean of $d$ is

$$E[d] = \begin{cases} \frac{\delta}{\gamma-1}, & \gamma > 1, \\ \infty, & \text{otherwise}. \end{cases} \quad (3.2)$$

Assume $\gamma > 1$, let $\omega$ be the Forward Recurrence Time (FRT) of a Lomax random variable with parameters, $\gamma$ and $\delta$. Its probability density function (PDF) is

$$f_\omega(x) = \frac{1 - P(d \leq x)}{E(d)} = \frac{P(d > x)}{E(d)} = \frac{\gamma - 1}{\delta} \left(1 + \frac{x}{\delta}\right)^{-\gamma}, \ x \geq 0, \quad (3.3)$$

and its CCDF can derived as

$$P(\omega > x) = \int_x^\infty f_\omega(tx)dt = \frac{\gamma - 1}{\delta} \int_x^\infty \left(1 + \frac{t}{\delta}\right)^{-\gamma} dt = (\gamma - 1) \left(0 - \frac{(1 + \frac{x}{\delta})^{-\gamma+1}}{-\gamma+1}\right) = \left(1 + \frac{x}{\delta}\right)^{1-\gamma}, \ x \geq 0. \quad (3.4)$$

It is understandable that $\delta$ is larger than the minimum flow size of some real networks where Lomax seems to be a good alternative for Pareto. Figure 3.2 shows one example when Lomax is a better choice than Pareto. The real traces are flows captured by CAIDA’s equinix-sanjose monitors on high-speed Internet backbone links in 2012 and 2014 [24]. For simplicity, we assume $r = 1$. The relationship between $\gamma$ and $\delta$ for Lomax is obtained via (3.2) in order to fit the mean flow size (8192.7 and 11015.8 [Byte]). Then, we fit the curve of the real traces as shown in
3.1 The Poisson Lomax Burst Process

Fig. 3.2 The complementary cumulative distribution, $P(d > x)$, of Pareto (with $\gamma = 1.108$) and Lomax (with $\gamma = 1.054$) vs. real traces.
the plot, by the curve fitting tool of Matlab which applies the nonlinear least squares method [157]. We obtain $\gamma = 1.108$ and $\delta = 884.8$ for the 2012 data, and $\gamma = 1.054$, $\delta = 969.5$ for the 2014 data. To ensure a similar tail behavior, we retain the same value for $\gamma$ of Pareto and the value of $\delta$ is estimated to fit the mean of the real traces. We observe in Figure 3.2 that Lomax fits better than Pareto for small flows ($< 10^4$ Bytes) while for larger flows, the tails of Pareto and Lomax are close to each other and reasonably close to the real traces (with Lomax slightly closer). Although Lomax is still very rough at small flows, it does provides improvements compared Pareto.

3.1.2 Statistics of PLBP

Let $B_t$ be the number of active bursts contributing work at time $t$. To make the burst process stationary, the system is initialized with $b_0$ initial bursts, where $b_0$ is a Poisson random variable with mean $\lambda E(d)$. The duration of each burst is the FRT of an i.i.d. Lomax random variables. By [158, p. 188], as the burst arrivals follow a Poisson process, $B_t$ are Poisson-distributed, with mean $E[B_t] = \lambda E[d]$. As illustrated in Figure 3.1, the work contributed by all bursts during the interval $(0,t]$ can be expressed as

$$A_t = r \int_0^t B_s ds,$$

with $A_t$ in steady state. Since we have assumed that all bursts contribute work at a constant rate, the mean of $A_t$ is

$$m(t) \triangleq E[A_t] = rt E[B_t] = rt \lambda E[d] = \frac{\lambda tr \delta}{\gamma - 1}. \quad (3.5)$$

In addition, the variance of $A_t$ can be obtained by repeatedly integrating the CCDF of the burst duration [141]:

$$\sigma^2(t) \triangleq \text{Var}[A_t] = 2 \lambda r^2 \int_0^t du \int_0^u dv \int_v^\infty P(d > x) dx. \quad (3.6)$$
Let \( f(v) \) be the first stage of the integration given in (3.6), we have

\[
f(v) = \int_v^\infty P(d > x) dx = \int_v^\infty \left(1 + \frac{x}{\delta}\right)^{-\gamma} dx = \delta \left[ \left(1 + \frac{x}{\delta}\right)^{1-\gamma} \right]_v^\infty = \frac{\delta}{\gamma - 1} (1 + \frac{v}{\delta})^{1-\gamma}.
\]

Let \( g(u) \) as the integral of

\[
f(v) = \frac{\delta(1 + \frac{v}{\delta})^{1-\gamma}}{\gamma - 1}, \quad v \geq 0,
\]

over the bounds required by (3.6), we have

\[
g(u) = \int_0^u f(v) dv = \frac{\delta}{\gamma - 1} \int_0^u \left(1 + \frac{v}{\delta}\right)^{1-\gamma} dv = \frac{\delta^2}{\gamma - 1} \left[ \left(1 + \frac{v}{\delta}\right)^{2-\gamma} \right]_0^u = \frac{\delta^2}{\gamma - 1} \left( \frac{(1 + \frac{u}{\delta})^{2-\gamma}}{2-\gamma} - \frac{1}{2-\gamma} \right) = \frac{\delta^2}{(\gamma - 1)(\gamma - 2)} \left(1 - (1 + \frac{u}{\delta})^{2-\gamma}\right).
\]

Then, we can express \( \sigma^2(t) \) as

\[
\sigma^2(t) = 2\lambda r^2 \int_0^t g(u) du = 2\lambda r^2 \int_0^t \frac{\delta^2}{(\gamma - 1)(\gamma - 2)} \left(1 - (1 + \frac{u}{\delta})^{2-\gamma}\right) du = \frac{2\lambda r^2 \delta^2}{(\gamma - 1)(\gamma - 2)} \int_0^t \left(1 - (1 + \frac{u}{\delta})^{2-\gamma}\right) du = \frac{2\lambda r^2 \delta^2}{(\gamma - 1)(\gamma - 2)} \left\{ t - \int_0^t (1 + \frac{u}{\delta})^{2-\gamma} du \right\}
\]
Finally, we obtain

\[
\sigma^2(t) = \frac{2\lambda r^2 \delta^3}{(\gamma - 1)(\gamma - 2)} \left( \frac{t}{\delta} - \frac{(1 + \frac{t}{\delta})^{3-\gamma}}{3-\gamma} + \frac{1}{3-\gamma} \right). 
\]  

(3.7)

From (3.7), we see that for large \( t \), the dominant term is

\[
\frac{2\lambda r^2 \delta^3 (1 + \frac{t}{\delta})^{3-\gamma}}{(\gamma - 1)(\gamma - 2)(\gamma - 3)}.
\]

Define

\[
H = \frac{3 - \gamma}{2},
\]

and observe that the growth of \( \sigma^2(t) \) is proportional to \( t^{2H} \). According to the properties of LRD shown in Section 2.1.1, this implies that for \( 1 < \gamma < 2 \), like PPBP, PLBP is LRD (asymptotically second-order self-similar) with Hurst parameter \( H \).

### 3.2 Performance Evaluation of the PLBP SSQ

Consider a single server queue (SSQ) with constant service rate, \( \mu \) [B/s], and an infinite buffer, fed by a PLBP input with parameters, \( \lambda, \gamma, \delta \) and \( r \), henceforth called PLBP SSQ, as shown in Figure 3.3.

An SSQ with constant service rate has been often used as a model of a statistical multiplexer used to multiplex variable bit rate (VBR) traffic from multiple sources into a single output (e.g. a wavelength). A related SSQ with constant service rate was used to study the performance of ATM buffers in [86]. Also, simulations of an SSQ with constant service rate fed by a VBR video trace were used in [10] to
explore the issue of resource allocation for statistically multiplexed video sources that revealed the LRD phenomenon of VBR video traffic. In this section we provide an algorithm for computing the buffer overflow probability of a PLBP SSQ by the Quasi-Stationary (QS) approximation. A satisfactory approximation was obtained for the PPBP SSQ in [95, 99]. Concise details of applying the QS idea are given here while more detailed justifications or concepts can be found in [95, 99]. In addition, the simulation method is introduced in Section 3.2.2.

3.2.1 The QS Approximation

The QS algorithm relies on an idea which was used in [136] to find the rate function for a large deviations characterization of multi-source heavy tailed on-off traffic as the number of sources increases. The idea is to separate the bursts into long and short bursts as shown in Figure 3.4. Given a time period, \( \tau \), those bursts existing for the entire period are identified as long bursts and the rest are considered to be short bursts, as shown in Figure 3.4. We separate PLBP into long and short burst processes, denoted: \( L_\tau \) and \( S_\tau \), respectively. Since the traffic contributed by each long burst is constant during the period \( \tau \), the performance of the PLBP SSQ is equivalent to that of a queue fed by \( S_\tau \) with service rate reduced by \( L_\tau \), which is equal to \( \mu - r\eta \). Given \( \eta \) and \( \tau \), we can obtain the CCDF of the steady state queue size \( Q_5 \), \( P(Q_5 > x) \), for this queue. Thus for a given \( \tau \), the overflow probability of a
Performance Evaluation of a PLBP Queue

PLBP SSQ, \( P_{\tau}(Q > x) \) is

\[
P_{\tau}(Q > x) = \sum_{n=0}^{\infty} P(\eta = n)P(Q_s > x),
\]

(3.8)

where \( \eta \) is the number of long bursts within the period.

[Diagram of PLBP Process]

Because the start-times for bursts form a Poisson process, and it is assumed that the duration of each burst is independent from all other aspects of the process, \( L_\tau \) is statistically independent from \( S_\tau \). The number of long bursts is Poisson distributed with mean \( \beta \), equal to the burst density (\( \lambda \)) times the probability that the forward recurrence time of the burst length distribution (Lomax) is longer than \( \tau \), which is

\[
\beta = \lambda E[d]P(\omega > \tau) = \frac{\lambda \delta}{\gamma - 1} \left( 1 + \frac{\tau}{\delta} \right)^{1-\gamma},
\]

by (3.4). Then \( P(\eta = n) \) can be expressed as

\[
P(\eta = n) = e^{-\beta} \frac{\beta^n}{n!}.
\]

(3.9)
3.2 Performance Evaluation of the PLBP SSQ

So, the mean and variance of the long bursts process \( (L_\tau) \) are

\[
m_L(t) \triangleq E[L_\tau] = rt\beta, \quad t < \tau, \tag{3.10}
\]

and

\[
\sigma_L^2(t) \triangleq \text{Var}[L_\tau] = r^2t^2\beta, \quad t < \tau. \tag{3.11}
\]

Now we consider the queue fed by the short burst process \( (S_\tau) \). The service rate of the queue, \( \mu_S \), is the original service rate reduced by the constant rate contributed by \( L_\tau \), thus \( \mu_S = \mu - r\eta \). Since \( S_\tau \) and \( L_\tau \) are independent, the mean/variance of \( S_\tau \) is equal to the mean/variance of PLBP reduced by the mean/variance of \( L_\tau \), giving

\[
m_S(t) \triangleq E[S_\tau] = m(t) - m_L(t) = m(t) - rt\beta, \tag{3.12}
\]

and

\[
\sigma_S^2(t) \triangleq \text{Var}[S_\tau] = \sigma^2(t) - \sigma_L^2(t) = \sigma^2(t) - r^2t^2\beta, \tag{3.13}
\]

for \( t < \tau \), where \( m(t) \) and \( \sigma^2(t) \) are given by (3.5) and (3.7). As in [95, 147], assuming that \( S_\tau \) is Gaussian, and using the approximation of [147], we obtain

\[
P(Q_S > x) \approx K \exp \left( -\frac{(x + (c - m_S(1))t^*)^2}{2\sigma^2(t^*)} \right), \tag{3.14}
\]

where \( K \) is chosen so that for \( x = 0 \), the right hand side in (3.14) is equal to a Gaussian estimate of the probability that \( \eta r \) exceeds the available capacity of the server, and \( t^* \) is a value of \( t (\geq 0) \) which minimizes the expression, \( (x + (c - m_S(1))t^*)^2 / 2\sigma^2(t^*) \).

Finally, we have the QS approximation for the overflow probability of the PLBP SSQ by exhausting all possible \( \tau \) for (3.8) to find a global maximum of \( P_\tau(Q > x) \), that is

\[
P(Q > x) \approx \sup_{\tau \geq 0} P_\tau(Q > x). \tag{3.15}
\]
3.2.2 A Fast Simulation Method

Instead of a straightforward slow simulation which is well-known to be time-consuming when it involves LRD processes, we adopt the fast simulation method provided by [95] for time efficiency. We now consider a discrete-time PLBP SSQ. The details of the queue has been provided in the previous section. By dividing the whole simulation time $T$ [s] into $M$ sampling intervals, we can obtain the amount of work buffered in the queue at the end of the $i$th interval by Lindley’s equation:

$$Q_i = \text{Max} \{ Q_{i-1} + \hat{A}_i - C, 0 \}, i = 1, 2, \ldots, M,$$

(3.16)

where $Q_0 = 0$ and $\hat{A}_i$ is the work contributed by PLBP within the $i$th interval. In this section, we will demonstrate the procedure of obtaining $Q_i$ by the fast simulation method.

