RESOURCE ALLOCATION AND PERFORMANCE GUARANTEES IN COMMUNICATION NETWORKS

by

Massieh Kordi Boroujeny
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Electrical and Computer Engineering

Committee:

_____________________ Dr. Brian L. Mark, Dissertation Director

_____________________ Dr. Yariv Ephraim, Committee Member

_____________________ Dr. Monson H. Hayes, Committee Member

_____________________ Dr. John F. Shortle, Committee Member

_____________________ Dr. Monson H. Hayes, Department Chair

Date: _____________________ Fall Semester 2022
George Mason University
Fairfax, VA

Resource Allocation and Performance Guarantees in Communication Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Massieh Kordi Boroujeny
Master of Science
Isfahan University of Technology, 2012
Bachelor of Science
Shahid Bahonar University of Kerman, 2009

Director: Dr. Brian L. Mark, Professor
Department of Electrical and Computer Engineering

Fall Semester 2022
George Mason University
Fairfax, VA

Copyright © 2022 by Massieh Kordi Boroujeny
All Rights Reserved

Dedication

I dedicate this dissertation to my niece and nephew, Heranoosh and Hirad. When I left to pursue my PhD they had just celebrated their first year birthday, and now that I have finished my thesis, they celebrated their seventh year birthday a couple of months ago. This dissertation is a summary of all the years I didn't get to be with them.

Acknowledgments

I would like to thank Dr. Brian L. Mark, my advisor, who made this work possible. I learned a lot from him during my PhD years, much more than just the academic knowledge I gained. He was more than an advisor to me, he was my mentor. I will be grateful to him indefinitely.

I would also like to thank Dr. Yariv Ephraim. I benefited a lot from his help and comments on our papers and learned much from the courses I took with him.

This work was supported in part by the U.S. National Science Foundation under Grant No. 1717033.

Table of Contents

List of Tables

List of Figures

Abstract

RESOURCE ALLOCATION AND PERFORMANCE GUARANTEES IN COMMUNICATION NETWORKS

Massieh Kordi Boroujeny, PhD

George Mason University, 2022

Dissertation Director: Dr. Brian L. Mark

One of the main challenges in modern communication networks, like the Internet, is providing performance guarantees, such as bounds on end-to-end delay, while avoiding underutilization of the network resources. In the early 1990s an approach to address this problem was proposed in which the input traffic was bounded either stochastically or deterministically by a so-called traffic envelope. Network calculus was developed to derive end-to-end delay bounds from the traffic burstiness bounds. Since deterministic network calculus can lead to loose bounds, our research focuses on stochastic network calculus.

In this dissertation, we address three open problems in applying stochastic network calculus to practical networks: 1) estimating an appropriate stochastic traffic envelope for an arbitrary traffic source; 2) enforcing conformance of a given traffic flow to a stochastic traffic envelope; 3) admission control based on an enforceable traffic envelope, while achieving statistical multiplexing gain. We develop a method to characterize an arbitrary traffic source by a traffic envelope that takes the form of a phase-type distribution. The versatility and generality of the phase-type distribution make it useful for obtaining tight bounds to characterize the traffic. We particularize a class of stochastic burtiness bounds using the proposed phase-type bounds. We also develop a stochastic traffic regulator that forces a

traffic flow to conform to a given traffic envelope from a class of traffic envelopes, including our proposed phase-type envelopes. We propose a new traffic envelope, referred to as the W-envelope, based on the moment generating function of the workload process obtained from offering the traffic to a constant service rate queue. We show how the W-envelope can be used in a QoS (quality of service) framework for providing stochastic end-to-end delay guarantees in conjunction with the proposed traffic characterization method and stochastic traffic regulator. Finally, we develop a new available bandwidth estimation (ABE) method that can provide accurate estimates of the available bandwidth on an end-to-end network path even in presence of packets dropped due to congestion. Our ABE method could be used to discover the amount of available bandwidth on an end-to-end, which could then be used to provide stochastic delay guarantees for time-sensitive traffic via our proposed QoS framework.

Chapter 1: Introduction

1.1 Motivation

One of the main challenges in data networks is providing end-to-end performance guarantees like, end-to-end delay. As the data network grows in size and provides service to a massive number of users, providing performance guarantee and avoiding under-utilization of the network resources becomes crucial. Using queuing theory we can extensively characterize the output traffic of a single node under general input and service processes. Such characterization, however, becomes intractable in the case of a network of nodes, unless unrealistic simplifying assumptions are made on the input traffic or service processes. Queuing theory fails to accurately address the case of bursty traffic, which is very common in today's Internet.

In an attempt to address the aforementioned shortcomes, network calculus was developed in pioneering work of Cruz [24, 25]. He bounded input traffic deterministically by the so-called (σ, ρ)-characterization. He then used such characterization and developed network calculus to derive a bound on end-to-end delay. However, since his method considered the worst case delay, in practice the end-to-end bounds were very loose. Therefore later on, Yaron and Sidi [97] developed stochastic bounds for the input traffic, specifically bounds in the form of exponential functions. The exponentially bounded burstiness (EBB) traffic envelope of Yaron and Sidi [97] was generalized to a larger class of bounding functions, resulting in a traffic envelope called stochastically bounded burstiness (SBB) [85]. This envelope was further modified in [51, 99] to the so-called generalized stochastically bounded burstiness (gSBB). Stochastic network calculus theorems associated with these traffic envelopes were developed to obtain stochastic bounds on end-to-end network delay. In theory, stochastic network calculus should provide much tighter end-to-end bounds and higher network utilization compared to deterministic network calculus.

In the literature on stochastic network calculus there has been relatively little work on specifying the bounding function itself, other then the exponential function proposed in [97], and the mixture of exponentials discussed in [85]. Motivated by the versatility of the family of the phase-type distributions, we specialize the bounding function into a form related to phase-type distribution. We also particularize a stochastic network calculus developed for general bounding functions, to this specific family of functions.

One of the main issues of the stochastic network calculus, which had not been addressed in prior work, is the question of how to estimate the traffic burstiness bounds for a given traffic trace. We develop two approaches to estimate phase-type bounds for a given traffic trace. Another problem that needs to be addressed is how to enforce a traffic stream to obey a given traffic burstiness bound. Based on the well-known of deterministic (σ, ρ) traffic shaper [24], we developed a stochastic traffic regulator that enforces the output traffic to conform to the so-called gSBB traffic envelope.

Stochastic network calculus involves the use of traffic envelopes to derive end-to-end performance metrics. To apply network calculus in practice, the traffic envelopes should: (i) be readily determined for an arbitrary traffic source, (ii) be enforceable by traffic regulation, and (iii) yield statistical multiplexing gain. Existing traffic envelopes typically satisfy at most two of these properties. The traffic envelope based on the moment generating function (MGF) of the arrival process satisfies only the third property. We propose a new traffic envelope, referred to as the W-envelope, based on the MGF of the workload process obtained from offering the traffic to a constant service rate queue. We show how the W-envelope is related to the gSBB traffic envelope. We show that W-envelope, together with its associated gSBB traffic envelope, satisfies all three properties and can be the basis of a framework for provisioning stochastic delay guarantees in a network.

The remainder of this dissertation is organized as follows. In Chapter 2, we review the literature on network calculus and we also cover phase-type distributions. In Chapter 3, we develop a network calculus based on phase-type distributions. In Chapter 4, we develop two methods of estimating the phase-type bounds for a given traffic source. In Chapter 5, we develop a stochastic traffic regulator to shape any given traffic flow into desired phase-type characterization. In Chapter 6, we propose and evaluate a framework, based on results from stochastic network calculus, for guaranteeing stochastic bounds on network delay at a statistical multiplexer. In Chapter 7, we propose a new traffic envelope, refereed to as W-envelope, based on the MGF of the workload process obtained from offering the traffic to a constant service rate queue. In Chapter 8, we propose a set of techniques to extend modern available bandwidth estimation methods in the presence of packet loss. Finally, the dissertation is concluded in Chapter 9.

1.2 Main Contributions

In this section we provide a summary of the main contributions of this dissertation.

1.2.1 Phase-type Traffic Bound

In Chapter 3, we developed a network calculus for PHBB and phase-type traffic bounds as particular cases of SBB and gSBB traffic envelopes developed in [51, 97, 99]. The main contributions in this chapter can be summarized as follows:

1. We develope phase-type bounded burstiness (PHBB) as a special cases of SBB traffic envelope using phase-type distribution.

2. We show that the network calculus theorems for SBB traffic envelopes hold for PHBB traffic envelopes. Therefore, characterization of all the input traffic flows to a network using PHBB bounds leads to characterization of all the traffic flows in the network using PHBB bounds. Such characterization can lead to stochastic bounds on end-to-end delay in the network.

3. We develop a new class of phase-type bounds as a special cases of gSBB traffic envelope using phase-type distribution. We introduced a threshold on the tail of the bound in

the definition and therefore making it possible to bound heavy-tailed traffic commonly seen in Internet using phase-type bounds.

4. We show that the network calculus theorems for gSBB traffic envelopes hold for phase-type traffic envelopes. Therefore, having characterized the input traffics to a network using phase-type bounds can result in characterization of all the traffics in the network using phase-type bounds. Such characterization can lead to stochastic bounds on end-to-end delay in the network. We Show the advantage of phase-type bounds versus PHBB bounds as the traffic is traversed through the network, the bounds will stay the same and will not become loose.

5. We show the versatility of phase-type bounds and the tightness they provide when the input traffic streams are assumed independent using a MMPP/G/1 traffic example.

1.2.2 Traffic Characterization Using Traffic Bounds

Having developed phase-type traffic bounds in Chapter 3, in Chapter 4 we addressed the first practical problem of using phase-type network calculus, namely, how to characterize an arbitrary traffic source. The main contributions of this chapter are summarized as follows:

1. We formulate the problem of finding the phase-type bound as an optimization problem and developed a least squares solution method. We simplified this optimization problem for the special cases of phase-type distribution, namely, hyperExponentials, mixture of Erlangs, and Canonical form I.

2. We develop an alternative algorithm for characterizing the traffic stream using phase-type bounds based on EM algorithm. The algorithm is based on generating the samples of the workload, when traffic is fed into a constant rate server, and fitting the samples to a phase-type distribution. In this algorithm we also use a special class of phase-type distribution, namely, mixture of Erlangs, which includes mixture of exponentials as a special case.

3. We develop an efficient EM algorithm for fitting a traffic trace to a special form of of phase-type distribution, namely, mixture of Erlangs. Our algorithm is a very efficient extension of the algorithm developed in [87].

4. We apply the two approaches to characterize a heavy-tailed traffic source using phase-type bounds. Our results showed that the least squares method outperformed the EM method.

1.2.3 Stochastic Traffic Regulator

In Chapter 5, we addressed the open problem of how to enforce conformance of an arbitrary traffic flow to a given gSBB bound. The main contributions of this chapter are summarized as follows:

1. We design a stochastic traffic regulator to shape an arbitrary traffic flow to conform to a given gSBB bound by extending the (σ, ρ) traffic regulator developed in [24].

2. We develop three alternative implementations of the designed stochastic traffic regulator. We show the first algorithms satisfies the desired bound asymptotically and second and the third ones satisfying the desired bound at all the times.

3. We present examples of shaping of Poisson traffic flow and a bursty traffic flow into the desired gSBB bound. We show that much lower delay is introduced by the stochastic traffic regulator in comparison to the (σ, ρ) traffic regulator.

1.2.4 A Framework for Providing Stochastic Delay Guarantee

In Chapter 6, we designed a practical framework for providing stochastic delay guarantees in communication networks utilizing stochastic network calculus developed in previous chapters. Specifically, our framework is based on the phase-type traffic envelope and the W-envelope. The main contributions of this chapter are summarized as follows:

1. We proposed the use of phase-type traffic envelopes to characterize traffic flows (Chapter 4) and the enforcement of these envelopes by a stochastic traffic regulator (Chapter

5). By doing so we can guarantee that, all traffic flows will conform to the negotiated stochastic bounds and the desired performance metrics of the traffic flows are not compromised by nonconforming traffic flows.

2. We propose an admission control scheme based on phase-type bounds and MGF bounds. By using network calculus and delay guarantees provided by phase-type and MGF bounds, such admission control only accepts new flows, when the desired performance metrics of the rest of the flows will not be violated.

3. We applied the hybrid admission control scheme to an example with traffic modeled by a Markov modulated Poisson process and demonstrate multiplexing gain achieved using MGF bounds.

1.2.5 Traffic Envelope and the Offered Multiplexing Gain

In Chapter 7, we introduced a new traffic envelope based on an exponential bound on MGF of the workload, when the traffic flow is offered as an input to a constant rate server. We showed this new traffic envelope, which we refer to as the W-envelope, can offer multiplexing gain when it is applied at a multiplexer, and more importantly as it is based on the workload, a given traffic flow can be characterized and enforced according to a W-envelope via an associated gSBB envelope. The main contributions of this chapter are summarized as follows:

1. We introduce the W-envelope and derive its properties, particularly the relationship between the W-envelope and the arrival MGF envelope.

2. We develop a framework for providing delay guarantees using the W-envelop. We show through the relationship between gSBB bounds and the W-envelop, how a given traffic flow can be characterized using a W-envelop. We also show by using a gSBB stochastic regulator we can enforce a given traffic to conform to its associated W-envelope.

3. We use the W-envelop to characterize two well-known traffic models, namely, Markov fluid model and MMPP model. We then showed how using the W-envelope we can achieve multiplexing gain for these traffic sources and how this envelop can be applied to admission control at a multiplexer to provide stochastic delay guarantee, meanwhile, providing multiplexing gain.

1.2.6 Available Bandwidth Estimation

In Chapter 8, we addressed the problem of estimating the available bandwidth (ABE) on an end-to-end path. Having such estimation can help us to provide the end-to-end delay guarantee using the methods introduced in previous chapters. Specifically, we aimed to address the problem of ABE using an active method by sending probes on the end-to-end path. However, one of the key issues that arises in such situation is losing some of the probes due to the congestion on the path. We formulated this problem mathematically and developed an approach to extend the existing estimation methods to address these situations. The main contributions of this chapter are summarized as follows:

1. We review the cases that lead to the lost probes, for example because of congested queues along the end-to-end path or because of aggressive automatic queuing managements (AQM) used along the path. We discuss how current ABE methods are not designed to address such cases.

2. We extend the formulation of the current decreasing-rate ABE methods to estimate the lost cross traffic during the estimation process, and therefore estimate the available bandwidth while accounting for the lost cross traffic.

3. We develop two alternative methods to solve the problem of ABE in the presence of lost packets. The first method is based on accounting for the received probes versus the sent probes. The second method however, estimates the lost cross traffic and leads to a more accurate estimate. The second method, however, needs an estimate of the bottleneck capacity.

4. We exhaustively tested our developed algorithm on a real testbed using Linux servers. Our tests were done on different range of channel bandwidth, i.e., 100 Mbps, 1 Gbps, and 10 Gbps. We show that our two methods significantly outperform the current existing ABE methods in all cases.

Chapter 2: Background and Literature Review

In this chapter we review some concepts that our work is based on: stochastic network calculus, the (σ, ρ) deterministic network calculus and (σ, ρ) traffic regulator, the phase-type distribution, and the EM algorithm.

The remainder of this chapter is organized as follows. In Section 2.1, we provide a review of stochastic network calculus definitions and theorems. In Section 2.3, we review the phase-type distribution and we cover some theorems about this class of random variables. Finally, in Section 2.4, we review the EM algorithm for hyper-Erlang distribution.

2.1 Stochastic Network Calculus

Stochastic network calculus was pioneered by Yaron and Sidi in [96, 97] and Chang in [22]. The main contribution of Yaron and Sidi was to characterize the input traffic as so-called EBB (exponential bounded burstiness), which in more accurate terms can be expressed in these two definitions.

Definition 2.1.1 (EB). A stochastic process $W = \{W(t) : t \geq 0\}$ is called exponentially bounded (EB) if there exist $\alpha \geq 0$ and $A \in [0, 1]$ such that

$$\mathsf{P}\{W(t) \geq \sigma\} \leq Ae^{-\alpha\sigma}, \tag{2.1}$$

for all $t \geq 0$ and all $\sigma \geq 0$.

Definition 2.1.2 (EBB). A traffic process $A(t)$ is said to have exponentially bounded burstiness (EBB) with upper rate ρ if there exist $\alpha \geq 0$ such that for all $t \geq s \geq 0$ and all $\sigma \geq 0$

$$\mathsf{P}\left\{A(s, t) \geq \rho(t - s) + \sigma\right\} \leq Ae^{-\alpha\sigma}, \tag{2.2}$$

where $A(s,t) := A(t) - A(s)$ is the amount of traffic that arrives in the interval $[s,t)$.

For a discrete-time traffic process $\{A_k : k = 0, 1, \ldots\}$, essentially the same definition of EBB applies, except that s and t are nonnegative integers. Using this definition, they were able to develop network calculus theorems and extend these characteristics to the output traffics and workload processes through the network. Therefore, they were able to bound the delay and other parameters of the interest in the network by exponential bounds.

For some traffic models, the exponential bound of EBB can be quite loose. Therefore, this bound was extended in [85] to employ a general bounding function.

Definition 2.1.3 (SB). A stochastic process $W(t)$ is called stochastically bounded (SB) if, for all $t \geq 0$ and all $\sigma \geq 0$

$$\mathsf{P}\{W(t) \ \geq \sigma\} \leq f(\sigma), \tag{2.3}$$

where $f(\sigma) \in \mathcal{F}$, and $\mathcal{F}$ is defined as the family of functions such that for every $n, \sigma \geq 0$, the n-fold integral $(\int_\sigma^\infty \mathrm{d}u)^n f(u)$ is bounded.

Definition 2.1.4 (SBB). A traffic process $A(t)$ is said to have stochastically bounded burstiness (SBB) with upper rate ρ and bounding function $f(\sigma)$ if, for all $t \geq s \geq 0$ and all $\sigma \geq 0$,

$$\mathsf{P}\left\{A(s,t) \geq \rho(t-s) + \sigma\right\} \leq f(\sigma), \tag{2.4}$$

where $f(\sigma) \in \mathcal{F}$, and $\mathcal{F}$ is defined in Definition 2.1.3.

Network calculus theorems developed for SBB generalize those for EBB. The network model considered in this context is a feedforward network that starts at $t = 0$ and all the network queues are empty at that time. Buffers are assumed to be infinite. The network is assumed to be a work-conserving system, which means that in every element of the network, no work is created or destroyed, and the server of the element never idles in the presence of a non-empty queue.

- The *SBB Characterization Theorem* [85, Theorem 1] considers a work-conserving system that transmits at a rate of ρ, fed with a traffic stream with traffic process $A(t)$.

If $W(t)$, the queue workload at time t, is SB with bounding function $f(\sigma)$, then the input traffic stream will have SBB with the same bounding function $f(\sigma)$ and upper rate ρ.

- The *SBB Sum Theorem* [85, Theorem 2] states that when two traffic streams $A_1(t)$ and $A_2(t)$ having SBB with bounding functions $f_1(\sigma)$ and $f_2(\sigma)$ and upper rates ρ_1 and ρ_2 are fed into a network element with constant service rate, the aggregate traffic process $A_1(t)+A_2(t)$ will also have SBB with upper rate $\rho_1+\rho_2$ and bounding function $g(\sigma) = f_1(p\sigma) + f_2((1-p)\sigma)$, where p is any value such that $0 < p < 1$.

- The *SBB Input-Output Relation Theorem* [85, Theorem 3] considers a traffic process $A_{\text{in}}(t)$ fed as input to a work-conserving network element that transmits at rate C. If $A_{\text{in}}(t)$ is having SBB with upper rate $\rho < C$ and bounding function $f(\sigma)$, then the queue workload process $W(t)$ and the output traffic process $A_{\text{out}}(t)$ have the following properties:

  i) $A_{\text{out}}(t)$ is having SBB with upper rate ρ and bounding function

$$g(\sigma) = f(\sigma) + \frac{1}{C-\rho}\int_{\sigma}^{\infty} f(u)\,\mathrm{d}u, \tag{2.5}$$

  ii) $W(t)$ is SB with the same bounding function as (2.5).

By the Sum Theorem, if a set of individual traffic streams are having SBB, their aggregated traffic stream will also have SBB. Then by the Input-Output Relation Theorem, $W(t)$ and $A_{\text{out}}(t)$ of these nodes will be SB and having SBB, respectively. Following the same steps, we can extend this further to other nodes, and eventually to the entire network. Thus, if the input traffic streams to the feedforward network can be characterized as having SBB, then the traffic streams in all links of the network and the queue workloads at all network elements can be characterized as having SBB and being SB, respectively.

The idea of SBB was further developed in [51, 99], by introducing the closely related concept of gSBB.

Definition 2.1.5 (gSBB). A traffic process $A(t)$ is said to have generalized stochastically bounded burstiness (gSSB) with upper rate ρ and bounding function $f(\sigma) \in \mathcal{BF}$ if, for all $t \geq 0$ and all $\sigma \geq 0$,

$$\mathsf{P}\left\{W(t) \geq \sigma\right\} \leq f(\sigma), \tag{2.6}$$

where,

$$W_\rho(t) = \max_{0 \leq s \leq t} \left\{A(s,t) - \rho(t-s)\right\}, \tag{2.7}$$

and $\mathcal{BF}$ is defined as the family of positive and non-increasing functions.

We should note that, by comparing Eqs. (2.4) and (2.6), the gSBB characterization is more restrictive than that of SBB. In other words, for a given bounding function, if a traffic process is gSBB then it is also SBB, but the converse may not hold. The inequivalence between SBB and gSBB concepts is explained in detail with a counter-example in Appendix A. There are some advantages for gSBB characterization over SBB, which makes it more useful. First of all, the class of bounding functions for gSBB is less restrictive than that for SBB. This is especially useful in characterizing heavy-tail traffics , which are very common in the Internet traffic [47, 49]. Secondly, in the definition of gSBB, the process $W(t)$ can be interpreted as the virtual workload of a constant rate queue with service rate ρ and input traffic R. This property is used later on in our work, as discussed in Chapter 4, to estimate the parameter of the gSBB traffic burstiness bound.

In [99] and [51], several network calculus theorems for gSBB traffics are developed that can be used to bound network delays using probabilistic bounds. The main ones are similar to the ones for SBB and are summarized as follows:

- The *gSBB Characterization Theorem* [99, Theorem 1] considers a work-conserving system that transmits at a rate of ρ, fed with a traffic process $A(t)$ and $W(t)$ is the queue workload at time t. Then $A(t)$ is having gSBB with upper rate ρ and bounding function f if and only if

$$\mathsf{P}\{W(t) \geq \sigma\} \leq f(\sigma). \tag{2.8}$$

- The *gSBB Sum Theorem* [99, Theorem 3] states that when two traffic streams $A_1(t)$ and $A_2(t)$, having gSBB with bounding functions $f_1(\sigma)$ and $f_2(\sigma)$ and upper rates ρ_1 and ρ_2 are fed into a network element, the aggregate traffic process $A_1(t) + A_2(t)$ will also have gSBB with upper rate $\rho_1 + \rho_2$ and bounding function $g(\sigma) = f_1(p\sigma) + f_2((1-p)\sigma)$, where p is any value such that $0 < p < 1$.

- The *gSBB Input-Output Relation Theorem* [99, Theorem 5] considers a traffic process $A_{\text{in}}(t)$ fed as input to a work-conserving network element that transmits at rate C. If $A_{\text{in}}(t)$ is having gSBB with upper rate $\rho < C$ and bounding function $f(\sigma)$, the output rate process $A_{\text{out}}(t)$ is gSBB with upper rate ρ and bounding function $f(\sigma)$.

One of the advantages of gSBB characterization lies in the last property, as the input and output traffics are having the same bounding function, whereas in SBB characterization, the bounding function of the output and workload will be added by an integration part as in (2.5). Therefore, as we go deeper in the network nodes, the bounding function in SBB characterization becomes looser, whereas in gSBB characterization, the same tightness of the bound is kept throughout the network.

2.2 (σ, ρ)-bounded Traffics and (σ, ρ) Regulator

2.2.1 (σ, ρ)-bounded Traffics

Cruz in his pioneering work [24] on deterministic network calculus, defined the (σ, ρ)-bounded traffic as a traffic stream which is bounded by a linear function in any interval. More precisely,

Definition 2.2.1. Given $\sigma \geq 0$, and $\rho \geq 0$, A traffic process $A(t)$ is said to (σ, ρ)-bounded, and is denoted as $A \sim\ (\sigma, \rho)$, if and only if

$$A(s,t) \leq \rho(t-s) + \sigma \tag{2.9}$$

for all $t \geq s \geq 0$.

In other words, if $A \sim\ (\sigma, \rho)$, there is an upper bound on the amount of the traffic received in any interval. If we define $W_\rho(t)$ as (2.7), therefore $W_\rho(t) \leq \sigma$, if and only if $A \sim\ (\sigma, \rho)$. This property is further used to regulate the traffic, or in other words, for a given $\sigma, \rho \geq 0$ enforce the traffic to $A \sim\ (\sigma, \rho)$.

2.2.2 (σ, ρ) Regulator

The (σ, ρ) regulator was introduced in [24] and was used in [25] to decrease the end-to-end delay by shaping the traffic. The concept of (σ, ρ) regulator is closely related to leaky bucket, which has been extensively studied in the literature [17, 18, 90, 91]. The idea of the (σ, ρ) regulator is to delay the incoming packets long enough, such that the delayed traffic or so-called output of the regulator is (σ, ρ)-bounded.

When a certain performance guarantee, such as a maximum end-to-end delay, is demanded in the network, the input traffic at different stages of the network should be shaped to enforce the desired guarantees. Except the work of the Cruz [24, 25], which is a case of deterministic traffic shaper, namely, it enforces the output traffic to always be (σ, ρ)-bounded, there has been relatively little work to develop different forms of traffic regulator. In Chapter 5, we propose a stochastic traffic shaper, which shapes the traffic such that the result is stochastically (σ, ρ)-bounded.

We assume in a (σ, ρ) traffic regulator the input and output link have a capacity of C bits/s. Therefore, the input traffic rate, $\frac{\mathrm{d}\,A(t)}{\mathrm{d}\,t}$, can be expressed as

$$\frac{\mathrm{d}\,A(t)}{\mathrm{d}\,t} = C \sum_{j=1}^{\infty} I_{\{s_j \leq t < s_j + L_j/C\}} \tag{2.10}$$

where s_j is the arrival time of the jth packet and L_j is the length of the jth packet in bits. It is assumed a packet will not arrive while the previous one is being received. In other words, we have $s_j + L_j/C \leq s_{j+1}$. Let t_j be the time that jth packet leave the traffic regulator, and $R_{\mathrm{o}}(t)$ the traffic process exiting the traffic regulator. A (σ, ρ) traffic regulator transmits the packets on the output link in FCFS order such as

$$W_\rho(t_j) \leq \sigma \tag{2.11}$$

$W_\rho(t_j)$ is defined in (2.7). If we define $d_j = t_j - s_j \geq 0$ as the delay which the jth packet suffers in a (σ, ρ) traffic regulator, it is shown in [24] that

$$d_j = \frac{1}{\rho}(W_\rho(s_j) - \sigma)^+, \tag{2.12}$$

for all j, where $x^+ := \max\{x, 0\}$. For the output traffic process, $A_{\mathrm{o}}(t)$, we have

$$A_{\mathrm{o}} \sim (\sigma + (1 - \rho/C)L, \rho), \tag{2.13}$$

where $L = \max_j L_j$.

2.3 Phase-Type Distribution

The phase-type distribution is defined in terms of a Markov chain $X = \{X(t) : t \geq 0\}$ with state space $E = \{1, 2, \ldots, n, n+1\}$, where states $1, 2, \ldots, n$ are transient states and $n+1$ is an absorbing state. The generator of X has the form [11]

$$\begin{pmatrix} \mathbf{Q} & \mathbf{q} \\ \mathbf{0} & 0 \end{pmatrix}, \tag{2.14}$$

where $\mathbf{Q} = [q_{ij} : i, j = 1, \ldots, n]$ is an $n \times n$ matrix such that q_{ij} is the transition rate from state i to state j and $\mathbf{q} = \mathrm{col}(q_1, \ldots, q_n)$ such that q_i is the transition rate from transient state i to the absorbing state $n + 1$. The submatrix $\mathbf{Q}$ is invertible and the vector $\mathbf{q}$ is related to $\mathbf{Q}$ as follows:

$$\mathbf{q} = -\mathbf{Q}\mathbf{1}, \tag{2.15}$$

where $\mathbf{1}$ denotes a column vector of ones of the appropriate dimension, which is n in this case. Define $\pi_i = \mathsf{P}(X(0) = i)$ for $i = 1, \ldots, n + 1$ and the vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$. Hence, the initial distribution of X is given by $(\boldsymbol{\pi}, \pi_{n+1})$, where π_{n+1} is the probability that the chain starts in the absorbing state. Let $\tau = \inf\{t \geq 0 | X(t) = n + 1\}$ be the time until absorption of the Markov process X. The random variable τ is phase-type with parameter $(\boldsymbol{\pi}, \mathbf{Q})$:

$$\tau \sim \mathrm{PH}_n(\boldsymbol{\pi}, \mathbf{Q}). \tag{2.16}$$

In this case, the probability density function, cumulative distribution function and survival function of τ are given, respectively, by

$$f_\tau(t) = -\boldsymbol{\pi} e^{\mathbf{Q}t}\mathbf{Q}\mathbf{1}, \quad t \geq 0 \tag{2.17}$$

$$F_\tau(t) = 1 - \boldsymbol{\pi} e^{\mathbf{Q}t}\mathbf{1}, \quad t \geq 0 \tag{2.18}$$

$$S_\tau(t) = \mathsf{P}(\tau > t) = 1 - F_\tau(t) = \boldsymbol{\pi} e^{\mathbf{Q}t}\mathbf{1}, \quad t \geq 0. \tag{2.19}$$

The Laplace transform of τ is given by

$$\begin{aligned} M_\tau(s) :&= \mathsf{E}\left\{e^{-s\tau}\right\} = \pi_{n+1} + \boldsymbol{\pi}[sI - \mathbf{Q}]^{-1}\mathbf{q} \\ &= \pi_{n+1} + \frac{N(s)}{D(s)}, \end{aligned} \tag{2.20}$$

where I denotes an identity matrix of appropriate dimension, in this case $n \times n$, and

$$D(s) = \det(sI - \mathbf{Q}) = \prod_{i=1}^{n}(s + \gamma_i), \tag{2.21}$$

where γ_i being the generally complex-values eigenvalues of $\mathbf{Q}$. On the other hand, $N(s)$ is in general, a polynomial of order $p-1$, therefore not considering the mass at absorbing state, π_{p+1}, we need $2p$ parameters to represent a phase-type random variable. However, by using $\boldsymbol{\pi}$ and $\mathbf{Q}$ we need p^2+p-1 parameters to represent the same phase-type random variable. Therefore, representation of a phase-type random variable using (2.16) is not unique [77]. The expected value of the phase-type random variable τ is given by

$$\mathsf{E}\{\tau\} = -\boldsymbol{\pi}\mathbf{Q}^{-1}\mathbf{1}. \tag{2.22}$$

The transition probabilities among the transient states of X are given by

$$\mathsf{P}(X(t) = j, \tau > t \mid X(0) = i) = \left[e^{\mathbf{Q}t}\right]_{ij}, \tag{2.23}$$

where $i, j \in \{1, 2, \dots, n\}$. As the states $1, 2, \dots, n$ are transient, we have

$$\lim_{t\to\infty}\left[e^{\mathbf{Q}t}\right]_{ij} = 0, \tag{2.24}$$

The family of phase-type distributions is closed under convolution and mixture operations (see [11], Theorems 3.1.26 and 3.1.27, respectively). Suppose, for example, that $\tau_1 \sim \mathrm{PH}_n(\boldsymbol{\alpha}, \mathbf{G})$ and $\tau_2 \sim \mathrm{PH}_m(\boldsymbol{\beta}, \mathbf{H})$ and τ_1 and τ_2 are independent. Then $\tau_{\text{sum}} = \tau_1 + \tau_2$ is a phase-type random variable with $n+m$ transient states such that

$$\tau_{\text{sum}} \sim \mathrm{PH}_{m+n}\left((\boldsymbol{\alpha}, \mathbf{0}), \begin{pmatrix} \mathbf{G} & \mathbf{g}\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{H} \end{pmatrix}\right), \tag{2.25}$$

where $\mathbf{g} = -\mathbf{G}\mathbf{1}$. Thus, if $X_1, X_2, \ldots, X_n$ are independent exponential random variables with $X_i \sim \exp(\lambda_i)$, $i = 1, \ldots, n$, then the distribution of the sum $\tau = X_1 + X_2 + \ldots + X_n$ is given by $\tau \sim \mathrm{PH}_n(\boldsymbol{\pi}, \mathbf{Q})$ where

$$\mathbf{Q} = \begin{pmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & -\lambda_3 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \lambda_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & -\lambda_n \end{pmatrix}, \qquad (2.26)$$

$$\boldsymbol{\pi} = (1, 0, 0, \cdots, 0).$$

Next consider a mixture of phase-type distributions defined by

$$\tau_{\text{mix}} = \begin{cases} \tau_1 & \text{with probability } p, \\ \tau_2 & \text{with probability } 1-p, \end{cases}$$

where $p \in [0, 1]$. Then

$$\tau_{\text{mix}} \sim \mathrm{PH}_{n+m}\left((p\boldsymbol{\alpha}, (1-p)\boldsymbol{\beta}), \begin{pmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{pmatrix} \right). \qquad (2.27)$$

In particular, if τ is a random variable such that with probability π_i, τ is exponentially distributed with parameter λ_i for $i = 1, \ldots, n$, then τ is called mixture of exponentials and

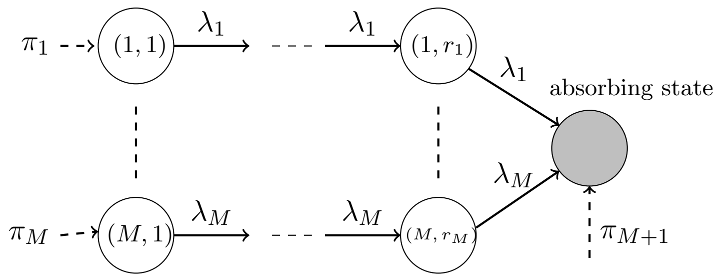


Figure 2.1: Hyper-Erlang form of a Phase-type random variable

we have $\tau \sim \mathrm{PH}_n(\boldsymbol{\pi}, \mathbf{Q})$, where

$$\mathbf{Q} = \mathrm{diag}\{-\lambda_1, -\lambda_2, \ldots, -\lambda_n\}, \tag{2.28}$$

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n). \tag{2.29}$$

By limiting $\mathbf{Q}$ to special forms we can achieve other special forms of phase-type random variable such as Erlang random variables, Hyper-Erlang random variable, or acyclic Phase-type random variables. In case of Hyper-Erlang random variable the associated generating Markov chain can be represented as Fig. 2.1. In this case every branch is a summation of independent identically distributed exponentials where results in Erlang random variables at every branch and the phase-type random variable itself is a mixture of these Erlang random variables. We should note in this form no mass is considered at absorbing state or $\pi_{n+1} = 0$. In this case $\mathbf{Q}$ will be

$$\mathbf{Q} = \mathrm{diag}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_M\}, \tag{2.30}$$

$$\mathbf{q}_i = \begin{pmatrix} -\lambda_i & \lambda_i & 0 & \ldots & 0 \\ 0 & -\lambda_i & \lambda_i & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & -\lambda_i & \lambda_i \\ 0 & 0 & \ldots & 0 & -\lambda_i \end{pmatrix}_{r_i \times r_i} \tag{2.31}$$

where r_i is the order of the Erlang distribution in the ith branch. On the other hand, $\boldsymbol{\pi}$ will be

$$\boldsymbol{\pi} = (\pi_1, \underbrace{0, \ldots, 0}_{r_1 - 1}, \pi_2, \underbrace{0, \ldots, 0}_{r_2 - 1}, \ldots, \pi_M, \underbrace{0, \ldots, 0}_{r_M - 1}, \pi_{M+1}) \tag{2.32}$$

In this case probability density function for τ is given by

$$f_\tau(t) = \sum_{i=1}^{M} \pi_i \frac{(\lambda_i t)^{r_i - 1}}{(r_i - 1)!} \lambda_i e^{-\lambda_i t} \tag{2.33}$$

Limiting the phase-type random distribution to Hyper-Erlang gives the representation of the phase-type distribution using $3M$ parameters of $\mathbf{r} = (r_1, r_2, \ldots, r_M) \in \mathbb{N}^M$, $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_M) \in \mathbb{R}^M$, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_M) \in \mathbb{R}^M$, and M as the number of mixture components.

On the other hand by limiting $\mathbf{Q}$ to an upper-triangular matrix we can achieve an Acyclic Phase-Type random variable. What is special about Hyper-Erlang random variables and Acyclic Phase-Type random variables is that, not only general phase-type distributions are dense in the set of densities with nonnegative support and every density function in this set can be approximated arbitrarily close by a general phase-type distribution [7, Theorem 4.2] [95, Theorem 5.2], also Hyper-Erlang distributions [33] [7, Corollary 4.4] and Acyclic Phase-type distributions [26] are dense in the set of densities with nonnegative support.

In [26] it is shown every Acyclic continuous-time phase-type random variable can be represented in the Canonical Form 1(CF1) such that the associated generating markov chain can be represented as Fig. 2.2. where,

$$\pi_i \geq 0 \quad i = 1, 2, \ldots, p+1 \qquad , \sum_{i=1}^{p+1} \pi_i = 1$$

$$\lambda_p \geq \lambda_{p-1} \geq \ldots \geq \lambda_2 \geq \lambda_1 \geq 0, \tag{2.34}$$

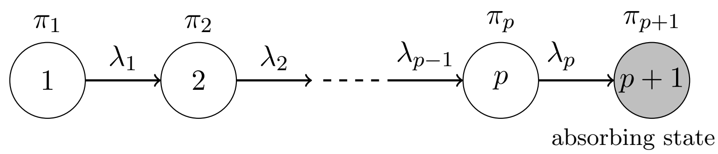


Figure 2.2: Canonical Form 1(CF1) of an Acyclic Phase-type random variable

In this case $\mathbf{Q}$ will be

$$\mathbf{Q} = \begin{pmatrix} -\lambda_1 & \lambda_1 & 0 & \dots & 0 \\ 0 & -\lambda_2 & \lambda_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -\lambda_{p-1} & \lambda_{p-1} \\ 0 & 0 & \dots & 0 & -\lambda_p \end{pmatrix} \tag{2.35}$$

When there is no mass at absorbing state $\pi_{p+1} = 0$. Laplace transform of the CF1 random variable will be [12]

$$M_\tau(s) = \pi_{p+1} + \frac{N(s)}{D(s)}, \quad D(s) = \prod_{i=1}^{p}(s + \lambda_i), \tag{2.36}$$

2.4 EM Algorithm for Hyper-Erlang Distribution

We have developed two methods to derive the phase-type bounding parameters. The second method is based on an EM algorithm for a special class of phase-type distributions, i.e., a mixture of Erlang distribution, also called the hyper-Erlang distribution.

The EM algorithm is essentially an optimization method trying to minimize the cross-entropy, $\int_0^\infty f(t)\log(\frac{f(t)}{\hat{f}(t)})\mathrm{d}t$, between a known probability density dunction, $f(t)$, and fiting probability density function, $\hat{f}(t)$. This minimization is equivalent to maximizing the likelihood function $\int_0^\infty \log(\hat{f}(t))\mathrm{d}t$ of the fitting pdf $\hat{f}(t)$. EM algorithm, however, does this by estimating the parameters of the $\hat{f}(t)$ from a given set of data trace which may be incomplete or have missing values [29, 73].

There are numerous works in the literature for fitting a data set to a phase-type distribution using EM algorithm such as pioneering work by Asmussen [8] for general forms of phase-type distribution. This work was further improved in [75]. Some authors also developed EM algorithm to fit the data to special forms of the phase-type distribution like the hyper-Erlang distribution [87] or mixture of exponentials [53]. There have been also some work on dividing the samples into different clusters and the fitting different clusters with special forms of phase-type distributions like the hyper-Erlang distribution [83, 93], or mixture of CF1, described in 2.3, and mixture of exponentials [44]. In this section we present a slight modification to the algorithm presented in [87] that fits the data with Hyper-Erlang distribution. This algorithm is further improved in section 4, for finding the parameters of the bounding function in form of phase-type distributions.

The pdf of an Erlang distribution with parameter (r, λ) is given by

$$f(x; r, \lambda) = \frac{(\lambda x)^{r-1}}{(r-1)!}\lambda e^{-\lambda x}, \quad x \geq 0, \tag{2.37}$$

which can be seen as the convolution of r pdfs of an exponential distribution with rate λ. The pdf of an Erlang mixture model with parameter $\mathbf{\Theta} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$ is given by

$$f(x; \mathbf{\Theta}) = \sum_{i=1}^{M} \pi_i f(x; r_i, \lambda_i). \tag{2.38}$$

In our case of traffic stream, however, when traffic is fed to a constant rate server with rate C and workload is sampled, sometimes the workload is empty or some samples are 0. In queuing theory usually we have $\mathsf{P}\{W(t) = 0\} = 1 - \rho$, where ρ is the utilization factor of the queue. Having a phase-type fit with a non-zero mass at absorbing state has not been addressed in the works in this area, such as [8,87]. To allow for a positive probability value at $x = 0$, the Erlang mixture model can be extended by introducing an additional mixture probability π_{M+1} such that $\sum_{i=1}^{M+1} \pi_i = 1$. In other words, in Fig. 2.1, $\pi_{M+1} \neq 0$. The pdf of the extended Erlang mixture model is given by

$$\tilde{f}(x; \mathbf{\Theta}) = \sum_{i=1}^{M} \pi_i f(x; r_i, \lambda_i) + \pi_{M+1}\delta(x), \tag{2.39}$$

where $\delta(x)$ denotes the Dirac delta function.

Due to the term involving $\delta(x)$, the pdf in (2.39) cannot be used as a likelihood function for parameter estimation. Instead, we shall use the Radon-Nikodym derivative of the probability law of the extended Erlang mixture model specified by (2.39) with respect to the measure $\nu = \lambda + \delta_x$, where μ and δ_x denote the Lebesgue and Dirac measures (see, e.g., [21]), respectively:

$$p(x; \mathbf{\Theta}) = \begin{cases} f(x; \mathbf{\Theta}), & x \neq 0, \\ \pi_{M+1}, & x = 0. \end{cases} \tag{2.40}$$

Therefore, the parameter of the hyper-Erlang distribution will be extended to $\mathbf{\Theta} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda}) = (\pi_1, \pi_2, \ldots, \pi_M, \pi_{M+1}, r_1, r_2, \ldots, r_M, \lambda_1, \lambda_2, \ldots, \lambda_M)$. Under the (extended) Erlang mixture model, the log-likelihood of a sample vector $\mathbf{x} = (x_1, \ldots, x_K)$ is given by

$$\begin{aligned} \ell(\mathbf{\Theta} \mid \mathbf{x}) = \log p(\mathbf{x}; \mathbf{\Theta}) &= \log \prod_{k=1}^{K} p(x_k; \mathbf{\Theta}) \\ &= \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log \left(\sum_{i=1}^{M} \pi_i f(x_k; r_i, \lambda_i) \right) + K_0 \log(\pi_{M+1}), \end{aligned} \tag{2.41}$$

where $K_0 = \#\{k : x_k = 0; k = 1, \ldots, K\}$ denotes the number of 0 samples. In a similar way, the Erlang mixture model can be extended to accommodate positive probabilities at any finite set of points on the positive real line.