**Conventional simulation method**

**Fast simulation method**

Fig. 3.5 Conventional simulation vs. fast simulation.

The only difference between the fast simulation and the slow simulation is that we assume there already exist $\eta$ long bursts when the simulation is initialized, as illustrated in Figure 3.5. By simulation, we can obtain the estimation of the overflow
3.3 Numerical Results

probability of the queue started with \( \eta \) long bursts, denoted: \( \hat{P}(Q_{(\eta,T)} > x) \). Then the overflow probability of the PLBP SSQ is

\[
P(Q > x) = \sum_{n=0}^{\infty} P(\eta = n) \hat{P}(Q_{(\eta,T)} > x), \tag{3.17}
\]

where \( P(\eta = n) \) is shown in (3.9). For practice, we set \( n \) going to \( N \) instead of \( \infty \); \( N \) is the smallest number that makes \( P(\eta = N) \leq \varepsilon \). Here, the tolerable error is set to be \( 10^{-9} \).

In sum, for a selected set of \( x \) values, \( X \), the fast simulation is accomplished by the following steps, starting with \( \eta = 0 \):

1. Start a thread of simulation with \( \eta \) initial long bursts, and total period, \( T \). Assign the short bursts to the queue. The number of short bursts follows a Poisson process with mean rate as \( \lambda E[d] - \beta \), and their durations are bounded Pareto FRT number with maximal value \( T \). Finally, feed the queue with normal PLBP input, monitor the queue size by (3.16), and calculating the probability of the occurrence for \( x \in X \) to obtain \( \hat{P}(Q_{(\eta,T)} > x) \).

2. Go to Step 3 if \( \eta = N \), else set \( \eta = \eta + 1 \) and go to 1).

3. Obtain \( P(Q > x), x \in X \), by (3.17).

The time to reach a consistent state for traditional simulation for a LRD queue is very long. The merit of the fast simulation method is that it reduces this time by initializing the system in a range of states. The details of Step 1 are provided in Appendix A.

3.3 Numerical Results

In this section, we validate the QS approximation by means of simulation results. Also, comparisons between the PLBP and the PPBP SSQs are presented. All
Table 3.1 Total times for obtaining $P(Q > x)$ for all selected $x$ values.

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approx.</td>
<td>0.2 s</td>
<td>0.3 s</td>
<td>0.4 s</td>
<td>0.6 s</td>
<td>0.7 s</td>
</tr>
<tr>
<td>Fast sim.</td>
<td>53 min</td>
<td>14 min</td>
<td>4 h</td>
<td>21 min</td>
<td>48 min</td>
</tr>
<tr>
<td>Slow sim.</td>
<td>93 min</td>
<td>46 min</td>
<td>6 days</td>
<td>33 min</td>
<td>2 h</td>
</tr>
</tbody>
</table>

Simulations results are provided with 95% confidence intervals based on Student’s t-distribution, which are presented as the coloured area in the figures.

In Figure 3.6, we compare the overflow probabilities of a PLBP SSQ obtained by the QS approximation with the simulation results from both the slow and fast simulation methods, for various cases involving a range of values for $H$, $r$ and $\mu$. Table 3.1 shows the total times required to obtain the overflow probabilities for the approximation and the simulations, for all $x$ values selected, in each of the six cases presented in Figure 3.6. Notice the advantage of the fast simulation in speed over the slow simulation, especially for $H = 0.85$. As expected, the approximation takes far less time, which does not vary much with $H$. The consistency of the approximation and the two simulation methods can be found in the plots. However, there exist discrepancies between the QS approximation and the simulations for some $x$ values. The approximation relies on modelling the short burst process as Gaussian. In cases where the short-burst process plays a major role in determining performance, this may lead to inaccuracy. This is probably the main cause of discrepancy between the QS approximation and simulation results. Observe that the results of the fast simulation, that do not rely on this assumption, are consistent with those of the slow simulation. However, for large $H$, both slow and fast simulations of LRD queues may require even longer (impractically long) run-times to achieve the required level of accuracy.

Figure 3.7 illustrates the differences in overflow probabilities obtained by PPBP and PLBP for the same fitted key parameters values. We set $\gamma = 1.5$ for both processes so that the Hurst parameter is given by $H = 0.75$. The plot shows two
3.3 Numerical Results

Fig. 3.6 Overflow probabilities based on QS approximation vs. simulation results for Case 1 to 6, with $\lambda = 10$ and $\delta = 1$ for all cases.

For each case, the parameters of both PPBP and PLBP are chosen to fit the same mean and variance as shown in Table 3.2. The capacities ($\mu$) of both queues are set as $m(1) + 3\sigma(1)$. The overflow probabilities for both processes are obtained by simulations. We see that the overflow probabilities of the PLBP SSQ are higher than those of the PPBP SSQ for both cases. The reason is that while there are more small bursts for PLBP, these extra small bursts must be balanced by more large bursts to fit the mean, compared to PPBP, and this leads to more overflows.
observe that the performance prediction results of PLBP can be significantly more conservative than PPBP. Recalling that the results in Figure 3.2 indicate that PLBP accurately represents measured traces, we propose PLBP as a useful tool in network performance modelling.

Table 3.2 Parameters of PPBP and PLBP.

<table>
<thead>
<tr>
<th>λ</th>
<th>Mean (m(1))</th>
<th>Variance (σ²(1))</th>
<th>PPBP</th>
<th>PLBP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>δ</td>
<td>r</td>
</tr>
<tr>
<td>1</td>
<td>1.836</td>
<td>1</td>
<td>1.537</td>
<td>0.597</td>
</tr>
<tr>
<td>10</td>
<td>5.82</td>
<td>1</td>
<td>1.545</td>
<td>0.188</td>
</tr>
</tbody>
</table>

Fig. 3.7 Overflow probabilities for PPBP and PLBP SSQs.

3.4 Summary

In this chapter, we have motivated the use of PLBP through demonstrating that Lomax overcomes a weakness of Pareto in better representing short bursts while
3.4 Summary

retaining similar heavy tail behavior, by real traces. Then, we provided an analytical approximation and a fast simulation method for evaluating the overflow probability of a PLBP SSQ. In all the cases considered, the accuracy of the fast simulation method was demonstrated by showing that its results are very close to those obtained by slow simulations. We have observed some inaccuracies of the approximation which were explained by its Gaussian assumption of the short bursts. One possible way to improve the QS approximation is to find a method to seek a proper \( \tau \) that can make the short burst process close to an existing model, which requires further work.
Chapter 4

Performance Evaluation of an FBm Queue

It has been well established that Internet traffic is long range dependent (LRD) [9, 11, 17–19, 159, 160]. Furthermore, core and metropolitan Internet links are shared by a large number of users, so, by the central limit theorem, the traffic on such links, which represents the multiplexing of traffic generated by many users, can be assumed to follow a Gaussian process for the purpose of performance evaluation and link capacity dimensioning [97]. A Gaussian self-similar process, the fractional Brownian motion (fBm), has been widely considered as the model of choice for heavily multiplexed LRD traffic for its simplicity and accuracy. The fBm process has been validated, as an accurate model, for such traffic using real measurements, e.g., see [9, 25, 102, 161].

In addition, Internet traffic is transported based on the store and forward principle, where packets are stored in router buffers before they are forwarded towards their destination. Therefore, a queue fed by fBm input has been considered a fundamentally important model for Internet queueing performance analysis and capacity assignment, and has attracted significant attention [34, 100, 102, 140, 142–144, 146, 161–164]. However, to date, despite considerable effort, only asymptotic results are available
for the queueing performance of fBm queues. Accordingly, it is important to obtain simple and accurate analytical results for fBm queueing performance and service provisioning. In addition, even though they are only asymptotically accurate, the best results for the performance of a queue fed by fBm available up to now take a form which is not sufficiently explicit to be readily applied for dimensioning. The approximation presented here is accurate and both mathematically and numerically simple.

In this chapter, a neat and accurate closed-form approximation for the stationary workload distribution of a single server queue (SSQ) fed by an fBm input is provided. The queueing problem of fBm is difficult because it does not lend itself to a classical Markov chain analysis. Therefore new methods have had to be developed for the analysis of an fBm queue. We derive a new formula arising from a new interpretation of an asymptotically accurate formula for the tail of the distribution derived in [34], the details of which are shown in Section 4.1. To verify the accuracy of our closed-form approximation, there is a need to simulate fBm queues. There are two difficulties associated with such simulations. The first difficulty is the adaption of the continuous-time concept of fBm to the discrete-time implementation of a computer simulation. The second one is the large execution time required for the generation of long fBm sequences. We overcome these difficulties by a new simulation approach that re-uses a single fBm sequence, as described in Section 4.2. Then in Section 4.3, we show the consistency between the simulation and analytical results. At last, we provide a link between the fBm queue size distribution and the Generalized Gamma distribution [165] (a special case of the Amoroso distribution [166]) which leads to very accurate closed-form approximations for important queueing statistics of an fBm queue, namely the mean, the variance, the second central moment, and the skewness. The approximations are validated by simulation results. Also, simplified expressions are provided for these results in certain cases when the Hurst parameter takes certain special values.
4.1 A New Analytical Result for an FBm Queue

We consider a single server queue with an infinite buffer fed by an fBm input process, \( X = \{X(t), t \geq 0\} \), with Hurst parameter \( H \), variance \( \sigma^2 \) and drift \( m \), as shown in Figure 4.1. Specifically, the variables \( \sigma^2 \) and \( m \) are the variance and the mean of the amount of work arriving during a time interval of length 1. Let \( X^{(j)} = \{X_i^{(j)}, i = 1, 2, \ldots\} \) be the aggregated process of \( X \) at level \( j > 0 \), that is

\[
X_i^{(j)} = \int_{j(i-1)}^{ji} X(t)dt, \ i = 1, 2, \ldots
\]

The mean and variance of \( X^{(m)} \) are denoted \( m(j) = E[X_i^{(j)}] \) and \( \sigma^2(j) = Var[X_i^{(j)}] \) for all \( i = 1, 2, \ldots \). Thus, \( m = m(1) \) and \( \sigma^2 = \sigma^2(1) \). The service rate, denoted by \( \mu \) [bit/sec.], is assumed constant. Let \( \iota \) be the mean net input during a time interval of length 1, i.e., \( \iota = m - \mu \). For stability we assume \( \iota < 0 \). Henceforth, a time interval of length 1 will be called 1 second [sec.]. Let \( Q \) denote the steady state queue size. Because fBm is a continuous-time process we can not define \( Q \) as the limit of a quantity defined by Lindley’s equation; however, making use of the formula of Reich [167], we can define

\[
Q = \sup_{i \geq 0} (X(t) - \mu t).
\]
For the case $H = 0.5$, the complementary cumulative distribution function (CCDF) of $Q$, namely, the probability that the fBm queue size exceeds $x$, denoted $P(Q > x)$, is well known [168] and is given by

$$P(Q > x) = \exp \frac{2t}{\sigma^2} x, \quad x \geq 0. \quad (4.1)$$

For $H \neq 0.5$, despite considerable efforts, there is no exact result for $P(Q > x)$.

Of the known results, a first approximation/bound for the CCDF was given in [100] as

$$P(Q > x) \approx \exp \left( - \frac{x^{2-2H}(1-H)2H-2|t|2H}{2H2H\sigma^2} \right), \quad x \geq 0. \quad (4.2)$$

In [140], the authors showed that (4.2) holds in the sense that

$$\lim_{x \to \infty} \frac{1}{x} \log \left( \frac{\text{LHS}}{\text{RHS}} \right) = 0. \quad (4.3)$$

A more precise approximation results from [34, Theorem 1, Equation (9)] with $\alpha = 2H$, $\beta = 1$, together with the asymptotic approximation for the tail of a Normal distribution,

$$1 - \Phi(x) \sim x^{-1} \exp \left( - \frac{x^2}{2} \right),$$

giving

$$P(Q > x) \approx Cx^{\frac{2H^2-3H+1}{2H}} e^{-\frac{x^{2-2H}(1-H)2H-2|t|2H}{2H2H\sigma^2}}, \quad x \geq 0, \quad (4.4)$$

where $C$ is a certain constant. This approximation holds in the sense that

$$\lim_{x \to \infty} \frac{\text{LHS}}{\text{RHS}} = 1. \quad (4.5)$$

No method for determining $C$ was provided in [34] although there is an extensive literature connecting $C$ with Pickands’ constants, and methods for bounding and
approximating this constant (which depends on $H$) are available [169]. In Section 4.3 we estimate this constant by numerically fitting (4.4) to simulation results for the purpose of comparing it with our analytical formula.

While the objective in [34] was to obtain a function that is accurate in the limit of $x \to \infty$, namely satisfies (4.5), our objective is to obtain an accurate approximation for $P(Q > x)$ for all values of $x$. If the limiting behaviour of the queueing system for $x \to \infty$ is actually typical even for moderate and small $x$, and if (4.4) provides a good way to characterise this limiting behaviour, then we can use (4.4) for our objective as a good approximation for the behaviour of an fBm queue.