The EM algorithm aims to find $\mathbf{\Theta}$ that maximizes the log-likelihood $\ell(\mathbf{\Theta} \mid \mathbf{x})$. Note that, $\mathbf{r}$ in $\mathbf{\Theta} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$ is fixed and chosen in advance. This maximization is complicated by the summation inside the logarithm in (2.41). Therefore, we introduce unobserved data $\mathbf{y} = (y_1, \ldots, y_K)$, where $y_k \in \{1, \ldots, M+1\}$ denotes the mixture component corresponding to the observed sample x_k. We shall assume that the unobserved sample $y_k = M+1$ when $x_k = 0$. The joint likelihood of (x_k, y_k) is given by

$$p(x_k, y_k; \mathbf{\Theta}) = \pi_{y_k} p(x_k; \mathbf{\Theta}). \tag{2.42}$$

The complete data log-likelihood is then given by

$$\begin{aligned}\ell(\mathbf{\Theta} \mid \mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}, \mathbf{y}; \mathbf{\Theta}) &= \log \prod_{k=1}^{K} p(x_k, y_k; \mathbf{\Theta}) \\ &= \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\pi_{y_k} f(x_k; r_{y_k}, \lambda_{y_k})) + K_0 \log(\pi_{M+1}). \end{aligned} \tag{2.43}$$

The EM algorithm involves maximization of the parameter $\mathbf{\Theta}$ over an auxiliary function defined as the expectation of the complete data log-likelihood given the observed data $\mathbf{x}$ with respect to the current parameter estimate $\hat{\mathbf{\Theta}} = (\hat{\boldsymbol{\pi}}, \mathbf{r}, \hat{\boldsymbol{\lambda}})$:

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}) := E\left[\ell(\mathbf{\Theta} \mid \mathbf{x}, \mathbf{Y}) \mid \mathbf{x}; \hat{\mathbf{\Theta}}\right] = \sum_{\mathbf{y} \in \{1, \ldots, M+1\}^K} p(\mathbf{y} \mid \mathbf{x}; \hat{\mathbf{\Theta}}) \ell(\mathbf{\Theta} | \mathbf{x}, \mathbf{y}) \tag{2.44}$$

where $\mathbf{Y}$ denotes the random vector corresponding to the realization $\mathbf{y}$ of the unobserved data. Posterior probability $\{\mathbf{Y} = \mathbf{y}\}$, given the observed sample vector $\mathbf{x}$, and current estimate of the parameters, $\hat{\mathbf{\Theta}}$ is given by,

$$p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{\Theta}}) = \prod_{k=1}^{K} p(y_k|x_k, \hat{\mathbf{\Theta}}) \tag{2.45}$$

When $x_k \neq 0$, the posterior probability $p(y_k \mid x_k; \hat{\mathbf{\Theta}})$ is given by

$$p(y_k \mid x_k; \hat{\mathbf{\Theta}}) = \frac{\hat{\pi}_{y_k} f(x_k; r_{y_k}, \hat{\lambda}_{y_k})}{\sum_{i=1}^{M} \hat{\pi}_i f(x_k; r_i, \hat{\lambda}_i)}, \tag{2.46}$$

where $y_k \in \{1, \ldots, M\}$. Substituting (2.45) into (2.44), and performing some algebraic manipulations (see [87, Appendix A.2]), the following expression for the auxiliary function can be obtained:

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}) = K_0 \log(\pi_{M+1}) + \sum_{i=1}^{M} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\mathbf{\Theta}}) \log(\pi_i)$$

$$+ \sum_{i=1}^{M} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\mathbf{\Theta}}) \log(f(x_k; r_i, \lambda_i)). \tag{2.47}$$

Considering the expression for $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ in (2.47), it can be seen that maximization with respect to (π_i, λ_i) is independent of (π_j, λ_j) for $j \neq i$. Further, maximization of $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ with respect to $\boldsymbol{\pi}$ is independent of maximization with respect to $\boldsymbol{\lambda}$. Thus, maximization of $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ can be performed separately for each mixture component. During each iteration of the EM algorithm, the auxiliary function $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ is optimized first with respect to $\boldsymbol{\pi}$ and then with respect to the parameter λ_i of each Erlang mixture component, for $i = 1, \ldots, M$.

Since $\mathbf{r}$ is fixed, like [87], maximization of $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ with respect to $\mathbf{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\lambda})$ can be done using the method of Lagrange multipliers subject to the normalization constraint for the mixture probabilities $\{\pi_1, \ldots, \pi_{M+1}\}$. The local optimum solution is given by

$$\pi_i = \frac{1}{K} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\mathbf{\Theta}}), \quad i = 1, \ldots, M, \tag{2.48}$$

$$\pi_{M+1} = \frac{K_0}{K}, \tag{2.49}$$

$$\lambda_i(r_i) = r_i \cdot \frac{\sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\mathbf{\Theta}})}{\sum_{\substack{k=1 \\ x_k \neq 0}}^{K} x_k p(i \mid x_k; \hat{\mathbf{\Theta}})}, \quad i = 1, \ldots, M, \tag{2.50}$$

where in (2.50), we have written λ_i explicitly as a function of r_i. We note that π_{M+1} corresponds to the probability of a 0 sample and remains fixed for all EM iterations.

In [87], number of Erlang branches, M, and order of each branch, $(r_1, r_2, \ldots, r_M)$, are chosen from set $\mathcal{R}_n$ defined as

$$\mathcal{R}_n = \{(r_1, r_2, \ldots, r_M) | r_1 + r_2 + \ldots + r_M = n, r_i \geq 0, \text{for } i = 1, 2, \ldots, M\} \tag{2.51}$$

For each branch branch order $\mathbf{r} = (r_1, r_2, \ldots, r_M)$, at each iteration of EM algorithm after (π_i, λ_i) has been determined for all mixture components, $i = 1, \ldots, M$, according to the procedure described above, the Erlang mixture parameter estimate is updated as $\hat{\mathbf{\Theta}} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$, and the EM algorithm iteration is completed. The EM algorithm is continued until the relative difference between the log-likelihoods, given by (2.41), of the last two parameter estimates falls below a threshold ϵ or limit, N, on the maximum number EM iterations is reached. The proposed EM algorithm is outlined in Algorithm 1. After going exhaustively through $\mathcal{R}_n$ and performing EM algorithm for each $\mathbf{r} \in \mathcal{R}_n$, the branch order with the highest log-likelihood, given by (2.41), is chosen. This algorithm is summarized in a pseudo-code in Algorithm 1.

Algorithm 1 Calculate $\mathbf{Q}, \boldsymbol{\pi}$ using EM algorithm

Input: Workload samples and n

Output: $\mathbf{Q}, \boldsymbol{\pi}$

Set $\pi_{M+1} = \frac{K_0}{K}$

repeat

  Choose $(r_1, r_2, \ldots, r_M)$ from $\mathcal{R}_n$ as in (2.51)

  Choose initial estimate $\hat{\Theta} = (\hat{\pi}_1, \hat{\pi}_2, \ldots, \hat{\pi}_M, r_1, r_2, \ldots, r_M, \hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_M)$

  repeat

    Compute $f(x_k; r_i, \hat{\lambda}_i)$ for $i = 1, 2, \ldots, M,\ k = 1, 2, \ldots, K,\ x_k \neq 0$ according to (2.37).

    E-step: Compute the pmf of the unobserved data for $i = 1, 2, \ldots, M,\ k = 1, 2, \ldots, K,\ x_k \neq 0$ as.

$$p(y_k \mid x_k; \hat{\mathbf{\Theta}}) = \frac{\hat{\pi}_{y_k} f(x_k; r_{y_k}, \hat{\lambda}_{y_k})}{\sum_{i=1}^{M} \hat{\pi}_i f(x_k; r_i, \hat{\lambda}_i)},$$

    M-step: Choose π_i and λ_i that maximizes (2.47) for $i = 1, 2, \ldots, M$ as

$$\pi_i = \frac{1}{K} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} q(i|x_k, \hat{\mathbf{\Theta}}) \text{ and}$$

$$\lambda_i = r_i \cdot \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} q(i|x_k, \hat{\mathbf{\Theta}}) / \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} q(i|x_k, \hat{\mathbf{\Theta}}) \cdot x_k$$

    $\hat{\mathbf{\Theta}} \leftarrow \mathbf{\Theta}$

  until relative difference of log-likelihood in (2.41) is less than $\epsilon = 10^{-6}$

until All order combinations of $(r_1, r_2, \ldots, r_M)$ from $\mathcal{R}_n$ are tried

Choose the one order combination $(r_1, r_2, \ldots, r_M)$ and the derived $\boldsymbol{\pi}, \boldsymbol{\lambda}$ with the highest log-likelihood as in (2.41)

$\mathbf{Q}$ is derived based on $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_M]$ according to (2.30), and (2.31) **return** $\mathbf{Q}, \boldsymbol{\pi}$

Chapter 3: Phase-Type Bounded Burstiness

In this section, we develop phase-type network delay bounds based on the SBB and gSBB calculus developed in [51, 85, 99]. The phase-type bounds provide a useful specialization of the SBB and gSBB bounds. The class of phase-type distributions has the important property of being dense in the family of distributions of nonnegative random variables; i.e., the distribution of any random variable taking values in $[0, \infty)$ can be approximated arbitrarily closely by a phase-type distribution [7, Theorem 4.2] [95, Theorem 5.2]. In addition, phase-type distributions are mathematically tractable and form a closed set with respect to operations such as convolutions or mixtures. We use properties of phase-type random variables to relate bounds on the input traffic to a network element to bounds on the queue workload as well as bounds on the output traffic. Parts of the work in this chapter were published in [56, 61].

The remainder of the chapter is organized as follows. In Section 3.1, we extend the concepts of SBB and gSBB to PHBB and phase-type bounds. In Section 3.2, we extend the theorems on SBB bounds to PHBB bounds. In Section 3.3, we provide some theorems on the application of PHBB bounds in providing stochastic network performance guarantees. In Section 3.4, we extend the theorems on gSBB to the case of phase-type bounds. In Section 3.5, we provide some theorems on the application of phase-type bounds in providing stochastic network performance guarantees. In Section 3.6, we investigate the stochastic performance guarantees provided by gSBB and phase-type bounds. Concluding remarks are given in Section 3.7.

3.1 Concepts

In this section we provide the definitions of the developed phase-type bounded burstiness (PHBB) and phase-type bounds. These concepts are particular cases of SBB and gSBB, respectively, with an additional parameter component to limit the tail of the bound.

Definition 3.1.1. A stochastic process $W(t)$ is phase-type bounded (PHB) with bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a, T)$ where $a \geq 0$, and $(\boldsymbol{\pi}, \mathbf{Q})$ are the parameters of a phase-type random variable such that

$$\mathsf{P}\{W(t) \ \geq \sigma\} \leq a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}, \tag{3.1}$$

for all $t \geq 0$ and all $T \geq \sigma \geq 0$. When $T = \infty$, the PHB bounding parameter is written simply as $(\boldsymbol{\pi}, \mathbf{Q}, a)$.

Definition 3.1.2. A traffic process $A(t)$ is said to have phase-type bounded burstiness (PHBB) with upper rate ρ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a, T)$ if

$$\mathsf{P}\left\{A(s,t) \geq \rho(t-s) + \sigma\right\} \leq A\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}, \tag{3.2}$$

for all $t \geq s \geq 0$ and all $T \geq \sigma \geq 0$, where $A(s,t) := A(t) - A(s)$ is the amount of traffic that arrives in the interval $[s,t)$. Also in this definition, when $T = \infty$, the PHBB bounding parameter is written simply as $(\boldsymbol{\pi}, \mathbf{Q}, a)$.

Next, we show that phase-type bounding functions belong to the family of functions $\mathcal{F}$ defined immediately after (2.1.3).

Theorem 3.1.1. Let $(\boldsymbol{\pi}, \mathbf{Q}, a)$ be a phase-type bounding parameter. Then $f(\sigma) = a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}$ is monotonically decreasing and $f \in \mathcal{F}$.

Proof. Since $(\boldsymbol{\pi}, \mathbf{Q})$ is the parameter of a phase-type distribution, the function $S(\sigma) = \boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}$ is the associated survival function, which by definition is monotonically decreasing. Therefore, $f(\sigma)$ is a monotonically decreasing function. To show that $f \in \mathcal{F}$, we need to show that $(\int_\sigma^\infty \mathrm{d}u)^n f(u)$ is bounded. For the phase-type bounding function we have

$$\int_\sigma^\infty a\boldsymbol{\pi} e^{\mathbf{Q}u}\mathbf{1}\mathrm{d}u = a\boldsymbol{\pi}\int_\sigma^\infty e^{\mathbf{Q}u}\,\mathrm{d}u\ \mathbf{1} = a\boldsymbol{\pi}\left[\mathbf{Q}^{-1}e^{\mathbf{Q}u}\right]_\sigma^\infty\ \mathbf{1} = -a\boldsymbol{\pi}\mathbf{Q}^{-1}e^{\mathbf{Q}\sigma}\mathbf{1}. \tag{3.3}$$

From (2.24), $\lim_{u\to\infty} e^{\mathbf{Q}u} = \mathbf{0}$. Hence, the right-hand side of (3.3) is bounded. Repeating this argument $n-1$ more times shows that $(\int_\sigma^\infty \mathrm{d}u)^n f(u)$ is bounded. □

Corollary 3.1.1. For phase-type bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a, T)$ and $f(\sigma) = a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$ for $T \geq \sigma \geq 0$, we also have $f \in \mathcal{F}$.

As mentioned earlier we also particularize the gSBB concept to bounds based on phase-type distributions.

Definition 3.1.3. A traffic process $A(t)$ has tail-limited general phase-type bounded burstiness (gPHBB) with upper rate ρ and bounding parameter $(a, \boldsymbol{\pi}, \mathbf{Q}, T)$ if

$$\mathsf{P}\left\{W_\rho(t; A) \geq \sigma\right\} = \mathsf{P}\left\{\max_{0\leq s\leq t}\left\{A(s,t) - \rho(t-s)\right\} \geq \sigma\right\} \leq a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}, \tag{3.4}$$

for all $t \geq 0$ and all $T \geq \sigma \geq 0$. Again, when $T = \infty$, the bounding parameter is written as $(a, \boldsymbol{\pi}, \mathbf{Q})$.

In this definition

$$W_\rho(t; A) = \max_{0\leq s\leq t}\left\{A(s,t) - \rho(t-s)\right\} \tag{3.5}$$

can be interpreted as the queue workload in a work-conserving system that transmits at a constant rate of ρ and is fed with an input traffic process $A(t)$.

Definition 3.1.4. A process $D(t)$ is tail-limited general phase-type bounded (gPHB) with bounding parameter $(a, \boldsymbol{\pi}, \mathbf{Q}, T)$ if

$$\mathsf{P}\left\{D(t) \geq \sigma\right\} \leq a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}, \tag{3.6}$$

for all $t \geq 0$ and all $T \geq \sigma \geq 0$.

3.2 PHBB Network Calculus

In this section we extend the network calculus theorems in [85] for SBB to the case of PHBB. For simplicity, we shall assume $T = \infty$, but the theorems also hold when T is finite. We will include T when we discuss gPHBB network calculus in Section 3.4.

Theorem 3.2.1 (*Characterization*). Consider a work-conserving system that transmits at a constant rate of ρ and is fed with a single traffic process $A(t)$. Let $W(t)$ be the workload in the system at time t. If $W(t)$ is PHB with parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$ then $A(t)$ is PHBB with upper rate ρ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$.

Proof. The result follows from [85, Theorem 1], where the bounding function is given by $f(\sigma) = a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}$. □

Theorem 3.2.2 (*Sum*). Let $A_1(t)$ be PHBB with upper rate ρ_1 and bounding parameter $(\boldsymbol{\alpha}, \mathbf{G}, a_1)$, and $A_2(t)$ be PHBB with upper rate ρ_2 and bounding parameter $(\boldsymbol{\beta}, \mathbf{H}, a_2)$. Then $A_1(t) + A_2(t)$ is PHBB with upper rate $\rho = \rho_1 + \rho_2$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$ where $a = a_1 + a_2$,

$$\boldsymbol{\pi} = \left[\frac{a_1}{a}\boldsymbol{\alpha}, \frac{a_2}{a}\boldsymbol{\beta}\right], \quad \mathbf{Q} = \begin{pmatrix} p\mathbf{G} & \mathbf{0} \\ \mathbf{0} & (1-p)\mathbf{H} \end{pmatrix}, \tag{3.7}$$

and p is a real number such that $0 < p < 1$.

Proof. As $A_1(t)$ and $A_2(t)$ are PHBB, a special case of SBB, we can apply the Sum theorem for SBB [85, Theorem 2]. In this case, a bounding function of the aggregated traffic is given by $g(\sigma) = f_1(p\sigma) + f_2((1-p)\sigma)$, where

$$f_1(\sigma) = a_1\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{1}, \quad f_2(\sigma) = a_2\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{1}. \tag{3.8}$$

We have

$$g(\sigma) = a_1\boldsymbol{\alpha}e^{p\mathbf{G}\sigma}\mathbf{1} + a_2\boldsymbol{\beta}e^{(1-p)\mathbf{H}\sigma}\mathbf{1} = \Big[a_1\boldsymbol{\alpha}, a_2\boldsymbol{\beta}\Big]\begin{pmatrix} e^{p\mathbf{G}} & \mathbf{0} \\ \mathbf{0} & e^{(1-p)\mathbf{H}} \end{pmatrix}\mathbf{1}, \tag{3.9}$$

from which the result follows. □

Theorem 3.2.3 (*Sum of Independent Traffic Streams*). Let $A_1(t)$ be PHBB with upper rate ρ_1 and bounding parameter $(\boldsymbol{\alpha}, \mathbf{G}, a)$, and $A_2(t)$ be PHBB with upper rate ρ_2 and bounding parameter $(\boldsymbol{\beta}, \mathbf{H}, b)$. Then if $A_1(t)$ and $R_2(t)$ are independent, $A_1(t) + A_2(t)$ is PHBB with upper rate $\rho = \rho_1 + \rho_2$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, c)$, where $c = a + b - ab$

$$\boldsymbol{\pi} = \left[\frac{a(1-b)}{a+b-ab}\boldsymbol{\alpha}, \frac{b(1-a)}{a+b-ab}\boldsymbol{\beta}, \frac{ab}{a+b-ab}\boldsymbol{\alpha}, \mathbf{0}\right], \tag{3.10}$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G} & \mathbf{g}\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H} \end{pmatrix},$$

where $\mathbf{g} = -\mathbf{G}\mathbf{1}$.

Proof. This proof is based on the proof of theorem 4 in [99]. If $A_1(t)$ and $A_2(t)$ are independent, therefore $\{A_1(s,t) - \rho_1(t-s)\}$ and $\{A_2(s,t) - \rho_2(t-s)\}$ are independent with the corresponding distribution functions of $G_1(\sigma)$ and $G_2(\sigma)$ and

$$\mathsf{P}\left\{A_1(s,t) - \rho_1(t-s) \geq \sigma\right\} = 1 - G_1(\sigma),$$

$$\mathsf{P}\left\{A_2(s,t) - \rho_2(t-s) \geq \sigma\right\} = 1 - G_2(\sigma).$$

Therefore, $\{A_1(s,t) + A_2(s,t) - (\rho_1 + \rho_2)(t-s)\}$ which is summation of two independent random variables has the distribution function of $G_1(\sigma)\star G_2(\sigma)$, where $\star$ denotes the Stieltjes convolution and is defined as: $F_1\star F_2 = \int_0^x F_1(x-y)\mathrm{d}F_2(y)$. Hence

$$\mathsf{P}\left\{A_1(s,t)+A_2(s,t)-(\rho_1+\rho_2)(t-s)\geq\sigma\right\}=1-G_1(\sigma)\star G_2(\sigma)$$

On the other hand,

$$\mathsf{P}\left\{A_1(s,t)-\rho_1(t-s)\geq\sigma\right\}\leq 1-F_1(\sigma)=a\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1},$$

$$\mathsf{P}\left\{A_2(s,t)-\rho_2(t-s)\geq\sigma\right\}\leq 1-F_2(\sigma)=b\boldsymbol{\beta}e^{\mathbf{H}\sigma}\mathbf{1}$$

Therefore, we have

$$1-G_1(\sigma)\leq 1-F_1(\sigma),\quad 1-G_2(\sigma)\leq 1-F_2(\sigma)\rightarrow F_1(\sigma)\star F_2(\sigma)\leq G_1(\sigma)\star G_2(\sigma).$$

Hence,

$$\mathsf{P}\left\{A_1(s,t)+A_2(s,t)-(\rho_1+\rho_2)(t-s)\geq\sigma\right\}\leq 1-F_1(\sigma)\star F_2(\sigma).$$

But if we consider X as a random variable with cumulative distribution function $F_1(\sigma)$ and pdf $f_1(\sigma)=\frac{\mathrm{d}F_1(\sigma)}{\mathrm{d}\sigma}$ and Y as another random variable with cumulative distribution function $F_2(\sigma)$ and pdf $f_2(\sigma)=\frac{\mathrm{d}F_2(\sigma)}{\mathrm{d}\sigma}$, and if we consider X and Y as independent random variables, then $F(\sigma)=F_1\star F_2(\sigma)$ is the cumulative distribution fucntion of the new random variable $Z=X+Y$ with pdf of $f(\sigma)=f_1(\sigma)*f_2(\sigma)$. We have

$$F_1(0)=1-a,\quad F_1(0)=1-b$$

therefore,

$$f_1(\sigma)=(1-a)\delta(\sigma)+a\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{g}\qquad\qquad f_1(\sigma)=(1-b)\delta(\sigma)+b\boldsymbol{\beta}e^{\mathbf{H}\sigma}\mathbf{h},$$

where $\mathbf{g} = -\mathbf{G}\mathbf{1}$, and $\mathbf{h} = -\mathbf{H}\mathbf{1}$. Therefore,

$$f(\sigma) = f_1(\sigma) * f_2(\sigma) = ((1-a)\delta(\sigma) + a\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{g}) * ((1-b)\delta(\sigma) + b\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{h})$$

$$= (1-a)(1-b)\delta(\sigma) + (1-a)b\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{h} + (1-b)a\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{g} + ab(\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{g}) * (\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{h})$$

but $\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{g}$ and $\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{h}$ are pdfs of the two phase-type random variables $\tau_1 \sim \mathrm{PH}_n(\boldsymbol{\alpha}, \mathbf{G})$ and $\tau_2 \sim \mathrm{PH}_n(\boldsymbol{\beta}, \mathbf{H})$, respectively, and $(\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{g}) * (\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{h})$ is the pdf of $\tau = \tau_1 + \tau_2$ when τ_1 and τ_2 are independent, which is according to (2.25) a phase-type random variable. Therefore,

$$\tau_1 + \tau_2 \sim \mathrm{PH}_{m+n}\left((\boldsymbol{\alpha}, \mathbf{0}), \begin{pmatrix} \mathbf{G} & \mathbf{g}\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{H} \end{pmatrix}\right)$$

where $\mathbf{g} = -\mathbf{G}\mathbf{1}$. Therefore,

$$f(\sigma) = (1-a)(1-b)\delta(\sigma) + (1-a)b\boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{h} + (1-b)a\boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{g} + ab\boldsymbol{\gamma} e^{\mathbf{K}\sigma}\mathbf{k},$$

where

$$\boldsymbol{\gamma} = (\boldsymbol{\alpha}, \mathbf{0}), \quad \mathbf{K} = \begin{pmatrix} \mathbf{G} & \mathbf{g}\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{H} \end{pmatrix}, \quad \mathbf{k} = -\mathbf{K}\mathbf{1}.$$

Therefore,

$$F(\sigma) = \int_0^\sigma f(\tau)\mathrm{d}\tau = (1-a)(1-b) + (1-a)b(1 - \boldsymbol{\beta} e^{\mathbf{H}\sigma}\mathbf{1}) + (1-b)a(1 - \boldsymbol{\alpha} e^{\mathbf{G}\sigma}\mathbf{1})$$

$$+ ab(1 - \boldsymbol{\gamma} e^{\mathbf{K}\sigma}\mathbf{1}),$$

Therefore,

$$g(\sigma) = (1-b)a\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1} + (1-a)b\boldsymbol{\beta}e^{\mathbf{H}\sigma}\mathbf{1} + ab\boldsymbol{\gamma}e^{\mathbf{K}\sigma}\mathbf{1} = (a+b-ab)\cdot$$

$$\left[\frac{(1-b)a}{a+b-ab}\boldsymbol{\alpha}, \frac{(1-a)b}{a+b-ab}\boldsymbol{\beta}, \frac{ab}{a+b-ab}\boldsymbol{\gamma}\right] \cdot \begin{pmatrix} e^{\mathbf{G}\sigma} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & e^{\mathbf{H}\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & e^{\mathbf{K}\sigma} \end{pmatrix} \cdot \mathbf{1}.$$

□

Theorem 3.2.4 (*Input-Output Relation*). Let $A_{\text{in}}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\text{in}}(t)$ is fed to the element on the input link with infinite capacity and is PHBB with upper rate $\rho < C$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$. Let $W(t)$ denote the queue workload process and let $A_{\text{out}}(t)$ denote the output traffic rate process. Then the following hold:

1. $W(t)$ is PHB with bounding parameter

$$\left(\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\} + C - \rho}, \mathbf{Q}, \frac{a(C-\rho+\mathsf{E}\{\tau\})}{C-\rho}\right), \tag{3.11}$$

   where $\mathsf{E}\{\tau\} = -\boldsymbol{\pi}\mathbf{Q}^{-1}\mathbf{1}$ is the mean of phase-type random variable $\tau \sim \text{PH}(\boldsymbol{\pi}, \mathbf{Q})$.

2. $A_{\text{out}}(t)$ is PHBB with upper rate ρ and bounding parameter as given in (3.11).

Proof. 1. Since $A_{\text{in}}(t)$ is PHBB with upper rate $\rho < C$, we can apply the general SBB input-output relation theorem given in [85, Theorem 3]. In this case, $W(t)$ will be bounded with bounding function $g(\sigma) = f(\sigma) + \frac{1}{C-\rho}\int_{\sigma}^{\infty} f(u)\ \mathrm{d}u$, where

$f(\sigma) = a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$. We have

$$g(\sigma) = a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1} - \frac{a\boldsymbol{\pi}\mathbf{Q}^{-1}e^{\mathbf{Q}\sigma}\mathbf{1}}{C-\rho} = a\left[\boldsymbol{\pi} - \frac{\boldsymbol{\pi}\mathbf{Q}^{-1}}{C-\rho}\right]e^{\mathbf{Q}\sigma}\mathbf{1}$$

$$= \frac{a(C-\rho+\mathsf{E}\{\tau\})}{C-\rho}\left[\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\}+C-\rho}\right]e^{\mathbf{Q}\sigma}\mathbf{1}.$$

The factor in square brackets represents a probability distribution since

$$\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{E\{\tau\}+C-\rho}\cdot\mathbf{1} = \frac{C-\rho-\boldsymbol{\pi}\mathbf{Q}^{-1}\mathbf{1}}{\mathsf{E}\{\tau\}+C-\rho} = 1,$$

where we have used (2.22). Therefore, $g(\sigma)$ is a phase-type bounding function for the output traffic rate process.

2. Since $A_{\text{in}}(t)$ is PHBB, following the same argument as above we can establish that $A_{\text{out}}(t)$ is bounded with upper rate ρ and bounding function $g(\sigma) = f(\sigma) + \frac{1}{C-\rho}\int_\sigma^\infty f(u)\,\mathrm{d}u$, where $f(\sigma) =$ is $\mathrm{PH}_n(\boldsymbol{\pi}, \mathbf{T}, a)$. The proof relies on the facts that $\int_\sigma^\infty f(u)\,\mathrm{d}u \in \mathcal{F}$ and $f(\sigma)$ is a decreasing function of σ, which are established in Theorem 3.1.1.

□

Theorem 3.2.5 (*Average workload in a work-conserving element*). Let $A_{\text{in}}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\text{in}}(t)$ is PHBB with upper rate $\rho < C$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$. Let $W(t)$ denote the queue workload process, then we have

$$\mathsf{E}\{W(t)\} \le a\boldsymbol{\pi}\left(\mathbf{I} - \mathbf{Q}^{-1}\left(1 + \frac{1}{C-\rho}\right) + \frac{\mathbf{Q}^{-2}}{C-\rho}\right)e^{\mathbf{Q}}\mathbf{1} \tag{3.12}$$

In order to prove this theorem at first we present its generalized form for SBB input traffic and then particularize the result to the case of PHBB.

Lemma 3.2.1. Let $A_{\text{in}}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\rm in}(t)$ is SBB with upper rate $\rho < C$ and bounding function $f(\sigma)$. Let $W(t)$ denote the queue workload process, then we have

$$\mathsf{E}\{W(t)\} \le f(1) + \left(1 + \frac{1}{C-\rho}\right)\int_1^\infty f(u)\,\mathrm{d}u + \frac{1}{C-\rho}\int_1^\infty\int_\tau^\infty f(u)\,\mathrm{d}u\mathrm{d}\tau \tag{3.13}$$

Proof. According to [85][Theorem Input-Output], $W(t)$ is SB with bounding function $g(\sigma)$ $= f(\sigma) + \frac{1}{C-\rho}\int_\sigma^\infty f(u)\,\mathrm{d}u$. Therefore, we have

$$\mathsf{E}\{W(t)\} = \sum_{\sigma=1}^{\infty} \mathsf{P}\{W(t) \ge \sigma\} \le \sum_{\sigma=1}^{\infty} g(\sigma) = g(1) + \sum_{\sigma=2}^{\infty} g(\sigma) \le g(1) + \int_1^\infty g(u)\,\mathrm{d}u$$

□

Theorem 3.2.5 can be proved by particularizing the result of Lemma 3.2.1.

Theorem 3.2.6 (*Delay in a work-conserving element*). Let $A_{\rm in}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\rm in}(t)$ is PHBB with upper rate $\rho < C$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$. Let $W(t)$ denote the queue workload process and $D(t)$ denote the maximum delay of the bits of a packet arriving at time t, then $D(t)$ is gPHB with bounding parameter

$$\left(\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\} + C - \rho}, \mathbf{Q}(C-\rho), \frac{a(C-\rho+\mathsf{E}\{\tau\})}{C-\rho}\right), \tag{3.14}$$

where $\mathsf{E}\{\tau\} = -\boldsymbol{\pi}\mathbf{Q}^{-1}\mathbf{1}$ is the mean of phase-type random variable $\tau \sim \mathrm{PH}(\boldsymbol{\pi}, \mathbf{Q})$.

In order to prove the following theorem at first we present its generalizations for SBB traffic and then particularize the result for the case of PHBB traffic.

Lemma 3.2.2. Let $A_{\rm in}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\rm in}(t)$ is SBB with upper rate $\rho < C$ and bounding function $f(\sigma)$. Let $W(t)$ denote the queue workload process and $D(t)$ denote the maximum delay of the bits of a packet arriving at time t, then $D(t)$ is SB with bounding function $g(\sigma)$

$$g(\sigma) = f((C-\rho)\sigma) + \frac{1}{C-\rho}\int_{(C-\rho)\sigma}^{\infty} f(\tau)\,\mathrm{d}\tau. \tag{3.15}$$

Proof. Let $d(t)$ denote the time that has passed since the last time the queue was empty prior to t. In other words

$$d(t) = \min\{u : W(t-u) = 0\} \tag{3.16}$$

Also Let $\mathcal{D}(t)$ denote the rest of the busy period that t resides in. In other words

$$\mathcal{D}(t) = \min\{u : W(t+u) = 0\} \tag{3.17}$$

Then the delay incurred on the packet arriving at t, $D(t)$, is bounded by $\mathcal{D}(t)$. Therefore, we have

$$\mathsf{P}\{D(t) \geq \sigma\} \leq \mathsf{P}\{\mathcal{D}(t) \geq \sigma\} = \sum_{i=0}^{t} \mathsf{P}\{\{d(t) = i\} \cap \{\mathcal{D}(t) \geq \sigma\}\}$$

$$\leq \sum_{i=0}^{\infty} \mathsf{P}\{A_{\text{in}}(t-i, t+\sigma) \geq C(i+\sigma)\} \leq f((C-\rho)\sigma)$$

$$+ \frac{1}{(C-\rho)} \sum_{i=1}^{\infty} (C-\rho) f\left((C-\rho)(i+\sigma)\right) \leq f((C-\rho)\sigma) + \frac{1}{(C-\rho)} \int_{(C-\rho)\sigma}^{\infty} f(\tau)\,\mathrm{d}\tau$$

□

Now we can proof Theorem 3.2.6 by particularizing the result of Lemma 3.2.2 to the case of phase-type bound.

Proof. As A_{in} is PHBB, a special case of SBB we can use the result of Theorem 3.2.2. As we have $f(\sigma) = a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}$, therefore $D(t)$ is PHB with bounding function $g(\sigma)$, where

$$g(\sigma) = a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1} - \frac{A\boldsymbol{\pi}\mathbf{Q}^{-1}e^{\mathbf{Q}\sigma}\mathbf{1}}{C-\rho} = a\left[\boldsymbol{\pi} - \frac{\boldsymbol{\pi}\mathbf{Q}^{-1}}{C-\rho}\right] e^{\mathbf{Q}(C-\rho)\sigma}\mathbf{1}$$

$$= \frac{a(C-\rho+\mathsf{E}\{\tau\})}{C-\rho}\left[\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\}+C-\rho}\right] e^{\mathbf{Q}(C-\rho)\sigma}\mathbf{1}.$$

□

Theorem 3.2.7 (*Input-Output Relation for Limited Capacity Input Link*). Let $A_{\text{in}}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\text{in}}(t)$ is fed to the element on the input link with capacity $C_1 > C$ and is PHBB with upper rate $\rho < C$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$. Let $W(t)$ denote the queue workload process and let $A_{\text{out}}(t)$ denote the output traffic rate process. Then the following hold:

1. $W(t)$ is PHB with bounding parameter

$$\left(\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\}+C-\rho}, \mathbf{Q}\left(\frac{C-\rho}{C_1-C}+1\right), \frac{a(C-\rho+\mathsf{E}\{\tau\})}{C-\rho}\right), \tag{3.18}$$

   where $\mathsf{E}\{\tau\} = -\boldsymbol{\pi}\mathbf{Q}^{-1}\mathbf{1}$ is the mean of phase-type random variable $\tau \sim \mathrm{PH}(\boldsymbol{\pi}, \mathbf{Q})$.

2. $R_{\text{out}}(t)$ is PHBB with upper rate ρ and bounding parameter as given in (3.18).

Note that in Theorem 3.2.7, if $C_1 < C$ then we have $W(t) = 0$ w.p 1. In order to prove Theorem 3.2.7 we first present its generalized form for SBB input traffic and then particularize the results for PHBB case.

Lemma 3.2.3. Let $A_{\text{in}}(t)$ be the input traffic rate process to a work-conserving element, which transmits at constant rate C. Suppose that $A_{\text{in}}(t)$ is fed to the element on the input link with capacity $C_1 > C$ and is SBB with upper rate $\rho < C$ and bounding function $f(\sigma)$. Let $W(t)$ denote the queue workload process and let $A_{\text{out}}(t)$ denote the output traffic rate process. Then the following hold:

1. $W(t)$ is SB with bounding function

$$f\left(\sigma\left(\frac{C-\rho}{C_1-C}+1\right)\right)+\frac{1}{C-\rho}\int_{\sigma\left(\frac{C-\rho}{C_1-C}+1\right)}^{\infty} f(u)\,\mathrm{d}u, \tag{3.19}$$

2. $A_{\text{out}}(t)$ is SBB with upper rate ρ and bounding parameter as given in (3.19).

Proof. We prove the part 1 in here. Part 2 is proved similarly. Let $d(t)$, as defined in (3.16), denote the time passed since the start of the busy period. Therefore we will have

$$\mathsf{P}\{W(t)\geq\sigma\}=\sum_{i=0}^{t}\{\{W(t)\geq\sigma\}\cap\{d(t)=i\}\}=\sum_{i=0}^{t}\{A_{\text{in}}(t-i,t)\geq Ci+\sigma\} \tag{3.20}$$

But we have

$$C_1 t\geq A_{\text{in}}(t-i,t)$$

and if $A_{\text{in}}(t-i,t)\geq\sigma+Ci$, then we should have $i\geq\frac{\sigma}{C_1-C}$. Therefore summation in (3.20) should modify to

$$\begin{aligned}\mathsf{P}\{W(t)\geq\sigma\}&=\sum_{i=\frac{\sigma}{C_1-C}}^{t}\{A_{\text{in}}(t-i,t)\geq Ci+\sigma\}\leq\sum_{i=\frac{\sigma}{C_1-C}}^{\infty} f((C-\rho)i+\sigma)\\&=f\left(\sigma\left(\frac{C-\rho}{C_1-C}+1\right)\right)+\frac{1}{C-\rho}\sum_{i=\frac{\sigma}{C_1-C}+1}^{\infty}(C-\rho)f((C-\rho)i+\sigma)\\&\leq f\left(\sigma\left(\frac{C-\rho}{C_1-C}+1\right)\right)+\frac{1}{C-\rho}\int_{\sigma\left(\frac{C-\rho}{C_1-C}+1\right)}^{\infty} f(u)\,\mathrm{d}u\end{aligned}$$

$\square$

Theorem 3.2.7 is proved by particularizing the results of Lemma 3.2.3. Details are omitted.

3.3 PHBB Network Calculus Application

In this section, we provide some applications of the PHBB network calculus theorems. Here, we consider discrete time input traffic. We assume there is a work-conserving server with capacity C which is fed by N input traffic processes $A_i(t)$, $1 \le i \le N$, such that each $A_i(t)$ is PHBB with upper rate ρ_i and bounding parameter $(\boldsymbol{\alpha}_i, \mathbf{G}_i, a_i)$, and $\sum_{i=1}^{N} \rho_i < C$. In this section, we characterize the output traffic and delay for each source under different service disciplines. Here, $D(t)$, denotes the maximal delay for packets arriving at time t and $D(t) = n$ means the last bit of input traffic $A(t)$ is transmitted at time $t+n$. On the other hand, $Q(t)$ denotes the queue length at time t, and $Q_i(t)$ denotes the portion of the queue which belongs to the source i.

If $A(t) = \sum_{i=1}^{N} A_i(t)$ is defined as the aggregate input traffic, then according to the Sum theorem $A(t)$ is PHBB with upper rate $\rho = \sum_{i=1}^{N} \rho_i$ and bounding parameter $(\boldsymbol{\beta}, \mathbf{H}, b)$ where

$$\boldsymbol{\beta} = \left[\frac{a_1}{b}\boldsymbol{\alpha}_1, \frac{a_2}{b}\boldsymbol{\alpha}_2, \dots, \frac{a_N}{b}\boldsymbol{\alpha}_N\right], \qquad \mathbf{H} = \begin{pmatrix} q_1\mathbf{G}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & q_2\mathbf{G}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & q_N\mathbf{G}_N \end{pmatrix} \tag{3.21}$$

$B = \sum_{i=1}^{N} a_i$ and where $0 < q_i < 1$, $i = 1, 2, \dots, N$ and $\sum_{i=1}^{N} q_i = 1$. In here and all other subsequent sections, q_i for $i = 1, 2, \dots, N$ should be chosen such that the resulting bound is the tightest. This aspect remains to be investigated further.

3.3.1 General results

The following theorem holds for all service disciplines.

Theorem 3.3.1. Suppose that we have N sources sharing a work-conserving system with transmission rate C. Let $A_j(t)$ be PHBB with upper rate ρ_j and bounding parameter

$(\boldsymbol{\alpha}_j, \mathbf{G}_j, a_j)$, where $R_j(t)$ is the input traffic process from source j, and $\sum_{j=1}^{N} \rho_j < C$. Let $A_j^{(\text{out})}(t)$ be the output process of source j. Then $A_j^{(\text{out})}(t)$ is PHBB with upper rate ρ_j and bounding parameter $(\boldsymbol{\zeta}, \mathbf{H}, c_j)$ where

$$\boldsymbol{\zeta} = \left[\frac{a_j}{c_j}\boldsymbol{\alpha}_j, \frac{b}{c_j}\boldsymbol{\Gamma}\right], \qquad \mathbf{H} = \begin{pmatrix} p\mathbf{G}_j & \mathbf{0} \\ \mathbf{0} & (1-p)\mathbf{Q} \end{pmatrix} \tag{3.22}$$

with

$$\boldsymbol{\Gamma} = \frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\} + C - \rho}, \qquad B = \frac{a(C - \rho + \mathsf{E}\{\tau\})}{C-\rho},$$

$$\boldsymbol{\pi} = \left[\frac{a_1}{a}\boldsymbol{\alpha}_1, \frac{a_2}{a}\boldsymbol{\alpha}_2, \ldots, \frac{a_N}{a}\boldsymbol{\alpha}_N\right] \qquad \mathbf{Q} = \begin{pmatrix} q_1\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & q_2\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & q_N\mathbf{G}_N \end{pmatrix}, \tag{3.23}$$

where $a = \sum_{j=1}^{N} a_j$ and $a_j = a_j + b$ and $0 < q_i < 1$ and $0 < p < 1$ and $\sum_{i=1}^{N} q_i = 1$. The q_i, for $i = 1, 2, \ldots, N$ and p, should be chosen to achieve the tightest bound.

Proof. This proof is based on the proof of Theorem 6 in [99]. We use $A(t)$ to denote the aggregate input process, i.e., $A(t) = \sum_{j=1}^{N} A_j(t)$. According to the Sum theorem $A(t)$ is PHBB with upper rate $\rho = \sum_{j=1}^{N} \rho_j$ and bounding parameter $(\boldsymbol{\pi}, \mathbf{Q}, a)$ where $a = \sum_{j=1}^{N} a_j$

$$\boldsymbol{\pi} = \left[\frac{a_1}{a}\boldsymbol{\alpha}_1, \frac{a_2}{a}\boldsymbol{\alpha}_2, \ldots, \frac{a_N}{a}\boldsymbol{\alpha}_N\right] \qquad \mathbf{Q} = \begin{pmatrix} q_1\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & q_2\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & q_N\mathbf{G}_N \end{pmatrix}, \tag{3.24}$$

with $0 < q_i < 1$ and $\sum_{i=1}^{N} q_i = 1$. Let $W(t)$ denote the queue workload process and $W_i(t)$

denote the portion of the workload process which belongs to source i. According to the Input-Output Theorem $W(t)$ is PHB with bounding parameter

$$\left(\underbrace{\frac{\boldsymbol{\pi}(C-\rho) - \boldsymbol{\pi}\mathbf{Q}^{-1}}{\mathsf{E}\{\tau\} + C - \rho}}_{\boldsymbol{\Gamma}}, \mathbf{Q}, \underbrace{\frac{a(C-\rho+\mathsf{E}\{\tau\})}{C-\rho}}_{b} \right), \tag{3.25}$$

where $\mathsf{E}\{\tau\} = -\boldsymbol{\pi}\mathbf{Q}^{-1}\mathbf{1}$ is the mean of a phase-type random variable $\tau \sim \mathrm{PH}(\boldsymbol{\pi}, \mathbf{Q})$. Because the system is work-conserving, we have

$$A_j^{(\mathrm{out})}(s,t) \le W_j(s) + A_j(s,t).$$

Clearly, the output of source j in the time interval $(s,t]$ cannot exceed the sum of the input in this interval and the amount of workload from the same source stored in the queue previously. Therefore,

$$A_j^{(\mathrm{out})}(s,t) - \rho_j(t-s) \le W_j(s) + A_j(s,t) - \rho_j(t-s) \le W(s) + A_j(s,t) - \rho_j(t-s).$$

But we have

$$\left\{ A_j^{(\mathrm{out})}(s,t) - \rho_j(t-s) \ge \sigma \right\} \subseteq \{W(s) \ge p\sigma\} \cup \{A_j(s,t) - \rho_j(t-s) \ge (1-p)\sigma\}$$

where $0 < p < 1$, therefore

$$\mathsf{P}\left\{ A_j^{(\mathrm{out})}(s,t) - \rho_j(t-s) \ge \sigma \right\} \le \mathsf{P}\{W(s) \ge p\sigma\} + \mathsf{P}\left\{A_j(s,t) - \rho_j(t-s) \ge (1-p)\sigma\right\}$$

But according to Theorem 3.2.4, $W(s)$ is PHB with bounding parameter $(\boldsymbol{\Gamma}, \mathbf{Q}, b)$ and $R_j(t)$ is PHBB with bounding parameter $(\boldsymbol{\alpha}_j, \mathbf{G}_j, a_j)$. Therefore, $A_j^{(\mathrm{out})}(t)$ is PHBB with bounding parameter $(\boldsymbol{\zeta}, \mathbf{H}, c_j)$ where $c_j = a_j + b$

$$\boldsymbol{\zeta} = \left[\frac{a_j}{c_j}\boldsymbol{\alpha}_j, \frac{b}{c_j}\boldsymbol{\Gamma}\right], \quad \mathbf{H} = \begin{pmatrix} p\mathbf{G}_j & \mathbf{0} \\ \mathbf{0} & (1-p)\mathbf{Q} \end{pmatrix},$$

□

3.3.2 First in first out (FIFO)

In this section we assume the service discipline is FIFO, which means if $Q(t-1) < C$ then any arrival at time t have the equal opportunity to be served.

3.3.3 Strict priority (SP)

In this service discipline, if $i < j$, source i has a higher priority over source j and source j will not be served as long as there is workload from source i in the system. However, for traffic from the same priority, the FIFO serving discipline is adopted.

Theorem 3.3.2. Suppose that we have N sources sharing a work-conserving system with transmission rate C, which serves according to SP discipline. Let $A_j(t)$ be PHBB with upper rate ρ_j and bounding parameter $(\boldsymbol{\alpha}_j, \mathbf{G}_j, a_j)$, where $A_j(t)$ is the input traffic process from source j, and $\sum_{j=1}^{N} \rho_j < C$. Let $A_j^{(\text{out})}(t)$ be the output process of source j. Then $A_1^{(\text{out})}(t)$ is PHBB with upper bound ρ_1 and bounding parameter

$$\left(\frac{\boldsymbol{\alpha}_1(C-\rho) - \boldsymbol{\alpha}_1\mathbf{G}_1^{-1}}{\mathsf{E}\{\tau\} + C - \rho}, \mathbf{G}_1, \frac{a(C - \rho + \mathsf{E}\{\tau\})}{C-\rho}\right), \tag{3.26}$$

where $\mathsf{E}\{\tau\} = -\boldsymbol{\alpha}_1\mathbf{G}_1^{-1}\mathbf{1}$ is the mean of the phase-type random variable $\tau \sim \text{PH}(\boldsymbol{\alpha}_1, \mathbf{G}_1)$. $A_j^{(\text{out})}(t)$ for $j = 2, \ldots, N$, on the other hand, is PHBB with upper rate ρ_j and bounding parameter $(\boldsymbol{\zeta}_j, \mathbf{H}_j, c_j)$ which are derived from (3.22) and (3.23), when we have just $A_1(t), A_2(t), \ldots, A_j(t)$ as input traffic processes.

Proof. Due to the SP service discipline, for the traffic from source 1, the system behaves exactly like a single input system with a PHBB input with upper rate ρ_1 and bounding parameter $(\boldsymbol{\alpha}_1, \mathbf{G}_1, a_1)$. Therefore $A_1^{(\text{out})}(t)$ will be PHBB and its parameters can be derived from (3.11). For $j = 2$, on the other hand, the system works as if we have just the aggregate input traffic $A_1(t) + A_2(t)$. Therefore $A_2^{(\text{out})}(t)$ is PHBB with upper rate ρ_2 and bounding parameter $(\boldsymbol{\zeta}_2, \mathbf{H}_2, c_2)$, which can be derived from (3.22) and (3.23) with $j = 2$. By a similar argument we can get $(\boldsymbol{\zeta}_j, \mathbf{H}_j, c_j)$ for $j = 3, \ldots, N$. □

3.3.4 Generalized processor sharing (GPS)

In this discipline, the ith flow is assigned a parameter $\phi_i > 0$, and if there is a backlog for flow i in the time interval $(s, t]$, or in other words $Q_i(\tau) > 0$, for all $s < \tau \leq t$, then we have

$$\frac{A_i^{(\text{out})}(s,t)}{A_j^{(\text{out})}(s,t)} \geq \frac{\phi_i}{\phi_j}, \qquad j = 1, 2, \ldots, N, \tag{3.27}$$

where $A_i^{(\text{out})}(s,t)$ is the amount of output traffic in the time interval $(s, t]$ for the ith flow [78]. In our work, without loss of generality, we assume $\sum_{i=1}^{N} \phi_i = 1$, and we call ϕ_i the ith service weight. It can be seen from (3.27) that if the ith traffic flow is backlogged all the time in the interval $(s, t]$, then the available service rate for source i is at least $\phi_i C$ during this time interval.

3.4 Phase-type Network Calculus

In this section we extend the network calculus theorems in [51, 99] developed for gSBB to the case of phase-type traffic bounds. Further, we shall incorporate a limit T on the tail of the bounding function in the results for phase-type traffic bounds.

Theorem 3.4.1 (*Characterization*). Consider a work-conserving system that transmits at a constant rate of ρ and is fed with a single traffic stream with traffic process $A(t)$ and $W(t)$ is the queue workload at time t. Then $A(t)$ is characterized by a *phase-type traffic descriptor* $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$ if and only if

$$\mathsf{P}\{W(t) \geq \sigma\} \leq a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}, \tag{3.28}$$

for all $t \geq 0$ and all $\sigma \in (0, T]$. We interchangeably say $A(t)$ is characterized by a phase-type traffic descriptor with upper rate ρ and bounding function $a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}$.

Proof. By using the relation

$$W(t) = \max_{0 \leq s \leq t} \left\{A(s,t) - \rho(t-s)\right\},$$

the theorem is easily proven. □

In Appendix A, we have shown that a gSBB traffic process is SBB with the same upper rate and bounding parameter, but the converse does not necessarily hold. The following theorem expresses relationships between the bounding parameter of a PHBB process and that of the counterpart phase-type traffic process.

Theorem 3.4.2.

1. If $A(t)$ is characterized by a phase-type traffic descriptor $[\rho; (a, \boldsymbol{\alpha}, \mathbf{G}, T)]$, then it is PHBB with the same upper rate and bounding parameter.

2. If $A(t)$ is PHBB with upper rate ρ and bounding parameters $(a, \boldsymbol{\alpha}, \mathbf{G})$, then for any $\epsilon > 0$, it characterized by a phase-type traffic descriptor $[\rho + \epsilon; (b, \boldsymbol{\beta}, \mathbf{H}, T = \infty)]$, where

$$(b, \boldsymbol{\beta}, \mathbf{H}) = \left(\frac{a(\epsilon + \mathsf{E}\{\tau\})}{\epsilon}, \mathbf{G}, \frac{\boldsymbol{\alpha}\epsilon - \boldsymbol{\alpha}\mathbf{G}^{-1}}{\mathsf{E}\{\tau\} + \epsilon}\right), \tag{3.29}$$

where $\mathsf{E}\{\tau\} = -\boldsymbol{\alpha}\mathbf{G}^{-1}\mathbf{1}$ is the mean of phase-type random variable $\tau \sim \mathrm{PH}(\boldsymbol{\alpha}, \mathbf{G})$.

These relationships between phase-type traffic bound and PHBB also hold for a finite tail bound T.

Proof. As PHBB and phase-type traffic bounds are special cases of SBB and gSBB, respectively, this theorem is a special case of [99, Theorem 2], therefore part one is easily verified. For the second part we have the bounding function of the phase-type traffic process as $g(\sigma) = f(\sigma) + \frac{1}{\epsilon}\int_\sigma^\infty f(u)\,\mathrm{d}u$, where $f(\sigma) = A\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1}$ is the bounding function of the PHBB process $R(t)$. We have

$$
\begin{aligned}
g(\sigma) &= A\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1} - \frac{1}{\epsilon}\int_\sigma^\infty A\boldsymbol{\alpha}e^{\mathbf{G}u}\mathbf{1}\,\mathrm{d}u = A\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1} - \frac{1}{\epsilon}A\boldsymbol{\alpha}\int_\sigma^\infty e^{\mathbf{G}u}\,\mathrm{d}u\;\mathbf{1} \\
&= A\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1} - A\boldsymbol{\alpha}\left[\mathbf{G}^{-1}e^{\mathbf{G}u}\right]_\sigma^\infty\;\mathbf{1} = A\left[\boldsymbol{\alpha} - \frac{\boldsymbol{\alpha}\mathbf{G}^{-1}}{\epsilon}\right]e^{\mathbf{G}\sigma}\mathbf{1} \\
&= \frac{A(\epsilon + \mathsf{E}\{\tau\})}{\epsilon}\left[\frac{\boldsymbol{\alpha}\epsilon - \boldsymbol{\alpha}\mathbf{G}^{-1}}{\mathsf{E}\{\tau\} + \epsilon}\right]e^{\mathbf{G}\sigma}\mathbf{1}.
\end{aligned}
\tag{3.30}
$$

where in deriving (3.30), we used, $\lim_{u\to\infty} e^{\mathbf{G}u} = \mathbf{0}$, which we have according to (2.24). The factor in square brackets represents a probability distribution since

$$
\frac{\boldsymbol{\alpha}\epsilon - \boldsymbol{\alpha}\mathbf{G}^{-1}}{E\{\tau\} + \epsilon}\cdot\mathbf{1} = \frac{\epsilon - \boldsymbol{\alpha}\mathbf{G}^{-1}\mathbf{1}}{\mathsf{E}\{\tau\} + \epsilon} = 1,
$$

where we have used (2.22). Therefore, $A(t)$ can be characterized by a phase-type traffic descriptor with upper rate $\rho + \epsilon$ and bounding function $g(\sigma)$.

$\square$

Theorem 3.4.3 (*Sum*). Let $A_1(t)$ be characterized by a phase-type traffic descriptor $[\rho_1; (a, \boldsymbol{\alpha}, \mathbf{G}, T_1)]$, and $A_2(t)$ be characterized by a phase-type traffic descriptor $[\rho_2; (b, \boldsymbol{\beta}, \mathbf{H}, T_2)]$. Then $A_1(t)+A_2(t)$ can be characterized by a phase-type traffic descriptor $[\rho_1+\rho_2; (c, \boldsymbol{\pi}, \mathbf{Q}, T)]$

where $T = \min(T_1, T_2)$, $c = a + b$,

$$\boldsymbol{\pi} = \left[\frac{a}{c}\boldsymbol{\alpha}, \frac{b}{c}\boldsymbol{\beta}\right], \quad \mathbf{Q} = \begin{pmatrix} p\mathbf{G} & \mathbf{0} \\ \mathbf{0} & (1-p)\mathbf{H} \end{pmatrix}, \tag{3.31}$$

and p is a real number such that $0 < p < 1$.

Proof. As $A_1(t)$ and $A_2(t)$ are characterized by phase-type traffic descriptors, a special case of gSBB, we can apply the Sum theorem for gSBB [99, Theorem 3]. In this case, a bounding function of the aggregated traffic is given by $g(\sigma) = f_1(p\sigma) + f_2((1-p)\sigma)$, where

$$f_1(\sigma) = a\boldsymbol{\alpha}e^{\mathbf{G}\sigma}\mathbf{1}, \quad \text{for} \quad T_1 > \sigma > 0, \qquad\qquad f_2(\sigma) = b\boldsymbol{\beta}e^{\mathbf{H}\sigma}\mathbf{1}, \quad \text{for} \quad T_2 > \sigma > 0.$$

We have

$$g(\sigma) = a\boldsymbol{\alpha}e^{p\mathbf{G}\sigma}\mathbf{1} + b\boldsymbol{\beta}e^{(1-p)\mathbf{H}\sigma}\mathbf{1} = (a+b)\left[\frac{a\boldsymbol{\alpha}}{a+b}, \frac{b\boldsymbol{\beta}}{a+b}\right]\begin{pmatrix} e^{p\mathbf{G}} & \mathbf{0} \\ \mathbf{0} & e^{(1-p)\mathbf{H}} \end{pmatrix}\mathbf{1},$$

for $T = \min(T_1, T_2) \geq \sigma > 0$. By setting $T = \min(T_1, T_2)$, $g(\sigma)$ is well-defined. □

Theorem 3.4.4 (*Sum of Independent Traffic Processes*). Let $A_1(t)$ be characterized by a phase-type traffic descriptor $[\rho_1; (a, \boldsymbol{\alpha}, \mathbf{G}, T_1)]$, and $A_2(t)$ be characterized by a phase-type traffic descriptor $[\rho_2; (b, \boldsymbol{\beta}, \mathbf{H}, T_2)]$. Then if $A_1(t)$ and $A_2(t)$ are independent, $A_1(t) + A_2(t)$ can be characterized by a phase-type traffic descriptor $[\rho_1 + \rho_2; (c, \boldsymbol{\pi}, \mathbf{Q}, T)]$, where $T = \min(T_1, T_2)$, $c = a + b - ab$,

$$\boldsymbol{\pi} = \left[\frac{a(1-b)}{c}\boldsymbol{\alpha}, \frac{b(1-a)}{c}\boldsymbol{\beta}, \frac{ab}{c}\boldsymbol{\alpha}, \mathbf{0}\right], \quad \mathbf{Q} = \begin{pmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G} & \mathbf{g}\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H} \end{pmatrix}, \tag{3.32}$$

where $\mathbf{g} = -\mathbf{G}\mathbf{1}$.

Proof. As $A_1(t)$ and $A_2(t)$ are characterized by phase-type traffic descriptors, a special case of gSBB, we can apply the Sum of Independent Traffic theorem for gSBB [99, Theorem 4]. Details of the proof are omitted as the proof is very similar to the proof of Theorem 3.2.3. □

Theorem 3.4.5 (*Input-Output Relation*). Let $A_{\text{in}}(t)$ be the input traffic rate process to a work-conserving element, which transmits at rate C. Suppose that $A_{\text{in}}(t)$ is characterized by a phase-type traffic descriptor $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$. Let $A_{\text{out}}(t)$ denotes the output traffic rate process. Then the following hold:

1. $A_{\text{out}}(t)$ is less bursty than $A_{\text{in}}(t)$. In other words,

$$\max_{0 \le s \le t} \{A_{\text{out}}(s,t) - \rho(t-s)\} \le \max_{0 \le s \le t} \{A_{\text{in}}(s,t) - \rho(t-s)\}, \tag{3.33}$$

2. $A_{\text{out}}(t)$ can be characterized by the same phase-type traffic descriptor $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$.

Proof.
1. This relation is based on [99, Theorem 5] and does not depend on the bounding function.
2. Since $A_{\text{in}}(t)$ is characterized by a phase-type traffic descriptor $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$, which is a special case of gSBB, therefore according to [99, Corollary (Input-Output Relation)] $A_{\text{out}}(t)$ can be characterized by the same phase-type traffic descriptor $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$.

□

Theorem 3.4.6. Assume that $A(t)$ is characterized by a phase-type traffic descriptor $[\rho; (a, \boldsymbol{\alpha}, \mathbf{G}, T)]$ and is the input traffic to a work-conserving system with transmission rate $C > \rho$. If $D(t)$ denotes the maximum delay of the bits of a packet arriving at time t, then the tail probability $D(t)$ is is bounded as follows,

$$\mathsf{P}\{D(t) \ge \sigma\} \le A\boldsymbol{\alpha} e^{(C-\rho)\mathbf{G}\sigma}\mathbf{1}, \tag{3.34}$$

for all $t \geq 0$ and all $\frac{T}{C-\rho} \geq \sigma \geq 0$.

Proof. Here, time is considered as discrete and σ is an integer value. As $A(t)$ is characterized by a phase-type traffic descriptor, a special case of gSBB, we can apply the corresponding theorem for gSBB [99, Theorem 7]. Therefore,

$$\mathsf{P}\{D(t) \geq \sigma\} \leq f(\sigma(C-\rho)),$$

where $f(\sigma(C-\rho)) = A\boldsymbol{\alpha}e^{(C-\rho)\mathbf{G}\sigma}\mathbf{1}$. This relation, however, holds if

$$\mathsf{P}\{W_C(t+\sigma-1;R) \geq \sigma(C-\rho)\} \leq f(\sigma(C-\rho)),$$

where $W_C(t+\sigma-1;R)$ is defined according to (3.5), and is valid for $\sigma(C-\rho) < T$ or $\sigma < \frac{T}{C-\rho}$. □

3.5 Phase-type Network Calculus Application

In this section, we provide some applications of the Phase-type network calculus theorems. In this section we consider the same work-conserving system described in Section 3.3. We assume there is a work-conserving server with capacity C which is fed by N input traffic processes $A_i(t)$, $1 \leq i \leq N$, such that each $A_i(t)$ is characterized by a phase-type traffic descriptor $[\rho_i; (a_i, \boldsymbol{\alpha}_i, \mathbf{G}_i, T_i)]$, and $\sum_{i=1}^{N} \rho_i < C$. In this section we characterize the output traffic and delay for each source under different service disciplines. In this section, $D(t)$ denotes the maximum delay of the bits of a packet arriving at time t and $D(t) = n$ means the last bit of input traffic $A(t)$ is transmitted at time $t+n$. On the other hand, $Q(t)$ denotes the queue length at time t, and $Q_i(t)$ denotes the portion of the queue which belongs to source i.

If $A(t) = \sum_{i=1}^{N} A_i(t)$ is the aggregate input traffic, then according to the Sum theorem $A(t)$

can be characterized by a phase-type traffic descriptor $[\rho; (b, \boldsymbol{\beta}, \mathbf{H}, T)]$ where $T = \min_{1\le j\le N} T_j$,

$$\boldsymbol{\beta} = \left[\frac{a_1}{b}\boldsymbol{\alpha}_1, \frac{a_2}{b}\boldsymbol{\alpha}_2, \ldots, \frac{a_N}{b}\boldsymbol{\alpha}_N\right], \qquad \mathbf{H} = \begin{pmatrix} q_1\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & q_2\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & q_N\mathbf{G}_N \end{pmatrix} \tag{3.35}$$

and $b = \sum_{i=1}^N a_i$, $0 < q_i < 1$, $i = 1, 2, \ldots, N$ and $\sum_{i=1}^N q_i = 1$.

3.5.1 General results

Theorem 3.5.1. Suppose that we have N sources, $A_j(t)$, $j = 1, 2, \ldots, N$, sharing a work-conserving system with transmission rate C. Let $A_j(t)$ be characterized by a phase-type traffic descriptor $[\rho_i; (a_i, \boldsymbol{\alpha}_i, \mathbf{G}_i, T_i)]$, where $A_j(t)$ is the input traffic process from source j, and $\sum_{j=1}^N \rho_j < C$. Let $A_j^{(\mathrm{out})}(t)$ be the output process of source j. Then for any $\epsilon > 0$, $A_j^{(\mathrm{out})}(t)$ can be characterized by a phase-type traffic descriptor $[\rho_i + \epsilon; (\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j, T)]$ where $T = \min_{1\le j\le N} T_j$,

$$(\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j) = \left(\frac{b_j(\epsilon + \mathsf{E}\{\tau_j\})}{\epsilon}, \frac{\boldsymbol{\pi}_j\epsilon - \boldsymbol{\pi}_j\mathbf{Q}_j^{-1}}{\mathsf{E}\{\tau_j\} + \epsilon}, \mathbf{Q}_j\right), \tag{3.36}$$

$\mathsf{E}\{\tau_j\} = -\boldsymbol{\pi}_j \mathbf{Q}_j^{-1}\mathbf{1}$ is the mean of phase-type random variable $\tau_j \sim \mathrm{PH}(\boldsymbol{\pi}_j, \mathbf{Q}_j)$, and

$$\boldsymbol{\pi}_j = \left[\frac{a_j}{b_j}\boldsymbol{\alpha}_j, \frac{a_1}{b_j}\boldsymbol{\alpha}_1, \frac{a_2}{b_j}\boldsymbol{\alpha}_2, \ldots, \frac{a_N}{b_j}\boldsymbol{\alpha}_N\right], \quad \mathbf{Q}_j = \begin{pmatrix} p_j\mathbf{G}_j & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & p_1\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & p_2\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & p_N\mathbf{G}_N \end{pmatrix}, \tag{3.37}$$

where $b_j = a_j + \sum_{i=1}^{N} a_i$, $0 < p_i < 1,\ i = 1, 2, \ldots, N$ and $p_j + \sum_{i=1}^{N} p_i = 1$. This theorem is valid for both cases of discrete and continuous-time processes[investigated further].

Proof. As $A_j(t),\ j = 1, 2, \ldots, N$ are characterized by a phase-type traffic descriptor, a special case of gSBB, we can apply the corresponding theorem for gSBB [51, Theorem 6]. Therefore $A_j^{(\mathrm{out})}(t)$ is PHBB with upper rate ρ_j and bounding function

$$g_j(\sigma) = \left[f_j(p_j\sigma) + \sum_{i=1}^{N} f_i(p_i\sigma)\right]$$

where $f_j(\sigma) = a_j\boldsymbol{\alpha}_j e^{\mathbf{G}_j\sigma}\mathbf{1}$ for $0 < \sigma < T_j$ and $0 < p_i < 1$ and $p_j + \sum_{i=1}^{N} p_i = 1$. Therefore, we have

$$g_j(\sigma) = \left[a_j\boldsymbol{\alpha}_j e^{\mathbf{G}_j p_j \sigma}\mathbf{1} + \sum_{i=1}^{N} a_i\boldsymbol{\alpha}_i e^{\mathbf{G}_i p_i \sigma}\mathbf{1}\right] = [a_j\boldsymbol{\alpha}_j, a_1\boldsymbol{\alpha}_1, a_2\boldsymbol{\alpha}_2, \ldots, a_n\boldsymbol{\alpha}_N] \cdot$$

$$. \exp\begin{pmatrix} p_j\mathbf{G}_j & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & p_1\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & p_2\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & p_N\mathbf{G}_N \end{pmatrix} \cdot \mathbf{1}.$$

$g_j(\sigma)$ is well-defined on the interval $0 < \sigma < \min_{1\le j\le N} T_j$. Therefore, for any $\epsilon > 0$, $A_j^{(\mathrm{out})}(t)$ can be characterized by a phase-type traffic descriptor $[\rho_i+\epsilon; (\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j, T)]$, which according to Theorem 3.4.2, are related to phase-type traffic descriptor $[\rho_i + \epsilon; (b_j, \boldsymbol{\pi}_j, \mathbf{Q}_j, T)]$ as (3.29). From this relation the results follows. □

3.5.2 First in first out (FIFO)

Theorem 3.5.2. Assume that $A(t)$ is characterized by a phase-type traffic descriptor $[\rho; (a, \boldsymbol{\alpha}, \mathbf{G}, T)]$ and is the input traffic to a work-convserving system with transmission rate $C > \rho$ and FIFO discipline. Then the tail probability $D(t)$ is bounded as follows,

$$\mathsf{P}\{D(t) \ge \sigma\} \le a\boldsymbol{\alpha}e^{(C-\frac{\rho}{\sigma})\mathbf{G}\sigma}\mathbf{1}, \tag{3.38}$$

for all $t \ge 0$ and all $\frac{T+\rho}{C} \ge \sigma \ge 0$.

Proof. As $A(t)$ is characterized by a phase-type traffic descriptor, a special case of gSBB, we can apply the corresponding theorem for gSBB [99, Theorem 8]. Therefore

$$\mathsf{P}\{D(t) \ge \sigma\} \le f(C\sigma - \rho),$$

where $f(C\sigma - \rho) = a\boldsymbol{\alpha}e^{(C\sigma-\rho)\mathbf{G}}\mathbf{1} = a\boldsymbol{\alpha}e^{(C-\frac{\rho}{\sigma})\mathbf{G}\sigma}\mathbf{1}$. Note that, this relation is valid for $C\sigma - \rho < T$ or $\sigma < \frac{T+\rho}{C}$. □

3.5.3 Strict priority (SP)

Theorem 3.5.3. Suppose that we have N sources sharing a work-conserving system with transmission rate C, and the SP service discipline. Assume that $A_j(t)$ is characterized by a phase-type traffic descriptor $[\rho_i; (a_i, \boldsymbol{\alpha}_i, \mathbf{G}_i, T_i)]$, where $A_j(t)$ is the input traffic process from source j, and $\sum_{j=1}^{N} \rho_j < C$. Let $A_j^{(\text{out})}(t)$ be the output process of source j. Then $A_1^{(\text{out})}(t)$ can be characterized by a phase-type traffic descriptor $[\rho_1; (a_1, \boldsymbol{\alpha}_1, \mathbf{G}_1, T_1)]$. On the other hand, for $j = 2, \ldots, N$, $A_j^{(\text{out})}(t)$ for any $\epsilon > 0$ can be characterized by a phase-type traffic descriptor $[\rho_j + \epsilon; \tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j, \mathbf{t}_j)]$, where $\mathbf{t}_j = \min\limits_{1 \le i \le j} T_i$,

$$(\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j) = \left(\frac{b_j(\epsilon + \mathsf{E}\{\tau_j\})}{\epsilon}, \frac{\boldsymbol{\pi}_j \epsilon - \boldsymbol{\pi}_j \mathbf{Q}_j^{-1}}{\mathsf{E}\{\tau_j\} + \epsilon}, \mathbf{Q}_j \right), \tag{3.39}$$

where $\mathsf{E}\{\tau_j\} = -\boldsymbol{\pi}_j \mathbf{Q}_j^{-1}\mathbf{1}$ is the mean of a phase-type random variable $\tau_j \sim \text{PH}(\boldsymbol{\pi}_j, \mathbf{Q}_j)$, and

$$\boldsymbol{\pi}_j = \left[\frac{a_j}{b_j}\boldsymbol{\alpha}_j, \frac{a_1}{b_j}\boldsymbol{\alpha}_1, \frac{a_2}{b_j}\boldsymbol{\alpha}_2, \ldots, \frac{a_j}{b_j}\boldsymbol{\alpha}_j \right], \quad \mathbf{Q}_j = \begin{pmatrix} p_{jj}\mathbf{G}_j & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & p_{j1}\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & p_{j2}\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & p_{jj}\mathbf{G}_j \end{pmatrix}, \tag{3.40}$$

where $b_j = a_j + \sum_{i=1}^{j} a_i$, $1 > p_{ji} > 0$ and $p_{jj} + \sum_{i=1}^{j} p_{ji} = 1,\ j = 2, \ldots, N$.

Proof. For $A_1(t)$, the SP system behaves exactly like a single input system that input is characterized by a phase-type traffic descriptor $[\rho_1; (a_1, \boldsymbol{\alpha}_1, \mathbf{G}_1, T_1)]$, therefore, $A_1^{(\text{out})}(t)$ can be characterized by the same phase-type traffic descriptor $[\rho_1; (a_1, \boldsymbol{\alpha}_1, \mathbf{G}_1, T_1)]$. For $j = 2$, on the other hand, the system works as if we have aggregate input of $A_1(t) + A_2(t)$, therefore, for any $\epsilon > 0$, $A_2^{(\text{out})}(t)$ can be characterized by a phase-type traffic descriptor $[\rho_2 + \epsilon; (\tilde{B}_2, \tilde{\boldsymbol{\pi}}_2, \tilde{\mathbf{Q}}_2, \mathbf{t}_2)]$ which are derived from (3.36) with $j = 2$. By a similar argument we can obtain $(\tilde{B}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j, \mathbf{t}_j)$ for $j = 3, \ldots, N$. $\square$

Theorem 3.5.4. Let $D_j(t)$ be the maximal delay of source j, then with the same assumptions of Theorem 3.5.3, the tail probability of $D_1(t)$ can be bounded as follows,

$$\mathsf{P}\{D_1(t) \geq \sigma\} \leq a_1 \boldsymbol{\alpha} e^{(C - \frac{\rho}{\sigma})\mathbf{G}_1 \sigma} \mathbf{1}, \tag{3.41}$$

for all $t \geq 0$ and all $\frac{T_1 + \rho}{C} \geq \sigma \geq 0$. For $j \geq 2$, the tail probability of $D_j(t)$ can be bounded as follows,

$$\mathsf{P}\{D_j(t) \geq \sigma\} \leq b_j \boldsymbol{\beta}_j e^{(C - \hat{\rho}_j)\mathbf{H}_j \sigma} \mathbf{1}, \tag{3.42}$$

for all $t \geq 0$ and all $\dfrac{\mathbf{t}_j}{C - \hat{\rho}_j} \geq \sigma \geq 0$, where $\mathbf{t}_j = \min\limits_{1 \leq i \leq j} T_i$,

$$\boldsymbol{\beta}_j = \left[\frac{a_1}{b_j}\boldsymbol{\alpha}_1, \frac{a_2}{b_j}\boldsymbol{\alpha}_2, \ldots, \frac{a_j}{b_j}\boldsymbol{\alpha}_j\right], \qquad \mathbf{H}_j = \begin{pmatrix} q_{j1}\mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & q_{j2}\mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & q_{jj}\mathbf{G}_j \end{pmatrix}, \tag{3.43}$$

and $b_j = \sum_{i=1}^{j} a_i$, $\hat{\rho}_j = \sum_{l=1}^{j} \rho_l$, $0 < q_{ji} < 1$, $i, j = 1, 2, \ldots, N$ and $\sum_{i=1}^{j} q_{ji} = 1$.

Proof. For $D_1(t)$, the system behaves exactly like a FIFO single input system with an input that is characterized by a phase-type traffic descriptor $[\rho_1; (a_1, \boldsymbol{\alpha}_1, \mathbf{G}_1, T_1)]$, therefore we can use Theorem 3.5.2. Hence, the tail probability of $D_1(t)$ can be bounded according to (3.41).

For $j \geq 2$, on the other hand, the system works as if we have aggregate input of $A_1(t) + \ldots + A_j(t)$, which according to Theorem 3.4.3 can be characterized by a phase-type traffic descriptor $[\hat{\rho}_j = \sum_{l=1}^{j} \rho_l; (B_j, \boldsymbol{\beta}_j, \mathbf{H}_j, T_j)]$, derived according to (3.35) for $N = j$. Therefore, we can apply Theorem 3.5.2, and the tail probability of $D_j(t)$ can be bounded as (3.42). $\square$

3.5.4 Generalized processor sharing (GPS)

Theorem 3.5.5. Suppose that we have N sources sharing a work-conserving system with transmission rate C, which adopts GPS serving discipline. For $1 \leq i \leq N$, traffic source i is assigned with serving weight $\phi_j > 0$. Assume that $A_j(t)$ is characterized by a phase-type traffic descriptor $[\rho_i; (a_i, \boldsymbol{\alpha}_i, \mathbf{G}_i, T_i)]$, where $A_j(t)$ is the input traffic process from source j, and $\sum_{j=1}^{N} \rho_j < C$. Let $A_j^{(\text{out})}(t)$ be the output process of source j. Then $A_j^{(\text{out})}(t)$ can be characterized by a phase-type traffic descriptor $[\rho_j; (\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j, t_j)]$. If $\phi_j C > \rho_j$ we have $(\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j, t_j) = (a_j, \boldsymbol{\alpha}_j, \mathbf{G}_j, T_j)$. Otherwise,

$$(\tilde{b}_j, \tilde{\boldsymbol{\pi}}_j, \tilde{\mathbf{Q}}_j) = \left(\frac{b_j(\epsilon + \mathsf{E}\{\tau_j\})}{\epsilon}, \frac{\boldsymbol{\pi}_j \epsilon - \boldsymbol{\pi}_j \mathbf{Q}_j^{-1}}{\mathsf{E}\{\tau_j\} + \epsilon}, \mathbf{Q}_j \right), \tag{3.44}$$

where $\mathsf{E}\{\tau_j\} = -\boldsymbol{\pi}_j \mathbf{Q}_j^{-1} \mathbf{1}$ is the mean of a phase-type random variable $\tau_j \sim \mathrm{PH}(\boldsymbol{\pi}_j, \mathbf{Q}_j)$, and

$$\boldsymbol{\pi}_j = \left[\frac{a_j}{b_j} \boldsymbol{\alpha}_j, \frac{a_1}{b_j} \boldsymbol{\alpha}_1, \frac{a_2}{b_j} \boldsymbol{\alpha}_2, \ldots, \frac{a_N}{b_j} \boldsymbol{\alpha}_N \right], \quad \mathbf{Q}_j = \begin{pmatrix} p_j \mathbf{G}_j & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & p_1 \mathbf{G}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & p_2 \mathbf{G}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & p_N \mathbf{G}_N \end{pmatrix}, \tag{3.45}$$

where $b_j = a_j + \sum_{i=1}^{N} a_i$, $0 < p_i < 1,\ i = 1, 2, \dots, N$, $p_j + \sum_{i=1}^{N} p_i = 1$ and $t_j = \min_{1 \le i \le N} T_i$.

Proof. If $\phi_j C > \rho_j$, for the jth traffic flow we can use Theorem 3.4.5, as if the jth flow is served by an isolated server of minimum rate of $\phi_j C$. On the other hand, for the case of $\phi_j C \le \rho_j$, the result follows from Theorem 3.5.1. □

Theorem 3.5.6. Let $D_j(t)$ be the maximal delay of source j, then with the same assumptions of Theorem 3.5.5, if $\phi_j C > \rho_j$, then for $j \ge 1$, the tail probability of $D_j(t)$ can be bounded as,

$$\mathsf{P}\{D(t) \ge \sigma\} \le a_j \boldsymbol{\alpha}_j e^{(C\phi_j - \frac{\rho_j}{\sigma})\mathbf{G}\sigma}\mathbf{1}, \tag{3.46}$$

for all $t \ge 0$ and all $\frac{T_j + \rho_j}{C\phi_j} \ge \sigma \ge 0$.

Proof. If $\phi_j C > \rho_j$, we again consider jth traffic flow as if it is serviced by an isolated server of minimum rate of $\phi_j C$. Then the result follows from Theorem 3.4.6. □

3.6 Case Study

In this section, we consider examples of Markov modulated Poisson process (MMPP) input traffic streams to investigate the PHBB and phase-type traffic bounds to characterize traffic streams. We try to bound the tail probability of workload samples.

We should note that all the derived results in this section are at steady-state for waiting time and queue length. On the other hand, for a traffic stream to be characterized by a phase-type traffic descriptor according to Definition 3.1.3, the bound should be valid for all $t \ge 0$. However, according to the following theorem, queue length (and also waiting time) are stochastically monotically increasing. Therefore, if the steady-state queue length (or waiting time) is stochastically bounded by a bounding function (in our case $a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$), this bound will be also valid for all $t \ge 0$.

Theorem 3.6.1. [51, 67] If $A_{\text{in}}(t)$ is ergodic and stationary and $E\{A_{\text{in}}(1)\} < \rho$ for some $\rho > 0$, then for all $t \geq 0$ we have,

$$W_\rho(t; A_{\text{in}}) \leq_{st} W_\rho(t+1; A_{\text{in}}) \leq_{st} \ldots \leq_{st} W_\rho(\infty; A_{\text{in}}) \tag{3.47}$$

, where $W_\rho(t; A_{\text{in}})$ is defined in (3.5), and $W_\rho(\infty; A_{\text{in}})$ denotes the random variable corresponding to the steady state of $W_\rho(t; A_{\text{in}})$ as $t \to \infty$.

We should note that, in all of the cases we have considered, the traffic is stationary and ergodic and Loynes' stability condition of $E\{A_{\text{in}}(1)\} < \rho$ for some $\rho > 0$ is also satisfied.

3.6.1 MMPP/G/1 queue

The MMPP is a common model for traffic with a high degree of burstiness [36]. A 2-state MMPP is parameterized by an arrival matrix

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \tag{3.48}$$

and a rate matrix

$$\mathbf{R} = \begin{bmatrix} -r_0 & r_0 \\ r_1 & -r_1 \end{bmatrix}, \tag{3.49}$$

which is the generator of the modulating Markov chain. On the other hand, when two identically independent 2-state MMPP are aggregated the result will be a 3-state MMPP [36], with arrival matrix

$$\mathbf{\Lambda} = \begin{bmatrix} 2\lambda_1 & 0 & 0 \\ 0 & \lambda_0 + \lambda_1 & 0 \\ 0 & 0 & 2\lambda_0 \end{bmatrix} \tag{3.50}$$

and a rate matrix

$$\mathbf{R} = \begin{bmatrix} -2r_1 & 2r_1 & 0 \\ r_0 & -r_0 - r_1 & r_1 \\ 0 & 2r_0 & -2r_0 \end{bmatrix}. \tag{3.51}$$

When the service times are independent and generally distributed, the resulting queue is denoted as MMPP/G/1. A relatively simple form for the Laplace transform of the virtual waiting time of a two-state MMPP/G/1 queue is given in [68] in terms of a transition probability matrix

$$\mathbf{G} = [G_{ij} : i, j = 0, 1] = \begin{bmatrix} 1 - d_0 & d_0 \\ d_1 & 1 - d_1 \end{bmatrix}, \tag{3.52}$$

where G_{ij} is the probability that a busy period starting in the underlying state i ends in underlying state j. When the MMPP has more than 2 states, deriving the virtual waiting time for MMPP/G/1 queue becomes more complicated. An MMPP with rate matrix $\mathbf{R}$ and arrival matrix $\boldsymbol{\Lambda}$ is a special case of Markov Arrival Process (MAP) with $\mathbf{D}_0 = \mathbf{R} - \boldsymbol{\Lambda}$ and $\mathbf{D}_1 = \boldsymbol{\Lambda}$, where this MAP process is a two-dimensional Markov process $\{N(t), J(t)\}$ on the state space $\{(i, j) : i \geq 0, 1 \leq j \leq m\}$, where m is the number of states of the MMPP process. The generator matrix of the MAP has the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_0 & \dots \\ \dots & & & & \end{bmatrix}. \tag{3.53}$$

Remark. The virtual waiting time at any given time is defined as the waiting time of a packet arriving at that exact given time. Therefore, at any given time the virtual waiting time is the summation of the service times of the unserviced packets that arrived before that time and the residual service time of the current packet in the server. But if we consider the system to be such that the packet lengths are equal to the service times (as opposed to a constant packet length of one), with the same arrival model, then if this input traffic is fed to a server with constant rate C, then the virtual waiting time will be the workload size divided by C. In the case of the MMPP/G/1 queue, the virtual waiting time is equivalent to the queue workload or buffer content divided by C, in a system with a constant rate server with rate C, where packets arrive according to an MMPP model and packet lengths are drawn from a common distribution denoted by G.

Let $H(s)$ denote the Laplace transform of the packet length, which in this interpretation is equivalent to the packet service time, assuming that the service rate is one unit (e.g., bit) per unit time (e.g., seconds). The generator matrix $\mathbf{G}$ for a 2-state MMPP is determined by numerically solving the following equations [68, Eqs. (86), (87)]:

$$d_0 + d_1 = 1 - H(r_0 + r_1 + \lambda_0 d_0 + \lambda_1 d_1) \tag{3.54}$$

$$d_0(r_1 + \lambda_1 d_1) = d_1(r_0 + \lambda_0 d_0). \tag{3.55}$$

For an MMPP with two or more states, generator matrix $\mathbf{G}$ is derived by the following equation

$$\mathbf{G} = \sum_{n=0}^{\infty} \gamma_n (I + \theta^{-1} D[\mathbf{G}])^n, \tag{3.56}$$

where $\gamma_n = \int_0^\infty e^{-\theta x} \frac{(\theta x)^n}{n!} \mathrm{d}\tilde{H}(x)$. In this equation $\tilde{H}(x)$ is the cumulative distribution function of the service process, $\theta = \max_i \{(-\mathbf{D}_0)_{ii}\}$ and $D(z)$ as matrix generating function $D(z) = \sum_{k=0}^{\infty} \mathbf{D}_k z^k = \mathbf{D}_0 + \mathbf{D}_1 z$. This equation can be solved iteratively in the following recursion,

$$H_{n+1,k} = [I + \theta^{-1} D[\mathbf{G}]] H_{n,k} \tag{3.57}$$

$$\mathbf{G}_{k+1} = \sum_{n=0}^{\infty} \gamma_n H_{n,k}, \tag{3.58}$$

where $H_{0,k} = I$. We can start with $\mathbf{G}_0 = \mathbf{0}$, but we can achieve faster convergence if we start with $\mathbf{G}_0 = \tilde{\boldsymbol{\pi}}\mathbf{1}$, where $\tilde{\boldsymbol{\pi}}$ is the stationary probability vector of the Markov process with generator matrix $\mathbf{D} = \sum_{k=0}^{\infty} \mathbf{D}_k = \mathbf{D}_0 + \mathbf{D}_1$, where

$$\tilde{\boldsymbol{\pi}}\mathbf{D} = \mathbf{0}, \quad \tilde{\boldsymbol{\pi}}\mathbf{1} = 1 \tag{3.59}$$

Average arrival rate for the arrival process is given by

$$\frac{1}{\lambda_{\text{avg}}} = \tilde{\boldsymbol{\pi}} \sum_{k=1}^{\infty} k\mathbf{D}_k \mathbf{1} = \tilde{\boldsymbol{\pi}}\mathbf{D}_1\mathbf{1} \tag{3.60}$$

When the queue utilization factor, $\rho = \frac{\lambda_{\text{avg}}}{\mu} < 1$, $\mathbf{G}$ will be a stochastic matrix and the invariant probability vector $\mathbf{g}$ associated with $\mathbf{G}$ is derived according to,

$$\mathbf{g}\mathbf{G} = \mathbf{g}, \quad \mathbf{g}\mathbf{1} = 1. \tag{3.61}$$

The Laplace transform of the queue workload is given as [68]:

$$W(s) = s(1-\rho)\mathbf{g}[sI + D(H(s))]^{-1} \tag{3.62}$$

$$W(0) = \tilde{\boldsymbol{\pi}}. \tag{3.63}$$

which for the special case of a 2-state MMPP simplifies to

$$W(s) = \frac{N(s)}{D(s)}, \tag{3.64}$$

where

$$N(s) = s(1-\rho)[s-r_0-r_1+(H(s)-1)(f_0\lambda_1+f_1\lambda_0)] \tag{3.65}$$

$$\begin{aligned} D(s) &= s^2 + [(H(s)-1)(\lambda_0+\lambda_1) - (r_0+r_1)]s \\ &\quad + (H(s)-1)[(H(s)-1)\lambda_0\lambda_1 - r_0\lambda_1 - r_1\lambda_0]. \end{aligned} \tag{3.66}$$

3.6.2 Numerical Examples

In this section we provide three examples to investigate the phase-type bounds to characterize the input traffic. These examples are based on queues with MMPP input traffic and different service time distributions described in Section 3.6.1. In these examples we derive the stationary virtual waiting time or stationary waiting time and by using the method mentioned in Section 3.6.1 we relate it to the workload in a virtual queue with a constant rate server. By bounding the tail probability of the workload and using Theorem 3.4.1, we characterize the input traffic. In these three examples virtual waiting time or stationary waiting times are replaced by workload size by considering the constant rate of the server in the virtual queue as 1 or 2, e.g. in the second example.

In the first example we consider a queue, denoted by MMPP/M/1, with MMPP input traffic and exponential service time distribution. In the second example, we use the same MMPP arrival model with Erlang-2 service time distribution denoted by MMPP/E_2/1. In the third example, however, we consider a queue with aggregated arrivals of two independent 2-state MMPP traffic streams and exponential service time distributions. Using Theorem 3.4.4, we will characterize the aggregate input traffic with a phase-type descriptor and compare the results with a gSBB bound based on [85, Theorem 2], which does not assume independence of the input traffic streams.