However, the particular form of (4.4) does not lend itself to this strategy because if $0.5 < H < 1$, which is usually the case, the RHS of (4.4) $\to \infty$ as $x \to 0$, whereas ideally it should tend to the limit 1. Hence, as an approximation, the formula (4.4) is certain to be inaccurate for small $x$.

In this thesis we revise this approach by supposing that it is not the CCDF, but rather the density whose character remains stable for $x$ near $\infty$ and is characterised according to (4.4). Note, by Theorem 11.11 in [170], the stationary waiting time distribution of an fBm queue has a density, for all $0 < H < 1$, except for one possible atom, which must occur at the minimal value for which the buffer level has non-zero probability density. Let

$$P(Q > u) \approx \int_u^\infty cf(x)dx.$$  \hspace{1cm} (4.6)

in the sense of (4.3), where $c$ is a certain constant provided below in (4.8) and $f$ is a function of the form

$$f(x) = x^\kappa e^{\left(-\frac{x^{2-2H(1-H)2H-2(1-2H)2H}}{2H^{2H}2^H\sigma^2}\right)}.$$ \hspace{1cm} (4.7)
By taking a derivative of the RHS of (4.4):

\[
\frac{d}{dx} \left( C x^{2H^2 - 3H + 1} e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \right)
\]

\[
= C e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \frac{d}{dx} \left( x^{2H^2 - 3H + 1} \right) e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)}
\]

\[
= C e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \frac{2H^2 - 3H + 1}{H} x^{2H^2 - 3H + 1} - e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \frac{2H^2 - 3H + 1}{H} x^{2H^2 - 3H + 1} 
\]

\[
= C e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \frac{2H^2 - 3H + 1}{H} x^{2H^2 - 3H + 1} + C e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \frac{2H^2 - 3H + 1}{H} x^{2H^2 - 3H + 1}
\]

we find \( \kappa = \frac{1 - 2H}{H} \). The derivative has another term involving \( x^{2H^2 - 4H + 1} \), but this takes the same form but with a smaller exponent for the initial power of \( x \) since \( \frac{2H^2 - 3H + 1}{H} - 1 < \frac{1 - 2H}{H} = \frac{2H^2 - 3H + 1}{H} - 1 + 2 - 2H \), for \( 0 < H < 1 \); omitting this term will not invalidate (4.3). Since we seek a function of the form specified in (4.7) (not a sum of functions each of this form) we must omit the term with the lower exponent of \( x \). For the same reason, we also omit the term \( C e^{\left( -\frac{\sigma^2 H (1-H)^2}{2} + \frac{\sigma^2 H^2}{2} \right)} \frac{2H^2 - 3H + 1}{H} x^{2H^2 - 3H + 1} \). Hence our approximation, (4.6), holds in the sense that

\[
\lim_{x \to \infty} \frac{\text{LHS}}{\text{RHS}} = B,
\]

where \( B \) is a certain constant. The approximations (4.6) and (4.4) are not the same since we omit some terms when deriving (4.7). When \( H = 0.5 \), (4.6) has the same result as (4.1).

We are not able to judge whether the assumption that the true density takes a form asymptotically like (4.7) is superior or inferior to the assumption that \( P(Q > x) \) takes a form like (4.4). Which of the two assumptions is closer to the exact solution
will, in general, vary from case to case. We present evidence below, however, which suggests that (4.7) provides a more accurate approximation than (4.4) in the present instance.

The density \( cf(x) \) has a finite integral for all values of \( H \); this overcomes the problem identified above in the use of (4.4) as an approximation. We can find \( c \) from

\[
c^{-1} = \int_0^\infty f(x)dx = \int_0^\infty x^{H-1}e^{-\alpha x^\nu}dx = \nu^{-1}\alpha^{-\frac{\beta}{\nu}}\Gamma\left(\frac{\beta}{\nu}\right),
\]

(4.8)

where

\[
\alpha = \frac{(1-H)^{2H-2}|t|^{2H}}{2H^2H^2},
\]

\[
\beta = \frac{1-H}{H}, \quad \nu = 2-2H \quad \text{and} \quad \Gamma \text{ denotes the Gamma function.}
\]

It is feasible that (4.6) is accurate for small \( x \), and perhaps for all \( x \). The simulation results do appear to confirm that this is the case. Another benefit of (4.6) over (4.4) is that we have a simple formula for \( c \) whereas [34] does not provide a way to derive \( C \). The new formula is simple to calculate and work with, has no unknown parameters, is asymptotically accurate in the sense of (4.3) rather than (4.5) but is potentially accurate over the full range of parameter values. Simulation tests have shown that this is, in fact, the case.

We propose this as a way to characterise the asymptotic behaviour of the buffer level distribution, for an fBm queue, which may have more inherent validity than characterising the behaviour of the model by its complementary cumulative distribution function. The fact that this regime appears to hold over a wide range of values, including the range where we normally wish to evaluate the model, is shown below by means of simulations.
4.2 A Sequence of Simulations of an FBm Queue

There are two challenges facing us in simulating an fBm queue: (1) how to adapt the continuous-time concept of fBm to the discrete-time implementation of a computer simulation that considers the queue size at the endpoints of consecutive intervals each of size $\Delta t$ [sec.], (2) the generation of long fBm sequences requires a great deal of computer time which becomes prohibitive beyond a certain sequence length. The first challenge forces us to consider the queueing performance limit as we perform a sequence of simulations where $\Delta t$ becomes smaller and smaller. The second challenge can be met if we are able to carry out this sequence of simulations with a single fBm sequence, because this will enormously reduce the computation time required.

There is a significant body of work on how to generate fBm traffic [44, 105, 108, 171, 172]. We use the sequence of its incremental process, fractional Brownian noise (fBn), obtained by the Hosking recursive method using the code of Dieker [173]. We re-use this same fBn sequence for each simulation, making use of the self-similarity of the fBm process to enable us to re-interpret the original sequence, after making an appropriate transformation, as a sequence of finer and finer views of the same process. In this way we can be confident that our results are not artefacts of the limited resolution of our simulation.

In addition, a check has also been made that the results were not overly dependent on the length of the simulations by repeating the whole process with a longer original fBn sequence.

For each scenario (i.e., a given $H$ value), we use one fBn sequence, $\{U_n\}_{n=0}^N$, of length $N = 2^{23}$ samples. The sequence for one scenario is independent of the sequences used for the others. The sequence $\{U_n\}_{n=0}^N$ is characterized by its Hurst parameter, $H$, and the variance $\nu_1$ of $U_1$. By definition, $E(U_1) = 0$. We interpret the time between samples as corresponding to a certain time interval, $\Delta t$, which
4.2 A Sequence of Simulations of an FBm Queue

we successively reduce while we re-use the same sequence, to represent the same process at a finer level of detail.

The basic algorithm of our discrete-time queue simulation is *Lindley’s equation*,

\[ Q_{n+1} = \max(0, Q_n + \hat{U}_n), \quad n = 0, 1, \ldots, N, \tag{4.9} \]

where \( Q_n \) denotes the queue length at the end of the \( n \)th time-interval and \( \hat{U}_n \) is the amount of work arriving in each interval in the discrete-time approximation to the continuous time model.

We now discuss how one sequence of standard fBn can be used to generate a whole family of different discrete-time approximations to the continuous time model, for different choices of the sampling interval. For this purpose it is sufficient to set \( \hat{U}_n = s(\Delta t) U_n + m(\Delta t) \) where \( s(\Delta t) \) and \( m(\Delta t) \) are chosen so that \( \hat{U}_n \) has the appropriate mean and variance. The Hurst parameter of \( \{\hat{U}_n\} \) will then be the same as that of \( \{U_n\} \).

According to [9], the variance of work arriving in an interval of length \( \Delta t \) should be \( \sigma^2 (\Delta t)^{2H} \), but the variance of \( U_1 \) is \( v_1 \), so \( s(\Delta t)^2 v_1 = \sigma^2 (\Delta t)^{2H} \); therefore

\[ s(\Delta t) = \frac{(\Delta t)^H \sigma}{\sqrt{v_1}}. \tag{4.10} \]

Similarly, the work arriving in an interval of length \( \Delta t \) should be \( t \Delta t \) so

\[ m(\Delta t) = t \Delta t. \tag{4.11} \]

Thus, using (4.10) and (4.11) to derive the family of sequences \( \{\hat{U}_n\} \) from a single fBn sequence \( \{U_n\} \), we are able to simulate the fBm queue with a range of different sampling intervals with no need to generate more than one fBn sequence. This approach is used in the simulations in the next section to ensure that the
discretisation of time is not compromising the accuracy of the simulations. The C codes for the simulations are provided in Appendix B

4.3 Validation of the Workload Distribution

In this section, we present numerical results that demonstrate the accuracy of both our simulation and approximation. All simulation results are provided with 95% confidence intervals based on the Student’s t-distribution, estimating the standard deviation by dividing the simulation into blocks.

4.3.1 Validation of the Simulation

To validate the simulation, in Figure 4.2 using the known exact result (4.1) for the case $H = 0.5$, we demonstrate how the sequence of simulations described above approach as the exact result using only one fBm sequence. We note that in the case $\Delta t = 0.01$, the analytical results are within the simulation confidence intervals.

4.3.2 Validation of the Analytical Formula

In Figures 4.3 – 4.7, we present results for $H = 0.3, 0.4, 0.6, 0.7$ and 0.8, respectively, to demonstrate the accuracy and robustness of our approximation. In most instances the approximation is within the confidence intervals of the simulation over the entire range and in all instances the existing discrepancies are quite small. As expected, the existing asymptotics, of Norros (4.2) and Hustler-Piterbarg (4.4), are not accurate for the full range of parameters in all cases.
4.3 Validation of the Workload Distribution

Fig. 4.2 Overflow probabilities based on exact formula vs. simulation results for $H = 0.5, \sigma^2 = 1, t = -0.5$ with $\Delta t = 1, 0.1$ and 0.01.

Fig. 4.3 Overflow probabilities based on the asymptotics of Norros [100] and Hustler-Piterbarg [34], and our approximation vs. simulation results for $H = 0.3, \sigma^2 = 1, t = -0.5$. 
Fig. 4.4 Overflow probabilities based on the asymptotics of Norros [100] and Hustler-Piterbarg [34], and our approximation vs. simulation results for $H = 0.4, \sigma^2 = 1, t = -0.5$.

Fig. 4.5 Overflow probabilities based on the asymptotics of Norros [100] and Hustler-Piterbarg [34], and our approximation vs. simulation results for $H = 0.6, \sigma^2 = 1, t = -0.5$. 
4.3 Validation of the Workload Distribution

![Graph showing overflow probabilities based on asymptotics of Norros and Hustler-Piterbarg, and our approximation vs. simulation results for $H = 0.7, \sigma^2 = 1, t = -0.5$.]

Fig. 4.6 Overflow probabilities based on the asymptotics of Norros [100] and Hustler-Piterbarg [34], and our approximation vs. simulation results for $H = 0.7, \sigma^2 = 1, t = -0.5$.

### 4.3.3 Discussion of Results

First of all, it should be observed that the formula (4.6) presented here, based on comparison with simulation, appears to be more accurate than the existing alternative formulae in all the situations which have been tested, up to now.

Second, the new formula displays good accuracy for the full range of parameters.

Third, although it might appear that the Hustler-Piterbarg asymptotics [34] also provides adequate accuracy over the full practical range of the parameters (for our purposes, for example, its weakness when $H$ and $x$ are small is not practically important), this formula cannot readily be used independently of simulations because it requires a constant for which there is no clear method of calculation. In order to use the Hustler-Piterbarg asymptotics in the above comparisons, this constant was estimated by simulations, as discussed in Section 4.1.
4.4 Statistics of an FBm Queue

The mean, the variance, the third central moments, and the skewness are four vital important statistics for queueing analysis and link dimensioning. In this section, we provide approximations for these four queueing statistics of an fBm queue by linking the probability density function of the steady-state queue size, $Q$ and the Generalised Gamma distribution.

4.4.1 The Model and Related Work

As shown in Section 4.1, we consider a single server queue with constant service rate, $\mu$ [B/s], fed by an fBm input process with Hurst parameter $H$, mean input rate $m$ [B/s], and variance $\sigma^2$. Introducing the mean net input, $\iota = m - \mu$, we can characterize the fBm queueing model by three parameters: $H, \iota$ and $\sigma^2$. By (4.6),

![Graph showing overflow probabilities based on the asymptotics of Norros and Hustler-Piterbarg, and our approximation vs. simulation results for $H = 0.8, \sigma^2 = 1, \iota = -0.5$.]
(4.7) and (4.8), the probability density function of $Q$ can be expressed as

$$
P(Q \in (x,x+dx)) \approx \frac{\nu \alpha^{\frac{\beta}{\nu}}}{\Gamma\left(\frac{\beta}{\nu}\right)} x^{\beta - 1} e^{-\alpha x^{\frac{\nu}{\beta}}},$$

(4.12)

where

$$\alpha = \frac{(1-H)^{2H-2}|t|^{2H}}{2H^2 \sigma^2},$$

(4.13)

$$\beta = \frac{1}{H} - 1,$$

and

$$\nu = 2(1-H).$$

Observe that this approximation to the fBm queue size distribution is known as the Generalised Gamma distribution. The Generalised Gamma distribution is a special case of the Amoroso distribution [166, 174, 175] used originally to model income rates. The Amoroso distribution admits the density

$$f_X(x; a, d, g, p) = \frac{p (x-g)^{d-1} \exp\left(-\frac{(x-g)^p}{a}\right)}{a^d \Gamma\left(\frac{d}{p}\right)}, a, d, p, g > 0, x \geq g,$$

(4.14)

where $\Gamma(\cdot)$ denotes the gamma function.

Stacy [165] defined the Generalised Gamma distribution as the special case of the Amoroso distribution where $g = 0$, and identified the cumulative distribution function (CDF) and moment-generating function (MGF) as

$$F(x; a, d, p) = \frac{\gamma\left(\frac{d}{p}, \left(\frac{x}{a}\right)^p\right)}{\Gamma\left(\frac{d}{p}\right)}, a, d, p, x > 0$$

(4.15)

and

$$M(t; a, d, p) = \sum_{k=0}^{\infty} \frac{a^k t^k}{k!} \frac{\Gamma\left(\frac{d+k}{p}\right)}{\Gamma\left(\frac{d}{p}\right)}, a, d, p, x > 0$$

(4.16)
respectively, where $\gamma(\cdot)$ in (4.15) denotes the lower incomplete gamma function, i.e.,

$$\gamma(s,x) = \int_0^x t^{s-1} e^{-t} dt. \quad (4.17)$$

The mean and variance are

$$E[Q] = a \frac{\Gamma((d+1)/p)}{\Gamma(d/p)}, \quad (4.18)$$

and

$$\text{Var}[Q] = a^2 \left[ \frac{\Gamma((d+2)/p)}{\Gamma(d/p)} - \left( \frac{\Gamma((d+1)/p)}{\Gamma(d/p)} \right)^2 \right], \quad (4.19)$$

respectively [174]. From the MGF, the third central moment of $Q$ can be derived, and is given by

$$E[(Q - E[Q])^3] = E[Q^3] - 3E[Q] \text{Var}[Q] + 2(E[Q])^3 = a^3 \frac{\Gamma(d+3/p)}{\Gamma(d/p)} + 2 \left( a \frac{\Gamma((d+1)/p)}{\Gamma(d/p)} \right)^3$$

$$- \frac{3a^3 \Gamma(d+1/p) \Gamma(d+2/p)}{\Gamma^2(d/p)}. \quad (4.20)$$

Substituting $g = 0$, $d = \beta$, $p = \nu$ and $a = \alpha^{-1/\nu}$ in (4.14) we obtain (4.12), so it is a Generalised Gamma density.

### 4.4.2 The Mean, Variance, Third Central Moment and Skewness

Replacing $d$, $p$ and $a$ with their expressions in terms of $\alpha$ and $H$, we find:

$$E[Q] \approx \alpha^{-1/2H} \frac{\Gamma\left(\frac{1-H}{2H}+\frac{1}{2}\right)}{\Gamma\left(\frac{1-H}{H(2-2H)}\right)} = \alpha^{-1/2H} \frac{\Gamma\left(\frac{1}{2H(1-H)}\right)}{\Gamma\left(\frac{1}{2H}\right)}, \quad (4.21)$$
4.4 Statistics of an FBm Queue

\[ \text{Var}[Q] \approx \frac{\alpha^{\frac{1}{2H}}}{\Gamma\left(\frac{1}{2H}\right)} \left[ \Gamma\left(\frac{1+H}{2H(1-H)}\right) - \frac{\Gamma^2\left(\frac{1}{2H}\right)}{\Gamma\left(\frac{1}{2H}\right)} \right], \quad \text{(4.22)} \]

and

\[ \mathbb{E}[(Q - \mathbb{E}[Q])^3] \approx \alpha^{\frac{3}{2H-2}} \left[ \frac{\Gamma\left(\frac{2H+1}{2H(1-H)}\right)}{\Gamma\left(\frac{1}{2H}\right)} + \frac{2\Gamma^3\left(\frac{1}{2H}\right)}{\Gamma^3\left(\frac{1}{2H}\right)} \right. \]
\[ \left. - \frac{3\Gamma\left(\frac{H+1}{2H(1-H)}\right)\Gamma\left(\frac{1}{2H}\right)}{\Gamma^2\left(\frac{1}{2H}\right)} \right]. \quad \text{(4.23)} \]

The skewness of \( Q \), can be expressed as

\[ \text{Skewness}[Q] = \frac{\mathbb{E}[(Q - \mathbb{E}[Q])^3]}{(\text{Var}[Q])^{3/2}}. \quad \text{(4.24)} \]

By (4.22) and (4.23), \( \alpha \) is canceled out in (4.24), so \( \text{Skewness}[Q] \) is a function of only one parameter, \( H \).

We can also obtain the mean and the variance of \( Q \) directly from (4.12). The mean of \( Q \) can be expressed as

\[ \mathbb{E}[Q] = \int_0^\infty xP(Q \in (x, x+dx)) \]
\[ \approx c \int_0^\infty xf(x)dx \]
\[ = c \int_0^\infty x\beta e^{-\alpha x^\nu} dx \]
\[ = c \cdot \nu^{-1} \alpha^{-\frac{\beta+1}{\nu}} \Gamma\left(\frac{\beta+1}{\nu}\right), \]

since \( c \) is determined by (4.8), we have

\[ \mathbb{E}[Q] = \frac{\nu \alpha^\nu}{\Gamma\left(\frac{\beta}{\nu}\right)} \nu^{-1} \alpha^{-\frac{\beta+1}{\nu}} \Gamma\left(\frac{\beta+1}{\nu}\right) \]
Performance Evaluation of an FBm Queue

\[ = \alpha^{\frac{1}{\nu}} \frac{\Gamma\left(\frac{\beta+1}{\nu}\right)}{\Gamma\left(\frac{\beta}{\nu}\right)}, \tag{4.25} \]

as \( \beta = 1/H - 1 \) and \( \nu = 2(1 - H) \),

\[ \mathbb{E}[Q] = \alpha^{\frac{1}{2-H}} \frac{\Gamma\left(\frac{(1-H)/H+1}{2-2H}\right)}{\Gamma\left(\frac{1-H}{H(2-2H)}\right)} \]
\[ = \alpha^{\frac{1}{2-H}} \frac{\Gamma\left(\frac{1}{2H(1-H)}\right)}{\Gamma\left(\frac{1}{2H}\right)}. \tag{4.26} \]

In the same way, we can obtain the second moment of \( Q \):

\[ \mathbb{E}[Q^2] = \int_0^\infty x^2 \mathbb{P}(Q \in (x,x+dx)) \]
\[ \approx c \int_0^\infty x^2 f(x) dx \]
\[ = c \cdot \nu^{-1} \alpha^{-\frac{\beta+2}{\nu}} \Gamma\left(\frac{\beta+2}{\nu}\right) \]
\[ = \alpha^{\frac{2}{\nu}} \frac{\Gamma\left(\frac{\beta+2}{\nu}\right)}{\Gamma\left(\frac{\beta}{\nu}\right)}, \tag{4.27} \]

as \( \beta = 1/H - 1 \) and \( \nu = 2(1 - H) \),

\[ \mathbb{E}[Q^2] = \alpha^{\frac{2}{2-H}} \frac{\Gamma\left(\frac{(1-H)/H+2}{2-2H}\right)}{\Gamma\left(\frac{1-H}{H(2-2H)}\right)} \]
\[ = \alpha^{\frac{1}{2-H}} \frac{\Gamma\left(\frac{1+H}{2H(1-H)}\right)}{\Gamma\left(\frac{1}{2H}\right)}. \tag{4.28} \]

The variance of \( Q \) can be obtained via (4.25) and (4.27) in the form

\[ \text{Var}[Q] = \mathbb{E}[Q^2] - (\mathbb{E}[Q])^2 \]

74
4.4 Statistics of an FBm Queue

\[
\approx \alpha^{2 \frac{\beta + 2}{\nu}} \Gamma\left(\frac{\beta + 2}{\nu}\right) - \left(\alpha^{\frac{\beta + 1}{\nu}} \Gamma\left(\frac{\beta + 1}{\nu}\right)\right)^2
\]

\[
= \alpha^{2 \frac{\beta + 2}{\nu}} \Gamma\left(\frac{\beta + 2}{\nu}\right) \left(\Gamma\left(\frac{\beta + 2}{\nu}\right) - \Gamma^2\left(\frac{\beta + 1}{\nu}\right)\right)
\]

as \(\beta = 1/H - 1\) and \(\nu = 2(1 - H)\),

\[
\text{Var}[Q] = \alpha^{2 \frac{1}{2H}} \Gamma\left(\frac{1}{2H}\right) \left(\Gamma\left(\frac{1 + H}{2H(1 - H)}\right) - \Gamma^2\left(\frac{1}{2H}\right)\right)
\]

(4.29)

The consistency between (4.21) and (4.26) ((4.22) and (4.29)) can be found.

4.4.3 Validation

We validate our mean, variance, third central moment and skewness results by simulations of fBm queues with \(\sigma = 1\) and different values of \(H\) and \(\tau\). We have used the simulation method described in [104]. The simulation results for \(E[Q]\), \(\text{Var}[Q]\), \(E[(Q - E[Q])^3]\) and \(\text{Skewness}[Q]\) are compared with the expressions from (4.21) - (4.24), as shown in Figure 4.8 - 4.11, where the 95% confidence intervals based on Student’s t-distribution are presented as the coloured area. The agreement between our analytical results and the simulation results can be observed in the figure. The setting \(\sigma = 1\) is without loss of generality because \(\sigma\) can be viewed as a scale parameter. For example, for any value of \(\sigma\), using \(\tau\) and \(\sigma\) in units of \(\sigma\) (i.e., dividing both by \(\sigma\)) will give the scaled mean and variance values as presented in figure 4.8 and 4.9 (i.e., the mean divided by \(\sigma\) and the variance divided by \(\sigma^2\)). Note that \(H\) is not affected by the unit size.
4.4.4 Simplifications of $E[Q]$ and $\text{Var}[Q]$ for Certain $H$ Values

In the Brownian case of $H = 0.5$, where $\sigma^2 = m$ as in a Poisson process, (4.13) reduces to

$$\alpha = \frac{2|t|}{\sigma^2}.$$ 

Since, we have

$$\Gamma\left(\frac{1}{2H}\right) = \Gamma(1) = 1,$$

$$\Gamma\left(\frac{1}{2H(1-H)}\right) = \Gamma(2) = 1,$$

and

$$\Gamma\left(\frac{1+H}{2H(1-H)}\right) = \Gamma(3) = 2,$$

(4.21) and (4.22) can be further simplified as

$$E[Q] = \left(\frac{2|t|}{\sigma^2}\right)^{-1} = \frac{\sigma^2}{2|t|},$$

(4.30)
and

$$\text{Var}[Q] = \left( \frac{2|\iota|}{\sigma^2} \right)^{-2} \left( 2 - \frac{1^2}{1} \right) = \left( \frac{\sigma^2}{2|\iota|} \right)^2 = (E[Q])^2. \quad (4.31)$$

An alternative way to obtain $E[Q]$ and $\text{Var}[Q]$ for the case $H = 0.5$ is to use the exact solution for $P(Q > x)$ from [168], which is

$$P(Q > x) = e^{\frac{2}{\sigma^2}x}. $$

Then the mean and variance of $Q$ are obtained by

$$E[Q] = \int_0^{\infty} P(Q > x) dx$$

$$= \frac{\sigma^2}{2\iota} \left. \frac{2e^{2/\sigma^2 x}}{2^2/\sigma^2} \right|_{0}^{\infty}$$

which, given $t < 0$

$$= \frac{\sigma^2}{2|\iota|^3}$$
Fig. 4.10 Approximations vs. simulation results for $E[(Q - E[Q])^3]$ with $\sigma = 1$.

and

$$\text{Var}[Q] = 2 \int_0^\infty x P(Q > x) dx - \left( \int_0^\infty P(Q > x) dx \right)^2$$

$$= 2 \left( \frac{\sigma_1^2}{2t} \right)^2 \left| e^{\frac{2t}{\sigma_1^2}x} (\frac{2t}{\sigma_1^2}x - 1) \right|_0^\infty - \left( \frac{\sigma_1^2}{2|t|} \right)^2$$

which, given $t < 0$

$$= 2 \left( \frac{\sigma_1^2}{2|t|} \right)^2 - \left( \frac{\sigma_1^2}{2|t|} \right)^2$$

$$= \left( \frac{\sigma_1^2}{2|t|} \right)^2.$$

They are consistent with (4.30) and (4.31). Since $|t| = C - m$, we can express $E[Q]$ with the help of (4.30) as

$$E[Q] = \frac{\sigma^2}{2(C - m)},$$

which is half of that for the equivalent M/M/1 queueing system, where $\sigma^2 = m$ and $E[Q] = m/(C - m)$. It is also equal to the mean of an equivalent M/D/1 queue under
heavy traffic (where the utilization approaches 1) based on the Pollaczeck Khintchine Formula.