MMPP/M/1 **Queue**

In the case of exponential service, the Laplace transform of the packet length distribution is given by

$$H(s) = \frac{\mu}{s+\mu}. \tag{3.67}$$

Let λ_{avg} denote the average arrival rate to the queue. Then the queue utilization is given by

$$\rho = \frac{\lambda_{\text{avg}}}{\mu} = \frac{r_1}{\mu(r_0+r_1)}\lambda_0 + \frac{r_0}{\mu(r_0+r_1)}\lambda_1. \tag{3.68}$$

For our numerical example, we set the parameters as follows: $r_0 = 2$, $r_1 = 10^{-2}$, $\lambda_0 = 12$, $\lambda_1 = 3$, $\mu = 7$. Using (3.68), we compute $\rho = 0.435$. Applying (3.64) and inverting the Laplace transform, the density of the queue workload process is obtained as follows:

$$f_W(\sigma) = 1.67e^{-4.01\sigma} + 0.0277e^{-1.61\sigma} + 0.565\delta(\sigma), \quad \sigma \geq 0, \tag{3.69}$$

where $\delta(\sigma)$ denotes the Dirac delta function. The delta function in the density function implies a discontinuity in the distribution and survival functions at $\sigma = 0$. Since we are interested in tail probabilities, we shall only consider the case $\sigma > 0$.

The tail probability of queue workload process $W(t)$ is then given by

$$\mathsf{P}\{W(t) \geq \sigma\} = 0.4165e^{-4.01\sigma} + 0.0172e^{-1.61\sigma}, \tag{3.70}$$

for $t \geq 0$, $\sigma > 0$.

Since a mixture of exponential distributions is a special case of the phase-type distribution, the right-hand side of (3.70) can be written in the form $a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$, where

$$\boldsymbol{\pi} = (0.9603, 0.0397), \quad \mathbf{Q} = \begin{bmatrix} -4.01 & 0 \\ 0 & -1.61 \end{bmatrix}, \tag{3.71}$$

and $a = 0.4337$.

MMPP/E_2/1 **Queue**

When the packet length has an Erlang-2 distribution, we have

$$H(s) = \left(\frac{2\mu}{s + 2\mu}\right)^2. \tag{3.72}$$

Note that the corresponding Erlang-2 random variable is the sum of two independent exponential random variables, each of mean $1/2\mu$. The queue utilization in this case is also given by (3.68). The parameters are set as follows: $r_0 = 2$, $r_1 = 10^{-3}$, $\lambda_0 = 20$, $\lambda_1 = 3$, $\mu = 4$. Applying (3.64), we obtain the following pdf for the queue workload:

$$f_W(\sigma) = 1.05e^{-1.38\sigma} - 0.318e^{-11.6\sigma} - 4.65 \cdot 10^{-5}e^{-14\sigma} + 0.007e^{-0.444\sigma} + 0.248\delta(\sigma), \tag{3.73}$$

for $\sigma \geq 0$. The survival function of the queue workload for $\sigma > 0$ is then given by

$$\mathsf{P}\{W(t) \geq \sigma\} = 0.7609e^{-1.38\sigma} - 0.0274e^{-11.6\sigma} - 0.3321 \cdot 10^{-5}e^{-14\sigma} + 0.0158e^{-0.444\sigma}. \tag{3.74}$$

Note that the tail probability of $W(t)$ is not a mixture of exponentials, nor a phase-type distribution due to the negative coefficients on the right-hand side of (3.74). Nevertheless the tail probability of $W(t)$ has a matrix exponential distribution [11] and can be bounded using phase-type bounds. The phase-type bound can be obtained by simply dropping the negative terms on the right-hand side of (3.74), i.e.,

$$\mathsf{P}\{W(t) \geq \sigma\} \leq 0.7609e^{-1.38\sigma} + 0.0158e^{-0.444\sigma}. \tag{3.75}$$

In this case the tail probability of $W(t)$ has a bound in the form $a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$, where ,

$$\boldsymbol{\pi} = (0.9806, 0.0194), \qquad \mathbf{Q} = \begin{bmatrix} -1.38 & 0 \\ 0 & -0.444 \end{bmatrix}, \tag{3.76}$$

and $a = 0.7760$.

Aggregate MMPP/M/1 **input traffics**

As shown in (3.70), when an MMPP/M/1 input traffic with parameters set as Section 3.6.2, is fed to a server with constant rate of 1, the tail probability of the queue length will be bounded by a mixture of exponentials. Therefore, according to the *Characterization Theorem* [99, Theorem 1] the input traffic is gSBB, and also can be characterized by a phase-type traffic descriptor $[\rho_i = 1; (a_i = 0.4337, \boldsymbol{\pi}_i, \mathbf{Q}_i, T = \infty)],\ i = 1, 2$, where $\boldsymbol{\pi}_i,\ i = 1, 2$ and $\mathbf{Q}_i,\ i = 1, 2$ are given in (3.71).

Now we consider two i.i.d. input traffic streams, as inputs to a server with constant service rate of 2. In this case we have $A_1(t)$ and $A_2(t)$ as gSBB traffic inputs with upper rate $\rho_1 = \rho_2 = 1$ and bounding function

$$f_1(\sigma) = f_2(\sigma) = 0.4165e^{-4.01\sigma} + 0.0172e^{-1.61\sigma}. \tag{3.77}$$

According to the *Sum Theorem* [99, Theorem 3] the aggregate input $A(t) = A_1(t) + A_2(t)$ will be also gSBB with upper rate $\rho = \rho_1 + \rho_2 = 2$ and bounding function

$$g(\sigma) = f_1(p\sigma) + f_2((1-p)\sigma) = 0.4165e^{-4.01p\sigma} + \\ 0.0172e^{-1.61p\sigma} + 0.4165e^{-4.01(1-p)\sigma} + 0.0172e^{-1.61(1-p)\sigma} \tag{3.78}$$

where p is any value such that $0 < p < 1$. As we want this upper bound to be the tightest we set $p = 0.5$, in which case

$$g(\sigma) = 0.833e^{-2.005\sigma} + 0.0344e^{-0.805\sigma}. \tag{3.79}$$

which is a weighted sum of exponentials. Note that $A_1(t)$ and $A_2(t)$ can also be characterized by a phase-type traffic descriptor $[\rho_i = 1; (a_i = 0.4337, \boldsymbol{\pi}_i, \mathbf{Q}_i, T = \infty)]$, $i = 1, 2$, where $\boldsymbol{\pi}_i$, $i = 1, 2$ and $\mathbf{Q}_i$, $i = 1, 2$ are given in (3.71).

Considering the independence of the input traffics in this case, and according to Theorem 3.4.4, $A(t) = A_1(t) + A_2(t)$ can be characterized by a phase-type traffic descriptor $[\rho = \rho_1 + \rho_2 = 2; (b, \boldsymbol{\alpha}, \mathbf{H})]$, where $b = a_1 + a_2 - a_1a_2 = 0.6793$,

$$\boldsymbol{\alpha} = \left[\frac{a_1(1-a_2)}{b}\boldsymbol{\pi}_1, \frac{a_2(1-a_1)}{b}\boldsymbol{\pi}_2, \frac{a_1a_2}{b}\boldsymbol{\pi}_1, \mathbf{0}\right]$$

$$= (0.2359, 0.0097, 0.2359, 0.0097, 0.1806, 0.0075, 0, 0),$$

$$\mathbf{H} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_1 & \mathbf{q}_1\boldsymbol{\pi}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_2 \end{pmatrix} = \begin{pmatrix} -4.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.61 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1.61 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4.01 & 0 & 3.851 & 0.159 \\ 0 & 0 & 0 & 0 & 0 & -1.61 & 1.5461 & 0.0639 \\ 0 & 0 & 0 & 0 & 0 & 0 & -4.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.61 \end{pmatrix}. \tag{3.80}$$

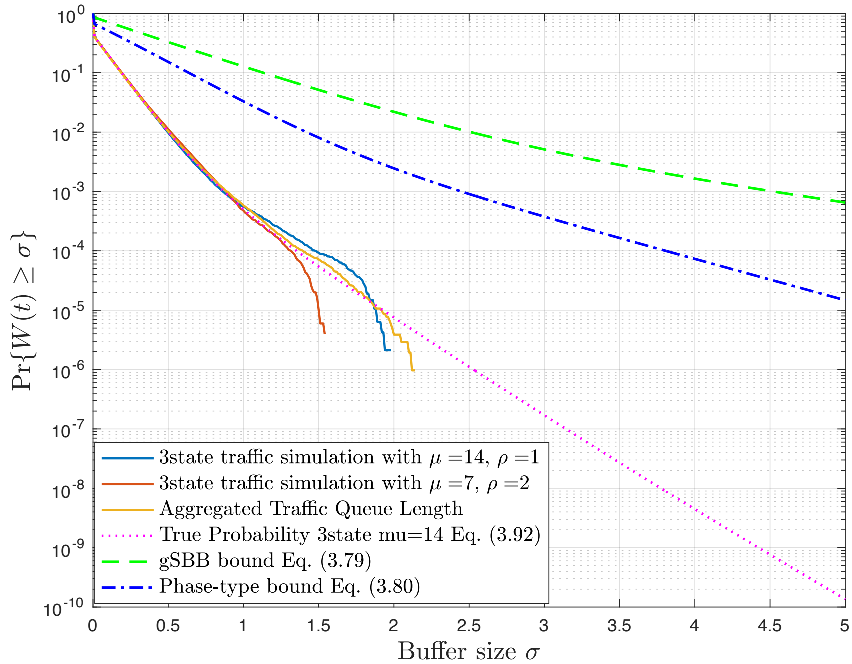


Figure 3.1: gSBB bound, phase-type bound, and true tail probability for aggregated MMPP/M/1 input traffic streams.

As we can see in Fig. 3.1, the phase-type bound is tighter than the gSBB bound. The true tail probability is derived through the following steps. As we have two identically independent 2-state MMPP's, with $\mathbf{R}$ and $\mathbf{\Lambda}$ as

$$\mathbf{R} = \begin{bmatrix} -r_0 & r_0 \\ r_1 & -r_1 \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ 10^{-2} & -10^{-2} \end{bmatrix}, \tag{3.81}$$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} = \begin{bmatrix} 12 & 0 \\ 0 & 3 \end{bmatrix}, \tag{3.82}$$

the aggregated arrival process will be according to (3.50) and (3.51) a 3-state MMPP process with with $\mathbf{R}$ and $\mathbf{\Lambda}$ as

$$\mathbf{R} = \begin{bmatrix} -2r_1 & 2r_1 & 0 \\ r_0 & -r_0 - r_1 & r_1 \\ 0 & 2r_0 & -2r_0 \end{bmatrix} = \begin{bmatrix} -0.02 & 0.02 & 0 \\ 2 & -2.01 & 0.01 \\ 0 & 4 & -4 \end{bmatrix}, \tag{3.83}$$

$$\mathbf{\Lambda} = \begin{bmatrix} 2\lambda_1 & 0 & 0 \\ 0 & \lambda_0 + \lambda_1 & 0 \\ 0 & 0 & 2\lambda_0 \end{bmatrix} = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 24 \end{bmatrix}. \tag{3.84}$$

In this case, the 3-state MMPP as a special case of MAP will give us $\mathbf{D}_0 = \mathbf{R} - \mathbf{\Lambda}$ and $\mathbf{D}_1 = \mathbf{\Lambda}$. Therefore, $D[\mathbf{G}] = \mathbf{D}_0 + \mathbf{D}_1\mathbf{G}$. Matrix $\mathbf{G}$ is derived according to (3.56), where in this case

$$\gamma_n = \int_0^\infty e^{-\theta x}\frac{(\theta x)^n}{n!}\mathrm{d}\tilde{H}(x) = \int_0^\infty e^{-\theta x}\frac{(\theta x)^n}{n!}\mu e^{-\mu x}\mathrm{d}x = \frac{\theta^n \mu}{(\theta+\mu)^{n+1}} \tag{3.85}$$

In the aggregate case we are trying to find

$$\max_{0 \le s \le t}\{A_1(s,t) + A_2(s,t) - (\rho_1 + \rho_2)(t-s)\} \tag{3.86}$$

which is the queue workload when the input traffic is aggregated as $A_1(t) + A_2(t)$ and the server constant rate is $\rho_1 + \rho_2$. In our model, that arrivals are MMPP and packet sizes are exponentially distributed with Laplace Transform $H(s) = \frac{\mu}{s+\mu} = \frac{7}{s+7}$, and constant server rate $\rho_1 = \rho_2 = 1$. When we are considering the aggregate case and we increase the server rate to $\rho_1 + \rho_2 = 2$, using (3.62) and (3.63), we incorporate this increase of the server rate in our calculation by doubling the μ to $\mu = 14$. In this case we have packet sizes that are exponentially distributed again, however, this time the average packet lengths are halved on average. Therefore, doubling the server rate to 2, which means the queue workload will be emptied in the half time, is equivalent to having a constant rate server with server rate

1, but packet lengths that are halved by setting $\mu = 14$. This change in the packet length instead of increasing the server rate, as we can see in Fig.3.1, is verified by simulation. Following the iterative equations of (3.58) and (3.58), we obtain the $\mathbf{G}$ matrix as

$$\mathbf{G} = \begin{pmatrix} 0.99866 & 0.00133 & 0 \\ 0.33254 & 0.66714 & 0.00031 \\ 0.37585 & 0.17597 & 0.44817 \end{pmatrix}. \tag{3.87}$$

We should note the following iteration is started with $\mathbf{G}_0 = \tilde{\boldsymbol{\pi}}\mathbf{1}$, where $\tilde{\boldsymbol{\pi}}$ is derived according to (3.59), which is

$$\tilde{\boldsymbol{\pi}} = (0.99, 0.0099, 10^{-5}). \tag{3.88}$$

The average arrival rate for this 3-state MMPP is derived according to (3.60) as $\lambda_{\text{avg}} = 6.089$ and the utilization factor of the queue will be $\rho = 0.4349$. Since $\rho < 1$, the invariant probability vector g associated with G is derived following (3.61) as

$$\mathbf{g} = (0.996, 0.00398, 3.66 \cdot 10^{-6}) \tag{3.89}$$

Then $W(s)$, the Laplace Transform of the virtual waiting time, is derived according to (3.62) and (3.63) as

$$W_v(s) = \frac{3.34}{s + 8.02} + \frac{0.0721}{s + 4.01} + \frac{0.00318}{s + 3.22} + 0.565 \tag{3.90}$$

In the time-domain we have

$$f_W(\sigma) = 3.34e^{-8.02\sigma} + 0.0721e^{-4.01\sigma} + 0.00318e^{-3.22\sigma} + 0.565\delta(\sigma), \quad \sigma \geq 0 \tag{3.91}$$

Also in this case virtual waiting time pdf has a discontinuity at $\sigma = 0$, which is as expected and is equal to $1 - \rho$. Therefore, the tail probability for the queue workload process $W(t)$

is given by

$$\mathsf{P}\{W(t) \geq \sigma\} = 0.4165e^{-8.02\sigma} + 0.0180e^{-4.01\sigma} + 9.8758 \cdot 10^{-4}e^{-3.22\sigma}, \tag{3.92}$$

for $t \geq 0$, $\sigma > 0$.

3.7 Conclusion

In this chapter we extended the concept of SBB and gSBB to the case of PHBB and phase-type bounds. We showed the PHBB or phase-type network calculus provides a closed network calculus for characterizing all the traffics inside the network, and can provide a mechanism to provide stochastic network performance guarantees. We showed how PHBB and phase-type bounds can be applied in a network with different service policies. The phase-type bounds were used in our case study to provide a stochastic bound on the workload size, and it was shown how this particular class of bounds have the advantage to provide tighter bounds compare to gSBB when the independence of the input traffics is utilized.

Chapter 4: Fitting Traffics to Phase-Type Bounds

One of the open questions in stochastic network calculus is how to obtain traffic burstiness bounds, in our case, phase-type bounds, for a given traffic stream. According to the Characterization theorem if the tail probability of the workload size, $\mathsf{P}\{W(t) \geq \sigma\}$, when the traffic is fed to a constant rate server of rate ρ , can be upper bounded by a bounding function of the form $a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$, for $\sigma \in [0, T]$, then the input traffic will be phase-type bounded with the corresponding bounding parameter. Thus, we feed the traffic stream to a server with constant rate and obtain a phase-type traffic burstiness bound from samples of the queue workload. In this chapter, we develop two methods for fitting these workload samples to a phase-type bound. The first approach is based on minimizing the squared error between the bounding function and the empirical tail probability of the workload. We refer to this approach as the least squares method. The second method is based on EM (expectation-maximization) algorithm, which was designed with the objective of maximizing the likelihood function. Parts of this chapter were published in [61].

The remainder of the chapter is organized as follows. In Section 4.1, we develop the first characterization method using a least squares method. In Section 4.2, we develop the second characterization method using EM algorithm. In Section 4.3, we investigate the bounds derived from the least squares and EM methods for a heavy-tailed traffic process. Concluding remarks are given in Section 4.4.

4.1 Least Squares Method

The objective of the least squares is to find an upper bound of the form $a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$ for tail probability, $\mathsf{P}\{W(t) \geq \sigma\}$, for every $\sigma \in [0, T]$, which is the tightest possible upper bound in the sense of minimizing the squared error of the bound. For a fixed number of phases, p, this is a semi-infinitely constrained optimization problem

$$\begin{aligned}
&\min_{a,\boldsymbol{\pi},\mathbf{Q}} \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma \\
&\text{subject to :} \\
&A \geq 0, \boldsymbol{\pi} \geq \mathbf{0} \\
&\sum_{i=1}^{p} \pi_i = 1 \\
&\det(\mathbf{Q}) \neq 0 \\
&\mathbf{Q}(i,i) \leq 0 \text{ for every } i = 1 \text{ to } p \\
&\mathbf{Q}(i,j) \geq 0 \text{ for every } i,j = 1 \text{ to } p, i \neq j \\
&\sum_{j=1}^{p} \mathbf{Q}(i,j) \leq 0 \text{ for every } i = 1 \text{ to } p \\
&a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1} \geq \mathsf{P}\{W(t) \geq \sigma\} \text{ for every } \sigma \in [0,T]
\end{aligned} \tag{4.1}$$

Solving this problem in case of a general form of $\mathbf{Q}$ or in other words general form of a phase-type random variable is very time consuming. However, there are methods to over come these difficulties by limiting the form of the phase-type. We can limit the phase-type to different forms of acyclic phase-type, hyper-Erlang, or even a simple case of mixture of exponentials. As mentioned previously, the class of hyper-Erlang distributions [33] [7, Corollary 4.4] and the class of acyclic phase-type distributions [26] are also dense in the set of distributions with nonnegative support. Therefore, by limiting the optimization search space of phase-type distributions to the class of hyper-Erlang or acyclic Phase-Type distributions we do not lose the denseness property of the phase-type distribution.

We note here that the goal is to find a bound on the survival function rather than a phase-type fit to empirical workload distribution. Due to the denseness property, in theory a phase-type distribution can be chosen to approximate the empirical workload distribution to an arbitrary degree of accuracy. Given such an approximation, an upper bound on the empirical workload distribution can be obtained by scaling the approximating phase-type distribution by a constant. Therefore, the concept of denseness is relevant in the context of optimization.

By limiting the phase-type distribution to the class of hyper-Erlang distributions, our optimization reduces to,

$$\begin{aligned}
&\min_{a,\boldsymbol{\pi},\mathbf{Q}} \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma \\
&\text{subject to :} \\
&A \geq 0, \boldsymbol{\pi} \geq \mathbf{0} \\
&\mathbf{Q} \text{ in the form of (2.30) and (2.31)} \\
&\lambda_i > 0 \text{ for every } i = 1 \text{ to } p \\
&a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1} \geq \mathsf{P}\{W(t) \geq \sigma\} \text{ for every } \sigma \in [0, T]
\end{aligned} \tag{4.2}$$

We should note we have omitted the constraint on $\sum_{i=1}^{p} \pi_i = 1$, because as the $\boldsymbol{\pi}$ is multiplied by a we can incorporate this multiplication factor into $\boldsymbol{\pi}$. By doing so, we do not need to account for the probability mass at the absorbing state $p+1$ since the components of $\boldsymbol{\pi}$ do not need to sum to one.

This optimization problem, is still a semi-infinite constrained optimization. We can simplify it further by omitting the last constraint, and by omitting a from optimization variables. In other words, we reduce the problem to that of fitting the empirical distribution with a hyper-Erlang distribution. In this case our optimization problem has the form

$$\begin{aligned}
&\min_{\boldsymbol{\pi},\mathbf{Q}} \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma \\
&\text{subject to :} \\
&\boldsymbol{\pi} \geq \mathbf{0} \\
&\mathbf{Q} \text{ in the form of (2.30) and (2.31)} \\
&\lambda_i > 0 \text{ for every } i = 1 \text{ to } p
\end{aligned} \tag{4.3}$$

Then whatever result we get we can make it an upper bound by multiplying it by a constant $a > 1$, and then use this bound as the initial point for (4.2) or simply use this result if it is tight enough. In our simulations we have used `fmincon` in MATLAB [72] for constrained optimization which is based on an interior-point algorithm [19, 20, 92] and `fseminf` in MATLAB for semi-infinite constrained optimization.

For a mixture of exponentials, the associated phase-type random variable can be represented by (2.28) and (2.29). In [34], the authors have shown that a mixture of exponentials can be a good fit to distributions with strictly decreasing pdfs. Therefore, for the case of mixture of exponentials our optimization problem will be as in (4.2) and (4.3) with $\mathbf{Q}$ in the form of (2.28).

On the other hand, for the case of CF1 acyclic phase-type random variable, depicted in

Figure 2.2, our optimization problem will reduce to,

$$\min_{a,\boldsymbol{\pi},\mathbf{Q}} \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma$$

subject to :

$\boldsymbol{\pi} \geq \mathbf{0}$

$\mathbf{Q}$ in the form of (2.35)

$\lambda_p \geq \lambda_{p-1} \geq \ldots \geq \lambda_2 \geq \lambda_1 \geq 0$

$$a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1} \geq \mathsf{P}\{W(t) \geq \sigma\} \text{ for every } \sigma \in [0, T] \tag{4.4}$$

In [44], this optimization method was used with the CF1 phase-type distribution, but their optimization method was based on the Frank-Wolfe algorithm [38]. This semi-infinite constrained optimization problem can also be further simplified into a constrained optimization problem in the same way. We omit the details.

We can improve the speed and accuracy of the optimization algorithms by providing the derivative of the objective function, $f(\boldsymbol{\pi}, \lambda_1, \ldots, \lambda_p) = \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma$ with respect to the parameters, π_i and λ_i for $i = 1, 2, \ldots, p$. A nice feature of limiting the phase-type random variable to the special cases of hyper-Erlang, CF1, or mixture of exponentials is that deriving the derivative is relatively simple. For π_i we have

$$\frac{\partial f(\boldsymbol{\pi}, \lambda_1, \ldots, \lambda_p)}{\partial \pi_i} = \frac{\partial}{\partial \pi_i} \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma$$

$$= \int_0^T \frac{\partial}{\partial \pi_i} (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1})^2 \mathrm{d}\sigma = -2\int_0^T \mathbf{e}_i e^{\mathbf{Q}\sigma}\mathbf{1}(\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}) \mathrm{d}\sigma, \tag{4.5}$$

where $\mathbf{e}_i$ is a p-element row vector with $\mathbf{e}_i(j) = 0$ for $j = 1, 2, \ldots, p,\ j \neq i$ and $\mathbf{e}_i(i) = 1$.

For λ_i, on the other hand, we have

$$\frac{\partial f(\boldsymbol{\pi}, \lambda_1, \ldots, \lambda_p)}{\partial \lambda_i} = \frac{\partial}{\partial \lambda_i} \int_0^T (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi} e^{\mathbf{Q}\sigma} \mathbf{1})^2 \mathrm{d}\sigma$$

$$= \int_0^T \frac{\partial}{\partial \lambda_i} (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi} e^{\mathbf{Q}\sigma} \mathbf{1})^2 \mathrm{d}\sigma = -2 \int_0^T \boldsymbol{\pi} \frac{\partial e^{\mathbf{Q}\sigma}}{\partial \lambda_i} \mathbf{1} (\mathsf{P}\{W(t) \geq \sigma\} - \boldsymbol{\pi} e^{\mathbf{Q}\sigma} \mathbf{1}) \mathrm{d}\sigma. \tag{4.6}$$

If we consider $\Phi(\boldsymbol{\lambda}, \sigma) = e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}$ and $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_p]$, we can write $\Phi(\boldsymbol{\lambda}, \sigma)$ and $\frac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$ in the form of Taylor series

$$\Phi(\boldsymbol{\lambda}, \sigma) = \sum_{k=0}^{\infty} \frac{\sigma^k}{k!} \mathbf{Q}^k(\boldsymbol{\lambda}) \tag{4.7}$$

$$\frac{\partial \Phi(\boldsymbol{\lambda}, \sigma)}{\partial \lambda_i} = \sum_{k=0}^{\infty} \frac{\sigma^k}{k!} \frac{\partial \mathbf{Q}^k(\boldsymbol{\lambda})}{\partial \lambda_i}. \tag{4.8}$$

We can write (4.8) recursively as,

$$\mathbf{Q}^k(\boldsymbol{\lambda}) = \mathbf{Q}(\boldsymbol{\lambda}) \mathbf{Q}^{k-1}(\boldsymbol{\lambda}) \tag{4.9}$$

$$\frac{\partial \mathbf{Q}^k(\boldsymbol{\lambda})}{\partial \lambda_i} = \frac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i} \mathbf{Q}^{k-1}(\boldsymbol{\lambda}) + \mathbf{Q}(\boldsymbol{\lambda}) \frac{\partial \mathbf{Q}^{k-1}(\boldsymbol{\lambda})}{\partial \lambda_i}. \tag{4.10}$$

Because of the term σ^k in (4.8), the summation will diverge. Therefore, we cannot substitute $\frac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i}$ as it is, for different cases of hyper-Erlang, CF1, or mixture of exponential. Hence, we apply the scaling and squaring algorithm described in [16,39] to derive $\dfrac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$. In this algorithm we first perform scaling as follows:

$$\mathbf{Q}(\boldsymbol{\lambda})\sigma \leftarrow \mathbf{Q}(\boldsymbol{\lambda})\sigma / r \qquad \text{s.t.:} \quad \|\mathbf{Q}(\boldsymbol{\lambda})\sigma / r\| < 0.2 \tag{4.11}$$

where $\|.\|$ is the Euclidean norm of a matrix and is defined as the maximum singular value of the matrix. In [16], r is chosen as

$$r = 2^M, \qquad M = \lceil \log_2(\|\mathbf{Q}(\boldsymbol{\lambda})\sigma\|_\infty) \rceil \tag{4.12}$$

where $\|.\|_\infty$ is the infinity norm of the matrix and is defined as

$$\|M_{m\times n}\|_\infty = \max_{1\le j\le m} \left(\sum_{j=1}^{n} |a_{ij}| \right) \tag{4.13}$$

Then by (4.7), (4.8), (4.9), and (4.10) we derive $\dfrac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma/r}}{\partial \lambda_i}$. Applying squaring for $m = M, M-1, \ldots, 1$ we have,

$$\Phi(\boldsymbol{\lambda}, \sigma/2^{m-1}) = \Phi^2(\boldsymbol{\lambda}, \sigma/2^m) \tag{4.14}$$

and based on (4.10) we obtain,

$$\frac{\partial \Phi(\boldsymbol{\lambda}, \sigma/2^{m-1})}{\partial \lambda_i} = \frac{\partial \Phi(\boldsymbol{\lambda}, \sigma/2^m)}{\partial \lambda_i} \Phi(\boldsymbol{\lambda}, \sigma/2^m) + \Phi(\boldsymbol{\lambda}, \sigma/2^m) \frac{\partial \Phi(\boldsymbol{\lambda}, \sigma/2^m)}{\partial \lambda_i}. \tag{4.15}$$

We can derive $\frac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$ recursively using (4.14) and (4.15). For the case of hyper-Erlang distributions we have

$$\frac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i} = \mathrm{diag}\left\{ \mathbf{0}_1, \mathbf{0}_2, \ldots, \mathbf{0}_{i-1}, \frac{\partial \mathbf{q}_i}{\partial \lambda_i}, \mathbf{0}_{i+1}, \ldots, \mathbf{0}_p \right\}, \tag{4.16}$$

where $\mathbf{0}_j$ for $j = 1, 2, \ldots, p,\ j \neq i$ are zero matrices with dimension of $r_j \times r_j$ and r_j as the order of the Erlang distribution in the jth branch. The partial derivative $\frac{\partial \mathbf{q}_i}{\partial \lambda_i}$, however, is derived based on (2.31) as follows:

$$\frac{\partial \mathbf{q}_i}{\partial \lambda_i} = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 \\ 0 & 0 & \dots & 0 & -1 \end{pmatrix}_{r_i \times r_i} \tag{4.17}$$

For the case of CF1, according to (2.35) we have

$$\frac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i} = \begin{cases} -\mathbf{e}_i^T \mathbf{e}_i + \mathbf{e}_i^T \mathbf{e}_{i+1}, & \text{for } i = 1, 2, \dots, p-1 \\ -\mathbf{e}_p^T \mathbf{e}_p, & \text{for } i = p \end{cases} \tag{4.18}$$

where $\mathbf{e}_i$ is defined before. Finally, for the case of mixtures of exponentials we have

$$\frac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i} = -\mathbf{e}_i^T \mathbf{e}_i \qquad \text{for } i = 1, 2, \dots, p \tag{4.19}$$

This scaling and squaring method is summarized in a pseudo-code in Algorithm 2. In Algorithm 2, $\boldsymbol{\epsilon}$ can be chosen as a matrix with appropriate size with $\epsilon = 10^{-16}$ as every element. The optimization method for different forms of phase-type random variables is summarized in Algorithm 3.

4.2 EM Method

Another method of deriving $a, \boldsymbol{\pi}, \mathbf{Q}$ such that

$$A\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1} \geq \mathsf{P}\{W(t) \geq \sigma\} \text{ for every } \sigma \in [0, T] \tag{4.20}$$

Algorithm 2 Calculate $\frac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$ using squaring and scaling

Input: σ, $\dfrac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i}$ as in (4.16) or (4.18) or (4.19), $\boldsymbol{\epsilon}$

Output: $\dfrac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$

$M = \lceil \log_2(\|\mathbf{Q}(\boldsymbol{\lambda})\sigma\|_\infty) \rceil$
$r = 2^M$
$\mathbf{Q}(\boldsymbol{\lambda})\sigma \leftarrow \mathbf{Q}(\boldsymbol{\lambda})\sigma / r$
$k = 2$
while $\sum_{l=0}^{k} \frac{\sigma^l}{l!} \frac{\partial \mathbf{Q}^l(\boldsymbol{\lambda})}{\partial \lambda_i} - \sum_{l=0}^{k-1} \frac{\sigma^l}{l!} \frac{\partial \mathbf{Q}^l(\boldsymbol{\lambda})}{\partial \lambda_i} > \boldsymbol{\epsilon}$ **do**
    $\mathbf{Q}^k(\boldsymbol{\lambda}) = \mathbf{Q}(\boldsymbol{\lambda})\mathbf{Q}^{k-1}(\boldsymbol{\lambda})$
    $\frac{\partial \mathbf{Q}^k(\boldsymbol{\lambda})}{\partial \lambda_i} = \frac{\partial \mathbf{Q}(\boldsymbol{\lambda})}{\partial \lambda_i}\mathbf{Q}^{k-1}(\boldsymbol{\lambda}) + \mathbf{Q}(\boldsymbol{\lambda})\frac{\partial \mathbf{Q}^{k-1}(\boldsymbol{\lambda})}{\partial \lambda_i}$
    $k \leftarrow k + 1$
end while
for $m = M, M-1, \ldots, 1$ **do**
    $\Phi(\boldsymbol{\lambda}, \sigma/2^{m-1}) = \Phi^2(\boldsymbol{\lambda}, \sigma/2^m)$
    $\frac{\partial \Phi(\boldsymbol{\lambda}, \sigma/2^{m-1})}{\partial \lambda_i} = \frac{\partial \Phi(\boldsymbol{\lambda}, \sigma/2^m)}{\partial \lambda_i}\Phi(\boldsymbol{\lambda}, \sigma/2^m) + \Phi(\boldsymbol{\lambda}, \sigma/2^m)\frac{\partial \Phi(\boldsymbol{\lambda}, \sigma/2^m)}{\partial \lambda_i}$
end for
Return $\dfrac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$

is using EM algorithm. As mentioned earlier, our objective is to find a bound in the form of $a\boldsymbol{\pi}e^{\mathbf{Q}\sigma}\mathbf{1}$, which is the survival function of a phase-type random variable multiplied by a constant a to bound the survival function of the workload, $\mathsf{P}\{W(t) \geq \sigma\}$. However, if we try to fit the workload distribution with a phase-type distribution we can always achieve an upper bound on the survival function by multiplying the derived survival function by a constant $a > 1$. Such a technique was also used in Section 4.1 to simplify the optimization problem. As in the least squares method, we particularize the phase-type distribution to a special case, in particular, the class of hyper-Erlang distributions.

4.2.1 Hyper-Erlang EM algorithm without fixed branch orders

In this section we develop an EM algorithm for fitting data samples to a special case of phase-type distribution, namely hyper-Erlangs. This algorithm is the core part our algorithm to derive the bounding parameter $(a, \boldsymbol{\pi}, \mathbf{Q})$, but it can be used generally in any

Algorithm 3 Calculate $a, \mathbf{Q}, \boldsymbol{\pi}$ using least squares method

Input: $\rho, T,$ Input traffic stream
Output: $a, \boldsymbol{\pi}, \mathbf{Q}$

Feed the traffic stream into a server with a constant rate ρ to get $W(t)$ samples
Derive $\mathsf{P}\{W(t) \geq \sigma\}$ for every $\sigma \in [0, T]$
Choose the form of phase-type random variable as hyper-Erlang, CF1, or mixture of exponentials
Choose p
if phase-type random variable is hyper-Erlang **then**
    Choose $r_1, r_2, \ldots, r_p$
end if
Choose initial values for $\boldsymbol{\pi}, \mathbf{Q}$ according to the chosen form
for every $\sigma \in [0, T]$ **do**
    Derive $\frac{\partial e^{\mathbf{Q}(\boldsymbol{\lambda})\sigma}}{\partial \lambda_i}$ using Algorithm 2
end for
Derive $\boldsymbol{\pi}, \mathbf{Q}$ by solving constrained optimization problem (4.3)for hyper-Erlang or its variant for CF1 or mixture of exponentails
Multiply $\boldsymbol{\pi}$ by an $a > 1$ to derive an upper bound
if upper bound is tight enough **then**
    Output $a = \sum_{i=1}^{p} \boldsymbol{\pi}$ and $\boldsymbol{\pi} = \frac{\boldsymbol{\pi}}{a}$ and $\mathbf{Q}$
else
    Use $a \cdot \boldsymbol{\pi}, \mathbf{Q}$ as the intitial values for semi-infinite constrained optimization problem
    Output $a = \sum_{i=1}^{p} \boldsymbol{\pi}$ and $\boldsymbol{\pi} = \frac{\boldsymbol{\pi}}{a}$ and $\mathbf{Q}$
end if
Return $A, \mathbf{Q}, \boldsymbol{\pi}$

fitting problem for fitting data samples to hyper-Erlang distributions. Some numerical examples of this algorithm are provided in Appendix C. This algorithm is a generalization of the algorithm described in Section 2.4. In the algorithm described in section 2.4, the order of the Erlang branches, $\mathbf{r} = (r_1, r_2, \ldots, r_M)$, are chosen in advance from the set $\mathcal{R}_n$, defined in (2.51), and then for every set of orders the hyper-Erlang parameter, $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\lambda}) = (\pi_1, \pi_2, \ldots, \pi_M, \pi_{M+1}, \lambda_1, \lambda_2, \ldots, \lambda_M)$, are derived using EM algorithm. This method, however, requires repeating these steps for every set of branch orders and then deriving the final result by choosing the best set of branch orders, based on the likelihood of the derived distribution function. The drawback of this method, as it is mentioned in [87], is that as n grows set $\mathcal{R}_n$ grows semi-exponentially in size, resulting in a huge processing burden. Another drawback of such exhaustive search is that, by forcing the phase-type order n, we impose a restriction on the branch orders. Instead of fixing the branch orders in advance and then trying the EM algorithm, we can however, incorporate deriving branch orders in the EM algorithm and make the algorithm more efficient in terms of time and required process. We should note that in this case our parameters will extend to $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$. In deriving order of branches in our EM algorithm we use mixture of gamma distributions and we derive the EM algorithm for this distribution. As Erlang distribution is a special case of gamma distribution, we then particularize the derived EM algorithm for mixture of Erlangs[1].

The gamma distribution is parameterized by (α, λ), where $\alpha > 0$ is called the *shape* parameter and $\lambda > 0$ is called the *rate* parameter. The pdf of this distribution is given by

$$f(x;\theta) = \frac{(\lambda x)^{\alpha-1}}{\Gamma(\alpha)}\lambda e^{-\lambda x}, \quad x \geq 0, \tag{4.21}$$

where the gamma function is defined by

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}\mathrm{d}x. \tag{4.22}$$

A gamma mixture is a convex combination of, say M, gamma distributions parameterized by $\theta_i = (\alpha_i, \lambda_i)$ and the mixing probabilities π_i, where $i = 1, \ldots, M$. Thus, the parameter of a gamma mixture can be written as $(\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\lambda})$, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_M)$ is the vector of mixing probabilities, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ is the shape vector, and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M)$ is the rate vector. Mixtures of gamma distributions have been used to model general service time distributions in networks of queues (cf. [52, p. 76]). The pdf of an Gamma mixture model

[1]In our work mthe terms *Erlang mixture* and *hyper-Erlang* distribution are used interchangeably

with parameter $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\theta}) = (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\lambda})$ is given by

$$f(x; \boldsymbol{\Theta}) = \sum_{i=1}^{M} \pi_i f(x; \theta_i). \tag{4.23}$$

In this section also we develop the algorithm in a way to account for the case of zero samples. Therefore, we introduce the additional mixture probability π_{M+1} such that $\sum_{i=1}^{M+1} \pi_i = 1$. The pdf of the extended Gamma mixture model is given by

$$\tilde{f}(x; \boldsymbol{\Theta}) = \sum_{i=1}^{M} \pi_i f(x; \theta_i) + \pi_{M+1}\delta(x), \tag{4.24}$$

An Erlang distribution may be characterized as a gamma distribution with parameter (r, λ) where the shape parameter r is a positive integer. The probability density function (pdf) of an Erlang distribution can be expressed as the convolution of r exponential pdfs of rate λ. An Erlang mixture may be viewed as a gamma mixture parameterized by $(\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$, where the shape vector $\mathbf{r} = (r_1, \ldots, r_M)$ consists of all positive integers. The number of parameter components in an Erlang mixture is only $3M$, compared to $n^2 + n - 1$ parameter components for the phase-type representation, where $n = \sum_{i=1}^{M} r_i$ is the number of *phases*. Moreover, since the phase-type representation is non-unique [77], general phase-type fitting algorithms are prone to getting stuck in sub-optimal solutions. These are among the reasons we are inclined to use the special case of hyper-Erlangs for deriving our bounds using EM algorithm.

The incorporation of the shape parameters or branch orders $\mathbf{r}$ in the set of parameter results in a dramatic improvement in computational efficiency and estimation accuracy compared to previous approaches, allowing distributions to be fitted easily with hundreds or thousands of phases. The accuracy of this new algorithm can seen from our numerical examples in Appendix C, which show the likelihood is higher and the first three moments of the fitted distribution are very close to those of the sample data. Kullback-Leibler divergence is also used as another measure of the accuracy of the fitted distributions.

Again in here, due to the term involving $\delta(x)$, the pdf in (4.24) cannot be used as a likelihood function for parameter estimation. Therefore, we use the Radon-Nikodym derivative of the probability law of the extended Gamma mixture model specified by (4.24) with respect to the measure $\nu = \lambda + \delta_x$, where μ and δ_x denote the Lebesgue and Dirac measures (see, e.g., [21]), respectively:

$$p(x; \mathbf{\Theta}) = \begin{cases} f(x; \mathbf{\Theta}), & x \neq 0, \\ \pi_{M+1}, & x = 0. \end{cases} \tag{4.25}$$

Under the (extended) Gamma mixture model, the log-likelihood of a sample vector $\mathbf{x} = (x_1, \ldots, x_K)$ is given by

$$\begin{aligned} \ell(\mathbf{\Theta} \mid \mathbf{x}) &= \log p(\mathbf{x}; \mathbf{\Theta}) = \log \prod_{k=1}^{K} p(x_k; \mathbf{\Theta}) \\ &= \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log \left(\sum_{i=1}^{M} \pi_i f(x_k; \theta_i) \right) + K_0 \log(\pi_{M+1}), \end{aligned} \tag{4.26}$$

where $K_0 = \#\{k : x_k = 0; k = 1, \ldots, K\}$ denotes the number of 0 samples. Similar to Section 2.4, in introducing unobserved data $\mathbf{y} = (y_1, \ldots, y_K)$, where $y_k \in \{1, \ldots, M+1\}$ is the mixture component corresponding to the observed sample x_k, it is assumed $y_k = M+1$ when $x_k = 0$. The joint likelihood of (x_k, y_k) is given by

$$p(x_k, y_k; \mathbf{\Theta}) = \pi_{y_k} p(x_k; \mathbf{\Theta}). \tag{4.27}$$

Similar to Section 2.4, the complete data log-likelihood is then given by

$$\ell(\boldsymbol{\Theta} \mid \mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}, \mathbf{y}; \boldsymbol{\Theta}) = \log \prod_{k=1}^{K} p(x_k, y_k; \boldsymbol{\Theta})$$

$$= \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\pi_{y_k} f(x_k; \theta_k)) + K_0 \log(\pi_{M+1}). \tag{4.28}$$

Posterior probability $\{\mathbf{Y} = \mathbf{y}\}$, where $\mathbf{Y}$ denotes the random vector corresponding to the realization $\mathbf{y}$ of the unobserved data, given the observed sample vector $\mathbf{x}$, and current estimate of the parameters, $\hat{\boldsymbol{\Theta}}$ is given by,

$$p(\mathbf{y}|\mathbf{x}, \hat{\boldsymbol{\Theta}}) = \prod_{k=1}^{K} p(y_k|x_k, \hat{\boldsymbol{\Theta}}) \tag{4.29}$$

When $x_k \neq 0$, the posterior probability $p(y_k \mid x_k; \hat{\boldsymbol{\Theta}})$ is given by

$$p(y_k \mid x_k; \hat{\boldsymbol{\Theta}}) = \frac{\hat{\pi}_{y_k} f(x_k; \hat{\theta}_{y_k})}{\sum_{i=1}^{M} \hat{\pi}_i f(x_k; \hat{\theta}_i)}, \tag{4.30}$$

where $y_k \in \{1, \ldots, M\}$. Doing the same algebraic manipulation as Section 2.4, the following expression for the auxiliary function can be obtained:

$$Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}) = \sum_{i=1}^{M} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\pi_i) \cdot q(i|x_k, \hat{\boldsymbol{\Theta}}) + \sum_{i=1}^{M} g_i(\theta_i) + N_0 \log(\pi_{M+1}), \tag{4.31}$$

where

$$g_i(\theta_i) = \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(p_i(x_k|\lambda_i)) \cdot q(i|x_k, \hat{\boldsymbol{\Theta}}) \tag{4.32}$$

and $\theta_i = (\alpha_i, \lambda_i)$, $i = 1, \ldots, M$.