When \( H = 1 - \frac{1}{2n} \), for \( n = 1, 2, \ldots \), we have \( \frac{1}{2H(1-H)} = \frac{1}{2H} + n \), and

\[
\Gamma\left( \frac{1}{2H(1-H)} \right) = \Gamma\left( \frac{1}{2H} + n \right) = \left( n - 1 + \frac{1}{2H} \right) \left( n - 2 + \frac{1}{2H} \right) \cdots \left( 1 + \frac{1}{2H} \right) \frac{1}{2H} \Gamma\left( \frac{1}{2H} \right).
\]

Then, (4.21) can be expressed without the Gamma function as

\[
E[Q] = \alpha \frac{\pi^{-\gamma}}{2} \left[ \frac{1}{2(1-H)} - 1 + \frac{1}{2H} \right] \left[ \frac{1}{2(1-H)} - 2 + \frac{1}{2H} \right] \cdots \left( 1 + \frac{1}{2H} \right) \frac{1}{2H}.
\]

(4.32)
4.5 Summary

In this chapter, we have considered a queue fed by an fBm input and derived new results for queueing performance. We have also described an efficient approach to simulate such a queue. Agreement between the analytical and the simulation results was demonstrated, and a numerical comparison with existing asymptotics was presented. In addition, we have established, for the first time, the link between the fBm queue length distribution and the Amoroso distribution (or its special case of the Generalized Gamma distribution). This adds an important application to the long list of applications of the Amoroso distribution [176–178]. This link has provided new closed-form approximations for the mean, variance, third central moment and skewness of an fBm queue which have been validated by simulations. We have also provided simplified expressions for the mean and variance for a range of cases.
Chapter 5

Service rate provisioning of an FBm Queue

In Chapter 4, a closed-form approximation for the stationary workload distribution of a single server queue (SSQ) fed by an fractional Brownian motion (fBm) input was provided. We will now turn to apply the approximation to solve a practical problem of network design, namely the link dimensioning problem. An important prerequisite is confirming that fBm is suitable for modeling Internet traffic, which may not always be the case.

Despite its appeal, the fBm process, as a traffic model, has limitations due to its characteristics and features. As mentioned in the preceding chapter, the fBm model is applicable to situations where traffic is highly aggregated. However, fBm exhibits the feature of negative traffic (periods of time during which the total bits arriving is negative), which is not a property of real Internet traffic. For certain ranges of the parameters, the impact of this negative traffic is insignificant; for certain other parameter values, it is not. It is important to investigate this feature of the fBm model, to understand how it affects the queueing performance predicted by the fBm model, and to determine whether it causes errors in the application of fBm to modelling the performance of real networks.
In Section 5.1, we compare the queueing performance of the fBm model with that of its truncated counterpart via simulations, and obtain the region where fBm overcomes its negative traffic weakness. Next, a service provisioning formula is provided in Section 5.2. Gamma and incomplete Gamma functions are used to derive the inverse of the performance formula of an fBm queue which leads to a concise service provisioning formula. Finally, we demonstrate the accuracy of our fBm-based closed-form formulae by numerically comparing the overflow probability and link dimensioning results with the corresponding results obtained by simulating a single server queue fed by a real traffic trace and fitted Markov modulated Poisson process (MMPP) input traffic in Section 5.3.

5.1 Constraints on the Suitability of the FBm Model

The fBm process is the most broadly applied LRD Gaussian model. Since traffic becomes closer and closer to Gaussian as the degree of aggregation increases \[97\], and the autocovariance of traffic appears to be approximately the same as that of fBm \[9\], we expect an fBm model to be suitable for Internet traffic for sufficient levels of aggregation. In this section we carry out numerical comparisons with an important benchmark, a truncated version of the fBm that avoids the negative arrivals. This model provides insights into the effect of negative traffic on the accuracy of the fBm as a traffic model. The fBm model has the unavoidable feature of negative traffic, and this becomes increasingly evident as we consider finer and finer detail of the model. The negative traffic property of fBm is a highly counter-intuitive feature, because for real Internet links, the traffic can never be negative. This feature of the traffic almost disappears, in practice, if the capacity of the system carrying the traffic is sufficiently large.
5.1 Constraints on the Suitability of the FBm Model

5.1.1 The Problem of Negative Traffic

Unavoidably, the negative arrivals phenomenon is part of the FBm model. It is not desirable because it is inconsistent with traffic behaviour in the real world. In this section, we use as benchmark a single server queue whose input is a truncated version of the FBm model so that negative arrivals are avoided. A particularly important issue that we address in this section is identifying the range of parameters for which the adverse effect of negative arrivals on the overflow probability is negligible, making the FBm model useful in practice.

We find that the influence of negative arrivals is highly related to the ratio of $m$ to $\sigma$, which is intuitively understandable. A series of experiments based on simulations has been done in order to obtain a bound by which we are able to know if the FBm is a realistic traffic model. Different $H$ values — i.e. $H = 0.3, 0.4, 0.5, 0.6, 0.7$ and $0.8$ — are chosen for each experiment, where we use various ratios of $m$ to $\sigma$ (with increment as 0.1) for both queues fed by an FBm input and its truncated counterpart. When the difference of the overflow probabilities of the two inputs is below a certain level (here we choose 10%), then we say the FBm model is realistic with the certain value of $m/\sigma$; otherwise, we say it is unrealistic. By recording all the results for $H = 0.3 – 0.8$, we obtain Figure 5.1, where the regions of when FBm is a realistic model or not have been shown. We see that as the arrival rate increases, the effect of negative arrivals diminishes. For convenience, a curve involving $m/\sigma$ and $H$ has been drawn as a threshold in Figure 5.1.

Notice that $P(Q > 0) = 1$ regardless of whether the FBm input is truncated or not. This is not valid for real traffic, which may exhibit empty buffers for an extended period of time thereby revealing a weakness of the FBm model – it is not a suitable traffic model when the buffer size is small. This weakness means that FBm is usually not applicable to an all-optical network, where buffering light is difficult. However,
5.1 Regions where fBm is a realistic model or unrealistic model due to the presence of negative traffic.

The fBm model is relevant to the current Internet, where the buffer sizes of routers are very large.

5.1.2 Discussion

The fBm model is not universally appropriate to Internet traffic. An important weakness of this model is that for certain ranges of the parameters there exists excessive negative traffic. A small amount of negative traffic will always be present in the fBm model, but since real networks have no negative traffic at all, it is inappropriate to use a model which has large amounts of it. When $\sigma$ is large relative to $m$, or when $H \ll 0.5$, and especially when both of these conditions hold, the model will exhibit large amounts of negative traffic when observed at small time scales.

It may be possible to re-interpret an fBm model in a manner which alleviates this problem and enables us to use this model for a wider range of parameters. Since the dimensioning formula, which will be introduced in the next section, has such
a natural interpretation, it may be useful to develop methods applicable to a wider range of situations. However, in the absence of such a strategy it will be necessary to confine the use of the fBm model to the region in Figure 5.1 labelled as Realistic. This constraint on $\sigma/m$ means that the fBm model is applicable only to the core links of networks where the traffic is relatively smooth.

5.2 Service Rate Provisioning

Having validated the analytical formula derived for the workload distribution of a queue fed by fBm traffic in Chapter 4, we are now in a position to evaluate the capacity required to serve a queue fed by fBm traffic such that the overflow probability for a given queue threshold is below a given margin. In the following theorem, we provide a service rate provisioning formula whereby the required service rate can be calculated to meet a certain QoS measures-overflow probability and queue threshold.

**Proposition 1.** Consider a single server queue fed by an fBm input characterised by the mean rate $m$, the variance $\sigma^2$, and the Hurst parameter $H$. Let $\varepsilon$ be the overflow probability of an fBm queue. Using the notations of (4.6), we assume that

$$\varepsilon = P(Q > q) = \int_q^\infty cf(x)dx.$$  \hspace{1cm} (5.1)

That is, the approximation (4.6) is assumed to be exact. Then, the capacity which exactly meets the performance requirement is

$$\mu^* = m + \left(\frac{2H^{2H}G^{-1}\left((2H)^{-1},\varepsilon\right)}{(1-H)^{2H-2}}\right)^{\frac{1}{2H}}(\sigma q^{1-H})^{\frac{1}{2H}}.$$  \hspace{1cm} (5.2)
Service rate provisioning of an FBm Queue

In this expression, \( G^{-1}(\alpha, y) \) is the inverse regularised incomplete Gamma function defined by the property

\[
G^{-1}(\alpha, G(\alpha, x)) = G(\alpha, G^{-1}(\alpha, x)) = x, \tag{5.3}
\]

where \( G(a, x) \) is the regularised incomplete Gamma function, given by

\[
G(a, x) = \frac{\int_{x/k}^{\infty} t^{a-1} e^{-t} dt}{\Gamma(a)}. \tag{5.4}
\]

Proof. By substituting \( kt \) for \( t \) in (5.4), we find

\[
\Gamma(a)G(a, x) = k^a \int_{x/k}^{\infty} t^{a-1} e^{-kt} dt.
\]

The function \( f(x) \) and \( c \) were expressed in (4.7) and (4.8), respectively:

\[
f(x) = x^{\beta-1} e^{-\alpha x^\nu},
\]

and

\[
c^{-1} = \nu^{-1} \alpha^{-\frac{\beta}{\nu}} \Gamma\left(\frac{\beta}{\nu}\right).
\]

We can derive a dimensioning formula from (5.1) as follows:

\[
\varepsilon = \int_{q}^{\infty} c f(x) dx
\]

\[
= \frac{\nu \alpha^{\frac{\beta}{\nu}}}{\Gamma\left(\frac{\beta}{\nu}\right)} \int_{q}^{\infty} y^{\beta-1} e^{-\alpha y^\nu} dy.
\]
5.2 Service Rate Provisioning

Using the substitution \( u = y^\nu \), in which case \( du = \nu y^{\nu-1} dy \), or putting it another way, \( du = \nu u^{\nu-1} dy \), we have

\[
\varepsilon = \frac{\alpha^\beta}{\Gamma\left(\frac{B}{\nu}\right)} \int_{q^\nu}^{\infty} u^\nu \nu^{1-\nu} e^{-\alpha u} du
\]

\[
= \frac{\alpha^\beta}{\Gamma\left(\frac{B}{\nu}\right)} \int_{q^\nu}^{\infty} u^\nu e^{-\alpha u} du.
\]

Recall that \( \beta = \frac{1-H}{H} \) and \( \nu = 2 - 2H \), we obtain

\[
\varepsilon = G\left(\frac{1}{2H}, \alpha q^{2-2H}\right).
\]

With (5.3) we obtain

\[
\alpha = G^{-1}\left(\frac{1}{2H}, \varepsilon\right) q^{2H-2}. \tag{5.5}
\]

Now we express \( \iota \) according to the defining equation for \( \alpha \):

\[
\alpha = \frac{(1-H)^{2H-2} |\iota|^{2H}}{2H^2 \sigma^2}
\]

\[
\implies (1-H)^{2H-2} |\iota|^{2H} = 2\alpha H^{2H} \sigma^2
\]

\[
\implies |\iota| = \left( \frac{2\alpha H^{2H} \sigma^2}{(1-H)^{2H-2}} \right)^{\frac{1}{2H}}.
\]

Now since \( \iota = m - \mu^* \), where \( \mu^* \) is the capacity and \( m \) denotes the mean rate of the input traffic,

\[
\mu^* - m = \left( \frac{2\alpha H^{2H} \sigma^2}{(1-H)^{2H-2}} \right)^{\frac{1}{2H}}.
\]

The conclusion (5.2) is established by elementary algebra operation. \( \square \)

Proposition 1 provides a simple and elegant formula in service rate provisioning in a queue loaded by fBm traffic on the basis of the assumption that (4.6) holds. The extensive simulations used to validate the approximation (4.6) therefore validate also our service rate provisioning formula. From (5.2) simple relationships can be
observed. One observation is that the required spare capacity beyond the arrival rate \( m \) is independent of \( m \). Another simple observation is that the spare capacity required is proportional to \( \sigma^2/2H \). In the following, we provide numerical results illustrating the application of this formula.

### 5.2.1 Numerical Results

Numerical results of service rate provisioning for a range of examples using the method developed above are presented in Figures 5.2–5.5. The results presented are for the total capacity \( \mu^* \), but it is important to keep in mind the concept of spare capacity \( \mu^* - m \). In all the figures we set \( m = 1 \). This could represent a data rate in the order of Gb/s, e.g., one OC-192 or OC-768 [179] with rate of 10 Gb/s or 40 Gb/s. Then, \( q = 0.1 \) would represent a QoS measure of 100 ms.

In Figure 5.2, we illustrate the total capacity \( \mu^* \) required as a function of the second QoS measure \( \epsilon \) within the range \( 10^{-5} – 1 \). As expected the spare capacity required, i.e. \( \mu^* - 1 \), reduces with the QoS measure \( \epsilon \). For the case \( m = 1 \), the spare capacity \( \mu^* - 1 \) approaches zero as the QoS is further and further relaxed.

In Figure 5.3, we illustrate that the total capacity \( \mu^* \) increases with the Hurst parameter \( H \). This is also expected as stronger correlation in the traffic stream means higher queueing delay. Notice, however, that the increase remains bounded as \( H \to 1 \).

In Figure 5.4, we illustrate that the total capacity \( \mu^* \) increases with the standard deviation parameter \( \sigma \). This is also expected as stronger variations in the traffic stream, usually, means higher queueing delay.

In Figure 5.5, we illustrate that the total capacity \( \mu^* \) decreases as the QoS parameter \( q \) increases. Thus, allowing more delay and more buffering relaxes delay requirement, and less capacity is therefore required. The required spare capacity
5.2 Service Rate Provisioning

Fig. 5.2 Capacity vs. $\varepsilon$ by (5.2) for $H = 0.8$, $\sigma = 0.05$, $m = 1$ and $q = 0.1$.

Fig. 5.3 Capacity vs. $H$ by (5.2) for $\sigma = 0.05$, $m = 1$, $q = 0.1$ and $\varepsilon = 0.001$. 
approaches zero as the queue size approaches infinity. This is consistent with known results in elementary Markovian queues.

**5.3 Validation Based on Real Traffic Data**

In this section, we provide evidence based on real traffic data concerning the accuracy of our fBm-based overflow probability approximation and the dimensioning formula. We also describe a useful process for fitting the parameters which are then used in our closed-form formulae. We also provide a comparison with a short range dependent (SRD) model, in particular, the Markov modulated Poisson process (MMPP) [7, 80], and illustrate the benefit of using the fBm.
5.3 Validation Based on Real Traffic Data

Fig. 5.5 Capacity vs. $q$ by (5.2) for $H = 0.8$, $\sigma = 0.05$, $m = 1$ and $\varepsilon = 0.001$.

5.3.1 Dataset Description

A real traffic input data stream is selected from an IP traffic trace provided by CAIDA’s equinix-sanjose monitors on high-speed Internet backbone links. The details provided in [24] enable the reader to reproduce the complete traffic trace that we use here. It contains around $1.478 \times 10^9$ packets with total size of about 1 Terabyte that were captured within one hour on February 16, 2012. Let that one hour, during which the traffic data is captured, be divided into consecutive fixed time intervals, each of duration $t$ [sec.]. Let $S_n$ be the total traffic [Mbytes] that arrives within the time interval $[(n-1)t, nt)$.

5.3.2 Fitting FBm Parameters to the Real Measurements

In order to evaluate queueing performance or perform link dimensioning for an fBm queue, we first need to set the values of its parameters, namely, the mean rate ($m$), the Hurst parameter ($H$) and the variance of the total arriving workload within an
interval of length 1 ($\sigma^2$). In the following, we show how to fit these three parameters to a real traffic input data stream.

To begin with, the mean rate, $m$, can be estimated as the mean of $S_n$ for $t = 1$ [sec.]. For the present case, $m$ is set to approx 270.87 [Mbyte/sec.]. The values of $H$ and $\sigma^2$ are estimated by fitting the variance-time curve of $S_n$. Figure 5.6 illustrates how we obtain $H$ and $\sigma$. We first find the variances of $S_n$ for various $t$ values, and then fit a variance-time curve to them in a log-log plot. For this case, we choose the following $t$ values: $t = 0.001, 0.01, \ldots, 100$ [sec.]. A straight fitted line gives a slope of 1.723, which is larger than 1, and therefore provides evidence of the LRD phenomenon for the real traffic (see the definition and property of LRD as described in Section 2.2). Since the slope of the v-t curve is equal to $2H$, the value of $H$ can be obtained as 0.8615. More sophisticated methods for fitting $H$ are available, such as the Whittle estimator [10] and the Abry-Veitch wavelet method [50, 180]. However in the present context it is more useful to check that the variance-time curve takes the expected form (linear in a log-log plot). Meanwhile, the estimation of $\sigma^2$ is found from the v-t curve for $t = 1$, namely $\sigma^2 = 457.09$. Therefore, the three fBm parameters are set as follows: $m = 270.87$, $H = 0.8615$ and $\sigma^2 = 457.09$. As the three fBm parameters are available, we can use the closed-form equations (4.6) and (5.2) to obtain the overflow probability and perform link dimensioning, respectively.

5.3.3 Comparisons Between FBm, MMPP and Real Measurements

We now consider three independent single server queues. They all have the same service rate of $\mu^*$, initially set at $\mu^* = 320$ [MByte/sec.]. The input traffic into the first queue is the real data input traffic stream described in Section 5.3.1. The input traffic into the second queue is the fitted fBm process as described in Section 5.3.2. The input traffic into the third queue is based on the MMPP process obtained by
Algorithm LAMBDA [80]. In the following, we provide information on how we generate the MMPP input traffic so that its parameters are fitted to the real traffic input data stream.

The MMPP input traffic is formed by the MMPP arrival process with general packet size distribution. Algorithm LAMBDA is applied to determine the MMPP state and state transition matrix, by which the MMPP process is generated. Detailed steps for fitting real traffic data can be found in Section II-B of [80]. Also a matlab code for Algorithm LAMBDA is available in [181]. Using it on the real traffic data, we obtain 12 states and the corresponding state transition matrix. For the packet size distribution, we use the actual packet size distribution of the real traffic trace. Since no analytical result is available for single server queues with MMPP input and general nonparametric service time distribution, the queueing performance for MMPP is obtained by simulation.
In the following, we compare the overflow probabilities and link dimensioning results for the above-mentioned three queues. The results for the fBm queue are obtained based on our approximation of the overflow probabilities (4.6) and the dimensioning formula (5.2). The results for the two equivalent queues fed by the fitted MMPP and the actual trace are obtained by simulations. Because the arrival process of both cases is not Poisson, a Poisson inspector is used to measure the queue size during the simulation.

Figure 5.7 presents the overflow probabilities for the three queues. As mentioned, $\mu^*$ is set as 320 [MByte/sec.], which is close to $m + 3\sigma$. During the simulations for the queues fed by the real traffic trace and the MMPP input traffic, we independently computed overflow probabilities for each 10 minutes’ period, and then used them to obtain 90% confidence intervals based on the Student’s t-distribution, presented as the coloured area in the figure. The results from our fBm formula are within the confidence interval of the simulation results of real traces for $q > 0.1$ [MBytes].
5.3 Validation Based on Real Traffic Data

Explanation of the differences between the fBm formula and the real trace simulation results for small $q$ is the existence of negative traffic, which has more effect on small queue size. On the other hand, the simulation results of the queues fed by the real trace and the MMPP input traffic are close to each other for $q < 0.1$ [MBytes]. However, there is almost no overlap between their confidence intervals for $q > 0.1$ [MBytes]. This is explained by the fact that MMPP is SRD while fBm is LRD.

Table 5.1 Mean and confidence intervals of the overflow probabilities.

<table>
<thead>
<tr>
<th>$q$ (MB)</th>
<th>Real measurements</th>
<th>MMPP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>95% Confidence Interval</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>83.54%</td>
<td>(79.12%, 87.97%)</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>76.60%</td>
<td>(70.53%, 82.66%)</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>54.61%</td>
<td>(44.49%, 64.74%)</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>23.28%</td>
<td>(12.44%, 34.13%)</td>
</tr>
<tr>
<td>$10^{0}$</td>
<td>4.49%</td>
<td>(0%, 9.28%)</td>
</tr>
<tr>
<td>$10^{1}$</td>
<td>0.72%</td>
<td>(0%, 1.91%)</td>
</tr>
</tbody>
</table>

Next we consider the link dimensioning problem. For each one of the queue, we seek the least link capacity $\mu^*$ that ensures that the overflow probability will not exceed 0.001. For the fBm queue, this is achieved based on the dimensioning formula (5.2). For the other two queues, based on the MMPP input and the input from the real trace, this is achieved by bisection on $\mu^*$ using repeated simulations. As shown in Figure 5.8, the fBm model leads to reasonably accurate dimensioning, with error of no more than 5% and appears to outperform MMPP for the full range of the queue threshold. Noticing that in Figure 5.7, we observe better accuracy for MMPP with small $q$ values, the explanation of the results presented in Figure 5.8 is that there we are considering much smaller overflow probabilities than the ones for which MMPP performs well in Figure 5.7. We have noticed from the results in Figure 5.7 that MMPP significantly underestimates the overflow probability in the range where the overflow probability is below 1%; therefore it also underestimates the required capacity as illustrated in Figure 5.8. In sum, we demonstrate the accuracy
Fig. 5.8 Comparison of the least link capacities for fBm and MMPP models, and real measurements for the overflow probability $\varepsilon \leq 0.001$.

and usefulness of our easy-to-use fBm-based overflow probability approximation and the dimensioning formulae.

### 5.4 Summary

In this chapter, constraints on applying the fBm model so as to use it for link dimensioning have been studied. We observe that our formulae can be applied in practical scenarios to network link dimensioning where traffic is heavily multiplexed, and the buffer size and the system service rate are sufficiently large. A service rate provisioning formula for an fBm queue has been derived based on the approximation provided in the preceding chapter. We have also presented numerical results for a range of examples for link dimensioning based on our queueing analysis. The accuracies and usefulnesses of the fBm-based overflow probability and link dimensioning formulae have been demonstrated by comparisons with simulation results of a single server.
queue fed by a real traffic trace. This shows the benefits over the MMPP model when the overflow probability is below 1%.
Chapter 6

FBm Versus PPBP and PLBP

In this chapter, we compare the fBm model with the PPBP and the PLBP models. There are several reasons for us to use PPBP and PLBP for comparison and benchmarking of the fBm model. First, compared with the fBm model, either PPBP or PLBP, as a traffic model, is more general because it is not limited to the heavy multiplexing regime. As a result, it can help us identify the parameter region where insufficient multiplexing introduces errors when we use fBm as a traffic model. Second, neither PPBP nor PLBP has negative arrivals, therefore it can help us understand the parameter region where the negative arrivals introduce errors when we use fBm to model the traffic. These features of the PPBP and PLBP make them superior to fBm as a traffic model in capturing the nature of the Internet traffic. Unfortunately, the PPBP/PLBP does not lend itself to a closed-form analytic solution. On the other hand, fBm and PPBP/PLBP are closely related when the traffic intensity of PPBP/PLBP is very high [29, 99].

6.1 Comparison Between FBm and PPBP

We consider a single server queue fed by a PPBP input. The service rate is denoted by $\mu$ [bit/sec.]. The rate at which data is generated during each burst is assumed to
be a constant $r$ [bit/sec.]. New bursts arrive according to a Poisson process with rate $\hat{\lambda}$. The time duration of a burst is a random variable denoted $d$. It follows a Pareto distribution, the CCDF of which is

$$P(d > x) = \begin{cases} 
\left(\frac{x}{\delta}\right)^{-\gamma}, & x \geq \delta, \\
1, & \text{otherwise}.
\end{cases} \quad (6.1)$$

Here $\delta$ is the scale parameter which determines the minimum value of $d$, and $\gamma$ is the shape parameter that controls the tail behaviour of the distribution. We have

$$E(d) = \frac{\delta \gamma}{\gamma - 1} \text{ for } \gamma > 1.$$  

The mean traffic, $m$, is given by:

$$m = \hat{\lambda} r E(d) = \frac{\hat{\lambda} r \delta \gamma}{\gamma - 1}. \quad (6.2)$$

We also have the expression for the variance of the amount of bits arriving during the time interval $t$ in a PPBP from [95]:

$$\sigma^2(t) = \begin{cases} 
2r^2 \hat{\lambda} t^2 \left( \frac{\delta \gamma}{2(\gamma - 1)} - \frac{t}{6} \right), & 0 \leq t \leq \delta, \\
2r^2 \hat{\lambda} \left( \frac{\delta^3 \gamma}{6(3-\gamma)} - \frac{\delta^2 \gamma t}{2(2-\gamma)} - \frac{t^2 \gamma t^2}{2(1-\gamma)(2-\gamma)(3-\gamma)} \right), & t > \delta.
\end{cases} \quad (6.3)$$

We compare the fBm and PPBP models using the overflow probabilities obtained by our approximation as described in Chapter 4 and the Quasi-Stationary (QS) approximation for the PPBP queue of [99]. To achieve a fair comparison, we aim to match the key statistical characteristics of the two processes. In the following we describe how we choose an fBm process, described by the three parameters 1) the mean net input, $\iota$, 2) the Hurst parameter, $H$, and 3) the variance, $\sigma^2$, for a given PPBP process described by $\hat{\lambda}$, $r$, $\delta$ and $\gamma$. 