Maximization of the auxiliary function results in a sequence of parameter estimates with non-decreasing incomplete data log-likelihood values given by (4.26). From (4.31) it can be seen that maximizing of $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ with respect to $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\theta})$ is equivalent to maximizing the two terms separately with respect to $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, respectively. Maximization of the first term in (2.44) can be done using Lagrange multipliers, which yields

$$\pi_i = \frac{1}{K} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\boldsymbol{\Theta}}), \quad i = 1, \ldots, M. \tag{4.33}$$

$$\pi_{M+1} = \frac{K_0}{K}. \tag{4.34}$$

We note that, similar to Section 2.4, π_{M+1} corresponds to the probability of a 0 sample and remains fixed for all EM iterations. For the gamma mixture, maximization of the second term in (4.31) can be carried out by maximizing $g_i(\theta_i)$ separately for each $i = 1, \ldots, M$. As noted in [10, p. 168], for the gamma mixture, a function of the form $g_i(\theta)$ in (4.32) is strictly concave with respect to (α, λ). For completeness, we provide a proof of the strict concavity of $g_i(\alpha, \lambda)$ in Appendix B.1, Proposition B.1.3. The strict concavity property of the auxiliary function guarantees convergence of the EM algorithm to a stationary point of the likelihood function [31, p. 1542], [10, Theorem 3.1].

When the shape parameter α is fixed, setting

$$\frac{\partial}{\partial \lambda} g_i(\alpha, \lambda) = 0 \tag{4.35}$$

yields the same result (2.50) as

$$\lambda(\alpha) = \alpha \cdot \frac{\sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\boldsymbol{\Theta}})}{\sum_{\substack{k=1 \\ x_k \neq 0}}^{K} x_k p(i \mid x_k; \hat{\boldsymbol{\Theta}})}, \tag{4.36}$$

where we have made use of (B.6). Since $\lambda(\alpha)$ is a positive multiple of α,

$$h_i(\alpha) := g_i(\alpha, \lambda(\alpha)) \tag{4.37}$$

is strictly concave with respect to α. Hence, the solution to

$$(\alpha_i, \lambda_i) = \underset{\alpha, \lambda > 0}{\arg\max} \;\; g_i(\alpha, \lambda) \tag{4.38}$$

can be written as

$$\alpha_i = \underset{\alpha > 0}{\arg\max} \;\; h_i(\alpha), \tag{4.39}$$

$$\lambda_i(\alpha_i) = \alpha_i \cdot \frac{\sum_{\substack{k=1 \\ x_k \neq 0}}^{K} p(i \mid x_k; \hat{\boldsymbol{\Theta}})}{\sum_{\substack{k=1 \\ x_k \neq 0}}^{K} x_k p(i \mid x_k; \hat{\boldsymbol{\Theta}})}. \tag{4.40}$$

Since $h_i(\alpha)$ is strictly concave, the maximizing value of α must be a stationary point. Using (4.37) and (4.36), we have

$$h_i(\alpha) = \sum_{k=1}^{K} p(i \mid x_k; \hat{\boldsymbol{\Theta}}) \left[\alpha \log \left(\frac{A_i}{B_i} \right) + \alpha \log \alpha + (\alpha - 1) \log x_k - \log \Gamma(\alpha) - \frac{A_i}{B_i} \alpha x_k \right], \tag{4.41}$$

where

$$A_i := \sum_{k=1}^{K} p(i \mid x_k; \hat{\boldsymbol{\Theta}}) \tag{4.42}$$

$$B_i := \sum_{k=1}^{K} x_k p(i \mid x_k; \hat{\boldsymbol{\Theta}}). \tag{4.43}$$

An expression for the derivative of $h_i(\alpha)$ can be obtained as follows:

$$h_i'(\alpha) = C_i + A_i \left[\log \alpha - \psi(\alpha)\right], \tag{4.44}$$

where

$$\psi(\alpha) := \frac{d}{d\alpha} \log \Gamma(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} \tag{4.45}$$

is known as the digamma function and

$$\begin{aligned} C_i : &= \sum_{k=1}^{K} p(i \mid x_k; \hat{\mathbf{\Theta}}) \left[\log\left(\frac{A_i}{B_i}\right) + 1 + \log x_k - \frac{A_i}{B_i} x_k\right] \\ &= A_i \log\left(\frac{A_i}{B_i}\right) + \sum_{k=1}^{K} p(i \mid x_k; \hat{\mathbf{\Theta}}) \log(x_k). \end{aligned} \tag{4.46}$$

We note that A_i, B_i and C_i are not functions of α. Clearly, $A_i \geq 0$ and $B_i \geq 0$. It can also be shown that $\log \alpha - \psi(\alpha) > 0$ for all $\alpha > 0$. It is shown in Appendix B.4, a positive stationary point always exists. Setting $h_i'(\alpha) = 0$ in (4.44), the stationary point α_i can be obtained as the root of the equation

$$C_i + A_i \left[\log \alpha - \psi(\alpha)\right] = 0. \tag{4.47}$$

In Appendix B.2, we show that for $\alpha > 1/6$,

$$\frac{1}{2\alpha} \leq \log \alpha - \psi(\alpha) \leq \frac{1}{2\left(\alpha - \frac{1}{6}\right)}, \tag{4.48}$$

Using (4.47) and (4.48), we can obtain bounds on the stationary point α_i as follows:

$$b_i \leq \alpha_i \leq b_i + \frac{1}{6}, \tag{4.49}$$

where

$$b_i = \frac{-A_i}{2C_i}, \tag{4.50}$$

Using $[b_i, b_i + 1/6]$ as an initial interval for a numerical root-finding method, the stationary point of $h_i(\alpha)$ can be computed very efficiently. We have used the built-in MATLAB function `fzero` to compute the root α_i in this manner. The `fzero` function uses a combination of bisection, secant, and inverse quadratic interpolation methods [2].

The EM algorithm for the gamma mixture model is summarized in Algorithm 4. In each iteration of the algorithm, the E-step and M-step are carried out in line 5 and lines 6–11, respectively. After $\{(\pi_i, \alpha_i, \lambda_i)\}_{i=1}^{M}$ have been determined, the mixture parameter estimate is updated as $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\lambda})$, completing one iteration of the EM algorithm. The EM algorithm is continued until either the relative difference between the incomplete data log-likelihoods of the last two parameter estimates falls below a threshold ϵ or a limit, N, or the maximum number EM iterations is reached.

The EM algorithm for the Erlang mixture model follows from that of the gamma mixture model. The function $h_i(\alpha)$ in (4.37), defined for the gamma mixture model, was shown to be strictly concave in $\alpha > 0$. Therefore, the sequence $\{h_i(r)\}_{r=1}^{\infty}$ is strictly concave. Hence, if α_i is as given in (4.39), the value of r that maximizes $h_i(r)$ must be one of the two integers closest in value to α_i, i.e.,

$$r_i := \operatorname*{arg\,max}_{r=1,2,\dots} \; h_i(r) = \operatorname*{arg\,max}_{r \in \{\lfloor \alpha_i \rfloor \vee 1, \lceil \alpha_i \rceil\}} \; h_i(r). \tag{4.51}$$

Although a closed-form expression for α_i is not available, we know that it lies within the interval of length $1/6$ given by (4.49). This implies that r_i can be computed in terms of b_i as follows:

$$r_i = \operatorname*{arg\,max}_{r \in \{\lfloor b_i \rfloor \vee 1, \lceil b_i \rceil, \lceil b_i \rceil + 1\}} \; h_i(r), \tag{4.52}$$

which can be simplified further as follows (see Appendix B.3):

$$r_i = \underset{r \in \{\lfloor b_i \rfloor \vee 1, \lceil b_i \rceil\}}{\arg\max} \; h_i(r). \tag{4.53}$$

Given the value of r_i, the associated rate parameter is given by (cf. (4.40))

$$\lambda_i(r_i) = r_i \cdot \frac{\sum_{k=1}^{K} p(i \mid x_k; \hat{\phi})}{\sum_{k=1}^{K} x_k p(i \mid x_k; \hat{\phi})}. \tag{4.54}$$

Algorithm 4 can be adapted for the Erlang mixture model by replacing $\boldsymbol{\alpha}$ with $\mathbf{r}$ in the **Output** line and in lines 1 and 12. Lines 10 and 11 are replaced with the following:

10. Compute r_i using (4.53)
11. Compute $\lambda_i = \lambda(r_i)$ using (4.40)

We shall refer to Algorithm 4 for the gamma mixture model as Alg-Ga and the modification of this algorithm for the Erlang mixture model as Alg-Er.

Selection of the initial parameter estimate is often crucial to obtaining good parameter estimates via an EM algorithm. Therefore, we have developed an initialization system for Alg-Ga and Alg-Er. It has been observed in [34] that a monotonically decreasing pdf can be well approximated by a mixture of exponential pdfs. For such cases, we have initialized the shape vector ($\boldsymbol{\alpha}$ for gamma mixture and $\mathbf{r}$ for Erlang mixture) to a vector of all ones, i.e., $(1, \ldots, 1)$. For other pdfs that are not monotonically decreasing, such as the Weibull$(1, 5)$ (see Table C.1), the initialization $(1, 10, \ldots, 10)$ has worked well. For a given setting of the shape vector, we have initialized $\boldsymbol{\pi}$ and $\boldsymbol{\lambda}$ such that the first moment of the initial parameter estimate matches the empirical mean $\hat{\tau} = \frac{1}{K}\sum_{k=1}^{K} x_k$, as was done in [87]:

$$\pi_i = \frac{1}{M}, \;\; \lambda_i = \frac{i r_i}{M\hat{\tau}} \sum_{l=1}^{M} \frac{1}{l}, \tag{4.55}$$

for $i = 1, \ldots, M$.

As it is shown in examples in Appendix C, this algorithm is much more efficient comparing to the previous methods in phase-type or hyper-Erlang fitting. The E-step has complexity $O(M)$, while the M-step has complexity $O(MK)$. Therefore, the complexity of each iteration of the innermost loop (*repeat-until*) loop in Algorithm 1 is $O(MK)$. Hence, the overall complexity of the algorithm can be stated as $O(MKN)$, where N is maximum number of EM iterations. That is, the complexity of the algorithm is linear in the number of data samples K and also in the number of mixture components M.

Alg-Er differs from the approach of [87] by incorporating the shape vector $\mathbf{r}$ as a component in the Erlang mixture paramter, i.e., $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$, to be optimized via an EM algorithm. In the approach of [87], $\mathbf{r}$ is treated as a constant while the parameter $(\boldsymbol{\pi}, \boldsymbol{\lambda})$ is optimized by an EM algorithm. The computational complexity of each EM iteration is $O(MK)$. Since $M \leq n$ and the total number of EM iterations is at most N, the overall complexity of the algorithm can be stated as $O(nKN)$. The EM algorithm is applied with all shape parameters $\mathbf{r}$ from the set, $\mathcal{R}_n$, of all distinct settings of $(r_1, \ldots, r_n)$ such that the number of phases n is equal to a predefined constant. The parameter $(\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$ associated with the highest log-likelihood value determined from the exhaustive search of $\mathcal{R}_n$ is chosen as the parameter estimate. We note that the set $\mathcal{R}_n$ is equivalent to the set of integer partitions of n. The number of partitions of n, i.e., $\#\mathcal{R}_n$, grows as $Ce^{\sqrt{n}}$ as $n \to \infty$, where $C = e^{\pi\sqrt{2/3}}$ (cf. [5, p. 70]). Therefore, the overall complexity of the approach of [87] can be stated as $O\left(ne^{\sqrt{n}}KN\right)$. The exponential factor $e^{\sqrt{n}}$ explains why the approach is not computationally feasible for large values of n.

4.2.2 EM-based algorithm to derive bounding parameter

The algorithm to derive $(a, \boldsymbol{\pi}, \mathbf{Q})$ using this developed EM algorithm is summarized in Algorithm 5. In this method also after deriving $(\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$ we need to find a, such that (4.20) holds. By increasing a from 1 we can get the smallest value of a, such that (4.20) holds.

Algorithm 4 EM algorithm for gamma mixture model.

Input: $\mathbf{x} = (x_1, \ldots, x_K)$; M; $\epsilon = 10^{-6}$; $N = 100$
Output: $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\lambda})$
Set $\pi_{M+1} = \frac{K_0}{K}$
Choose initial estimate $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\lambda})$ [see (4.55) and the discussion before it]
$\ell \leftarrow \log p(\mathbf{x}; \boldsymbol{\Theta})$ using (4.26); $\iota \leftarrow 1$
repeat
    $\hat{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}$; $\hat{\ell} \leftarrow \ell$
    E-step: Compute $\{p(i \mid x_k; \hat{\boldsymbol{\Theta}})\}_{i=1}^{M}$ using (4.30)
    M-step: Compute $\{\pi_i\}_{i=1}^{M}$ using (4.34)
    for $i = 1$ to M **do**
        Compute b_i using (4.50)
        Compute α_i as the root of (4.47) using the initial interval $\left[b_i, b_i + \frac{1}{6}\right]$
        Compute $\lambda_i = \lambda(\alpha_i)$ using (4.40)
    end for
    $\boldsymbol{\Theta} \leftarrow (\boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{\lambda})$; $\ell \leftarrow \log p(\mathbf{x}; \boldsymbol{\Theta})$; $\iota \leftarrow \iota + 1$
until $(\ell - \hat{\ell})/\hat{\ell} < \epsilon$ or $\iota = N$
Return $\boldsymbol{\Theta}$

4.3 Case Study: M/G/1 Heavy-Tailed Queue

One of the commonly used models to describe the traffic streams in modern application of teletraffic theory, like Ethernet Local Area Networks [94], Wide Area Networks [80] is non-exponential tail models of the service time. In these models traffic streams show a striking similarity when is considered over time periods of hours, minutes, or milliseconds. These self-similarity or the consequent long-range dependencies of the traffic has been modeled by one or more on-off sources in a fluid queue, with heavy-tailed on periods [15] in at least one of the sources, or by ordinary single server queues, M/G/1, with heavy-tailed distributed service time [14].

In this section we use the model of the M/G/1 queue in [14]. In this model service time distribution, $B(t)$, has the following asymptotic behavior

$$1 - B(t) = O(t^{-v}), \qquad t \to \infty \tag{4.56}$$

Algorithm 5 Calculate $a, \mathbf{Q}, \boldsymbol{\pi}$ using EM algorithm

Input: Workload samples; T; M; $\epsilon = 10^{-6}$; $N = 100$
Output: $a, \mathbf{Q}, \boldsymbol{\pi}$

Set $\pi_{M+1} = \frac{K_0}{K}$
Choose initial estimate $\boldsymbol{\Theta} = (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$ [see (4.55) and the discussion before it]
$\ell \leftarrow \log p(\mathbf{x}; \boldsymbol{\Theta})$ using (4.26); $\iota \leftarrow 1$
repeat
    $\hat{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}$; $\hat{\ell} \leftarrow \ell$
    E-step: Compute $\{p(i \mid x_k; \hat{\boldsymbol{\Theta}})\}_{i=1}^{M}$ using (4.30)
    M-step: Compute $\{\pi_i\}_{i=1}^{M}$ using (4.34)
    for $i = 1$ to M **do**
        Compute b_i using (4.50)
        Compute r_i using (4.53)
        Compute $\lambda_i = \lambda(r_i)$ using (4.54)
    end for
    $\boldsymbol{\Theta} \leftarrow (\boldsymbol{\pi}, \mathbf{r}, \boldsymbol{\lambda})$; $\ell \leftarrow \log p(\mathbf{x}; \boldsymbol{\Theta})$; $\iota \leftarrow \iota + 1$
until $(\ell - \hat{\ell})/\hat{\ell} < \epsilon$ or $\iota = N$
return $\boldsymbol{\Theta}$
$\mathbf{Q}$ is derived based on $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_M]$ and $\mathbf{r} = [r_1, r_2, \ldots, r_M]$ according to (2.30), and (2.31)
Choose smallest a such that we have $a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1} \geq \mathsf{P}\{W(t) \geq \sigma\}$ for every $\sigma \in (0, T]$, where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_M]$.
return $a, \mathbf{Q}, \boldsymbol{\pi}$

where $1 < v < 2$. More precisely, service time, $\tau_{\boldsymbol{\theta}}$, has the following distribution function

$$\mathsf{P}\{\tau_{\boldsymbol{\theta}} < t | \boldsymbol{\theta} = \theta\} = 1 - \delta(\frac{\theta}{\theta + t})^v, \qquad t \geq 0 \tag{4.57}$$

where $0 < \delta \leq 1$ and $\boldsymbol{\theta}$ is a random variable with Gamma density function $f_{\boldsymbol{\theta}}(\theta)$ as

$$f_{\boldsymbol{\theta}}(\theta) = \frac{s^{2-v}}{\Gamma(2-v)} \theta^{1-v} e^{-s\theta}, \qquad \theta > 0, 1 < v < 2. \tag{4.58}$$

where s is positive constant. For this service time random variable, $\tau_{\boldsymbol{\theta}}$, we have

$$\beta := E\{\tau_{\boldsymbol{\theta}}\} = \frac{2-v}{v-1} \frac{\delta}{s} \tag{4.59}$$

One of the cases considered in [14] is when $\upsilon = \frac{3}{2}$. For this particular case we have,

$$B(t) \equiv \mathsf{P}\{\tau_{\boldsymbol{\theta}} \leq t\} = 1 + \delta \left[\frac{2\sqrt{st}}{\sqrt{\pi}} - (1 + 2st)e^{st}\mathrm{erfc}(\sqrt{st}) \right], \tag{4.60}$$

for $t \geq 0$, and with complementary error function given as

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-u^2} \mathrm{d}u \tag{4.61}$$

This service time distribution for $t \to \infty$ can be approximated using the following relation

$$1 - B(t) = \frac{2\delta}{\pi} \sum_{n=1}^{H} (-1)^{n-1} \frac{n\Gamma(n + \frac{1}{2})}{(st)^{n+\frac{1}{2}}} + O\left[\left(\frac{1}{st} \right)^{H+3/2} \right] \tag{4.62}$$

for every finite $H \in \{0, 1, \ldots\}$. Therefore, with this service time distribution in a single server queue with Poisson arrivals, M/G/1, and utilization factor $\rho = \lambda\beta < 1$, where λ is the arrival rate, stationary waiting time distribution will be

$$\begin{aligned} W(t) := \mathsf{P}\{w \leq t\} = 1 - \frac{1 + \sqrt{\rho}}{2} \sqrt{\rho} e^{(1-\sqrt{\rho})^2 st} \cdot \mathrm{erfc}\left[(1 - \sqrt{\rho})\sqrt{st} \right] \\ + \frac{1 - \sqrt{\rho}}{2} \sqrt{\rho} e^{(1-\sqrt{\rho})^2 st} \cdot \mathrm{erfc}\left[(1 + \sqrt{\rho})\sqrt{st} \right]. \end{aligned} \tag{4.63}$$

for $t > 0$. This stationary waiting time distribution for $t \to \infty$ can be approximated using the following relation

$$\begin{aligned} 1 - W(t) = \frac{\sqrt{\rho}}{2\pi}(1 - \rho) \sum_{m=0}^{H} (-1)^m \cdot \left[\frac{1}{(1 - \sqrt{\rho})^{2m+2}} - \frac{1}{(1 + \sqrt{\rho})^{2m+2}} \right] \cdot \frac{\Gamma(m + \frac{1}{2})}{(st)^{m+1/2}} \\ + O\left[\left(\frac{1}{st} \right)^{H+3/2} \right] \end{aligned} \tag{4.64}$$

for every finite $H \in \{0, 1, \ldots\}$.

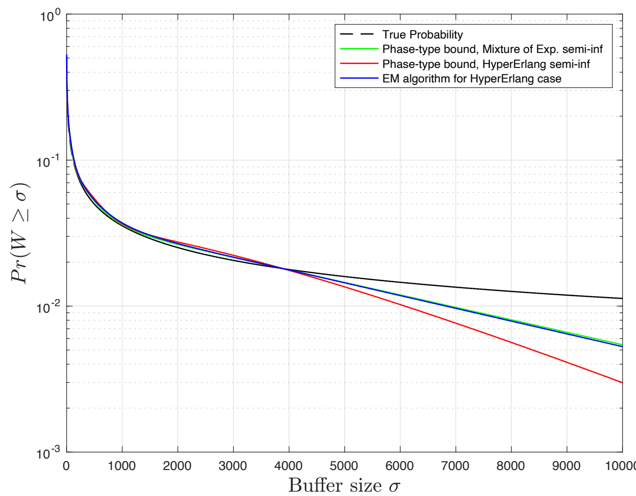


Figure 4.1: Phase-type bound, and true tail probability for a heavy-tailed workload in M/G/1 queue.

Remark. In this section we use the same technique used in 3.6.1 to interpret the stationary waiting time as the workload size divided by C, in a queue with a constant rate server with rate C.

Now we try to bound heavy-tailed workload in a M/G/1 queue described previously, with heavy-tailed service time by a phase-type bounds.

In our case study we consider $s = \delta = 1$ and $\lambda = 0.5$. Therefore, we will have a heavy-tailed workload which is derived by (4.63) for $\sigma \in (0, 120]$ and by (4.64) for $\sigma \in (120, \infty)$, where $\sigma > 0$ is the workload size. We have also

$$\mathsf{P}\{\sigma = 0\} = 1 - \rho = 1 - \lambda\beta = 0.5 \tag{4.65}$$

Survival function of this waiting time is shown in Fig. 4.1.

Here the workload distribution is heavy-tailed therefore, it cannot be bounded for all σ by a bound in the form of a phase-type survival function, unless the number of phases goes to infinity. Nevertheless, we can bound this distribution by a phase-type bound with practically

Table 4.1: Phase-type bound parameters for M/G/1 queue with heavy-tailed service time

	Method	a	$\boldsymbol{\pi}$	$\boldsymbol{\Lambda}$
1	**Opt. Mix. of Exp.**	0.543	$[0.1836, 0.140, 0.276, 0.079, 0.320]$	$[-1.096, -3.2e-3, -0.0219, -2.24e-4, -0.139]$
2	**Opt. Mix. of Exp. semi-inf**	0.5092	$[0.1694, 0.1347, 0.2822, 0.0762, 0.3375]$	$[-1.1052, -0.0028, -0.0197, -1.966e-4, -0.1385]$
3	**Opt. CF1**	0.5931	$[0.0772, 0.1520, 0.3308, 0.3401, 0.01]$	$[-2.57e-4, -0.0042, -0.0371, -0.3571, -47.54]$
4	**Opt. CF1 semi-inf**	0.5115	$[0.0747, 0.1388, 0.2828, 0.4252, 0.0785]$	$[-2.12e-4, -0.0033, -0.0257, -0.188, -47.589]$
5	**Opt. hyper-Erlang**	0.5255	$[0.0767, 0.1106, 0.2309, 0.3781, 0.2037]$	$[-4.74e-4, -0.0045, -0.0262, -0.172, -1.75]$
6	**Opt. hyper-Erlang semi-inf**	0.52	$[0.0652, 0.0979, 0.2104, 0.3543, 0.2722]$	$[-4.047e-4, -0.0039, -0.0242, -0.1606, -1.755]$
7	**EM hyper-Erlang**	1.058	$[0.0373, 0.0588, 0.1218, 0.1438, 0.1384]$	$[-2.0147e-4, -0.2.6e-3, -0.0149, -0.0756, -0.3756]$
8	**EM General Phase-Type**	0.5158	$[0.0474, 4.43e-9, 0.953, 8.12e-9, 5.69e-7]$	ref. to (4.66)

Table 4.2: Phase-type bound performance for M/G/1 queue with heavy-tailed service time

1	**Method**	**Objective function value**($\times T$)	**Log-likelihood**($\times 10^6$)	**Order**	
2	**Opt. Mix. of Exp.**	0.051	-3.259	5	
3	**Opt. Mix. of Exp. semi-inf**	0.01	-3.257	5	
4	**Opt. CF1**	0.1336	-3.3037	5	
5	**Opt. CF1 semi-inf**	0.0192	-3.294	5	
6	**Opt. hyper-Erlang**	0.224	-3.263	$[2, 2, 2, 2, 2]$	
7	**Opt. hyper-Erlang semi-inf**	0.0395	-3.259	$[2, 2, 2, 2, 2]$	
8	**EM hyper-Erlang**	0.0493	-3.248	$[1, 1, 1, 1, 1]$	
9	**EM General Phase-Type**	0.5158	0.0337	-3.251	5

limited number of phases for a limited interval of $\sigma \in [0, T]$. In practice also workload size cannot be considered unlimited, because buffer size in reality is always bounded.

In deriving a phase-type bound we have used both methods of least squares and EM algorithm described in this chapter.

In this example we have chosen $T = 3890$. In least squares method, as it is explained in Algorithm 3 we have found $a, \boldsymbol{\pi}, \mathbf{Q}$ by solving constrained optimization problem and then we have used these solutions as the initial values to solving the semi-infinite optimization problem. In EM Algorithm method number of samples generated based on (4.63) and (4.64) are $N = 10^6$. We have compared objective function value of different method, which is square of area difference between the phase-type bound and the tail probability of the workload and is expressed in (4.1). In the case of hyper-Erlang using EM method the result is a simple case of mixture of exponentials. Values of $a, \boldsymbol{\pi}, \mathbf{Q}$ for different solutions are presented in Table 4.1 and 4.2. For the case of hyper-Erlang using EM algorithm which is derived based on Algorithm 1 we have presented $\hat{\boldsymbol{\pi}}$. In the case of hyper-Erlang in least squares method order of the phase-type is considered as $r = [2, 2, 2, 2, 2]$ to be close enough to the hyper-Erlang solution of the EM algorithm with highest likelihood with order $r = [1, 1, 1, 1, 1]$. In other cases order of the phase-type random variable is considered as $r = 5$. In case of hyper-Erlang, although $\boldsymbol{\pi}$ is in the form of (2.32), we have omitted the 0's for brevity.

For the case of general phase-type bound we have a phase-type bound with 5 phases and $\mathbf{Q}$ as,

$$\mathbf{Q} = \begin{pmatrix} -4.87e-02 & 1.74e-05 & 3.29e-02 & 1.75e-05 & 1.22e-02 \\ 3.08e-06 & -1.49e-02 & 2.57e-07 & 1.37e-02 & 1.18e-03 \\ 1.344e-01 & 1.036e-06 & -2.59e-02 & 1.28e-07 & 2.347e-04 \\ 1.103e-06 & 3.266e-03 & 7.49e-09 & -3.41e-03 & 1.43e-04 \\ 4.3e-03 & 8.72e-04 & 3.461e-06 & 5.053e-04 & -5.681e-03 \end{pmatrix} \tag{4.66}$$

As we can see from table. 4.2, in least squares method algorithms going from constraint optimization to the semi-infinite case, as expected, decreases the objective function value or makes the bound tighter. As it is shown in the table the best result is achieved by a simple case of mixture of exponentials. This can be justified considering the fact workload density is a monotonically decreasing one and as it is pointed in [34], such a density can be well represented using a mixture of exponentials. In EM algorithm method, we achieve the highest likelihood, which is expected , as the EM algorithm tries to maximize the likelihood of the samples. However, the objective function value of the EM algorithm is higher than the least squares method results, and therefore, EM algorithm does not give us the tightest bound comparing to the other methods. As it can be seen from the results, the case of general phase-type using the EM algorithm also does not give us the tightest bound. To the contrary, such a general case of phase-type random variable does not improve the likelihood in compare to the limited case of hyper-Erlang. This decrease in the likelihood by deploying the general case of phase-type random variable can be justified by over-parameterization of the EM algorithm for the case of general phase-type random variable.

Phase-type bounds for the case of mixture of exponentials in the semi-infinite least squares method, which gives us the best result in this example and also hyper-Erlang in the semi-infinite least squares method, and hyper-Erlang for the case of EM algorithm method are shown in Fig. 4.1. These bounds are compared with the true survival function. As it can be seen from the figure and it can be verified by the results in Table 4.2, mixture of exponentials in semi-infinite least squares method gives us very close results to hyper-Erlang in EM algorithm method, which are both tight enough. Hyper-Erlang in semi-infinite least squares method, however, as it can be seen in the figure does not provide us a tight bound.

4.4 Conclusion

In this chapter we developed two methods to characterize a traffic process with phase-type bounds. The first method is based on minimizing the squared error of the bound and the empirical tail probability of the workload. In this method, the class of phase-type bounds was limited to the cases of hyper-Erlang, mixture of exponentials, and acyclic phase-type distributions. The second method for characterizing a traffic process with phase-type bounds was based on EM algorithm. A very efficient standalone EM algorithm for phase-type fitting based on hyper-Erlang class was developed and was used in deriving the phase-type bounds. This EM algorithm was shown to be very efficient in terms of complexity. The two developed method was used in a numerical study to characterize a heavy-tailed traffic. As it was shown the least squares method outperforms the EM method and can result in a tighter bound.

Chapter 5: Stochastic Traffic Regulator

In Chapter 3, we showed how having a phase-type characterization for an input traffic stream is related to the bounds on delay and workload size. In Chapter 4, we showed how to derive a phase-type characterization and therefore obtain a probabilistic bound on the delay and workload size of the data network fed by that traffic stream. Another question from a traffic admission point-of-view that should be answered is, given a desired probabilistic bound on the delay or workload size, how a given traffic can be regulated to comply with the desired performance measures. In this chapter we address this problem by developing a stochastic traffic regulator. Parts of the work in this chapter were published in [62], [60], and [57].

The remainder of this chapter is organized as follows. In Section 5.1, we provide an introduction and motivation about traffic regulation. In Section 5.2, we review basic concepts in deterministic and stochastic network calculus. In Section 5.3, we review key properties of the deterministic (σ, ρ) regulator and develop some new results for its analysis, which are applied in Section 5.4 to the design and implementation of the proposed stochastic (σ^*, ρ) regulator. Numerical results demonstrating the performance of the (σ^*, ρ) regulator are presented in Section 5.5. Concluding remarks are given in Section 5.6.

5.1 Introduction

Currently, the Internet does not provide end-to-end delay guarantees for traffic flows. Even if the path taken by a given traffic flow is fixed, e.g., via mechanisms such as software-defined networking or multi-protocol label switching, network congestion arising from other flows can result in highly variable delays. The variability and random nature of traffic flows in a packet-switched network make it very challenging to provide performance guarantees. End-to-end delay guarantees would improve the user experience provided by real-time applications such as video streaming and video conferencing, as well as enable emerging applications involving virtual reality and augmented reality. Bounds on network delay are of particular relevance to age of information (AoI), a performance metric whereby the freshness of data is assessed from the receiver's perspective [63].

The standard approach to providing network performance guarantees consists of two basic elements:

1. *Admission control:* A new flow is admitted only if performance guarantees can be maintained for all admitted flows with the available network resources.

2. *Traffic regulation:* Each admitted traffic flow is regulated to ensure that it does not consume more resources than what was negotiated by the admission control scheme.

Admission control relies on a means of characterizing the traffic to determine how to allocate network resources to the new flow. The random and bursty nature of traffic in packet-switched networks make them difficult to characterize. Even if a traffic flow were to be modeled as a random arrival process, e.g., a Markov modulated Poisson process, the problem of resource allocation to provide end-to-end performance guarantees is intractable. Moreover, traffic regulation to ensure that a flow conforms to the parameter of a traffic model is infeasible.

In his seminal work, Cruz [24, 25] proposed the (σ, ρ) characterization of traffic, which imposes a deterministic bound on the burstiness of a traffic flow. The bound ensures that the long-term average arrival rate a (σ, ρ)-bounded traffic source does not exceed the rate parameter ρ and its maximum burst size does not exceed the burst size parameter σ. By bounding traffic flows according to (σ, ρ) parameters, Cruz developed a network calculus which determined how these parameters propagate through network elements and derived associated end-to-end delay bounds.

An important feature of the (σ, ρ) characterization is that it can be enforced by a traffic regulator. The (σ, ρ) traffic bound can be defined operationally in terms of a (σ, ρ) traffic regulator. For bursty traffic sources, however, the (σ, ρ) bound can lead to worst-case end-to-end delay bounds that are very conservative, which in turn results in very low network resource utilization. To achieve higher utilization, the approach in [71] estimates a (σ, ρ)-based parameter for an arbitrary traffic source by minimizing a cost function derived from the concept of effective bandwidths [23, Chap. 9], subject to a constraint on the shaping delay incurred on the source. Given the traffic parameter, worst-case delay bounds for a traffic source could be computed using deterministic network calculus. Alternatively, resource allocation could be performed using effective bandwidths, but in this case true performance guarantees would not be provided.

The (σ, ρ) traffic bound of Cruz was the basis for further research into stochastic bounds on traffic burstiness and stochastic network calculus to provide probabilistic end-to-end delay guarantees as opposed to worst-case delay bounds. Stochastic network calculus remains an active topic of research [35]. To our knowledge, however, a traffic regulator to enforce a *stochastic* traffic bound based on stochastic network calculus has not been addressed previously, despite the fact that such a regulator is necessary to provide performance guarantees in real networks. Stochastic network calculus is based on the assumption that all traffic flows entering the network satisfy stochastic traffic bounds. In this chapter, we develop a traffic regulator to enforce the so-called *generalized Stochastically Bounded Burstiness* (gSBB) traffic burstiness bound in [51, 99]. The gSBB bound is closely related to the SBB and EBB bounds in [85] and [96], respectively. Formal definitions of these bounds are given in Section 5.2.2. These bounds are also related to the moment generating function (MGF) traffic bounds discussed in [22, 23]. We focus on the gSBB bound primarily because it is more amenable to traffic regulation than the other traffic bounds (see Section 5.4).

We refer to our proposed regulator as a stochastic (σ^*, ρ) regulator, since the burst size parameter σ^* varies over time and takes on values in a finite set $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$. We describe the design and basic properties of the stochastic (σ^*, ρ) regulator and develop two practical implementations. Our analysis establishes that the output traffic always conforms to the gSBB bound.

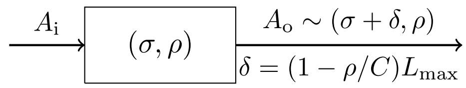


Figure 5.1: (σ, ρ) regulator with input/output links of capacity C.

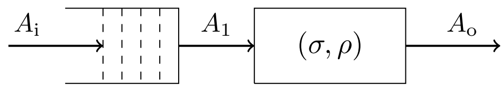


Figure 5.2: (σ, ρ) traffic shaper with front-end buffer.

5.2 Background on Network Calculus

Our proposed stochastic traffic regulator builds on the (σ, ρ) network calculus of Cruz [24,25] as well as stochastic network calculus. In this section, we review relevant aspects of both types of network calculus.

5.2.1 Deterministic (σ, ρ) Network Calculus

Let $A = \{A(t) : t \geq 0\}$ denote a traffic process or flow, where $A(t)$ represents the amount of traffic arriving in the interval $[0, t)$. We shall assume that t is a continuous-time parameter, although our results carry over to the discrete-time case as well.

Definition 5.2.1 ((σ, ρ) traffic bound). A traffic flow A is said to be (σ, ρ)-bounded, denoted as $A \sim (\sigma, \rho)$, if

$$A(t) - A(s) \leq \rho(t - s) + \sigma, \quad \forall s \in [0, t], \tag{5.1}$$

where $\sigma, \rho \geq 0$.

In conjunction with Definition 5.2.1, Cruz [24] introduced a traffic regulator to enforce conformance to the (σ, ρ) bound. For an idealized fluid model of input traffic, a (σ, ρ) traffic regulator ensures that the *output* traffic $A_\mathrm{o} \sim (\sigma, \rho)$ and traffic departs the regulator in the same order as it arrives to the regulator, i.e., the service discipline is first-come first-served (FCFS). When the traffic consists of discrete packets of maximum length $L_{\max}$ and the input/output links to the regulator have finite capacity C, the output traffic satisfies $A_{\mathrm{o}} \sim (\sigma + \delta, \rho)$, where (see Fig. 5.1)

$$\delta = (1 - \rho/C)L_{\max}. \tag{5.2}$$

Traffic regulation can be accomplished by packet dropping, packet tagging (as lower priority), or delaying of packets. In the first two cases, the traffic regulator is sometimes referred to as a *traffic policer* whereas in the third case it is referred to as a *traffic shaper*. The traffic regulators discussed in this chapter will be of the traffic shaper variety. A traffic shaper includes a front-end buffer, which stores packets that are delayed in the process of forcing the output traffic to conform to (σ, ρ) (see Fig. 5.2). A traffic policer is equivalent to a traffic shaper with no front-end buffer; i.e., packets that do not conform to (σ, ρ) are dropped or tagged immediately rather than placed in a buffer.

5.2.2 Stochastic Network Calculus

The (σ, ρ) bound in (5.1) tends to be very conservative for bursty traffic. This is illustrated in Fig. 5.12 (see Section 5.5) for a sample path of bursty traffic fed to a queue with service rate ρ. The queue size reaches the burstiness bound σ, but is far below σ most of the time. End-to-end delay bounds based on worst-case (σ, ρ) bounds will be overly conservative for bursty traffic flows. Admission control based on such bounds will lead to low network utilization. Moreover, deterministic network calculus cannot exploit the phenomenon of statistical multiplexing. These considerations motivated the development of *stochastic* traffic burstiness bounds and stochastic network calculus to allow the derivation of stochastic end-to-end delay bounds [35, 50].

An early proposal for a stochastic traffic burstiness bound was the Exponentially Bounded Burstiness (EBB) of Yaron and Sidi [97], which involves an exponential bounding function. A related traffic bound based on moment generating functions was proposed by Chang [22]. In this chapter, we focus on the *generalized* Stochastically Bounded Burstiness (gSBB) proposed in [51].

Definition 5.2.2 (gSBB). A traffic process A is gSBB with upper rate ρ and bounding function $f \in \mathcal{BF}$ if

$$\mathsf{P}\{W_\rho(t; A) \geq \sigma\} \leq f(\sigma), \quad \forall t \geq 0,\ \forall \sigma \geq 0, \tag{5.3}$$

where $\mathcal{BF}$ denotes the family of positive non-increasing real-valued functions and $W_\rho(t; A)$ is the virtual workload at time t of an infinite-buffer FCFS (First Come First Served) queue with constant service rate ρ with input traffic A. The virtual workload is given by

$$W_\rho(t; A) = \max_{0 \leq s \leq t} [A(t) - A(s) - \rho(t - s)]. \tag{5.4}$$

Intuitively, the virtual workload at an arbitrary time t is the amount of work (e.g., in units of bits) remaining in the system for the server to process. The gSBB concept is based on Stochastically Bounded Burstiness (SBB) [85].

Definition 5.2.3 (SBB). A traffic process A is SBB with upper rate ρ and bounding function $f \in \mathcal{F}$ if

$$\mathsf{P}\{A(t) - A(s) - \rho(t - s) \geq \sigma\} \leq f(\sigma), \quad \forall t \geq 0,\ \forall \sigma \geq 0, \tag{5.5}$$

where $\mathcal{F}$ is the family of functions f such that for every $n, \sigma \geq 0$, the n-fold integral $\left(\int_\sigma^\infty du\right)^n f(u)$ is bounded.

A traffic process is EBB if it is SBB with an exponentially decaying bounding function, i.e., $f(\sigma) = ae^{-\alpha\sigma}$, where $a, \alpha > 0$. For a given SBB bounding function $f \in \mathcal{F}$, a traffic process that is gSBB with respect to f is also SBB. Thus, the gSBB bound is more conservative than the SBB bound. Nevertheless, the gSBB concept has two important advantages over SBB, which we leverage in designing the (σ^*, ρ) regulator (see Section 5.4): 1) $\mathcal{BF} \supset \mathcal{F}$; 2) The gSBB bound is defined in terms of $W_\rho(t; A)$ rather than $A(t)$.

The phase-type traffic bound proposed in Chapter 3, is closely related to gSBB.

Definition 5.2.4. A traffic process A is characterized by a *phase-type traffic descriptor* $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$ if

$$\mathsf{P}\left\{W_\rho(t; A) \geq \sigma\right\} \leq a \boldsymbol{\pi} e^{\mathbf{Q}\sigma} \mathbf{1}, \tag{5.6}$$

for all $t \geq 0$ and all $\sigma \in (0, T]$. Here, $\mathbf{1}$ is a column vector of all ones, $a \geq 0$, $T > 0$, and $(\boldsymbol{\pi}, \mathbf{Q})$ denotes the parameter of a phase-type distribution [11, 54].

When $T = \infty$, the phase-type traffic bound is a particular case of gSBB. Using Algorithm 1 in Chapter 4, any given traffic flow can be characterized by a phase-type traffic descriptor.

Analogous to the deterministic network calculus, a stochastic network calculus can be developed based on a given stochastic traffic burstiness bound [23, 85, 97]. By applying results from the stochastic network calculus based on gSBB (see [51]), the admissibility of a given set of traffic flows with respect to a certain probabilistic end-to-end delay constraint can be determined. More general stochastic traffic bounds have since been developed in conjunction with notions of statistical arrival envelopes, service curves, and min-plus algebra in the context of stochastic network calculus [35]. However, end-to-end delay guarantees via stochastic network calculus can only be provided if the user traffic flows that are offered as input to the network conform to their negotiated traffic burstiness bounds. The stochastic traffic regulator developed in this chapter can be applied at the network edge to ensure that a user's traffic does not violate the gSBB bound provided to the admission control unit. Additional performance benefits can be obtained by applying stochastic traffic regulation in internal network elements to reshape traffic flows to their negotiated parameters, since the traffic parameter of a flow may be altered after it passes through a network element.

5.3 Analysis of Deterministic (σ, ρ) Regulator

The (σ^*, ρ) regulator may be viewed as an extension of the deterministic (σ, ρ) regulator, in which the burst size parameter σ^* takes on values from a finite set $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$ while adapting to the input traffic flows. In particular, the operation of a (σ, ρ) regulator can be viewed as a special case of a (σ^*, ρ) regulator. In Section 5.3.1, we review some relevant results on the virtual workload process $W_\rho(t)$ from the (σ, ρ) calculus. In Section 5.3.2, we develop some new results related to $W_\rho(t)$, which we shall use in the design and analysis of the stochastic (σ^*, ρ) regulator in Section 5.4.

5.3.1 Input/Output Workload Analysis

Suppose a traffic flow A is offered to an infinite-buffer FCFS system with constant service rate ρ. Clearly, the virtual workload $W_\rho(t; A)$ is a decreasing function of ρ. It can easily be shown that $A \sim (\sigma, \rho)$ if and only if

$$W_\rho(t; A) \leq \sigma, \quad \forall t \geq 0. \tag{5.7}$$

Equation (5.7) provides a useful alternative characterization of a (σ, ρ)-bounded traffic flow.

Now suppose that the input and output traffic links to and from a (σ, ρ) regulator have a finite capacity $C > \rho$. Consider an input traffic flow A_i to the regulator. Let s_j denote the arrival time of the jth packet, t_j its departure time, and L_j its length in bits. The jth packet begins arriving at time s_j and is received completely at the regulator at time $a_j = s_j + L_j/C$. We assume that a packet does not arrive when the previous one is being received. i.e., $a_j < s_{j+1}$.

The operation of the regulator can be described in terms of the workload $W_\rho(s_j; A_\mathrm{i})$. At time s_j, if $W_\rho(s_j; A_\mathrm{i}) > \sigma$, the regulator delays the packet such that at its departure time t_j, the condition $W_\rho(t_j; A_\mathrm{o}) \leq \sigma$ holds. Hence, the departure time of the jth packet is derived as [24]

$$t_j = [W_\rho(s_j; A_\mathrm{i}) - \sigma]^+/\rho + s_j, \tag{5.8}$$

where $[x]^+ := \max\{x, 0\}$. The packet completely departs the regulator at time

$$b_j = t_j + L_j/C. \tag{5.9}$$

At times other than departures, the workload may not necessarily be bounded by σ, but always satisfies [24]

$$W_\rho(t; A_\mathrm{o}) \leq \sigma + (1 - \rho/C)L_{\max}, \quad \forall t \geq 0. \tag{5.10}$$

Thus, $A_\mathrm{o} \sim (\sigma + \delta, \rho)$, where δ, given by (5.2), can be viewed as the maximum error margin in regulating packetized traffic when the input/output links have capacity C (see Fig. 5.1).