100
First, by Equation (6.2) we obtain

\[ t = \frac{\lambda r \delta \gamma}{\gamma - 1} - \mu. \]

Second, the value of the Hurst parameter \( H \) is obtained by the formula [95]

\[ H = \frac{3 - \gamma}{2}. \]

Finally, we derive an equation for the variance \( \sigma^2 \) of the fBm process in terms of the parameters of the PPBP process such that the variance-time curves of the two processes are close to each other. To achieve a close match of the variance-time curves, we first note that the variance-time function of fBm is given by

\[ \sigma^2(t) = \sigma^2 t^{2H}. \]  \hspace{1cm} (6.4)

Next, we choose an appropriate \( t_1 \) value such that

\[ \sigma_{fBm}^2(t_1) = \sigma_{PPBP}^2(t_1), \]

where the right-hand side of this equation is given by (6.3). If we have made a good choice of \( t_1 \), we can obtain the required fBm parameter, i.e. the variance \( \sigma^2 \), in terms of \( H \) by (6.4). Then, the fBm variance-time curve can be obtained via (6.4) as

\[ \sigma^2(t) = \sigma^2(t_1) \left( \frac{t}{t_1} \right)^{2H}. \]  \hspace{1cm} (6.5)

The remaining problem is therefore the choice of \( t_1 \). This is done by observing the behaviour of the variance-time curves for various choices of the parameter \( t_1 \). Let us illustrate our approach through the following example. We set the PPBP parameters as follows: \( \gamma = 1.5, \delta = 1 \) and \( r^2 \hat{\lambda} = 1 \). Note that by (6.3), the family of
PPBP processes that obey the relationship $r^2 \hat{\lambda} = 1$, all have the same variance-time curve. These parameters settings are sufficient to uniquely define the variance-time curve of the PPBP. Then, we choose three values for $t_1$: 1, 10, and 100. To obtain the corresponding fBm time curves, we use $H = (3 - \gamma)/2 = 0.75$. Having obtained $H$ and $t_1$, we use (6.5) to uniquely determine the variance-time curve. The three corresponding variance-time curves of the fBm are plotted in Figure 6.1. We observe that for reasonably large time scales choosing $t_1 \geq 10$ will give reasonably accurate match, so we choose $t_1 = 10$.

We set $\gamma = 1.5$ so that $H = 0.75$, which is a realistic value for Internet traffic. The values for mean net input, $t$, and $r$ are the same here as in [99], with

$$t = -\frac{3\sigma(\delta)}{\delta} \text{ and } r = 1.$$
The values of $\hat{\lambda}$ are chosen as 1, 64, 1024 and 1000000 to model the steady growth of traffic intensity, as might be expected in a real network. Since it is known theoretically that the Poisson-Pareto burst model converges to the behaviour of a Gaussian model with almost the same variance-time curve as fractional Brownian motion as $\hat{\lambda} \to \infty$ [98], checking that this convergence holds through our new formula for the stationary buffer level in an fBm queue provides further confirmation of the validity of the analyses of both models. To undertake this test thoroughly we should check that it holds over the full range of choices for the parameters of these models. It is not necessary to repeat the comparison for choices of parameters which are only trivially different (in the sense of scaling), and consequently the three figures 6.2–6.4 are sufficient to explore a fairly full range of parameter values.

These figures show a comparison of overflow probabilities of PPBP model and fBm model for $\delta = 0.1$, 0.2 and 0.5, for figures 6.2, 6.3, 6.4, respectively. We choose comparatively small $\delta$ because this represents the minimum burst duration, because application flows in practice, here represented by bursts, can be relatively small. In addition, note that the smaller the $\delta$ is, the less the discrepancy between the variance-time curves of the PPBP model and fBm. The three parameters of the PPBP model, $\delta$, $\hat{\lambda}$ and $r$, linearly depend on the time unit selected. Increasing these three parameters at the same rate is equivalent merely to a change of time unit, and their queueing behaviours are therefore identical.

The $x$-axis of the plots in Figure 6.2–Figure 6.4 is in units of $x/\sqrt{\hat{\lambda}}$ where $x$ is the buffer threshold level. This is because when the central limit theorem is applied to these PPBP processes convergence to a certain common Gaussian process will occur only if the arriving work is measured in these (or some equivalent) units.

As expected, we see the convergence between the two models for high aggregation ($\hat{\lambda} = 1000000$) on a large buffer threshold, for all three graphs. Compared with the fBm model, the PPBP model has the following differences: 1) it is a non-Gaussian process, 2) it has a slightly different variance-time curve (with the
FBm Versus PPBP and PLBP

Fig. 6.2 Overflow probabilities for PPBP model with various arrival rates $\hat{\lambda}$, estimated by the QS algorithm vs. the fBm model by our approximation $\delta = 0.1$

difference reducing as $\delta \to 0$) and 3) it contains no negative arrivals. According to the central limit theorem, PPBP tends to be Gaussian when $\hat{\lambda}$ is very large. Also the difference of the variance-time curves of fBm and PPBP is minimised by choosing a suitable value of $t_1$—which has been explained in details earlier in this section. So, we consider the negative arrival feature of the fBm model as the main reason for disagreements in the range associated with small buffer thresholds even for large $\hat{\lambda}$, i.e. $\hat{\lambda} = 1000000$. Intuitively, the smaller the buffer is, the higher the effect of negative arrivals will be. In the following we focus our attention on the effect of the negative arrival feature of fBm.

6.2 Comparison Between FBm and PLBP

Either the PLBP model or the fBm model has its pros and cons; which one to choose depends on the situation encountered. PLBP is designed to model Internet traffic. Its construction mimics the generation of real Internet traffic flows. Compared with the
6.2 Comparison Between FBm and PLBP

Fig. 6.3 Overflow probabilities for the PPBP model with various arrival rates $\hat{\lambda}$, estimated by the QS algorithm vs. the fBm model by our approximation for $\delta = 0.2$

Fig. 6.4 Overflow probabilities for the PPBP model with various arrival rates $\hat{\lambda}$, estimated by the QS algorithm vs. the fBm model by our approximation $\delta = 0.5$
fBm model, there are no constraints on applying the PLBP model. This model has a wide scope of application for Internet traffic modelling. It contains four parameters providing more flexibility, however, increasing the complexity of analysis. In this thesis, the approximation we have for the PLBP queue is not in closed-form. It implies that we can only obtain numerical solution, and no dimensioning formula can be derived. The fBm model was known before the Internet. It is not intended to model the Internet traffic. One of its weaknesses is negative traffic, which is inconsistent with the behaviour in the real world. Nevertheless, fBm is a good model for modelling highly multiplexed traffic, and is amenable to analysis. In this thesis, closed-form approximations for overflow probability and link dimensioning of the fBm queue have been provided. Considering that fBm is much easier to analyze, it is always better to choose fBm when the traffic are highly multiplexed, while the PLBP model can be used when fBm is not applicable.

Other than the differences between the two models, they do share some common features. Both of them are LRD, and the PLBP process converges to fBm as the traffic intensity, $\lambda \to \infty$. One example has been shown in Figure 6.5, where the overflow probabilities for the queues fed by these two processes are compared. We set the PPBP parameters as follows: $\gamma = 1.6$, $\delta = 0.7344$, and $r^2 \lambda = 1$. According to the variance function of PLBP, the family of PLBP processes obeying the relationship $r^2 \lambda = 1$, all have the same variance-time curve. The values of $\lambda$ are chosen as 1, 100, 10000 and 20000. The service rate of the queue $\mu$ is set such that the mean net input, $\iota = m - \mu$, is equal to $-0.5$, with $m$ as the mean of the process. For the fBm queue, we fit its parameters ($H$, $\iota$ and $\sigma^2$) to the PLBP queues. The value of $H$ is equal to 0.7 by $H = \frac{3 - \gamma}{2}$. By fitting the variance-time curve of fBm to that of PLBP, we can obtain $\sigma^2 = 2.509$ as shown in Figure 6.5b. The overflow probabilities of the PLBP queues obtained by simulations and of the fBm queue obtained by our approximations have been shown in Figure 6.5a. We see the convergence between the two models for large $\lambda$. 

106
6.2 Comparison Between FBm and PLBP

(a) Overflow probabilities

![Overflow probabilities graph]

- **our approx. for fBm**
- sim. for PLBP ($\lambda = 1$)
- sim. for PLBP ($\lambda = 100$)
- sim. for PLBP ($\lambda = 10000$)
- sim. for PLBP ($\lambda = 20000$)

Log($P(Q>x)$) vs. Log($\sigma^2(t)$)

(b) Variance-time curves

![Variance-time curves graph]

Log($\sigma^2(t)$) vs. Log($t$)

$\sigma^2$ for fBm = 2.509

Fig. 6.5 PLBP vs. fBm for $\gamma = 1.6$ ($H = 0.7$) and $t = -0.5$. 

107
Chapter 7

Conclusions

In this thesis, we aim to evaluate the performance of queues fed by LRD inputs. The main focuses are the queues fed by two specific LRD processes, namely the PLBP process and the fBm process. Approximate results for the overflow probabilities of SSQs with both processes have been provided.

In an attempt to improve the ability of the PPBP model to capture the small flows, we have proposed the PLBP model. The two models are similar except that the burst durations of the PLBP process are distributed according to the Lomax distribution instead of the Pareto distribution. We have shown that, in some cases, Lomax can be a better choice for the distribution of the flow sizes than Pareto, by real measurements. Based on the QS algorithm, we have obtained an analytical approximation for the overflow probabilities of a PLBP queue. To shorten the simulation time, a fast simulation method has been applied. We have demonstrated that the results obtained by this method are very close to those of conventional simulations. There are some discrepancies between the approximation and the simulation results. We believe the reason is the Gaussian assumption of the shorts bursts. Once we find a better solution for the queue fed by the short bursts, we can improve the accuracy of the approximation.
Conclusions

Next, we have derived a new approximation for the overflow probabilities of fBm queues. This approximation arises from a new interpretation of the asymptotic result by Hustler and Piterbarg. Although the expression of our approximation is based on this asymptotics, the two are not the same because we omitted some terms during the derivation. The approximation is simple to calculate and work with, and is accurate over full range of parameter, which has been shown by the comparison with simulation results. In order to save time for generating fBm sequences, we have invented a novel approach of simulating fBm queues by reusing a single fBm sequence. Linking the expression of the approximation and the Generalized Gamma distribution, we have provided important statistics of an fBm queue, i.e., the mean, the variance, the second central moment, and the skewness.

We have further studied the constraints on applying the fBm model in order to use it for link dimensioning. One counter-intuitive feature of the fBm model is that it contains negative traffic. By investigating the effect of this feature on queueing performance, we have observed that the fBm model is applicable when traffic is heavily multiplexed, and the buffer size and the system service rate are sufficiently large. In such situation, our approximation for the overflow probabilities can also be applied. Then, a service provisioning formula based on this approximation has been derived. The accuracies and usefulnesses of the service provisioning and the overflow probability formulae have been demonstrated by comparisons with simulation results of a SSQ fed by a real traffic trace.

Finally, we have compared the queueing performance of the fBm model with the PPBP and the PLBP models. The convergence between the fBm model and the PPBP/PLBP model has been found when traffic is highly multiplexed. The pros and cons of the two key models of this thesis - PLBP and fBm - have been discussed.
References


References


References


116


References


References


References


Appendix A

Simulation of a PLBP Queue

A.1 Java Code for One Observation

```java
package PLBP;

import ppbpplbp.RVGenerator;

/**
 * @author CHEN Jiongze
 */

public class PLBPSimInstance {
    private double T;
    private final int intervalNo;
    private final double intervalLength;
    private final int longBurstNo;
    private final int shortBurstNo;
    private double[] works;  // works of each interval
```
/**
 * 
 * @param T total simulation time
 * @param intervalNo number of intervals
 * @param longBurstNo number of initial long bursts
 * @param shortBurstNo number of initial short bursts
 */

public PLBPSimInstance(double T, int intervalNo, int longBurstNo,
                        int shortBurstNo){
    this.T = T;
    this.longBurstNo = longBurstNo;
    this.shortBurstNo = shortBurstNo;
    this.intervalNo = intervalNo;

    // obtain the number of time intervals
    intervalLength = T/(double)intervalNo;

    // initial the array of works at all time intervals
    works = new double[intervalNo];
    for(int i = 0; i < intervalNo; i++){
        works[i] = 0.0;
    }
}

/**
 * 
 * @param thresholds a double array containing the buffer threshold, x
 * @param lambda arrival rate lambda
 */
A.1 Java Code for One Observation

```java
* @param gamma shape parameter of the Lomax distribution
* @param delta scale parameter of the Lomax distribution
* @param r the constant rate at which each burst generate work
* @param C capacity/bandwidth
* @return an array of overflow probabilities, P(Q>x), for all x
* values in the thresholds
*/

public double[] runForOverflowProb(double[] thresholds, double lambda,
        double gamma, double delta, double r, double C){
    double currentTime = RVGenerator.getExpRV(1.0/lambda);
    //index of current time interval
    int checkPointNo = 1;
    //the work left in the queue at the end of the interval
    double workLeft = 0.0;
    //array of countors corresponding to x values
    int[] counts = new int[thresholds.length];
    for(int i = 0; i < counts.length; i++){
        counts[i] = 0;
    }

    //adding the initial long and short bursts into the queue
    initialQueue(gamma, delta, r);

    //start simulation
    while(currentTime < T){ //simulation does not end
        while(currentTime >= (double)checkPointNo * intervalLength){
            //a interval ends, start to monitor the queue
            workLeft = Math.max(workLeft + works[checkPointNo-1]
```