As shown Fig. 5.3, when a packet is being received by the regulator, e.g., during $[s_j, a_j]$, the workload $W_\rho(t; A_\mathrm{i})$ increases linearly with slope $C - \rho$. Conversely, during the time between the complete arrival of a packet and the initial arrival of the next packet to the system, e.g., during $[a_j, s_{j+1}]$, the workload $W_\rho(t; A_\mathrm{i})$ decreases linearly with slope $-\rho$. Similarly, when a packet departs the regulator, e.g., during $[t_j, b_j]$, the workload $W_\rho(t; A_\mathrm{o})$ increases linearly with slope $C - \rho$. When packets are not departing the system, e.g., during $[b_j, t_{j+1}]$, $W_\rho(t; A_\mathrm{o})$ decreases linearly with slope $-\rho$. Assume that the buffer of the regulator is empty at $t = s_1$. Let

$$\delta_j = (1 - \rho/C)L_j \tag{5.11}$$

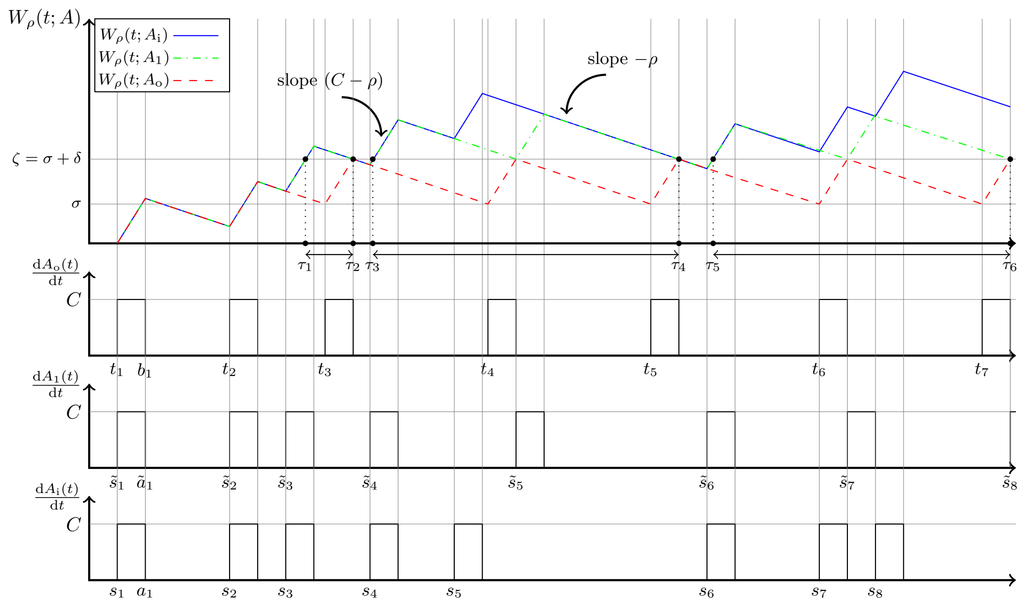


Figure 5.3: Example of the operation of a (σ, ρ) traffic regulator.

denote the error margin due to regulating the jth packet. We present the governing equations for a (σ, ρ) regulator in terms of the workloads $W_\rho(t; A_{\mathrm{i}})$ and $W_\rho(t; A_{\mathrm{o}})$ as follows:

$$W_\rho(t; A_{\mathrm{i}}) = [W_\rho(a_{j-1}; A_{\mathrm{i}}) - \rho(t - a_{j-1})]^+, \quad t \in [a_{j-1}, s_j]; \tag{5.12}$$

$$W_\rho(t; A_{\mathrm{i}}) = W_\rho(s_j; A_{\mathrm{i}}) + (t - s_j)(C - \rho), \quad t \in [s_j, a_j]; \tag{5.13}$$

$$W_\rho(t_j; A_{\mathrm{o}}) = \begin{cases} \sigma, & \text{if} \;\; W_\rho(s_j; A_{\mathrm{i}}) > \sigma, \\ W_\rho(s_j; A_{\mathrm{i}}), & \text{if} \;\; W_\rho(s_j; A_{\mathrm{i}}) \le \sigma; \end{cases} \tag{5.14}$$

$$W_\rho(t; A_{\mathrm{o}}) = W_\rho(t_j; A_{\mathrm{o}}) + (t - t_j)(C - \rho), \quad t \in [t_j, b_j]; \tag{5.15}$$

$$W_\rho(t; A_{\mathrm{o}}) = [W_\rho(b_{j-1}; A_{\mathrm{o}}) - \rho(t - b_{j-1})]^+, \quad t \in [b_{j-1}, t_j]; \tag{5.16}$$

for $j = 1, 2, \ldots$. Equations (5.12)–(5.16) provide a complete characterization of the virtual workloads of the traffic flows A_{i} and A_{o} and can be used to construct the corresponding workload curves shown in Fig. 5.3.

5.3.2 Internal Traffic Workload Analysis

To analyze the stochastic (σ^*, ρ) regulator developed in Section 5.4, it will be convenient to introduce the internal traffic flow A_1 shown in Fig. 5.2 for the (σ, ρ) regulator and in Fig. 5.4 for the (σ^*, ρ) regulator. We shall develop some new results for the (σ, ρ) regulator involving the internal flow A_1, which will be useful in the design of the (σ^*, ρ) regulator. Figure 5.2 can be viewed as a more detailed depiction of the (σ, ρ) regulator shown as a single box in Fig. 5.1. The diagrams in Figs. 5.2 and 5.4 represent single-server, infinite buffer queueing systems. The box represents the server, which imposes a variable service delay on an arriving packet. The service delay will be zero if no shaping is needed. Only one packet can reside in the server at any given time. A new packet j can arrive to the server at the instant packet $j-1$ leaves the server. Packets that arrive when the server is occupied are stored in the front-end buffer in FCFS order. The traffic flow A_1 consists of the sequence of packets arriving to the server.

Let $\tilde{s}_j$ denote the arrival time of the jth packet at the buffer and let $\tilde{a}_j$ denote the complete arrival time to the buffer, i.e., $\tilde{a}_j := \tilde{s}_j + L_j/C$. The server incurs a delay on the jth packet such that it begins departing the buffer at time t_j and completely leaves the regulator at time b_j. Since the front-end buffer delays each packet until the complete departure time of the previous packet from the regulator, we have

$$\tilde{s}_j = \max\{s_j, b_{j-1}\}. \tag{5.17}$$

Therefore, the operation of (σ, ρ) regulator can also be described in terms of the workload $W_\rho(\tilde{s}_j; A_1)$. In other words, we have the following result, which is proved in Appendix.

Proposition 5.3.1. The departure time t_j for the jth packet in the (σ, ρ) regulator is given by (cf. (5.8)):

$$t_j = [W_\rho(\tilde{s}_j; A_1) - \sigma]^+/\rho + \tilde{s}_j. \tag{5.18}$$

An example sample path of the workloads of traffic flows A_i, A_1, and A_o for a deterministic (σ, ρ) regulator is shown in the top graph of Fig. 5.3. If the input traffic flow A_i conforms to the (σ, ρ) traffic burstiness parameter at arrival times, then the workloads of A_i, A_1, and A_o will all coincide, which occurs in the interval $[s_1, s_3]$ in the figure. Within this interval, for packets $j = 1$ and 2, we have $s_j = \tilde{s}_j = t_j$ and $a_j = \tilde{a}_j = b_j$, since both packets arrive when the workload $W_\rho(t; A_\mathrm{i}) \leq \sigma$. At time $s_3 = \tilde{s}_3$, the workloads of A_1 and A_o diverge because packet 3 arrives when $W_\rho(t; A_\mathrm{i}) > \sigma$. Thus, the packet is delayed in the server and $t_3 > \tilde{s}_3$. However, the workloads of A_1 and A_o once again coincide at time b_3, i.e., the complete departure time of packet 3 from the regulator.

The workload curves of A_1 and A_o form a parallelogram in the interval $[\tilde{s}_3, b_3]$. The other points of this parallelogram occur at $\tilde{a}_3$, i.e., when packet 3 completely arrives to the server and at t_3, i.e., when packet 3 starts to depart the server. Then the two workload curves coincide in the interval $[b_3, \tilde{s}_4]$. In general, the workloads of A_1 and A_o form a (possibly degenerate) parallelogram during the interval $[\tilde{s}_j, b_j]$ and coincide during the interval $[b_j, \tilde{s}_{j+1}]$, for $j = 1, 2, \ldots$.

In Fig. 5.3, we see that the workload curves of A_i and A_1 coincide until time s_5, which is the start time of the arrival of packet 5 to the regulator. At this time, packet 4 is at the server, so packet 5 waits until time $\tilde{s}_5 > s_5$ to go into service. At time $\tilde{a}_5$, when packet 5 has arrived completely to the server, the two curves coincide once again. In the interval $[s_5, \tilde{a}_5]$, the two curves form a parallelogram. This is not true in general, but in the interval $[s_j, \tilde{a}_j]$ a (possibly degenerate) parallelogram can be formed in which the sides consists of $W_\rho(t; A_\mathrm{i})$ for $t \in [s_j, a_j]$, $W_\rho(t; A_1)$ for $t \in [\tilde{s}_j, \tilde{a}_j]$, $W_\rho(t; A_\mathrm{i})$ for $t \in [a_j, \tilde{a}_j]$, and $W_\rho(t; A_1)$ for $t \in [s_j, \tilde{s}_j]$ for $j = 1, 2, \ldots$. Thus, the workload curves of A_i and A_1 are separated by a *sequence* of possibly degenerate parallelograms. Each such parallelogram corresponds to a packet delayed in the buffer of the regulator. A similar type of relationship holds between the workload curves of A_i and A_o. The workload curves of A_1 and A_o are separated by at most *one* parallelogram because the server can hold at most one packet.

Based on the above analysis and Proposition 5.3.1, the operation of the (σ, ρ) regulator can be characterized in terms of the internal traffic flow A_1 and the output traffic A_o. Analogous to equations (5.12)–(5.16) the following equations involving A_1 can be derived:

$$W_\rho(t; A_o) = W_\rho(t; A_1) = [W_\rho(b_{j-1}; A_o) - \rho(t - b_{j-1})]^+, \quad t \in [b_{j-1}, \tilde{s}_j]; \tag{5.19}$$

$$W_\rho(t; A_1) = W_\rho(\tilde{s}_j; A_1) + (t - \tilde{s}_j)(C - \rho), \quad t \in [\tilde{s}_j, \tilde{a}_j]; \tag{5.20}$$

$$W_\rho(t; A_1) = W_\rho(\tilde{a}_j; A_1) - \rho(t - \tilde{a}_j), \quad t \in [\tilde{a}_j, b_j]; \tag{5.21}$$

$$W_\rho(t_j; A_o) = \begin{cases} \sigma, & \text{if } \; W_\rho(\tilde{s}_j; A_1) > \sigma, \\ W_\rho(\tilde{s}_j; A_1), & \text{if } \; W_\rho(\tilde{s}_j; A_1) \le \sigma; \end{cases} \tag{5.22}$$

$$W_\rho(t; A_o) = W_\rho(t_j; A_o) + (t - t_j)(C - \rho), \quad t \in [t_j, b_j]; \tag{5.23}$$

$$W_\rho(t; A_o) = W_\rho(\tilde{s}_j; A_o) - \rho(t - \tilde{s}_j), \quad t \in [\tilde{s}_j, t_j]; \tag{5.24}$$

for $j = 1, 2, \ldots$. Equation (5.19) follows from the following equality

$$W_\rho(b_{j-1}; A_o) = W_\rho(b_{j-1}; A_1), \tag{5.25}$$

which can be verified using (5.20)-(5.24) and (5.9). Intuitively, (5.25) holds because at most one packet is in the server of the regulator at any given time.

5.4 Stochastic (σ^*, ρ) Regulator

To our knowledge, the problem of regulating a traffic flow to force conformance to a stochastic traffic bound has not been addressed in the literature. This motivates the development of a *stochastic* traffic regulator, which enforces a probabilistic bound on a traffic source as follows:

$$\mathsf{P}\left\{W_\rho(t; A_o) \ge \gamma\right\} \le f(\gamma), \qquad \forall t \ge 0, \; \forall \gamma \in [0, T], \tag{5.26}$$

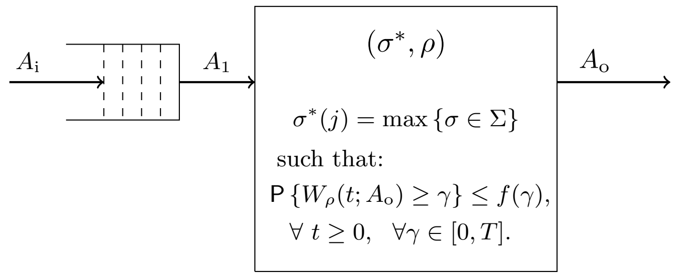


Figure 5.4: Idealized stochastic (σ^*, ρ) traffic regulator.

where f is a non-increasing positive bounding function and T is a limit on the tail distribution of the workload (see Chapter 3). As $T \to \infty$, (5.26) becomes equivalent to the gSBB bound in (5.3).

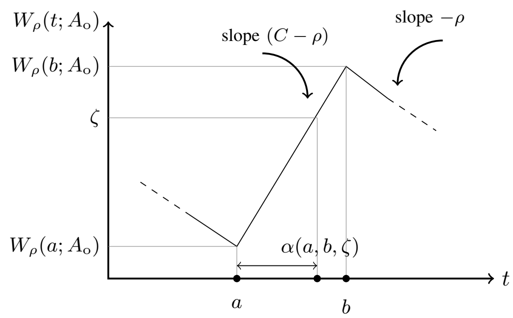
5.4.1 Operational Principles

We shall show that enforcement of (5.26) can be achieved under steady-state conditions using a regulator with a constant rate parameter ρ and a variable burstiness parameter σ^* which is chosen from a finite set Σ for each arriving packet. We refer to such a regulator as a stochastic (σ^*, ρ) regulator. A schematic of an idealized (σ^*, ρ) regulator is shown in Fig. 5.4. The input and output links of the regulator are assumed to have capacity C. A buffer at the front-end of the regulator delays incoming packets until all previous packets have departed, thus ensuring a FCFS service discipline. Let A_{i} and A_{o} denote, respectively, the input traffic to and output traffic from the regulator. We denote the internal traffic departing from the front-end buffer by A_1. Let s_j and $\tilde{s}_j$ denote, respectively, the arrival and departure times of the jth packet at the buffer.

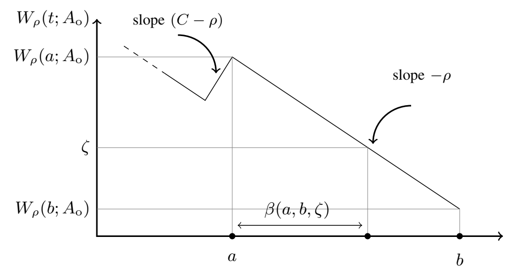
For each packet j, the (σ^*, ρ) regulator chooses a burstiness parameter $\sigma^*(j)$ such that a delay d_j is incurred, where (cf. (5.8))

$$d_j = t_j - s_j = [W_\rho(s_j; A_{\mathrm{i}}) - \sigma^*(j)]^+/\rho, \tag{5.27}$$

and t_j denotes the time at which the packet starts departing the traffic regulator. The



(a) $\alpha(a, b, \zeta)$



(b) $\beta(a, b, \zeta)$

Figure 5.5: Calculating the increment in the overshoot duration.

packet completely leaves the regulator at time b_j. The front-end buffer acts as in the deterministic (σ, ρ) regulator (see Section 5.3); therefore $\tilde{s}_j$ can be derived from (5.17). As in a deterministic (σ, ρ) traffic regulator, the rate parameter ρ should be greater than the long-term average input traffic rate, i.e.,

$$\rho > \lim_{t \to \infty} \frac{A(t) - A(s)}{t - s}, \quad \forall s \geq 0, \tag{5.28}$$

to avoid incurring unbounded packet delay.

5.4.2 Overshoot Probability and Overshoot Ratio

To design a practical (σ^*, ρ) regulator, the *overshoot probability* $\mathsf{P}\{W_\rho(t; A_o) \geq \gamma\}$ in (5.26) can be approximated by a time-averaged *overshoot ratio*.

Definition 5.4.1. Given a threshold value $\zeta > 0$ and a traffic flow A, an *overshoot interval* with respect to A and ζ is a maximal interval of time η such that $W_\rho(\tau; A) \geq \zeta$ for all $\tau \in \eta$. Let $|\eta|$ denote the length of interval η. Let $\mathcal{O}(t)$ denote the set of overshoot intervals contained in $[0, t]$. Then the *overshoot duration* up to time t is defined as

$$O_\zeta(t; A) = \sum_{\eta \in \mathcal{O}(t)} |\eta|. \tag{5.29}$$

In Fig. 5.3, the overshoot set with respect to threshold value ζ until the end of time domain depicted in the figure consists of three intervals $[\tau_1, \tau_2]$, $[\tau_3, \tau_4]$ and $[\tau_5, \tau_6]$. Given a time interval $[a, b]$, let $w_1 = W_\rho(a; A_\mathrm{o})$ and $w_2 = W_\rho(b; A_\mathrm{o})$. We define the increment in overshoot duration when the workload of the output process is *increasing* due to a packet departure from the regulator as follows:

$$\alpha(a, b, \zeta) = \begin{cases} b - a, & \zeta \leq w_1, \\ (w_2 - \zeta)/(C - \rho), & w_1 \leq \zeta \leq w_2, \\ 0, & w_2 < \zeta. \end{cases} \tag{5.30}$$

We define the increment in overshoot duration when the workload is *decreasing* due to the packet inter-departure time as follows:

$$\beta(a, b, \zeta) = \begin{cases} b - a, & \zeta \leq w_2, \\ (W_1 - \zeta)/\rho, & w_2 \leq \zeta \leq w_1, \\ 0, & w_1 < \zeta. \end{cases} \tag{5.31}$$

Figure 5.5 illustrates $\alpha(a, b, \zeta)$ and $\beta(a, b, \zeta)$. The following proposition follows immediately from the definitions and shows how to compute $O_\zeta(t; A_\mathrm{o})$ at time $t = b_j$ for packet j.

Proposition 5.4.1.

$$O_\zeta(b_1; A_\mathrm{o}) = \alpha(t_1, b_1, \zeta),$$

$$O_\zeta(b_j; A_\mathrm{o}) = O_\zeta(b_{j-1}; A_\mathrm{o}) + \beta(b_{j-1}, t_j, \zeta) + \alpha(t_j, b_j, \zeta),$$

for $j = 2, 3, \ldots$.

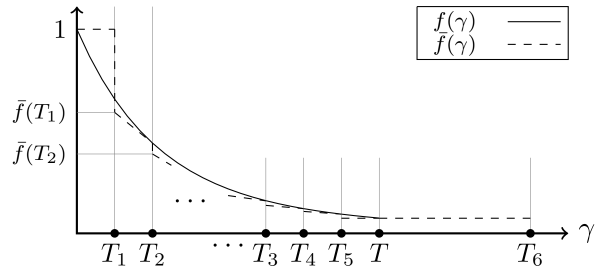


Figure 5.6: Piecewise-linear approximating function $\bar{f}$, $M = 6$.

We define the *overshoot ratio* of the regulator at time t with respect to a threshold ζ by

$$o_\zeta(t) = O_\zeta(t; A_\mathrm{o})/t. \tag{5.32}$$

For sufficiently large t, the virtual workload $W_\rho(t)$ can be modeled as a stationary and ergodic process [55]. This is a reasonable assumption when ρ satisfies (5.28), since the queueing system is stable in this case. Under this assumption, the overshoot ratio asymptotically approaches the overshoot probability, i.e.,

$$o_\zeta(t) \sim \mathsf{P}\left\{W_\rho(t; A_\mathrm{o}) > \zeta\right\} \text{ as } t \to \infty. \tag{5.33}$$

Using the overshoot ratio as a proxy for the overshoot probability in (5.26), we design a (σ^*, ρ) regulator that selects the burstiness parameters $\sigma^*(j)$, $j = 1, 2, \ldots$, from a finite set Σ such that

$$o_\gamma(t) \le f(\gamma), \quad \forall\, t \in [b_{j-1}, b_j],\ \forall \gamma \in [0, T], \tag{5.34}$$

while minimizing the incurred packet delay. Note that the bound (5.34) is satisfied at any time t, whereas the approximation for the overshoot probability in (5.33) holds asymptotically.

5.4.3 Piecewise-Linear Bounding Function

Next, we address the issues of selecting the set Σ of burstiness parameter values and verification of the condition (5.34). We replace the bounding function f by a piecewise-linear function $\bar{f}$ defined in terms of a set of values $T_1 < T_2 < \ldots < T_M$ and the value δ given by (5.2) satisfying the following constraints:

$$T - T_{M-1} \geq \delta;\ T_M \gg T,\ T_1 \geq \delta,\ T_{i+1} - T_i \geq \delta, \tag{5.35}$$

for $i = 1, 2, \ldots, M-1$. For given T and δ, the maximum possible value of M is given by

$$M_{\max} = \lfloor T/\delta \rfloor - 1. \tag{5.36}$$

The values $\{T_1, \ldots, T_M\}$ determine the set of burstiness parameter values

$$\Sigma = \{\sigma_i := T_i - \delta : i = 1, \ldots, M\}. \tag{5.37}$$

Note that $\sigma_1 < \sigma_2 < \ldots < \sigma_M$.

Without loss of generality, we assume $f(0) = 1$. The function $\bar{f}$ is designed[1] such that $f \leq \bar{f}$ in $[0, T_1)$, $\bar{f} \leq f$ in $[T_1, T]$ and $||f - \bar{f}||$ is small in $[T_1, T]$, where $||\cdot||$ denotes a norm on the space of bounding functions $\mathcal{BF}$, e.g., the L^2-norm $||\cdot||_2$. Since $\bar{f}$ is chosen from the class of piecewise-linear functions with a finite number M of linear pieces, it cannot be chosen arbitrarily close to f, although $||f - \bar{f}||$ decreases as M is increased. In particular, we set $\bar{f}(\gamma) = f(0) = 1$ for $\gamma \in [0, T_1)$. Since $\bar{f} \geq f$ in this interval, traffic regulation with respect to $\bar{f}$ may result in violation of (5.26). However, the violation probability is upper bounded by T_1/T, which can be made arbitrarily small by suitable choices of T_1 and/or T. We also set $\bar{f}(\gamma) = \bar{f}(T)$ for $T_M > \gamma \geq T_{M-1}$, and we choose a large value for T_M such that the burst size of the output traffic is not limited by the stochastic (σ^*, ρ) regulator.

[1] For technical reasons, a slightly different definition of $\bar{f}$ for values of $M < M_{\max}$ is used in the proofs of Theorems 1–3 given in Appendix D.

In the interval $[T_i, T_{i+1})$ let

$$g_i(\gamma) = f(T_{i+1}) + \omega_i(\gamma - T_{i+1}) \tag{5.38}$$

represent the line connecting the points $(T_i, f(T_i))$ and $(T_{i+1}, f(T_{i+1}))$ with slope

$$\omega_i = \frac{f(T_{i+1}) - f(T_i)}{T_{i+1} - T_i} \tag{5.39}$$

for $i = 1, \ldots, M-2$. If $f(\gamma) \geq g_i(\gamma)$ for all $\gamma \in [T_i, T_{i+1})$ we set $\bar{f} = g_i$ in this interval. Otherwise, we set $\bar{f} = h_i$ on $[T_i, T_{i+1})$, where

$$h_i(\gamma) = f(T_{i+1}) + f'(T_{i+1})(\gamma - T_{i+1}). \tag{5.40}$$

This ensures that $\bar{f} \leq f$ on $[T_1, T_{M-1})$. We then set $\bar{f}(\gamma) = f(T)$ for $\gamma \in [T_{M-1}, T_M]$ and $\bar{f}(\gamma) = 0$ for $\gamma > T_M$. To summarize, we define

$$\bar{f}(\gamma) = \begin{cases} 1, & \gamma \in [0, T_1), \\ f(T_{i+1}) + m_i(\gamma - T_{i+1}), & \gamma \in [T_i, T_{i+1}), \\ f(T), & \gamma \in [T_{M-1}, T_M], \\ 0, & \gamma > T_M, \end{cases} \tag{5.41}$$

for $i = 1, \ldots, M-2$ and the slopes m_i are given by

$$m_i = \begin{cases} \omega_i, & \text{if } f \geq g_i \text{ on } [T_i, T_{i+1}), \\ f'(T_{i+1}), & \text{otherwise}, \end{cases} \tag{5.42}$$

for $i = 1, \ldots, M-2$.

5.4.4 Canonical (σ^*, ρ) Regulator

Based on the definition of $\bar{f}$ in (5.41), we modify the constraint in (5.34) to hold only for $\gamma \in [T_1, T]$, i.e.,

$$o_\gamma(t) \leq f(\gamma), \quad \forall\ t \in [b_{j-1}, b_j],\ \forall \gamma \in [T_1, T]. \tag{5.43}$$

Towards a practical implementation, we further replace the bounding function f by $\bar{f}$ to obtain the following burstiness constraint:

$$o_\gamma(t) \leq \bar{f}(\gamma), \quad \forall\ t \in [b_{j-1}, b_j],\ \forall \gamma \in [T_1, T]. \tag{5.44}$$

To incur minimal packet delay, $\sigma^*(j)$ should be chosen as the largest value in Σ such that the constraint (5.44) is maintained. We then define a *canonical* (σ^*, ρ) regulator as follows:

$$\begin{aligned} \mathcal{A}_j = \{ & \sigma \in \Sigma : o_\gamma(t) \leq \bar{f}(\gamma), \forall t \in [b_{j-1}, b_j(\sigma)], \\ & \forall \gamma \in [T_1, T]\} \end{aligned}$$

$$\sigma^*(j) = \begin{cases} \sigma_{\max \mathcal{A}_j}, & \text{if } \mathcal{A}_j \neq \emptyset, \\ \sigma_1, & \text{otherwise.} \end{cases} \tag{5.45}$$

Equations (5.18)-(5.24) are the governing equations for a stochastic (σ^*, ρ) in which σ is replaced by $\sigma^*(j)$ according to (5.45). The canonical regulator cannot be implemented directly, since the condition for $\mathcal{A}_j$ in (5.45) is impractical to verify for all values of $t \in [b_{j-1}, b_j]$ and $\gamma \in [T_1, T]$. Next, we develop practical implementations of the canonical (σ^*, ρ) regulator.

5.4.5 Basic Implementation

We assume that T_M is chosen sufficiently large such that for every packet j the set

$$\mathcal{B}_j = \{1 \leq \ell \leq M : \sigma_\ell \geq W_\rho(\tilde{s}_j; A_1)\}, \tag{5.46}$$

is non-empty. Let

$$\mathcal{I}_j = \left\{2 \leq \ell \leq \min \mathcal{B}_j : o_{T_{\ell-1}}(b_j(\sigma_\ell)) \leq \bar{f}(T_\ell)\right\} \tag{5.47}$$

where $t_j(\sigma_\ell)$ and $b_j(\sigma_\ell)$ are given by (5.18) and (5.9), respectively. Let

$$\sigma^*(j) = \begin{cases} \sigma_{\max \mathcal{I}_j}, & \text{if } \mathcal{I}_j \neq \emptyset, \\ \sigma_1, & \text{otherwise.} \end{cases} \tag{5.48}$$

Equations (5.46)–(5.48) are used to develop approximate implementations of the canonical (σ^*, ρ) regulator given by (5.45). For a given value of $\sigma_\ell \in \Sigma$, the condition in (5.47) is checked only at $t = b_j(\sigma_\ell)$ and $\gamma = T_{\ell-1}$. Therefore, as shown in Section 5.5, the constraint (5.43) may be violated for some values of t. However, these violations will not occur when t is sufficiently large.

Theorem 5.4.1. The (σ^*, ρ) regulator defined by (5.46)–(5.48) produces output traffic that satisfies (5.43) for sufficiently large t.

A proof of Theorem 5.4.1 is given in Appendix D.4. Thus, the proposed regulator ensures that the overshoot ratio $o_\gamma(t)$ of the output traffic is bounded by the function $f(\gamma)$ for sufficiently large t. Since, by (5.33), $o_\gamma(t) \to \mathsf{P}\{W_\rho(t) \geq \gamma\}$, we have that $\mathsf{P}\{W_\rho(t) \geq \gamma\} \leq f(\gamma)$ for sufficiently large t i.e., the gSBB bound (5.3) is satisfied by the output traffic. The proof of Theorem 5.4.1 can be found in Appendix D. A pseudo-code implementation of the stochastic (σ^*, ρ) regulator is given in Algorithm 6. The input traffic A_i is represented as a sequence $\{(s_1, L_1), \ldots, (s_N, L_N)\}$, where the s_i's are the arrival times of the packets and the

L_i's are the packet lengths. The (σ^*, ρ) regulator consists of the rate ρ, the bounding function f, the range T over which the bound is applied, the set Σ, and the values $\{T_1, \ldots, T_M\}$, which determine the piecewise-linear bounding function $\bar{f}$. The input and output links for the regulator are assumed to be of capacity $C > \rho$. The output traffic A_o is represented by the sequence $\{(t_1, L_1), \ldots, (t_N, L_N)\}$, where the t_i's are packet departure times. The **for** loop starting in line 11 finds the largest $\ell \in \{2, \ldots, k = \min \mathcal{B}_j\}$ such that the inequality in (5.47) is satisfied with $\sigma = \sigma_\ell$. If such σ_ℓ exists, then $\sigma^*(j) = \sigma_\ell$; otherwise, $\sigma^*(j) = \sigma_1$, in accordance with (5.48).

Computation of the departure time, t_j, of the jth packet requires updates to $o_{T_i}(b_j)$ for $i = 1, \ldots, M-1$. Once t_j is determined, the values of $o_{T_i}(b_j)$, for $i = 1, \ldots, M-1$, need to be updated. Thus, the overall computational complexity is $O(M)$ per packet. Since the updates to $o_{T_i}(b_j)$ are independent of each other, they could be executed in parallel using a hardware accelerator such as a graphics processing unit (GPU). In particular, a parallel implementation of the **for** loop at line 11, can effectively reduce the computational complexity per packet to constant time, i.e., $O(1)$.

5.4.6 Alternative Implementation

The requirement of sufficient large t in Theorem 5.4.1 can be avoided by modifying the definition of $\mathcal{I}_j$ in (5.47) to include additional checks. Let $\mathcal{B}_j$ be as defined in (5.46). We re-define $\mathcal{I}_j$ as follows:

$$\mathcal{I}_j = \big\{2 \le \ell \le \min \mathcal{B}_j : o_{T_i}(b_j(\sigma_\ell)) \le \bar{f}(T_i) - \epsilon_{i,j}(\sigma_\ell), \quad \forall i = 1, \ldots, \ell - 1\big\}, \tag{5.49}$$

where

$$\epsilon_{i,j}(\sigma_\ell) := \begin{cases} \frac{W_\rho(b_j(\sigma_\ell); A_\text{o}) - T_i}{\rho b_j(\sigma_\ell)} (1 - \bar{f}(T_i)), & i = 1, \ldots, \ell - 2, \\ \bar{f}(T_{\ell-1}) - \bar{f}(T_\ell), & i = \ell - 1. \end{cases} \tag{5.50}$$

Algorithm 6 (σ^*, ρ) stochastic regulator

Input: $A_\text{i} \leftarrow \{(s_1, L_1), \ldots, (s_N, L_N)\}$; ▷ Input traffic
Input: ρ; $f(\cdot)$; T; M; L_max; C ▷ Regulator parameters
Output: $A_\text{o} \leftarrow \{t_1, t_2, \ldots, t_N\}$ ▷ Output traffic

$\delta \leftarrow (1 - \rho/C)L_\text{max}$
Compute T_i, σ_i for $i = 1, 2, \ldots, M$ ▷ (5.35), (5.37)
Compute $\bar{f}(\cdot)$ ▷ (5.41)
$t_1 \leftarrow \tilde{s}_1 \leftarrow s_1$; $b_1 \leftarrow t_1 + L_1/C$
$W_\rho(\tilde{s}_1; A_1) \leftarrow W_\rho(t_1; A_\text{o}) \leftarrow 0$
Compute $W_\rho(b_1; A_\text{o})$ ▷ (5.23)
Compute $o_{T_i}(b_1)$; $i = 1, 2, \ldots, M-1$ ▷ Prop. 5.4.1
for $j = 1, \ldots, N$ **do** ▷ Packet j arrives at time s_j
    Compute $\tilde{s}_j$, $W_\rho(\tilde{s}_j; A_1)$, $\mathcal{B}_j$ ▷ (5.17), (5.19), (5.46)
    found $\leftarrow$ **false**; $k \leftarrow \min \mathcal{B}_j$
    for $\ell = k, \ldots, 2$ **do** ▷ $k \geq 2$
        $\sigma \leftarrow \sigma_\ell$; Compute $t_j(\sigma)$, $b_j(\sigma)$ ▷ (5.18), (5.9)
        Compute $W_\rho(t_j; A_\text{o})$, $W_\rho(b_j; A_\text{o})$ ▷ (5.22), (5.23)
        Compute $o_{T_{\ell-1}}(b_j)$ ▷ Prop. 5.4.1
        if $o_{T_{\ell-1}}(b_j) \leq \bar{f}(T_\ell)$ **then** ▷ (5.47)
            found $\leftarrow$ **true**; **break**
        end if
    end for
    if not found **then**
        $\sigma \leftarrow \sigma_1$; Compute $t_j(\sigma)$, $b_j(\sigma)$ ▷ (5.18), (5.9)
    end if
    Compute $o_{T_i}(b_j)$; $i = 1, 2, \ldots, M-1$ ▷ Prop. 5.4.1
end for

The modified definition of $\mathcal{I}_j$ in (5.49) involves additional checks for the jth packet, which may result in a smaller value of $\sigma^*(j)$ and hence higher delay incurred on the packet. Interestingly, our numerical simulations show that this results in slightly smaller *average* delay incurred on the input traffic. This can be explained as follows. By incurring more delay on *some* input packets at an earlier stage, the output traffic may be better shaped to the desired bound; therefore, on average, less delay will need to be incurred on future packets.

The overshoot ratio $o_{T_i}(t)$ at $t = b_j(\sigma_\ell)$ is checked against $\bar{f}(T_i) - \epsilon_{i,j}(\sigma_\ell)$ rather than $\bar{f}(T_i)$, for $i = 1, \ldots, \ell - 2$. The reasoning behind this stricter condition is illustrated in

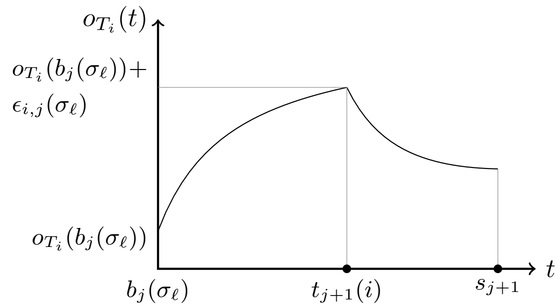


Figure 5.7: Overshoot ratio $o_{T_i}(t)$ for $t > b_j$, when $W_\rho(s_{j+1}; A_\mathrm{o}) = 0$.

Fig. 5.7. In choosing $\sigma^*(j) = \sigma_\ell$, the overshoot ratios $o_{T_i}(t)$, for $i = 1, \ldots, \ell - 2$, will be increasing functions of t, as shown in Fig. 5.7, up to time $t = t_{j+1}(i)$, which is defined as the time at which

$$W_\rho(t; A_\mathrm{o}) = T_i \quad \text{for} \quad i = 1, 2, \ldots, T_{\ell-2}, \tag{5.51}$$

and the $(j+1)$th packet arrives sufficiently late that $W_\rho(s_{j+1}; A_\mathrm{o}) = 0$. Enforcing the condition in (5.49) with the lower values $\bar{f}(T_i) - \epsilon_{i,j}(\sigma_\ell)$ ensures that the overshoot ratio stays less than $\bar{f}(T_i)$ for all $t \geq b_j$. In this implementation, for a given value of $\sigma_\ell \in \Sigma$, the condition (5.43) is checked only at $t = b_j(\sigma_\ell)$ and for $\gamma \in \{T_1, \ldots, T_{\ell-1}\}$. These extra checks compared to Algorithm 6, as stated in the following theorem and shown in Section 5.5, guarantee that there will be no violation of the constraint (5.43).

Theorem 5.4.2. The (σ^*, ρ) regulator defined by (5.46), (5.49), and (5.48) produces output traffic that satisfies (5.43) for all $t \geq 0$.

Algorithm 7 Replacement for lines 14–17 of Algorithm 6

14: Compute $o_{T_i}(b_j(\sigma_\ell))$; $i = 1, \ldots, \ell - 1$ $\triangleright$ Prop. 5.4.1
15: Compute $\epsilon_{i,j}(\sigma_\ell)$; $i = 1, \ldots, \ell - 1$ $\triangleright$ (5.50)
16: **if** $o_{T_i}(b_j) \leq \bar{f}(T_i) - \epsilon_{i,j}(\sigma_\ell)\ \forall i \in \{1, \ldots, \ell-1\}$ **then**
17: found $\leftarrow$ **true**; **break**
18: **end if**

Proof of Theorem 5.4.2 is given Appendix D.2. By modifying Algorithm 6 in accordance with Theorem 5.4.2, we obtain an alternative implementation that satisfies (5.43) for all $t \geq 0$ at the expense of some additional computation. The modified implementation is obtained by replacing lines 15–18 in Algorithm 6 with the pseudo-code shown in Algorithm 7. In lines 15 and 16, $\ell - 1$ values of $o_{T_i}(b_j(\sigma_\ell))$ and $\epsilon_{i,j}(\sigma_\ell)$ need to be computed. Therefore, the complexity of the **for** loop at line 12 in Algorithm 6 is $O(M^2)$ and the overall complexity of the modified algorithm is $O(M^2)$ per packet.

With further algorithmic modifications, the complexity of Algorithm 7 can be reduced to $O(M)$, i.e., the same time complexity as Algorithm 6. Let $\mathcal{B}_j$ be as in (5.46). Let $k = \min \mathcal{B}_j$ and

$$\mathcal{J}_j = \left\{1 \leq \ell \leq k-1 : o_{T_\ell}(b_j(\sigma_k)) \leq \bar{f}(T_\ell) - \epsilon_{\ell,j}(\sigma_k)\right\}, \tag{5.52}$$

where $\epsilon_{i,j}(\sigma_k)$ is defined in (5.50). If $1 \in \mathcal{J}_j$ let

$$m = \max\left\{\ell \in \mathcal{J}_j : i \in \mathcal{J}_j,\ \forall 1 \leq i \leq \ell\right\}, \tag{5.53}$$

and let

$$\mathcal{K}_j = \left\{2 \leq \ell \leq m+1 : o_{T_{\ell-1}}(b_j(\sigma_\ell)) \leq \bar{f}(T_\ell)\right\}, \tag{5.54}$$

where $b_j(\sigma_\ell)$ and $o_{T_{\ell-1}}(b_j(\sigma_\ell))$ are given as follows:

$$b_j(\sigma_\ell) = \tilde{s}_j + (W_\rho(\tilde{s}_j; A_1) - \sigma_\ell)/\rho + L_j/C, \tag{5.55}$$

$$b_j(\sigma_\ell) o_{T_{\ell-1}}(b_j(\sigma_\ell)) = b_j(\sigma_k) o_{T_{\ell-1}}(b_j(\sigma_k)) + (W_\rho(\tilde{s}_j; A_1) - \sigma_\ell)/\rho. \tag{5.56}$$

We now present a third implementation of the canonical (σ^*, ρ) regulator given by

$$\sigma^*(j) = \begin{cases} \sigma_{\max \mathcal{K}_j}, & \text{if } 1 \in \mathcal{J}_j \ \text{ and } \ \mathcal{K}_j \neq \emptyset, \\ \sigma_1, & \text{otherwise.} \end{cases} \tag{5.57}$$

Theorem 5.4.3. The (σ^*, ρ) regulator defined by (5.46) and (5.52)–(5.57) produces the same output traffic as the (σ^*, ρ) regulator of Theorem 5.4.2 for a given input flow and hence the output flow satisfies (5.43) for all $t \geq 0$.

Algorithm 8 Replacement for lines 11–18 of Algorithm 6

11: $m \leftarrow 0$; found $\leftarrow$ **false**
12: **for** $\ell = 1, \ldots, k-1$ **do** $\triangleright$ $k \geq 2$
13:     $\sigma \leftarrow \sigma_k$; Compute $t_j(\sigma)$, $b_j(\sigma)$, $\triangleright$ (5.18), (5.9)
14:     Compute $\epsilon_{\ell,j}(\sigma)$, $o_{T_\ell}(b_j)$ $\triangleright$ (5.50), Prop. 5.4.1
15:     **if** $o_{T_\ell}(b_j) > \bar{f}(T_\ell) - \epsilon_{\ell,j}(\sigma)$ **then** $\triangleright$ (5.52)
16:         **break**
17:     **end if**
18:     $m \leftarrow m + 1$
19: **end for**
20: **for** $\ell = m+1, \ldots, 2$ **do**
21:     $\sigma \leftarrow \sigma_\ell$; Compute $b_j(\sigma)$, $o_{T_{\ell-1}}(b_j)$ $\triangleright$ (5.55), (5.56)
22:     **if** $o_{T_{\ell-1}}(b_j) \leq \bar{f}(T_\ell)$ **then** $\triangleright$ (5.49)
23:         found $\leftarrow$ **true**; **break**
24:     **end if**
25: **end for**

A proof of Theorem 5.4.3 is given in Appendix D.3. The (σ^*, ρ) regulator corresponding to Theorem 5.4.3 can be implemented by replacing lines 11-18 in Algorithm 6 with the lines shown in in Algorithm 8. The **for** loops at lines 11 and 19 in Algorithm 8 both have complexity $O(M)$. Therefore, the overall complexity of Algorithm 8 is $O(M)$ per packet. As with Algorithm 6, using a suitable parallel implementation, the complexity per packet can be further reduced to $O(1)$.

To summarize, Algorithm 8 provides a theoretical guarantee that the sample path bound (5.43) holds for all $t \geq 0$. Algorithm 6 provides the weaker guarantee that the sample path bound is satisfied asymptotically. However, our empirical studies have shown that Algorithm 6 achieves the bound (5.43) very quickly in practice. Both implementations have computational complexity $O(M)$. Our simulation results (see Tables 5.1 and 5.2 in Section 5.5), show that Algorithm 1 incurs higher average delay and greater variance in shaping the traffic to the desired gSBB bound.

5.5 Numerical Results

We evaluate the performance of the (σ^*, ρ) regulator first in a basic scenario with Poisson-like traffic and then a more realistic example with bursty traffic.

5.5.1 Basic Scenario

First, we consider a system similar to one studied in [24]. The packets sizes L_j are drawn randomly according to

$$L_j \sim \mathrm{U}\{L_{\min}, L_{\min}+1, \ldots, L_{\max}\}, \tag{5.58}$$

where $\mathrm{U}(\mathcal{A})$ denotes a uniform distribution over the set $\mathcal{A}$. The inter-arrival times of the packets, $s_{j+1} - s_j$, are determined as follows:

$$s_{j+1} - s_j \sim U_j + L_j/C, \tag{5.59}$$

where $U_j \sim \mathrm{Exp}(\lambda)$, i.e., $\{U_j\}$ is an i.i.d. sequence of exponentially distributed random variables with rate parameter λ. By adopting (5.59) to model the inter-arrival times, we ensure that packets are received after the previous ones have been fully received, i.e., the packets will not overlap with each other. In a system described by (5.58)–(5.59), $\rho^{-1}W_\rho(s_j; A_\mathrm{i})$ is equal to the waiting time experienced by the jth customer in a $G/G/1$ queueing system in which the service time of the jth customer is given by $S_j = (\rho^{-1} - C^{-1})L_j$ and the inter-arrival time between the jth and $(j+1)$th customer is U_j [24, 54, 64].

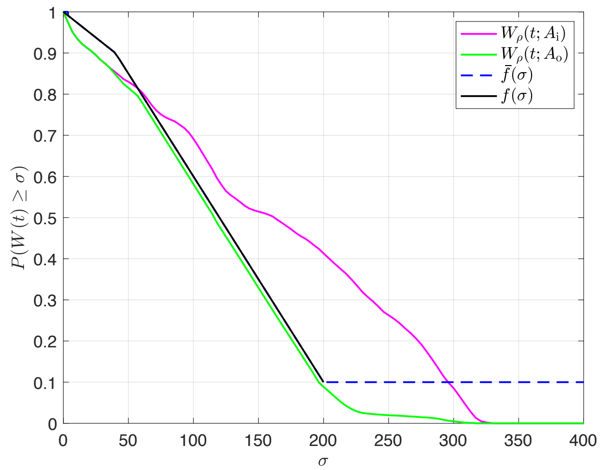


Figure 5.8: Performance of the stochastic (σ^*, ρ) traffic regulator.

In this example, we set $L_{\min} = 5$, $L_{\max} = 10$, and $\lambda = 0.25$, and $\rho = 0.65$. We use the following bounding function:

$$f(\sigma) := \begin{cases} -2.5 \times 10^{-3}\sigma + 1, & 0 \leq \sigma \leq 40, \\ -5 \times 10^{-3}\sigma + 1.1, & 40 < \sigma \leq T = 200. \end{cases} \tag{5.60}$$

In Fig. 5.8, $\bar{f}$ is defined by approximating f by a piecewise-linear function according to (5.41) with $M = 20$, $T_M = 400$ and $T_{i+1} - T_i = 20$ for $i = 1, \ldots, M-2$. Note that, as $f(\gamma)$ is also piecewise-linear, $\bar{f}(\gamma) = f(\gamma)$ for $\gamma \in [T_1, T]$. Observe that the output traffic is shaped to satisfy the desired bound.

Using the same model for inter-arrival and packet lengths, we have investigated the impact of the parameter M on traffic shaping of the input traffic. From Fig. 5.9, we see that as M is increased, a closer fit of the output traffic to the desired bound can be achieved. In our example, the maximum possible value of M, given by (5.36), is $M_{\max} = 56$, for which a very close fit to the bound is achieved. Figures 5.8 and 5.9 were obtained using Algorithm 8. In Fig. 5.8, the gSBB bound given by (5.60) is represented by the black curve. The pink curve, representing the input traffic, violates this bound. The green curve, which

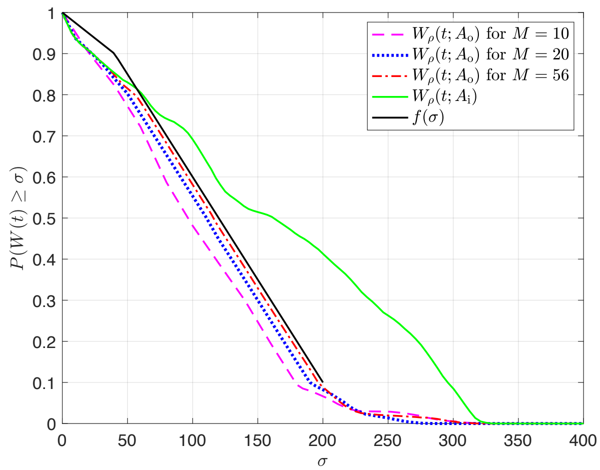


Figure 5.9: Traffic regulator performance with different M values.

Table 5.1: Traffic shaping delay with different M values for Algorithm 6 and Algorithm 8.

	Average Delay		Std. Dev. of Delay	
M	Alg. 6	Alg. 8	Alg. 6	Alg. 8
10	89	89	115	115
20	78	78	109	109
56	72	71	100	99

corresponds to the output traffic, shows that the (σ^*, ρ) regulator succeeds in enforcing the gSBB bound.

Table 5.1 presents the average delay and standard deviation of the delay for the packets using Algorithms 6 and 8. Note that as M increases the average delay decreases and the standard deviation of the packet delay also decreases. An increase in M implies that the delay incurred on a packet can increase in smaller increments, resulting in smaller overall variance. A larger value of M results in a smaller average delay since in this case the piecewise-linear function $\bar{f}$ given by Fig. 9 also shows the impact of increasing the value of M. With larger values of M, the workload tail probability of the output is closer to the gSBB bound, i.e., the bound becomes less conservative, since f is better approximated by $\bar{f}$.

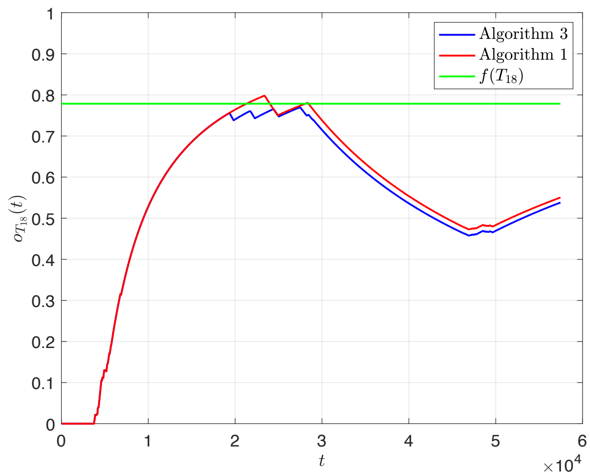


Figure 5.10: Overshoot ratio $o_{T_{17}}(t)$ for $M = 56$.

Algorithm 8 slightly outperforms Algorithm 6 for larger values of M with respect to mean and standard deviation of shaping delay, in particular, $M = 56$, as shown in Table 5.1. In Fig. 5.10, a sample path of the overshoot ratio $o_{T_{17}}(t)$ is shown for Algorithms 6 and 8. Observe that some violations of (5.34) occur with Algorithm 6 but there are no violations with Algorithm 8, which confirms Theorem 5.4.2.

5.5.2 Bursty Traffic Scenario

Next, we consider a more realistic scenario with bursty traffic. The purpose of this example is to show how the (σ^*, ρ) regulator can be applied to guarantee a delay bound for a traffic flow at a multiplexer and contrast this with an equivalent delay bound that can be provided using a (σ, ρ) regulator. This scenario can be generalized to a multi-hop network providing end-to-end stochastic delay guarantees by applying results from stochastic network calculus [35, 50, 51].

The packet inter-arrival times are given by (5.59), where the sequence $\{U_j\}$ is generated according to the inter-arrival times of a three-state Markov modulated Poisson Process (MMPP) with arrival matrix $\mathbf{\Lambda}$ and rate matrix $\mathbf{R}$ [36, 54]. In our example, the parameters of MMPP process are chosen according to [27, p. 79], with arrival matrix

$$\mathbf{\Lambda} = \mathrm{diag}\{116, 274, 931\} \tag{5.61}$$

in units of packets/s and rate matrix

$$\mathbf{R} = \begin{bmatrix} -0.12594 & 0.12594 & 0 \\ 0.25 & -2.22 & 1.97 \\ 0 & 2 & -2 \end{bmatrix} \tag{5.62}$$

in units of s^{-1}. These values were derived from matching arrival process of the I, P and B frames of an MPEG-4 encoded video to the three states of the MMPP. For the given MMPP, the average arrival rate is 358 packets/s.

The packet sizes L_j are generated according to the phase-type distribution referred to as G3 in [27, Table 1] as follows:

$$L_j \sim 0.54\ \mathrm{Er}(5, 26) + 0.46\ \mathrm{Er}(5, 956), \tag{5.63}$$

in units of bytes, where $\mathrm{Er}(r, 1/\mu)$ denotes an $r-$stage Erlang distribution with mean $1/\mu$ [11, 54]. This particular phase-type distribution is a mixture of Erlang distributions, which closely approximates the empirical distribution of measured Internet packet sizes obtained in [37]. We truncate the phase-type distribution at 1500 bytes, which is the MTU (Maximum Transmission Unit) for Ethernet. In addition, since the packet lengths are integer values, the random values generated according to the truncated phase-type distribution are quantized. The average packet size according to (5.63) is 454 bytes. In our case, however, due to truncation and quantization, the average packet size is 438 bytes. We have set the input link capacity to 10 Mbps. Because the packet inter-arrival times are given by (5.59), with this choice of C, the average packet arrival rate is 319 packets/s. The average bit rate of the traffic is 1.06 Mbps.

We consider a scenario in which the available bandwidth at a multiplexer is $C_o = 2$ Mbps. Since C_o exceeds the average packet rate of the traffic source, the flow can be supported. We consider two traffic descriptors:

1) phase-type descriptor $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$ (Definition 5.2.4);

2) (σ, ρ) descriptor (Definition 5.2.1).

For both traffic descriptors we set $\rho = C_o = 2$ Mbps. The black curve in Fig. 5.11 shows the phase-type bound obtained using Algorithm 3 in Chapter 4, with a 10-component hyperexponential distribution. The parameter T is set as the maximum value of $W_\rho(t; A_\mathrm{i})$, i.e., $T = 75$ KBytes. This ensures that the bounding function f corresponding to the phase-type descriptor bounds $\mathsf{P}\{W_\rho(t; A_\mathrm{i}) \geq \sigma\}$ for all values of $\sigma > 0$. For the (σ, ρ) descriptor, we set $\sigma = T = 75$ KBytes to ensure that no shaping delay will be incurred by the traffic regulator.

Along with the traffic descriptor, the user specifies a maximum delay bound requirement at the multiplexer. An admission controller determines whether or not the delay requirement can be satisfied with the available bandwidth C_o. If the traffic flow is admitted, a traffic regulator is applied to ensure conformance of the traffic flow to the traffic descriptor provided by the user. In case 1), a (σ^*, ρ) regulator is determined by finding a piecewise-linear approximation $\bar{f}$ to the bounding function f associated with the phase-type traffic descriptor (see Section 5.4). Fig. 5.11 shows the bounding function f and its approximation $\bar{f}$ (dashed blue curve). Then Algorithm 6 or 8 can be applied to enforce $\bar{f}$. Because the bounding function $\bar{f}$ is a close upper bound to $W_\rho(t, A_\mathrm{i})$, the workload curves at the input and output of the regulator are very close to each other. Hence, in Fig. 5.11, just $W_\rho(t, A_\mathrm{o})$ is shown. Consequently, the (σ^*, ρ) regulator incurs negligible shaping delay.

We next consider the delay bound that can be guaranteed at the multiplexer for the two cases. Let $Q(t)$ denote the queue size at the multiplexer with constant service rate C_o and let $D(t)$ denote the delay at the multiplexer experienced by a bit arriving at time t. Note

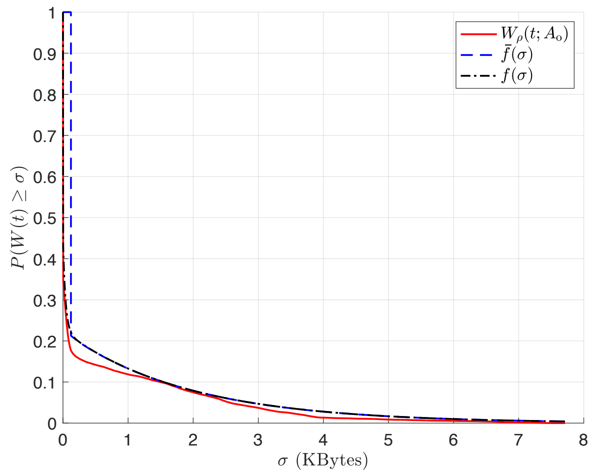


Figure 5.11: $\mathsf{P}\{W_\rho(t;A_\mathrm{o}) \geq \sigma\}$, $f(\sigma)$ and $\bar{f}(\sigma)$ vs. σ for bursty traffic source.

that if $C_o \geq \rho$,

$$Q(t) = W_{C_o}(t;A_\mathrm{i}) \leq W_\rho(t;A_\mathrm{i}). \tag{5.64}$$

Since $\rho = C_\mathrm{o}$, (5.64) holds with equality. For a FCFS server, $D(t) = Q(t)/C_o$. For the (σ^*, ρ) regulator,

$$\mathsf{P}\{D(t) \geq d\} = \mathsf{P}\{Q(t) \geq d\ C_o\} \leq \mathsf{P}\{W_\rho(t;A_\mathrm{i}) \geq d\ C_o\} = \bar{f}(d\ C_o) = \epsilon. \tag{5.65}$$

Setting $\epsilon = 0.02$ and using the bounding curve $\bar{f}$ from Fig. 5.11, we find that the smallest value of d that can satisfy (5.65) is 22 ms. For the (σ, ρ) regulator, the smallest delay bound that can be guaranteed is $d = \sigma/C_o = 294$ ms. Thus, with the (σ^*, ρ) regulator, a much smaller delay guarantee can be provided compared to that for the (σ, ρ) regulator.

In Fig. 5.12, the performance of the (σ, ρ) regulator is compared with that of a (σ^*, ρ) regulator with $M = 63$ for a sample path of the bursty traffic source. The output queue length attains the bound σ on the burst size, but is far smaller than σ most of the time. By contrast, the value of σ^* closely tracks the output queue length. Clearly, the (σ, ρ) parameter provides an overly conservative bound on the traffic.

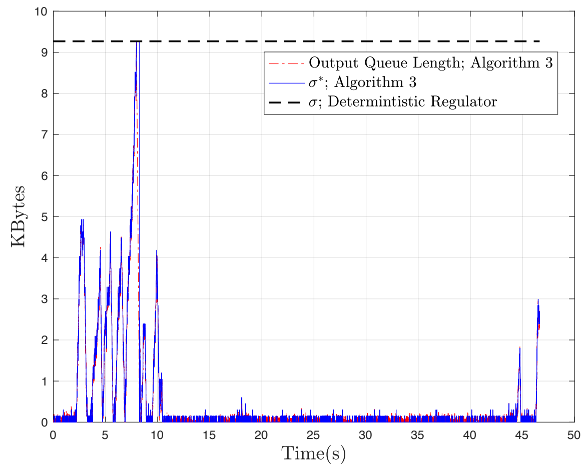


Figure 5.12: Workload profile of bursty traffic fed to server with service rate ρ and comparison of fixed bound σ vs. dynamic bound σ^*.

Table 5.2: Traffic shaping delay incurred by Algorithms 6 and 8 for bursty traffic ($M = 63$).

	Alg. 1	**Alg. 3**
Average Delay [ms]	27	1.2
Std. Dev. of Delay [ms]	50	15
$\mathsf{P}\{\text{shaping delay} > 0\}$	0.35	0.008

In Table 5.2, Algorithms 1 and 3 are compared in terms of the shaping delay incurred on the bursty traffic. Algorithm 3 shapes the input traffic to the desired gSBB bound while inducing significantly less traffic shaping delay. In particular, the value of $\mathsf{P}\{\text{shaping delay} > 0\}$ shows that the stochastic guarantee in (5.65) is achieved by Algorithm 3.

5.6 Conclusion

The stochastic traffic regulator developed in this chapter addresses an open problem in the application of stochastic network calculus to real networks: enforcement of stochastic traffic bounds. Given an input traffic source, our proposed stochastic (σ^*, ρ) regulator inserts delays, as necessary, to ensure that the output traffic conforms to the gSBB traffic bound [51]. Operationally, the (σ^*, ρ) regulator works similarly to a deterministic (σ, ρ) regulator, except that the burstiness parameter σ^* is chosen, for each arriving packet, from among a finite set of burst size parameters, $\Sigma = \{\sigma_1, \ldots, \sigma_M\}$. We showed, through both analysis and simulation, that the (σ^*, ρ) regulator ensures conformance of a traffic source to a given gSBB bound. Such a bound would be negotiated between the user and the network during the admission control phase and potentially renegotiated during the lifetime of the flow (cf. [71]). A closer fit to the gSBB bound can be achieved by increasing the value of M.

Chapter 6: A Framework for Providing Stochastic Delay Guarantees in Communication Networks

The provisioning of delay guarantees in packet-switched networks such as the Internet remains an important, yet challenging open problem. In this chapter, we propose and evaluate a framework, based on the results from stochastic network calculus, for guaranteeing stochastic bounds on network delay at a statistical multiplexer. The framework consists of phase-type traffic bounds and moment generating function traffic envelopes, stochastic traffic regulators to enforce the traffic bounds, and an admission control scheme to ensure that a stochastic delay bound is maintained for a given set of flows. Through numerical examples, we show that a stochastic delay bound is maintained at the multiplexer, and contrast the proposed framework to an approach based on deterministic network calculus. Parts of the work in this chapter were published in [58].

The remainder of the chapter is organized as follows. In Section 6.1, we provide an introduction and motivation about the framework for providing stochastic delay guarantees. In Section 6.2, we discuss the phase-type traffic bound and its use as a traffic descriptor, as well as the MGF traffic envelope. In Section 6.3, we discuss a scheme to enforce both a phase-type bound and an MGF envelope for a traffic process. In Section 6.4, we discuss an admission control scheme for a statistical multiplexer based on the phase-type bounds and results from stochastic network calculus. Numerical results, which demonstrate the proposed framework are presented in Section 6.5. Concluding remarks are given in Section 6.6.

6.1 Introduction

Currently, the Internet does not provide end-to-end delay guarantees for traffic flows. Even if the path taken by a given traffic flow is fixed, e.g., via mechanisms such as software-defined networking (SDN) or multi-protocol label switching (MPLS), network congestion arising from other flows can result in highly variable delays. The variability and random nature of traffic flows in a packet-switched network make it very challenging to provide performance guarantees.

The standard approach to providing network performance guarantees consists of two basic elements:

1. *Admission control:* A new flow is admitted to the network only if sufficient resources are available to maintain a given performance guarantee.

2. *Traffic regulation:* Each traffic flow must be regulated to ensure that it does not use more resource than what was negotiated by the admission control scheme.

Admission control is challenging due to the random and bursty nature of traffic flows, which makes them difficult to characterize and regulate. Even when flows are modeled as random arrival processes, provisioning for end-to-end performance guarantees in a multi-hop network is generally intractable.

In his seminal work, Cruz [24, 25] proposed the so-called (σ, ρ) characterization of traffic, which imposes a deterministic bound on the burstiness of a traffic flow. By bounding traffic flows according to (σ, ρ) parameters, Cruz developed a network calculus which determined how these parameters propagate through network elements, from which end-to-end delay bounds could be derived. An important feature of the (σ, ρ) characterization is that it could be enforced by a traffic regulator. In practice, however, the deterministic (σ, ρ) characterization leads to very loose end-to-end delay bounds, which leads to very low utilization of the network resources. Nevertheless, the (σ, ρ) characterization was the basis for further research into stochastic bounds on traffic burstiness and stochastic network calculus to provide tighter, probabilistic end-to-end delay guarantees. Stochastic network calculus and associated performance bounds remains an active topic of research, with ongoing efforts aimed at improving the tightness of the stochastic delay bounds. To our knowledge, however, stochastic network calculus has not previously been applied within a practical framework to provide performance guarantees.

We present a practical framework for providing performance guarantees based on concepts from stochastic network calculus. Traffic flows are characterized by the phase-type traffic traffic descriptor proposed in Chapter 4 as well as a moment generating function (MGF) traffic envelope [35]. The phase-type traffic bound for each traffic flow is enforced by a stochastic traffic regulator, see Chapter 5. An admission control scheme decides whether or not to admit a new traffic flow on the basis of both the phase-type descriptor and the MGF envelope to guarantee a stochastic delay bound for all admitted traffic flows. The main contribution of this chapter is to demonstrate that stochastic delay guarantees can be achieved for admitted flows while maintaining relatively high traffic utilization in the network.

6.2 Stochastic Traffic Bounds

Let $A = \{A(s,t) : 0 \leq s \leq t\}$ denote a traffic arrival process, where $A(s,t)$ denotes the amount of traffic arriving in time interval $[s,t)$. For simplicity, we shall assume that the time parameters s and t are discrete unless otherwise specified, but the results that follow also carry over to the continuous-time case. Our proposed framework involves two types of bounds on a traffic process: a phase-type traffic bound and an MGF traffic envelope.

6.2.1 Phase-type Traffic Bound

We consider stochastic bounds on the *burstiness* of a traffic flow, with respect to an *upper rate* ρ, which is chosen to be larger than or equal to the long term average traffic rate, i.e., $\rho \geq \lim_{t\to\infty} \frac{A(0,t)}{t}$, The concept of phase-type bounded traffic is defined as follows Chapter 3.

Definition 6.2.1. A traffic process A is characterized by a *phase-type traffic descriptor* $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$ if

$$\mathsf{P}\{W_\rho(t; A) \geq \sigma\} \leq a\boldsymbol{\pi} e^{\mathbf{Q}\sigma}\mathbf{1}, \tag{6.1}$$

for all $t \geq 0$ and all $\sigma \in (0, T]$. Here, $\mathbf{1}$ is a column vector of all ones, $a \geq 0$, $T > 0$. The virtual workload of a constant rate queue with service rate ρ and input traffic A is defined by

$$W_\rho(t; A) := \max_{0 \leq s \leq t} [A(s,t) - \rho(t - s)], \tag{6.2}$$

and $(\boldsymbol{\pi}, \mathbf{Q})$ denotes the parameter of a phase-type distribution [11].

When $T = \infty$, the phase-type traffic bound is a particular case of generalized stochastically bounded burstiness (gSBB), which was developed in [51, 99]. In Chapter 3, it was shown that the phase-type bound defined above is closed with respect to a stochastic network calculus based on the gSBB concept.

The concept of gSBB is closely related to the Stochastically Bounded Burstiness (SBB) concept introduced in [85], which in turn is a generalization of Exponentially Bounded Burstiness (EBB) [96]. A key feature gSBB vs. SBB is that it is based on the workload process $W_\rho(t; A)$, which can be reasonably assumed to be stationary and ergodic, rather than the arrival process A, which is neither stationary nor ergodic. Consequently, as discussed in Section 6.3, a stochastic traffic regulator can be designed based on enforcement of a time-average approximation of the left-hand side of (6.1).

The problem of finding a phase-type traffic descriptor to fit a given traffic trace can be formulated as a semi-infinitely constrained optimization problem Chapter 4, which can be solved numerically for special phase-type distributions such as the hyperexponential distribution. In particular, the hyperexponential distribution provides a tight phase-type bound for a large class of traffic flows. Given the procedure developed in Chapter 4, we shall assume that each traffic flow that requests admission to the network has an associated phase-type traffic descriptor.

6.2.2 MGF Traffic Envelope

An alternative approach to characterizing a traffic process is to bound the moment generating function (MGF)

$$M_A(\theta; s, t) := E\left[e^{\theta A(s,t)}\right]$$

where $\theta > 0$ is a free parameter [35].

Definition 6.2.2. The MGF traffic envelope of traffic process A is defined by

$$E\left[e^{\theta A(s,t)}\right] \leq e^{\theta(\hat{\rho}(\theta)(t-s)+\hat{\sigma}(\theta))}, \tag{6.3}$$

where the parameters $\hat{\rho}(\theta) > 0$ and $\hat{\sigma}(\theta) \geq 0$ are functions of $\theta > 0$.

The MGF traffic envelope is analogous to the deterministic (σ, ρ) characterization in that it involves analogous parameters $\hat{\sigma}(\theta)$ and $\hat{\rho}(\theta)$ and it can be related to the EBB characterization via the Chernoff bound.

The MGF traffic envelope, however, has some advantages compared to the phase-type bound traffic descriptor, the most important being the following,

Theorem 6.2.1 (Sum of MGF envelopes). When n independent flows $A_1, \ldots, A_n$, with MGF envelope parameters $(\hat{\sigma}_1, \hat{\rho}_1), \ldots (\hat{\sigma}_n, \hat{\rho}_n)$, respectively, are superposed, the aggregate traffic process $A = A_1 + \ldots + A_n$ can be characterized by the MGF parameter $(\hat{\sigma}, \hat{\rho})$, where $\hat{\sigma} = \sum_{i=1}^{n} \hat{\sigma}_i$ and $\hat{\rho} = \sum_{i=1}^{n} \rho_i$.

This property of the MGF traffic envelope not only simplifies the computations involved in admission control, but more importantly, it captures the effect of statistical multiplexing gain. For this reason, our proposed framework uses *both* the phase-type bound traffic descriptor and the MGF traffic envelope. The problem of finding a MGF traffic envelope can be simplified by defining a finite set Θ of values to consider for the free parameter θ in (6.3). Then a set of MGF envelope parameters, $\{(\hat{\sigma}(\theta), \hat{\rho}(\theta)) : \theta \in \Theta\}$, could be determined using an approach similar to the procedure in Chapter 4 for fitting the phase-type traffic descriptor.

6.3 Stochastic Traffic Regulation

Next, we discuss methods for enforcing both a phase-type bound traffic descriptor and the MGF traffic envelope.

6.3.1 (σ^*, ρ) Regulator

The deterministic (σ, ρ) regulator tends to provide a very loose bound on the traffic or to incur unnecessarily large delays on the traffic. To address these issues, a *stochastic* traffic regulator was proposed in Chapter 5, which enforces a probabilistic bound on a traffic process A:

$$\mathsf{P}\{W_\rho(t; A) \geq \gamma\} \leq f(\gamma), \qquad \forall \gamma \in [0, T], \tag{6.4}$$

where $f(\gamma)$ is a non-increasing positive bounding function and T is a limit on the tail distribution of the workload. We refer to a regulator that enforces (6.4) as a stochastic (σ^*, ρ) regulator, where the burstiness parameter σ^* is variable.

Users specify their traffic flows with a descriptor $[\rho; (f(\gamma), T)]$ in terms of a bound of the form (6.4). In particular, for the phase-type bound the bounding function has the form $f(\gamma) = a\boldsymbol{\pi} e^{\mathbf{Q}\gamma}\mathbf{1}$ (cf. (6.1)). By applying results from stochastic network calculus, the admissibility of a given set of traffic flows with respect to a certain probabilistic end-to-end delay constraint can be determined. However, such an end-to-end delay guarantee can only be provided if the traffic flows conform to their negotiated traffic parameters. The (σ^*, ρ) regulator can be applied at the network edge to force compliance of each traffic flow to a negotiated phase-type bound parameter. Optionally, the regulator could be applied at internal nodes of the network to reshape traffic flows to their negotiated phase-type traffic bounds. This has the benefit of maintaining the negotiated traffic profile for each traffic flow over a multi-hop path, but requires the additional overhead of traffic regulation within the network.

6.3.2 MGF Traffic Envelope Regulator

According to Definition 6.2.2, the MGF envelope parameters $\hat{\rho}(\theta)$ and $\hat{\sigma}(\theta)$ satisfy (6.3). However, verification of (6.3), requires estimation of the MGF $E\left[e^{\theta A(s,t)}\right]$, which presents difficulties because the traffic process A is non-stationary and non-ergodic. Therefore, we introduce an alternative MGF envelope characterization.

Definition 6.3.1. The MGF workload envelope (or w-envelope) of traffic process A is defined by

$$E\left[e^{\theta W_{\hat{\rho}(\theta)}(t;A)}\right] \leq e^{\theta\hat{\sigma}(\theta)}, \tag{6.5}$$

where $\theta > 0$ is a free parameter, $\hat{\rho}(\theta) > 0$ and $\hat{\sigma}(\theta) \geq 0$, and $W_{\hat{\rho}}(A;t)$ is the workload defined in (6.2).

The MGF w-envelope provides an upper bound on the MGF traffic envelope in the following sense.

Theorem 6.3.1. Suppose a traffic process A has an MGF w-envelope with parameter $\{(\hat{\sigma}(\theta), \hat{\rho}(\theta)) : \theta \in \Theta\}$, i.e.,

$$E\left[e^{\theta W_{\hat{\rho}(\theta)}(t;A)}\right] \leq e^{\theta\hat{\sigma}(\theta)}, \quad \theta \in \Theta. \tag{6.6}$$

Then it is also characterized by an MGF traffic envelope $\{(\hat{\sigma}(\theta), \hat{\rho}(\theta)) : \theta \in \Theta\}$, i.e.,

$$E\left[e^{\theta A(s,t)}\right] \leq e^{\theta[\sigma+\rho(t-s)]}, \quad 0 \leq s \leq t, \quad \theta \in \Theta. \tag{6.7}$$

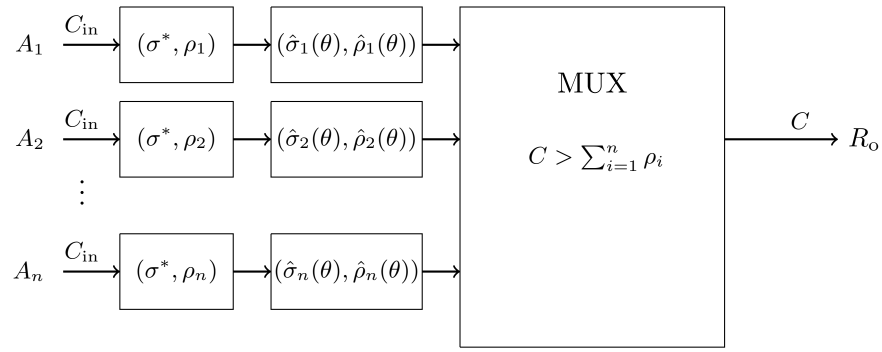


Figure 6.1: Multiplexer with n independent traffic flows.

Proof. For $0 \leq s \leq t$,

$$A(s,t) - \rho(t-s) \leq \max_{0 \leq s \leq t} [A(s,t) - \rho(t-s)] = W_\rho(t; A).$$

Therefore,

$$E[e^{\theta[A(s,t)-\rho(t-s)]}] \leq E[e^{\theta W_\rho(t;A)}],$$

and the result follows immediately. □

Theorem 6.3.1 implies that a traffic regulator which enforces an MGF w-envelope with parameter $(\hat{\sigma}, \rho)$ also enforces an MGF traffic envelope with the same parameter. To enforce an MGF traffic envelope for a traffic process A, we can estimate the left-hand side of (6.6), for each value of $\theta \in \Theta$, via a time-average and regulate it to ensure that the inequality is maintained. This can be accomplished by designing a stochastic regulator along the lines of the (σ^*, ρ) regulator in Chapter 5.

6.4 Admission Control

We develop an admission control scheme based on a stochastic delay bound derived from the phase-type traffic bound and MGF traffic envelope. The phase-type bound provides a tighter delay bound when a small to moderate number of flows is considered. When the number of flows becomes larger, the MGF envelope can yield a tighter bound due to the statistical multiplexing effect. Therefore, we propose a *hybrid* admission control scheme which uses both types of traffic bounds.

6.4.1 Admission Control via Phase-Type Bound

Consider a multiplexer of capacity C with a set of n independent traffic flows, $\mathcal{A} = \{1, \ldots, n\}$, as inputs characterized by phase-type traffic descriptors $\Delta_i = [\rho_i, (a_i, \boldsymbol{\pi}_i, \mathbf{Q}, T_i)]$, $i = 1, \ldots, n$. The essential task of the admission controller is to determine whether or not a stochastic delay bound of the following form can be satisfied:

$$\mathsf{P}\{D \geq d\} < \epsilon, \tag{6.8}$$

where D represents the delay experienced by a packet in the multiplexer, $\epsilon > 0$ is a small number, e.g., $\epsilon = 10^{-3}$, and d represents a "maximum" tolerable delay for a packet from any of the admitted flows. Clearly, a necessary condition for (6.8) to be satisfied is $\sum_{i=1}^{n} \rho_i < C$.

A phase-type traffic bound for the aggregate traffic input to the multiplexer can be determined by repeated application of the following theorem.

Theorem 6.4.1 (*Independent Sum*). Let A_1 and A_2 be independent traffic processes characterized by phase-type traffic descriptors $\Delta_1 = [\rho_1, (a, \boldsymbol{\alpha}, \mathbf{G}, T_1)]$, and $\Delta_2 = [\rho_2, (b, \boldsymbol{\beta}, \mathbf{H}, T_2)]$, respectively. The aggregate process $A = A_1 + A_2$ is bounded by the phase-type traffic descriptor $\Delta = [\rho, (c, \boldsymbol{\pi}, \mathbf{Q}, T)]$. where $\rho = \rho_1 + \rho_2$, $T = \min(T_1, T_2)$, $c = a + b - ab$,

$$\boldsymbol{\pi} = \left[\frac{a(1-b)}{c}\boldsymbol{\alpha}, \frac{b(1-a)}{c}\boldsymbol{\beta}, \frac{ab}{c}\boldsymbol{\alpha}, \mathbf{0} \right], \tag{6.9}$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{G} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G} & \mathbf{g}\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H} \end{pmatrix},$$

where $\mathbf{g} = -\mathbf{G}\mathbf{1}$.

This theorem can be derived from [99, Theorem 4] for gSBB flows by applying properties of the phase-type distribution. In Theorem 6.4.1, if the number of phases represented by the phase-type traffic descriptors Δ_1 and Δ_2 equals m, then the number of phases in Δ will be $4m$. To avoid this expansion in the size of the phase-type traffic descriptor, we can apply the numerical method developed in Chapter 4, to determine a phase-type traffic descriptor $\tilde{\Delta}$, which approximates the true aggregate descriptor Δ, but only has m phases. Thus, we obtain a practical procedure for obtaining a phase-type traffic descriptor for the superposition of an arbitrary set of independent flows characterized by phase-type traffic descriptors.

Given the above procedure for determining a phase-type traffic descriptor for an aggregate traffic flow A at the input to a multiplexer, a relationship between ϵ and d in (6.8) can be derived from following theorem:

Theorem 6.4.2. Let A be a traffic process with phase-type descriptor $[\rho; (a, \boldsymbol{\pi}, \mathbf{Q}, T)]$ that is input to a FIFO system with constant transmission rate $C > \rho$. Then the steady-state delay D through the system can be bounded as follows:

$$\mathsf{P}\{D \geq d\} = \mathsf{P}\{W_C(t; A) \geq Cd\} \leq \mathsf{P}\{W_\rho(t; A) \geq Cd\} \leq a\boldsymbol{\pi} e^{\mathbf{Q}Cd}\mathbf{1}, \tag{6.10}$$

for all $t \geq 0$ and all $\frac{T}{C} \geq d \geq 0$.

6.4.2 Admission Control via MGF Envelope

In conjunction with Theorem 6.2.1, the following result can be used to perform admission control based on MGF traffic envelopes [35].

Theorem 6.4.3. Suppose a traffic process A with MGF envelope $(\hat{\sigma}(\theta), \hat{\rho}(\theta))$, $\theta \in \Theta$, is offered as input to a constant rate server of capacity $C > \hat{\rho}(\theta)$, $\theta \in \Theta$. Then the steady-state system delay D can be bounded as follows:

$$P\{D \geq d\} = \mathsf{P}\{W_C(t; A) \geq Cd\} \leq \mathsf{P}\{W_{\hat{\rho}(\theta)}(t; A) \geq Cd\} \leq e^{\theta(\hat{\sigma}(\theta) - d)}, \quad \theta \in \Theta. \tag{6.11}$$

The parameter θ can be optimized to minimize the right-hand side of (6.11) .

6.4.3 Hybrid Admission Control Scheme

We proposed a hybrid admission control scheme that combines the phase-type traffic descriptor and MGF traffic envelope characterizations of the input traffic flows. The basic setup is depicted in Fig. 6.1. Each flow passes through a (σ^*, ρ) stochastic regulator (see Section 6.3.1), which enforces a phase-type traffic descriptor negotiated between the network and the traffic flow. Similarly, an MGF traffic w-envelope for each flow is enforced by an MGF traffic regulator (see Section 6.3.2).

Given a set of flows $\mathcal{A} = \{1, \ldots, n\}$, the hybrid admission control scheme checks two admission criteria with respect to the stochastic delay constraint (6.8):

1. Using the procedure based on the phase-type traffic descriptors outlined in Section 6.4.1, determine whether or not $\mathcal{A}$ is admissible.

2. Using the procedure based on MGF envelope parameters outlined in Section 6.4.2, determine whether or not $\mathcal{A}$ is admissible.

If $\mathcal{A}$ is admissible under criterion 1 *or* criterion 2, then $\mathcal{A}$ is considered admissible.

6.5 Numerical Study

In this section, we demonstrate key aspects of the proposed framework using traffic flows modeled as MMPPs and discrete-time Markov on-off fluid processes.

6.5.1 Markov Modulated Poisson Process

The MMPP is a popular continuous-time model for traffic flows possessing a high degree of burstiness [36]. An m-state MMPP is a doubly-stochastic Poisson point process $N(t)$ parameterized by a diagonal arrival matrix $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_m\}$, where $\lambda_i \geq 0$ is the Poisson arrival rate when the underlying Markov chain is in state i and a rate matrix $\mathbf{R} = [r_{ij}]$, $1 \leq i, j \leq m$, where $r_{ij} \geq 0$ is the departure rate of the Markov chain from state i to $j \neq i$. For $1 \leq i \leq m$, $r_{ii} < 0$ and $-r_{ii}$ is the departure rate of the Markov chain from state i. The rate matrix $\mathbf{R}$ is the generator matrix of the modulating Markov chain. For example, a 2-state MMPP is parameterized by arrival and rate matrices given, respectively, by

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} -r_1 & r_1 \\ r_2 & -r_2 \end{bmatrix}. \tag{6.12}$$

The superposition of n independent MMPPs is again an MMPP. The rate matrix $\mathbf{R}$ and the arrival rate matrix $\mathbf{\Lambda}$ of the aggregated process are given, respectively, by

$$\mathbf{R} = \mathbf{R}_1 \oplus \ldots \oplus \mathbf{R}_n, \quad \mathbf{\Lambda} = \mathbf{\Lambda}_1 \oplus \ldots \oplus \mathbf{\Lambda}_n, \tag{6.13}$$

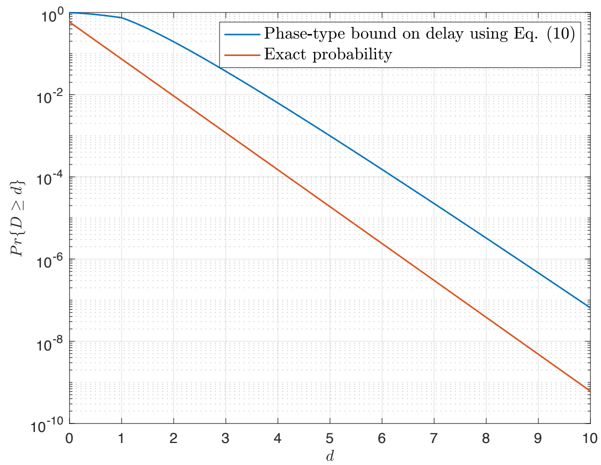


Figure 6.2: Stochastic delay bound via phase-type traffic bounds.

where $\oplus$ denotes the Kronecker-sum [36]. For example, the superposition of two independent, identically distributed 2-state MMPPs results in a 3-state MMPP with arrival matrix

$$\mathbf{\Lambda} = \begin{bmatrix} 2\lambda_2 & 0 & 0 \\ 0 & \lambda_1 + \lambda_2 & 0 \\ 0 & 0 & 2\lambda_1 \end{bmatrix} \tag{6.14}$$

and rate matrix

$$\mathbf{R} = \begin{bmatrix} -2r_2 & 2r_2 & 0 \\ r_1 & -r_1 - r_2 & r_2 \\ 0 & 2r_1 & -2r_1 \end{bmatrix}. \tag{6.15}$$

Assume that the packet lengths times are independent and generally distributed. Then the MMPP $N(t)$ together with the packet lengths specifies a continuous-time traffic arrival process A. When the process A is fed as input to a multiplexer with constant service rate, the system can be modeled as an MMPP$/G/1$ queue. A closed-form expression for the Laplace transform of the virtual waiting time of a MMPP$/G/1$ queue is given in [68].

6.5.2 Admission Control via Phase-Type Bounds

We consider the scenario shown in Fig. 6.1, in which five statistically independent traffic flows A_i, $i = 1, \cdots, 5$ arrive on input links with capacity C_{in} to a multiplexer with constant service rate C. All flows are identically distributed 2-state MMPPs characterized by Poisson arrival rates $\lambda_1 = 0.75$ and $\lambda_2 = 0.5$ and rate matrix given by $r_1 = 2$ and $r_2 = 1$. Hence, the average arrival rate of each flow is 0.583 packets/unit of time. All packet lengths are assumed to be exponentially distributed with mean $\mu^{-1} = 1$. The input link capacity is set as $C_{\text{in}} = 5$ and the multiplexer has a constant rate server of rate $C = 5$.

The superposition of traffic flows A_i, $i = 1, \ldots, 5$, is a 6-state MMPP whose parameter can be determined using (6.13). With the 6-state MMPP as the input traffic, the multiplexer can be modeled as an MMPP/M/1 queue. Therefore, a closed-form solution for the virtual waiting time distribution at the multiplexer can be determined using results from [68]. The orange curve in Fig. 6.2 shows the tail distribution of the incurred delay in this case. Using definition (6.1), a phase-type bound can be obtained for each input traffic stream by considering the virtual waiting time distribution of a 2-state MMPP/M/1 queue. Using results from [68], this distribution has the form of an hyperexponential distribution. For this scenario, the procedure for fitting a traffic flow to a phase-type traffic descriptor (see Section 6.2.1) can be bypassed. We shall set the parameter ρ equal to mean rate of the MMPP, i.e., 0.583. In this case, the phase-type bounding parameters of A_i, for $i = 1, \ldots, 5$ can be chosen to exactly match the tail of the workload distribution. Thus, we can assume $T = \infty$, and we obtain $a = 0.583$,

$$\boldsymbol{\pi} = [0.0.9982, 0.0018], \quad \mathbf{Q} = \begin{bmatrix} -0.413 & 0 \\ 0 & -0.858 \end{bmatrix}. \tag{6.16}$$

Since the traffic flows are MMPPs, they automatically satisfy the derived phase-type bounds and hence do not need to be regulated, although (σ^*, ρ) regulators are shown in Fig. 6.1 for the general case. Using the phase-type descriptor in (6.16), and Theorem 6.4.1,

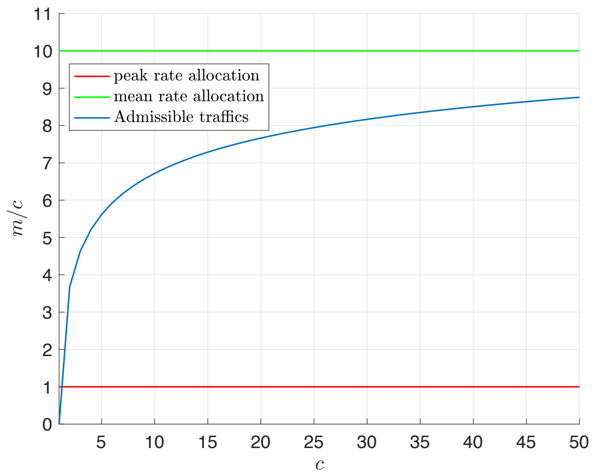


Figure 6.3: Statistical multiplexing gain via MGF traffic envelopes.

the phase-type descriptor of the aggregate arrival traffic can be derived. In this case, the aggregate traffic is characterized by a phase-type descriptor with a 92-state phase-type parameter. In this example, the approximation procedure described in Section 6.4.1 to limit the number of phases in the phase-type descriptor was not performed. Using this phase-type descriptor for the aggregate traffic, a bound on the delay, as shown in Fig. 6.2, can be derived via Theorem 6.4.2. The blue curve in Fig. 6.2 shows the bound on the delay. From Fig. 6.2 the phase-type bounds can be used to provide the following stochastic delay guarantee: $\mathsf{P}\{D \geq 5\} < 10^{-3}$. The output link utilization in this case is $5(0.583)/C = 0.583$.