127
Simulation of a PLBP Queue

- C*intervalLength,0.0);

//counter++ for the corresponding x if the queue size >= x
int tmpPos = searchPosition(thresholds, 0, workLeft);
for(int i = 0; i < tmpPos; i++){
    counts[i]++;
}
checkPointNo++;

//assign work to the intervals
double duration = RVGenerator.getLomaxRV(gamma, delta);
double departureTime = currentTime + duration;
addWork(checkPointNo, currentTime, departureTime, duration, r);

currentTime += RVGenerator.getExpRV(1.0/lambda);
}

//monitor the queue at the end of the simulation
int tmpPos = searchPosition(thresholds, 0, Math.max(workLeft +
    works[checkPointNo-1] - C*intervalLength,0.0));
for(int i = 0; i < tmpPos; i++){
    counts[i]++;
}

//calculate the overflow probabilities
double[] results = new double[counts.length+1];
for(int i = 0; i < counts.length; i++){
    results[i] = (double)counts[i] / (double)intervalNo;
}
double mean = 0.0;
for(int i = 0; i < works.length; i++){
    mean += works[i];
}
mean = mean / intervalLength / (double) works.length;

//add mean of the workloads for one interval to the array
results[counts.length] = mean;
return results;
}

/**
 * This function initializes the queue by adding the
 * initial long and short bursts into the queue
 * @param gamma shape parameter of the Lomax distribution
 * @param delta scale parameter of the Lomax distribution
 * @param r the constant rate at which each burst generate work
 */
private void initialQueue(double gamma, double delta, double r){
    double duration;
    for(int i = 0; i < works.length; i++){
        works[i] += (double)longBurstNo * r * intervalLength;
    }
    for(int i = 0; i < shortBurstNo; i++){
        duration = initialShortBurst(gamma, delta);
        addWork(1, 0, duration, duration, r);
    }
}
/**
 * @param checkPointNo index of the current interval, belonging to [1, intervalLength]
 * @param arrivalTime arrival time of the burst
 * @param departureTime departure time of the burst
 * @param duration the duration of the burst
 * @param r the constant rate at which each burst generate work
 */
private void addWork(int checkPointNo, double arrivalTime,
                      double departureTime, double duration, double r){
    if(departureTime <= (double)checkPointNo * intervalLength){
        //end before current interval
        works[checkPointNo-1] += duration * r;
    }else{ //end after current interval
        if(checkPointNo == intervalNo){ //last interval
            works[checkPointNo-1] += (T - arrivalTime) * r;
        }else{ //not last interval
            works[checkPointNo-1] += ((double)checkPointNo
                                * intervalLength - arrivalTime) * r;
            if(works[checkPointNo-1]<0){
                System.out.println("error");
            }
            int no = checkPointNo+1;
            while(departureTime > (double)no * intervalLength
                  && no < intervalNo){ //end after "no" interval
                works[no-1] += intervalLength * r;
            }
        }
    }
}
A.1 Java Code for One Observation

```java
    no++; 
    if(works[no-1]<0){
        System.out.println("error");
    }
}
}
if(departureTime > T){
    works[no-1] += intervalLength * r;
}else{
    works[no-1] += (departureTime - (double)(no - 1)
        * intervalLength) * r;
}
if(works[no-1]<0){
    System.out.println("error");
}
}
}
```
### Simulation of a PLBP Queue

```java
while (pos < array.length && array[pos] < x) {
    pos++;
}
return pos;
```

```java
/**
 * @param gamma shape parameter of the Lomax distribution
 * @param delta scale parameter of the Lomax distribution
 * @return the duration of an initial short burst
 */
private double initialShortBurst(double gamma, double delta) {
    double burst = RVGenerator.getFRTLomaxRV(gamma, delta);
    while (burst > T) {
        // the duration is longer than total simulation time
        burst = RVGenerator.getFRTLomaxRV(gamma, delta);
    }
    return burst;
}
```

### A.2 Flowcharts for the Java Code
A.2 Flowcharts for the Java Code

Start

Initialize the queue:
· Assign initial long and short bursts to the queue
· Set current time $= 0$
· Set total simulation time as $T$
· Set interval index $= 1$

Current time $< T$?

True

interval ends?

True

Observe the queue

Observe the queue

False

Interval index ++

Add a burst to the queue

Current time $+= \text{an exponential RV}$

False

End

Fig. A.1 One observation of the simulation.
Work left = works[interval index] - (interval length * service rate)

Work left < 0?

Work left = 0

i = 0

i < length of thresholds[]?

x = thresholds[i]

Work left > x?

count[i] ++

End

Fig. A.2 Observe the queue.
A.2 Flowcharts for the Java Code

Start

duration = a Lomax RV

Assign work to works[] by current time, the duration and the rate

End

Fig. A.3 Add a burst to the queue.
Appendix B

Simulations of an FBm Queue

B.1 C Code for Generating FBm Sequences

```c
#include <stdio.h>
#include "hosking.h"

/* hosking.h is obtained from */
/* http://www2.isye.gatech.edu/~adieker3/ */

void genFBM(long n, double H, double L, int cum, char *out){
    /* function that writes a fractional Brownian motion */
    /* or fractional Gaussian noise sample into a file. */
    /* n determines the sample size N by N=2^(*n) */
    /* H the Hurst parameter of the trace */
    /* L the sample is generated on [0,L] */
    /* cum = 0: fractional Gaussian noise is produced */
    /* = 1: fractional Brownian motion is produced */
    /* *out name of the output file */

    int i;
```

137
double *output;
long seed1, seed2, NUM;
FILE *out1, *out2;

out1 = fopen(out,"w");
seed1 = 1;
seed2 = 2;
NUM = (long)pow(2.0,n);
output = (double *) calloc[NUM, sizeof(double)];

hosking(&n, &H, &L, &cum, &seed1, &seed2, output);
for(i=0;i<NUM;i++){
    fprintf(out1, "%lf\t", output[i]);
}
fclose(out1);
}

void main(){
char output[100];
int cum;
int n;
double H, L, var;
printf("n: ");
scanf("%d", &n);
printf("H: ");
scanf("%lf", &H);
//printf("%d %.3f\n", n, H);
B.2 C Code for Obtaining Confidence Intervals Based on the Student's T-Distribution

```c
sprintf(output, "n%dv1H%.2f.txt", n, H);
printf("Saving to %s ...", output);

var = 1.0; //variance of the sequence
L = pow(2.0,(double)n)*var;
cum = 0; //generate fGn sequences
genFBM(n,H,L,cum,output);
```

B.2 C Code for Obtaining Confidence Intervals Based on the Student’s T-Distribution

double calConf(double *upper, double *lower, double prob[]){
    /* upper: upper bound of the confidence interval */
    /* lower: upper bound of the confidence interval */
    /* prob[]: array of the input values */
    /* OBS: no. of the observations (length og prob[]) */
    /* SD: degree of freedom as t_{0.025,OBS-1} */

    int i;
double eve, secondM, conf;
    //eve = mean; secondM = second moment; conf = confidence interval
eve = 0.0;
    for(i=0; i<OBS; i++){
        eve += prob[i];
    }
eve /= OBS;
```
Simulations of an FBm Queue

secondM = 0.0;
for(i=0; i<OBS; i++){  
    secondM += (prob[i]-eve) * (prob[i]-eve);
}
secondM /= (OBS-1);
conf = SD*sqrt(secondM/OBS);
*upper = eve + conf;
*lower = eve - conf;
return eve;  //return the mean of the input array
}

double calConf(double *error, double prob[]){  
    /* error: the confidence interval */  
    /* prob[]: array of the input values */  
    /* OBS: no. of the observations (length og prob[]) */  
    /* SD: degree of freedom as t_{0.025,OBS-1} */  
    int i;
    double eve, secondM;
    //eve = mean; secondM = second moment
    eve = 0.0;
    for(i=0; i<OBS; i++){  
        eve += prob[i];
    }
eve /= OBS;
    secondM = 0.0;
    for(i=0; i<OBS; i++){  
        secondM += (prob[i]-eve) * (prob[i]-eve);  
    }
B.3 C Code for Obtaining Overflow Probabilities

```c
secondM /= (OBS-1);
*error = SD*sqrt(secondM/OBS);
return eve; //return the mean of the input array
}
```

B.3 C Code for Obtaining Overflow Probabilities

```c
void getProb(int n, double H, double mu, FILE *fc, FILE *ft, char* inF,
    int xN, double iniX, double v, double sigmaS, double deltaT){
    /* function that gets the probability for given datafile */
    /* n determines the sample size N as N=2^(*n) */
    /* mu net mean input */
    /* *fc output file with extension csv with type FILE */
    /* *ft output file with extension txt with type FILE */
    /* *inF name of the input file containing fGn sequences */
    /* xN size of buffer threshold x */
    /* iniX initial value/scale of x */
    /* v variance in one unit time */
    /* sigmaS the variance in one second */
    /* deltaT one unit of time representing deltaT second */
    /* OBS number of observations desired */

    int i, j, m, bound, **prCount, intN;
    double fgn, prob, error, que, **pr, *x, nomUnit, mW;
    FILE *in;

    in = fopen(inF,"r");
    bound = pow(2.0,n);
```
Simulations of an FBm Queue

\[ \text{intN} = \text{floor}(\text{pow}(2.0, n)/(\text{OBS}+1.0)); \]
\[ \text{nomUnit} = \sqrt{\text{sigmaS}/v}\ast\text{pow}(\text{deltaT}, H); \]
\[ mW = \mu\ast\text{deltaT}/\text{nomUnit}; \]

\[ \text{prCount} = (\text{int **}) \text{calloc}(\text{xN}, \text{sizeof(int*));} \]
\[ \text{pr} = (\text{double **}) \text{calloc}(\text{xN}, \text{sizeof(double*));} \]
\[ \text{for}(i=0; i<\text{xN}; i++){ \]
\[ \text{prCount}[i] = (\text{int *}) \text{calloc}(\text{OBS}, \text{sizeof(int));} \]
\[ \text{pr}[i] = (\text{double *}) \text{calloc}(\text{OBS}, \text{sizeof(double));} \]
\[ } \]
\[ x = (\text{double *}) \text{calloc}(\text{xN}, \text{sizeof(double));} \]

/* form the array of x values */
\[ x[0] = \text{iniX}; \]
\[ \text{for}(i=1; i<\text{xN}; i++){ \]
\[ x[i] = x[i-1] +\text{iniX}; \]
\[ } \]

\[ \text{for}(j=0; j<\text{xN}; j++){ \]
\[ \text{for}(i=0; i<\text{OBS}; i++){ \]
\[ \text{prCount}[j][i] = 0; \]
\[ } \]
\[ } \]

/* start with an empty queue and do \text{intN} intervals without observation */
\[ \text{que} = 0.0; \]
\[ \text{for}(i=0; i<\text{intN}; i++){ \]
\[ \text{fscanf}(\text{in}, "%lf", &\text{fgn}); \]
}\]
B.3 C Code for Obtaining Overflow Probabilities

```c
que = que + fgn + mW;
if(que<0.0)
    que = 0.0;
}

/* start to observe the queue with around intN*OBS intervals */
m = 0; // index of the observation
for(i=intN; i < bound; i++){
    fscanf(in,"%lf",&fgn);
    que = que + fgn + mW;
    /* ensure that queue size is not negative */
    if(que<0.0)
        que = 0.0;
    /* start a new observation */
    if(i%intN==0 && i!=intN)
        m++;
    for(j=0; j < xN; j++){
        /* queue size is larger than the threshold */
        if(que>(x[j]/nomUnit))
            prCount[j][m]++;
    }
}

for(j=0; j<xN; j++){
    /* obtain the overflow probability for each observation */
    for(i=0;i<OBS;i++){
        pr[j][i] = (double)prCount[j][i]/(double)intN;
    }
```
Simulations of an FBm Queue

for(j=0;j<xN;j++){
    /* obtain the confidence intervals */
    prob = calConf(&error, pr[j]);

    /* write the output files */
    fprintf(fc, "%lf,%lf,%lf,%lf\n", mu, x[j], prob, error);
    for(i=0;i<OBS;i++)
        fprintf(ft, "%lf,%lf,%e\n", mu, x[j], pr[j][i]);
}
fclose(in);
}

void main(){
    FILE *fc, *ft;
    /* path of the file containing the fGn sequence */
    char *in = "n22v1H0.80.txt";

    /* path of the output files: fc (csv file), ft (txt file) */
    fc = fopen("E:/WORK/Sim data/PR(Qt)/2/H08 s1 d0_01.csv","w");
    ft = fopen("E:/WORK/Sim data/PR(Qt)/2/H08 s1 d0_01 detail.csv","w");

    /* one example of obtaining the overflow probability */
    getProb(22,0.8,-0.5,fc,ft,in,1000,0.1,1,1,0.01);
    fclose(fc);
B.3 C Code for Obtaining Overflow Probabilities

fclose(ft);
}