To compare the stochastic delay guarantee with a deterministic guarantee, such as that provided by the (σ, ρ) characterization of Cruz [24], we can increase the value of C such that $\mathsf{P}\{D \geq 5\}$ is close to zero, say 10^{-10}. As shown in Fig. 6.2, we can derive the exact tail probability $\mathsf{P}\{D \geq 5\}$ for every C. By increasing the value of C, we have that $\mathsf{P}\{D \geq 5\} \leq 10^{-10}$ when $C > 8.5$. Therefore, the link utilization that can be achieved when a deterministic guarantee is provided can be most $5(0.583)/8.5 \approx 0.34$.

6.5.3 Admission Control via MGF Traffic Envelope

As mentioned in Section 6.2.2, an advantage of the MGF traffic envelope representation is that it can capture statistical multiplexing gain. Here, we shall consider a discrete-time Markov on-off fluid flow as a model for traffic flows. Such a process consists of an underlying discrete-time 2-state Markov process. In state 1 (On-state) the source generates a constant fluid flow of packets at rate r and in state 2 (Off-state), the source does not generate packets. The underlying Markov process has transition probability matrix $\mathbf{P}$ given by

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}, \tag{6.17}$$

where p_{ij} for $i, j = 1, 2$ are the transition probabilities from state i to state j. The steady-state probability of states On and Off are given, respectively, by

$$p_{\text{on}} = \frac{p_{12}}{p_{12} + p_{21}}, \qquad p_{\text{off}} = \frac{p_{21}}{p_{12} + p_{21}}. \tag{6.18}$$

The mean arrival rate of the process is $p_{\text{on}} r$. This process is also characterized by a burst parameter $\beta = 1/p_{12} + 1/p_{21}$. According to [35], the MGF traffic envelope of the Markov On-Off process is given by $\hat{\sigma}(\theta) = 0$ and $\hat{\rho}(\theta) =$

$$\frac{1}{\theta} \ln\left(\frac{p_{11} + p_{22} e^{\theta r} + \sqrt{(p_{11} + p_{22} e^{\theta r})^2 - 4(p_{11} + p_{22} - 1) e^{\theta r}}}{2} \right), \tag{6.19}$$

for $\theta > 0$. For the special case of a *memoryless* On-Off process, we have $p_{11} = p_{21}$ and $p_{22} = p_{12}$. In this case, $p_{\text{on}} = p_{22}$ and the MGF traffic envelope simplifies to the form $\hat{\sigma}(\theta) = 0$ and

$$\hat{\rho}(\theta) = \frac{1}{\theta} \ln\left(p_{\text{on}} e^{\theta r} + 1 - p_{\text{on}} \right), \qquad \theta > 0. \tag{6.20}$$

Consider a multiplexer with constant rate C. Suppose a maximum of M identically distributed and statistically independent input Markov on-off fluid flows can be supported at the multiplexer while satisfying (6.8) for some specific d and ϵ. We are interested in evaluating the number of admissible flows per unit capacity, given by M/C, as C increases. As an example, we shall assume that each Markov on-off fluid flow, A_i, for $1 \leq i \leq M$, has average rate $p_{\mathrm{on}}r = 0.1$, peak rate $r = 1$, and burst parameter $\beta = 300$. Each Markov on-off fluid flow can be characterized by an MGF traffic envelope $(\hat{\sigma}(\theta), \hat{\rho}(\theta))$, which can be enforced by a regulator, as discussed in Section 6.2.2. In this case, $\hat{\rho}_i(\theta)$ for $1 \leq i \leq M$, is given by (6.19), where $\theta > 0$ is a free parameter. Also as mentioned before, $\hat{\sigma}_i(\theta) = 0$ for $1 \leq i \leq M$.

According to Theorem 6.2.1, the aggregate traffic process, $A = A_1 + A_2 + \ldots + A_M$ can be characterized by the MGF parameter $\rho = M\hat{\rho}_1(\theta)$ and $\hat{\sigma} = 0$. The admission control scheme imposes a stochastic delay constraint of the form (6.8) with $d = 100$ and $\epsilon = 10^{-3}$. For each value of M and C, by using Theorem 6.4.3, and by optimizing the free parameter $\theta > 0$ we can derive a statistical bound on the delay for $d = 100$. If the derived statistical bound on the delay is less than ϵ, then such a choice of M and C is acceptable. For each value of C, we try to find the maximum value of acceptable M such that the desired statistical bound on the delay is satisfied.

With *mean rate* allocation, $1/0.1 = 10$ flows can be supported per unit capacity, whereas with *peak rate* allocation, only one flow can be supported per unit capacity. By performing admission control according to the MGF bound parameters, statistical multiplexing gain is achieved as C increases, as shown in Fig. 6.3. In particular, as C increases, the number of admissible flows per unit capacity, M/C increases and approaches the mean rate allocation of 10 flows per unit capacity. This shows that statistical multiplexing gain is achieved.

6.6 Conclusion

We presented a practical framework for providing stochastic delay guarantees based on results from stochastic network calculus. Key elements of our approach are the phase-type traffic bound, Chapter 3, the MGF traffic envelope [35], a method for fitting a traffic flow to a phase-type bound, Chapter 4, stochastic traffic regulators to enforce compliance of a traffic flow to a negotiated traffic descriptors, Chapter 5, and an admission control scheme. Each flow characterizes its traffic process by a phase-type traffic descriptor, which can be determined using the procedure developed in Chapter 4. Similarly, an MGF traffic envelope can be determined for each traffic flow. Both types of traffic descriptors are enforced by stochastic regulators and are used in the proposed admission control scheme.

Our numerical study showed that much higher traffic utilization can be achieved compared to the deterministic (σ, ρ) framework, while providing a stochastic delay guarantee. Moreover, even higher utilization can be achieved by taking into account statistical multiplexing gain. The main contribution of this work is to show how results from stochastic network calculus can be applied in a practical framework to provide performance guarantees. In ongoing work, we are extending the proposed framework to multi-hop networking scenarios.

Chapter 7: Traffic Workload Envelope for Network Performance Guarantees with Multiplexing Gain

Stochastic network calculus involves the use of traffic bounds to make admission control and resource allocation decisions for providing end-to-end quality-of-service guarantees. To apply network calculus in practice, the traffic bounds should: (i) be readily determined for an arbitrary traffic source, (ii) be enforceable by traffic regulation, and (iii) yield statistical multiplexing gain. Existing traffic bounds typically satisfy at most two of these properties. The traffic envelope based on the moment generating function (MGF) of the arrival process satisfies only the third property. We propose a new traffic envelope, refereed to as ‘ We show that the traffic workload envelope satisfies all three properties and propose a framework for a network service that provides stochastic delay guarantees. We demonstrate the performance of the traffic workload envelope with two bursty traffic models: Markov on-off fluid and Markov modulated Poisson Process (MMPP). Parts of the work in this chapter were published in [59].

The remainder of this chapter is organized as follows. Section 7.1 provides an introduction and motivation about traffic workload envelope for network performance guarantees with multiplexing gain. Section 7.2 provides relevant background on traffic bounds and network calculus. In Section 7.3, we introduce the W-envelope and establish its key properties. In Section 7.4, we outline a framework, based on the W-envelope, for providing delay guarantees. The W-envelopes for two widely used traffic models are obtained in Section 7.5 and numerical examples are presented in Section 7.6. The chapter is concluded in Section 7.7.

7.1 Introduction

Provisioning quality-of-service (QoS) guarantees in a resource-efficient manner remains a challenging, yet important open problem for future networking. To this day, the Internet does not have the capability to provide end-to-end delay guarantees, which are required for time-critical multimedia applications. The packet-switching paradigm of the Internet enables very efficient network resource utilization due to statistical multiplexing, but makes provisioning for performance guarantees extremely challenging. The difficulties in achieving delay guarantees in packet-switched networks lie in how to bound bursty traffic, enforce such bounds, and achieve statistical multiplexing gain using these bounds.

Stochastic network calculus is a theoretical framework for leveraging the statistical characteristics of traffic for efficient resource allocation while providing end-to-end QoS guarantees in the form of stochastic delay bounds. As yet, howewver, stochastic network calculus has not been successfully applied to practical networks. To apply stochastic network calculus in practice, it is necessary: 1) to fit a given flow to a suitable traffic bound on the user side; 2) to enforce the traffic bound on the network side; 3) to achieve statistical multiplexing gain using the traffic bound. If traffic flows can be characterized by traffic bounds that are enforceable by the network, results from stochastic network calculus can be applied to perform admission control, which enables provisioning for end-to-end QoS guarantees. To our knowledge, such a traffic characterization is lacking in the networking literature.

Various traffic bounds have been proposed in conjunction with stochastic network calculus. The traffic arrival envelope [22], which we refer to as the A-envelope, imposes a bound on the moment generating function (MGF) of the traffic arrival process. An important feature of the A-envelope, in contrast to other traffic bounds such as the stochastically bounded burstiness (SBB) and its variants [85,99], is that significant statistical multiplexing gain can be achieved for a large number of independent flows. On the other hand, characterizing an arbitrary traffic source by an A-envelope is not straightforward since it is based on the traffic arrival process, which increases without bound. For a similar reason, the A-envelope is not amenable to traffic regulation. Thus, the A-envelope fails as a practical traffic bound with respect to the first two requirements given above.

The main contribution of this chapter is a new traffic bound, referred to as the W-envelope, which meets the three requirements for practical application of stochastic network calculus to achieve end-to-end quality-of-service. The W-envelope is a bound on the MGF of the workload process that results from offering the traffic to a constant rate server. The stationarity and ergodicity of the workload process facilitates characterization of an arbitrary traffic source in terms of the W-envelope traffic bound. In addition, a traffic regulator can be designed (see Chapter 5) to enforce conformance of a traffic flow to a given W-envelope parameter. We demonstrate through numerical examples with bursty traffic models that admission control based on W-envelopes can also achieve multiplexing gain.

7.2 Background and Motivation

We shall refer to a traffic process $A(t)$, which represents the number of arrivals in the interval $(0, t]$. The time parameter t is assumed continuous, but our results are also applicable to the discrete-time case. In this section, we discuss existing stochastic traffic bounds that are most relevant to our approach.

7.2.1 Stochastically Bounded Burstiness

In the exponentially bounded burstiness (EBB) concept of Yaron and Sidi [96], the tail distribution of the arrival process is bounded by an exponential function. The EBB was later generalized to the SBB by Starobinski and Sidi [85]:

Definition 7.2.1 (SBB). A traffic process $A(t)$ is said to be stochastically bounded bursty (SBB) with upper rate ρ and bounding function $f(\sigma) \in \mathcal{F}$ if, for all $t \geq 0$ and $0 \leq \tau < t$, and all $\sigma \geq 0$,

$$\mathsf{P}\left\{A(\tau, t) - \rho(t - \tau) \geq b\right\} \leq f(b), \tag{7.1}$$

where $A(\tau, t) := A(t) - A(\tau)$ is the amount of traffic that arrives in the interval $[\tau, t)$, $\mathcal{F}$ is defined as the family of functions such that for every $n, b \geq 0$, the n-fold integral

$(\int_b^\infty \mathrm{d}u)^n f(u)$ is bounded.

A traffic process is EBB if it is SBB with a bounding function of the form $f(b) = \alpha e^{-\theta b}$, where $\alpha, \theta \geq 0$.

The SBB was further extended in [51, 99] with the concept of *generalized* stochastically bounded burstiness (gSBB). Let

$$W_\rho(t) := \max_{0 \leq \tau \leq t} \left\{ A(\tau, t) - \rho(t - \tau) \right\}, \tag{7.2}$$

denote the virtual workload process (or workload) of a queue with input traffic $A(t)$ and constant service rate ρ.

Definition 7.2.2 (gSBB). The arrival process $A(t)$ is generalized stochastically bounded bursty (gSSB) with upper rate ρ and bounding function $f(\sigma) \in \mathcal{BF}$ if, for all $t \geq 0$ and all $\sigma \geq 0$,

$$\mathsf{P}\left\{ W_\rho(t) \geq \sigma \right\} \leq f(\sigma), \tag{7.3}$$

where $\mathcal{BF}$ is the family of positive, non-increasing functions.

Significantly, gSBB is defined in terms of the virtual workload $W_\rho(t)$, rather than the arrival process $A(t)$. If the rate ρ exceeds the average arrival rate, the constant service rate queue will be stable and in this case, $W_\rho(t)$ will be a stationary and ergodic process. This key property of the workload process was exploited in Chapter 4 to develop a method for fitting a given traffic source to a particular gSBB bound in the form of a phase-type distribution, which generalizes an exponential bounding function. Importantly, the stationarity and ergodicity of $W_\rho(t)$ was used in Chapter 5 to design a stochastic traffic regulator, which can enforce conformance of a given traffic source to an arbitrary gSBB bounding function.

Next, we characterize the superposition of gSBB traffic processes by extending [51, Theorem 4] to N traffic sources.

Theorem 7.2.1. Suppose the traffic processes $A_i(t)$, $1 \leq i \leq N$, are independent, and are gSBB with upper rate ρ_i and bounding function $f_i(\sigma)$. Then the aggregate arrival process

$A(t) = \sum_{i=1}^{N} A_i(t)$ is also gSBB with upper rate $\rho = \sum_{i=1}^{N} \rho_i$ and bounding function $g(\sigma)$ defined as follows:

$$g(\sigma) = 1 - F_1 \star F_2 \star \cdots \star F_N(\sigma), \tag{7.4}$$

$$F_i(\sigma) = 1 - f_i(\sigma), \quad i = 1, \ldots, N, \tag{7.5}$$

where $\star$ denotes Stieltjes convolution and is defined by

$$F_1 \star F_2(x) := \int_0^x F_1(x - y) \mathrm{d}F_2(y). \tag{7.6}$$

7.2.2 Moment Generating Function Traffic Envelope

As the number of traffic sources N increases, computation of the N-fold Stieltjes convolution in (7.6) becomes impractical for real-time admission control. An alternative traffic bound based on the moment generating function (MGF) of the arrival process was proposed by Chang [22].

Definition 7.2.3. The MGF traffic arrival envelope (A-envelope) of the traffic process $A(t)$ is given as follows [35]:

$$E\left[e^{\theta A(\tau,t)}\right] \leq e^{\theta[\rho(t-\tau)+\sigma]}, \tag{7.7}$$

where $\sigma \geq 0$ and

$$\rho > \lim_{t \to \infty} \frac{A(t)}{t} \tag{7.8}$$

are functions of a free parameter $\theta \geq 0$.

Equivalently, from (7.7), the A-envelope can be expressed as follows:

$$E\left[e^{\theta[A(\tau,t)-\rho(t-\tau)]}\right] \leq e^{\theta\sigma}. \tag{7.9}$$

By applying the Chernoff bound, one can show (cf. [35]) that a source with A-envelope parameterized by $(\sigma(\theta), \rho(\theta))$ is also EBB/SBB with upper rate $\rho(\theta)$ and bounding function $f(b) = e^{\theta\sigma(\theta)}e^{-\theta b}$.

7.2.3 Statistical Multiplexing and Admission Control

Suppose that a set of statistically independent and identically distributed traffic flows is offered as input to a constant service rate queue (i.e., the multiplexer) of capacity C. We consider an admission control scheme, which admits traffic flows subject to a quality-of-service (QoS) constraint:

$$P\{D > d\} < \epsilon, \tag{7.10}$$

where D represents the steady-state delay through the multiplexer, d is a delay threshold, ϵ is a small positive number.

An important feature of the A-envelope is that it leads to a relatively straightforward admission control scheme that can achieve statistical multiplexing gain for a given QoS constraint. For the SBB/gSBB families of traffic bounds, statistical multiplexing gain can be achieved in principle, but admission control requires convolution of the bounding functions, which is not scalable to large numbers of flows.

7.3 Workload-based Traffic Envelope

In this section, we develop an MGF traffic envelope based on the workload process $W_\rho(t)$ given in (7.2), which we refer to as the *W-envelope*.

7.3.1 Definition and basic properties

Definition 7.3.1. The W-envelope of traffic process is given as follows:

$$E\left[e^{\theta W_\rho(t)}\right] \leq e^{\theta\sigma(\theta)}, \tag{7.11}$$

where $W_\rho(t)$ is defined in (7.2) and $\sigma(\theta)$ is a function of the free parameter $\theta \geq 0$ such that $\sigma \geq 0$ and (7.8) holds.

The W-envelope is more suitable for traffic characterization and traffic regulation than the A-envelope because it is based on $W_\rho(t)$, which in practice will have a steady-state distribution, as opposed to $A(t)$.

Theorem 7.3.1. Suppose a traffic process A has a W-envelope with parameter (σ, ρ) as in (7.11). Then it is also characterized by an A-envelope with parameter (σ, ρ) as in (7.9).

$$E\left[e^{\theta[A(\tau,t)-\rho(t-\tau)]}\right] \leq e^{\theta\sigma}. \tag{7.12}$$

Proof. For $0 \leq \tau \leq t$,

$$A(\tau, t)-\rho(t-\tau) \leq \max_{0\leq\tau\leq t}[A(\tau, t)-\rho(t-\tau)]=W_\rho(t). \tag{7.13}$$

Therefore,

$$E[e^{\theta[A(\tau,t)-\rho(t-\tau)]}] \leq E[e^{\theta W_\rho(t)}], \tag{7.14}$$

and the result follows immediately. □

Theorem 7.3.1 implies that a traffic regulator that enforces a W-envelope with parameter (σ, ρ) also enforces an A-envelope with the same parameter. Consider N traffic processes $A_1(t), \ldots, A_N(t)$. Assume each process $A_i(t)$ is characterized by a W-envelope of the form (7.11) specified by parameters (σ_i, ρ_i), i.e.,

$$E\left[e^{\theta W_{\rho_i}(t)}\right] \leq e^{\theta[\rho_i(t-\tau)+\sigma_i]}, \quad i = 1, \ldots, N. \tag{7.15}$$

Let

$$A(t) = \sum_{i=1}^{N} A_i(t). \tag{7.16}$$

denote the aggregate traffic process. We will make use of the following inequality.

Lemma 7.3.1.

$$W_\rho(t) \le \sum_{i=1}^{N} W_{\rho_i}(t). \tag{7.17}$$

Proof.

$$W_\rho(t) = \max_{0\le\tau\le t}[A(\tau,t) - \rho(t-\tau)]. \tag{7.18}$$

Let

$$\tau^* = \text{argmax}_{0\le\tau\le t}[A(\tau,t) - \rho(t-\tau)], \tag{7.19}$$

such that

$$W_\rho(t) = A(\tau^*,t) - \rho(t-\tau^*). \tag{7.20}$$

Then, clearly,

$$A_i(\tau^*,t) - \rho_i(t-\tau^*) \le \max_{0\le\tau\le t}[A_i(\tau,t) - \rho(t-\tau)], \tag{7.21}$$

for $i = 1, \dots, N$, Summing both sides of (7.21) for $i = 1, \dots, N$, we obtain

$$A(\tau^*,t) - \rho(t-\tau^*) \le \sum_{i=1}^{N} \max_{0\le\tau\le t}[A_i(\tau,t) - \rho_i(t-\tau)]. \tag{7.22}$$

Applying (7.20) and the definition of $W_{\rho_i}(t)$ in the above inequality, we obtain

$$W_\rho(t) \leq \sum_{i=1}^{N} W_{\rho_i}(t). \tag{7.23}$$

$\square$

Theorem 7.3.2. Let $W_\rho(t)$ denote the workload at a multiplexer with constant service rate ρ with aggregate input traffic $A(t)$ given by (7.16). Each of the traffic processes $A_i(t)$ has a W-envelope given by (7.15). Let $\sigma = \sum_{i=1}^{N} \sigma_i$ and $\rho = \sum_{i=1}^{N} \rho_i$. Then, the traffic process A has a W-envelope of the form (7.11) with parameter (σ, ρ).

Proof of Theorem 7.3.2. We have

$$E\left[e^{\theta W_\rho(t)}\right] = E\left[e^{\theta \max_{0\leq\tau\leq t}[A(\tau,t)-\rho(t-\tau)]}\right]. \tag{7.24}$$

Applying lemma 7.3.1, we have

$$E\left[e^{\theta W_\rho(t)}\right] \leq E\left[e^{\theta \sum_{i=1}^{N} W_{\rho_i}(t)}\right] = E\left[\prod_{i=1}^{N} e^{\theta W_{\rho_i}(t)}\right]. \tag{7.25}$$

Next, we use induction to show that

$$E\left[\prod_{i=1}^{N} e^{\theta W_{\rho_i}(t)}\right] \leq \prod_{i=1}^{N} e^{\theta\sigma_i} = e^{\theta \sum_{i=1}^{N} \sigma_i}. \tag{7.26}$$

Note that (7.26) holds trivially when $N = 1$. Suppose that (7.26) holds for $N = k \geq 1$. Let

$$Y(t) = \sum_{i=1}^{k} W_{\rho_{i}(t)}. \tag{7.27}$$

Applying the Cauchy-Schwarz inequality for random variables, we have

$$E\left[e^{\theta Y(t)} \cdot e^{\theta W_{\rho_{k+1}}(t)}\right] \leq \left(E\left[e^{2\theta Y(t)}\right]\right)^{\frac{1}{2}} \cdot \left(E\left[e^{2\theta W_{\rho_{k+1}}(t)}\right]\right)^{\frac{1}{2}}. \tag{7.28}$$

We have

$$E\left[e^{\theta W_{\rho_{k+1}}(t)}\right] \leq e^{\theta\sigma_{k+1}} \tag{7.29}$$

for all $\theta > 0$. By the induction hypothesis.

$$E\left[e^{\theta Y(t)}\right] = E\left[\prod_{i=1}^{k} e^{\theta W_{\rho_i}(t)}\right] \leq e^{\theta \sum_{i=1}^{k} \sigma_i}. \tag{7.30}$$

Applying (7.29) and (7.30) into (7.28), we obtain

$$E\left[\prod_{i=1}^{k+1} e^{\theta W_{\rho_i}(t)}\right] \leq e^{\theta \sum_{i=1}^{k} \sigma_i} \cdot e^{\theta\sigma_{k+1}} = e^{\theta \sum_{i=1}^{k+1} \sigma_i}. \tag{7.31}$$

This establishes (7.26) by the principle of induction. Combining (7.25) and (7.26), we obtain (7.11). $\square$

Applying Theorem 7.3.2, we see that the A-envelope of the aggregate process $A(t)$ satisfies

$$E\left[e^{\theta[A(\tau,t)-\rho(t-\tau)]}\right] \leq E\left[e^{\theta W_\rho(t)}\right] \leq e^{\theta\sigma}. \tag{7.32}$$

Thus, a traffic regulator that enforces the individual W-envelope of the traffic stream $A_i(t)$ according to the parameter (σ_i, ρ_i), for $i = 1, \ldots, N$, enforces both the A-envelope and W-envelope of the aggregate process A according to the parameter (σ, ρ), where $\sigma = \sum_{i=1}^{N} \sigma_i$ and $\rho = \sum_{i=1}^{N} \rho_i$.

7.4 Framework for Providing Delay Guarantees

We outline a framework for providing stochastic delay guarantees using the W-envelope consisting of three components: traffic characterization, traffic regulation, and admission control.

7.4.1 Traffic Characterization

Consider a traffic process A that is gSBB with upper rate ρ and bounding function $f(\sigma)$. The following theorem[1], provides a characterization of the W-envelope of A.

Theorem 7.4.1. Let traffic process A be gSBB with upper rate $\rho > 0$ and bounding function $f(\sigma)$ and assume that the workload process $W_\rho(t)$ is upper bounded by $\sigma_{\max}$. Then A has W-envelope $(\sigma(\theta), \rho)$ given by

$$\sigma(\theta) = \frac{1}{\theta} \ln\left[1 + \theta C(\theta)\right], \tag{7.33}$$

where,

$$C(\theta) = \int_0^{\sigma_{\max}} f(y) e^{\theta y} \mathrm{d}y, \tag{7.34}$$

and $\theta \geq 0$ is a free parameter.

To prove this theorem we use the following lemma.

Lemma 7.4.1. Let X be a nonnegative random variable taking values in the interval $[a, b]$, where $0 \leq a < b < \infty$. Then

$$E[X] = a + \int_a^b P\{X > x\}\ dx. \tag{7.35}$$

[1]The proof is omitted due to space constraints.

Proof. The expectation of a nonnegative random variable X can be computed as follows (e.g., [55, Eq. (4.11)]):

$$E[X] = \int_0^\infty P\{X > x\}\, dx. \tag{7.36}$$

By partitioning the integral into three intervals $[0, a)$, $[a, b)$, and $[b, \infty)$, (7.35) follows readily from (7.36). □

Proof of Theorem 7.4.1. Let $X = e^{\theta W_\rho(t)}$. Then

$$P\{W_\rho(t) = 0\} = P\{X = 1\} > 0$$

and X satisfies the conditions of Lemma 7.4.1 with $a = 1$ and $b = e^{\theta \sigma_{\max}}$. Using Definition 2.1.5,

$$P\left\{e^{\theta W_\rho(t)} > x\right\} = P\left\{W_\rho(t) > \frac{1}{\theta} \ln x\right\} \leq f\left(\frac{1}{\theta} \ln x\right). \tag{7.37}$$

Applying (7.37) in (7.35), we have

$$\begin{aligned} E\left[e^{\theta W_\rho(t)}\right] &= 1 + \int_1^b P\left\{e^{\theta W_\rho(t)} > x\right\}\, dx \\ &\leq 1 + \int_1^b f\left(\frac{1}{\theta} \ln x\right)\, dx \\ &= 1 + \theta \int_0^{\sigma_{\max}} f(y) e^{\theta y}\, dy = 1 + \theta C(\theta). \end{aligned} \tag{7.38}$$

Using (7.38) we can derive $\sigma(\theta)$ as given by (7.33). □

In Chapter 4, an efficient method for fitting an arbitrary traffic source to a gSBB is developed. Given a gSBB characterization of a traffic process, a W-envelope characterization can then be obtained using Theorem 7.4.1.

7.4.2 Traffic Regulation

Using the gSBB traffic regulator developed in Chapter 5, and Theorem 7.4.1, the network can regulate an arbitrary traffic process A such that it conforms to a negotiated W-envelope parameter (σ_W, ρ_W), as defined in (7.11), where σ_W is a functions of the free parameter $\theta \geq 0$. Through the relationship between gSBB bounding function, $f(\sigma)$ and the desired $\sigma_W(\theta)$ obtained using Theorem 7.4.1, we find the required $f(\sigma)$ for the gSBB characterization. Then, using the gSBB traffic regulator in Chapter 5, the traffic flow can be forced to conform to a gSBB bound with bounding function $f(\sigma)$. The desired ρ_W will be used in the gSBB traffic regulator, which will force the traffic flow to conform to the negotiated W-envelope parameter (σ_W, ρ_W).

7.4.3 Admission Control

Consider a set of statistically independent and identically distributed traffic flows is offered as input to a constant service rate queue (i.e., the multiplexer) of capacity C. We consider an admission control scheme, which admits new traffic flows subject to the following quality-of-service (QoS) constraint:

$$P\{D > d\} < \epsilon. \tag{7.39}$$

Suppose a set of N independent traffic flows is characterized by W-envelope parameters $(\sigma_{W_i}, \rho_{W_i})$, $i = 1, \ldots, N$, such that $\rho_W := \sum_{i=1}^{N} \rho_{W_i} < C$. Almost surely,

$$Q(t) = W_C(t) \leq W_{\rho_W}(t), \tag{7.40}$$

where $Q(t)$ denotes the queue size at the multiplexer. For a FCFS server, $D(t) = Q(t)/C$; hence,

$$\mathsf{P}\{D(t) \geq d\} \leq \mathsf{P}\{W_{\rho_W}(t) \geq d\ C\} \leq e^{\theta(\sigma_W - Cd)}, \tag{7.41}$$

where the last inequality follows from the Markov inequality. In (7.41), the free parameter θ can be optimized to obtain the best upper bound on $\mathsf{P}\{D > d\}$. Network calculus results (cf. [35]) can be used to extend (7.41) to an end-to-end delay guarantee for flows traversing multi-hop paths.

7.5 Workload Envelope for Two Traffic Models

We now consider the workload envelopes for Markov fluid models and Markov modulated Poisson Processes (MMPPs).

7.5.1 Markov Fluid Model

We consider an Markov on-off fluid model, which consists of an underlying Markov chain with two states: 0 (*off*) and 1 (*on*) [6]. In the *on* state, the source generates fluid at a constant rate of one unit of information per unit time, while in the *off* state, no fluid is generated. The sojourn time in each *on* state is exponentially distributed with mean one, while that in each *off* state is exponentially distributed with mean λ^{-1}.

An A-envelope for a Markov on-off fluid can be obtained from the *minimum envelope rate* defined in [22]. We shall use the A-envelope given by $(\sigma_A = 0, \rho = a^*(\theta))$, where $a^*(\theta)$ is the minimum envelope rate given by [22, Eq. (46)] (with $\mu = \nu = 1$):

$$a^*(\theta) = \left[\theta - 1 - \lambda + \sqrt{(\theta - 1 + \lambda)^2 + 4\lambda}\right] / 2\theta. \tag{7.42}$$

We now derive the W-envelope for a Markov on-off fluid source. Applying [6, Eq. (46)] with $C = \rho_W < 1$ and $N = 1$,

$$\mathsf{P}\{B > x\} = -a_0(\mathbf{1}'\boldsymbol{\phi}_0)e^{z_0 x}, \tag{7.43}$$

where $\mathbf{1}$ is a column vector of all ones, $'$ denotes transpose,

$$z_0 = \frac{\lambda(1-\rho_W)-\rho_W}{\rho_W(1-\rho_W)}, \ \boldsymbol{\phi}_0 = \left[\frac{1-\rho_W}{\rho_W}, 1\right]', \ a_0 = \frac{-\lambda}{1+\lambda}. \tag{7.44}$$

Note that the traffic utilization for a single Markov on-off fluid source fed to a constant rate server with rate ρ_W is $U := p_{\mathrm{on}}/\rho_W = \lambda/(\rho_W(1+\lambda))$. The traffic utilization should be less than 1; otherwise, the workload process will grow without bound. We also pick $\rho_W < 1$ since otherwise, the workload would be almost surely 0. Therefore, $\frac{\lambda}{1+\lambda} < \rho_W < 1$. Applying (7.43) and (7.44), for a single source fed into a constant rate server with rate ρ_W, we get

$$\mathsf{P}\{B > x\} = \frac{\lambda}{\rho_W(1+\lambda)} e^{z_0 x}. \tag{7.45}$$

Hence,

$$E\left[e^{\theta W_\rho(t)}\right] = 1 - \frac{\lambda}{\rho_W(1+\lambda)} + \frac{\lambda z_0}{\rho_W(1+\lambda)(\theta+z_0)}, \tag{7.46}$$

for $\theta \in (0, -z_0)$. Then σ_W can be derived by equating $e^{\theta\sigma_W}$ and $E\left[e^{\theta W_\rho(t)}\right]$, which yields

$$\sigma_W(\theta) = \frac{1}{\theta} \log E\left[e^{\theta W_\rho(t)}\right]. \tag{7.47}$$

7.5.2 MMPP Traffic Model

We next consider a model of bursty traffic generated by a three-state MMPP with parameter values as in [27, p. 79], which are derived from matching the arrival process of I, P and B frames in an MPEG-4 encoded video stream to the three states of the MMPP. The packet sizes are modeled according to a special phase-type distribution referred to as "G3" in [27, Table 1] with probability density function (pdf)

$$f_L(l) = p \ \mathrm{Er}(r_1, 1/\mu_1) + (1-p) \ \mathrm{Er}(r_2, 1/\mu_2), \tag{7.48}$$

where $\mathrm{Er}(r, 1/\mu)$ denotes the pdf of an $r-$stage Erlang distribution with mean r/μ. This phase-type distribution is a mixture of Erlangs, which closely approximates the empirical distribution of measured Internet packet sizes obtained in [37]. The Laplace transform (LT) of the packet length pdf is

$$\tilde{H}(s) = p\left(1 + \frac{s}{\mu_1}\right)^{-r_1} + (1-p)\left(1 + \frac{s}{\mu_2}\right)^{-r_2}. \tag{7.49}$$

Suppose N independent MMPP traffic sources are offered as input to a multiplexer with capacity C. Let $\mathbf{\Lambda}_i$ and $\mathbf{R}_i$ denote the rate matrix and generator matrix of the ith source, $i = 1, \ldots, N$. Then the input traffic is an MMPP with arrival matrix and rate matrix given by [36]

$$\mathbf{\Lambda} = \mathbf{\Lambda}_1 \oplus \ldots \oplus \mathbf{\Lambda}_N \text{ and } \mathbf{R} = \mathbf{R}_1 \oplus \ldots \oplus \mathbf{R}_N,$$

respectively, where $\oplus$ represents the Kronecker sum. When the service times are independent and generally distributed, the resulting queue is denoted by MMPP/G/1.

The distribution of the virtual waiting time $V(t)$ in steady-state can be obtained from results in [69, 82] for a MAP/G/1 queue, since an MMPP is a special case of a Markovian Arrival Process (MAP). Assuming that the service rate of the queue is normalized to one, the LT of the steady-state virtual waiting time pdf of an MMPP/G/1 queue is given by [82]

$$\tilde{V}(s) = s(1-\gamma)\mathbf{g}[sI + \mathbf{D}(\tilde{H}(s))]^{-1}\mathbf{1}, \tag{7.50}$$

where $\gamma = \lambda_{\text{avg}}/\mu_{\text{avg}}$ is the utilization of the queue, λ_{avg} is the average packet arrival rate, μ_{avg}^{-1} is the mean packet length, and $\tilde{H}(s)$ is given in (7.49). The matrix function $\mathbf{D}(z)$

is given by $\mathbf{D}(z) = \mathbf{D}_0 + \mathbf{D}_1 z$, where $\mathbf{D}_0 = \mathbf{R} - \mathbf{\Lambda}$ and $\mathbf{D}_1 = \mathbf{\Lambda}$. The row vector $\mathbf{g}$ is the invariant probability vector associated with the stochastic matrix $\mathbf{G}$ defined in [69, Eq. (22)], i.e., $\mathbf{g}$ is the solution to

$$\mathbf{g}\mathbf{G} = \mathbf{g}, \quad \mathbf{g}\mathbf{1} = 1. \tag{7.51}$$

By relating the virtual waiting time for the MAP/G/1 system considered in [69] to the workload of the desired system, with server rate C, we obtain the LT of the workload pdf as follows:

$$\tilde{W}(s) = sC(1 - \gamma/C)\mathbf{g}[sCI + \mathbf{D}(\tilde{H}(s))]^{-1}\mathbf{1}. \tag{7.52}$$

Using (7.52), with $C = \rho_W$ and the MMPP parameter, we can derive the W-envelope parameter σ_W as follows:

$$\sigma_W(\theta) = \frac{1}{\theta} \log E\left[e^{\theta W_{\rho_W}(t)}\right] = \frac{1}{\theta} \log \tilde{W}(-\theta). \tag{7.53}$$

7.6 Numerical Examples

Using the analytical results from Section 7.5, we numerically investigate the performance of the W-envelope for particular Markov fluid and MMPP traffic sources.

Markov Fluid Model

Let M denote the maximum number of Markov on-off fluid sources that can be supported at a multiplexer with capacity C. The statistical multiplexing gain can be quantified by ratio $g = M/C$. Under *peak rate* allocation, the number of sources that can be supported is $M_p = C$ and in this case, the multiplexing gain is $g_p = 1$. Under *mean rate* allocation, the workload at the multiplexer will grow without bound, but the number $M_m = C/p_{\text{on}} = C(1+\lambda)/\lambda$ provides an upper bound on the number of sources that can be supported under

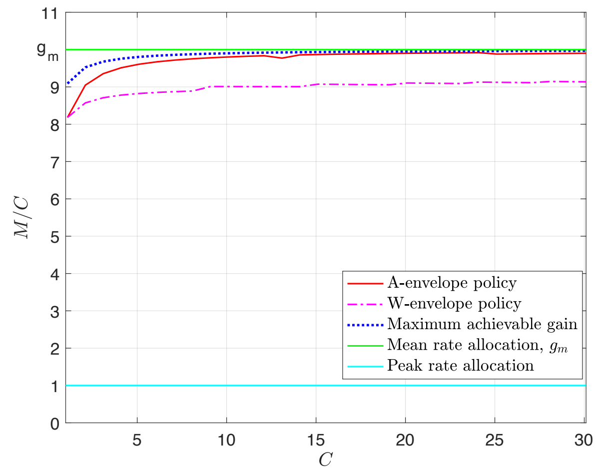


Figure 7.1: Statistical multiplexing gain vs. C using A-envelope and W-envelope: $p_{\text{on}} = 0.1$, $d = 100$, $\epsilon = 10^{-3}$.

any admission control policy, and we define $g_m := M_m/C = (1+\lambda)/\lambda$. In general,

$$g \in [g_p, g_m) = \left[1,\ \frac{1+\lambda}{\lambda}\right). \tag{7.54}$$

Fig. 7.1 shows the statistical multiplexing gain under the A-envelope and W-envelope admission control policies when the QoS constraint (7.39) is specified by $\epsilon = 10^{-3}$ and $d = 100$. The duty cycle of each on-off source is given by $p_{\text{on}} = 0.1$. The achievable gain curve is derived from the result in [6, Eq. (46)]. As shown in Fig. 7.1, both policies achieve high levels of multiplexing gain even for moderate values of the capacity C.

MMPP Bursty Traffic Model

Under a mean rate allocation scheme, for this traffic model, the number $M_m = C/\gamma = C\mu_{\text{avg}}/\lambda_{\text{avg}}$ provides an upper bound on the number of sources that can be supported under any admission control policy, and we define $g_m := M_m/C = \mu_{\text{avg}}/\lambda_{\text{avg}}$. The gain g of a general policy satisfies $g \le g_m$.

To model a bursty traffic source, the MMPP parameter values are chosen according

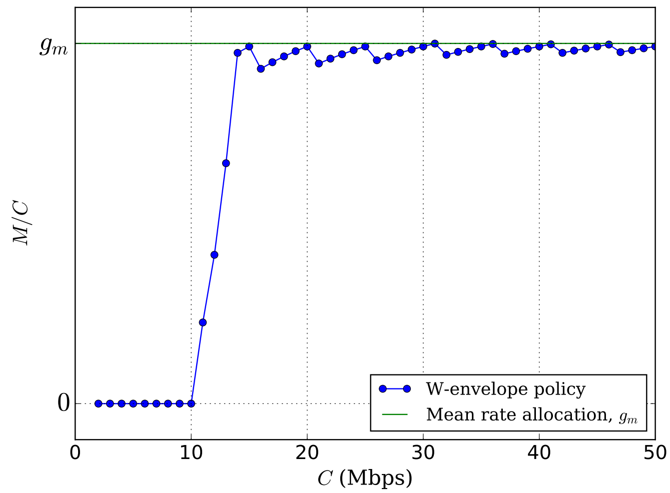


Figure 7.2: Statistical multiplexing gain vs. C using W-envelope for MMPP bursty traffic: $d = 4$ ms, $\epsilon = 10^{-3}$.

to [27, p. 79], with arrival matrix $\mathbf{\Lambda} = \text{diag}\{116, 274, 931\}$ in units of [packets/s] and rate matrix

$$\mathbf{R} = \begin{bmatrix} -0.12594 & 0.12594 & 0 \\ 0.25 & -2.22 & 1.97 \\ 0 & 2 & -2 \end{bmatrix} \tag{7.55}$$

in units of [s^{-1}]. The packet length parameters in (7.48) are set as $p = 0.54$, $r_1 = r_2 = 5$, $\mu_1^{-1} = 5.2$ bytes, $\mu_2^{-1} = 191.2$ bytes. This Erlang mixture distribution closely approximates the empirical distribution of measured Internet packet sizes obtained in [37]. The average packet length is 454 bytes, which yields an average bit rate of 1.24 Mbps.

Fig. 7.2 shows the multiplexing gain under the W-envelope policy when the QoS constraint (7.39) is specified by $\epsilon = 10^{-3}$ and $d = 4$ ms. Observe that the W-envelope policy achieves a gain close to the upper bound g_m even for moderate values of C. Although the gain g can decrease slightly with increasing C, the number of admitted flows is always monotonically increasing.

7.7 Conclusion

Motivated by the desire to provide performance guarantees for time-critical applications in future networks, we proposed the W-envelope, a traffic bound on the MGF of the workload process resulting from offering a traffic source to a constant rate server. In contrast to the MGF traffic arrival envelope (A-envelope), the W-envelope is amenable to traffic regulation (Chapter 5), and traffic fitting (Chapter 4). Our numerical results showed that the W-envelope policy for admission control leads to significant statistical multiplexing gain.

We showed that the W-envelope framework can provide end-to-end network performance guarantees in a resource-efficient manner. Although our numerical results were based on analytical W-envelope expressions for Markov fluid and MMPP traffic models, an empirical W-envelope can be obtained for an arbitrary traffic trace using fitting methods along the lines of Chapter 4. We also remark that by leveraging network softwarization and virtualization, a service for time-critical applications requiring delay guarantees can be realized within a network slice using the proposed framework.

Chapter 8: Available Bandwidth Estimation in the Presence of Lost Packets

Accurate available bandwidth estimation (ABE) along an end-to-end path in a network is crucial for a wide range of applications including traffic engineering, multimedia streaming, and path selection in software-defined wide area networks. Many existing ABE methods rely on active measurements of end-to-end packet delay using probe packets. In general, ABE performance degrades severely when packet losses occur, since they can adversely affect the probing process. Such packet losses may arise, for example, as a result of queue overflow at a bottleneck link of the path. Most ABE methods assume that no packet losses occur or simply discard entire trains of probe packets if they contain lost packets. The most recent class of ABE methods, decreasing-rate methods, are more susceptible to packet loss as they perform measurement throughout the period of high congestion. We propose a set of techniques to extend modern ABE methods such as PathCos++, SLDRT or Voyager-D. These techniques significantly improves estimation accuracy in the presence of packet loss. The proposed techniques estimates the amount of probe traffic and cross traffic dropped at the bottleneck to correct the original estimate. It also uses an estimate of the bottleneck link capacity, but is relatively insensitive to the accuracy of this estimate. Our experimental results show that even with an inaccurate estimate of the bottleneck capacity, our approach achieves satisfactory available bandwidth estimates.

The remainder of this chapter is organized as follows. In Section 8.1, we provide an introduction and motivation about available bandwidth estimation problem. In Section 8.2, we discuss the relevant background and related work in the ABE space. Our ABE formulation to account for packet losses is developed in Section 8.3. In Section 8.4, we describe the experimental setup of our network testbed and present numerical results. The chapter is concluded in Section 8.5.

8.1 Introduction

Accurate and efficient estimation of the available bandwidth (ABW) along an end-to-end network is important for achieving high performance and reliability for delay sensitive, bandwidth hungry Internet applications such as multimedia streaming, network gaming, and virtual/augmented reality. Over the past couple of decades, various ABW estimation (ABE) methods have been proposed and studied in the literature [41]. Most ABE schemes apply active measurement techniques using trains of packet probes, also known as *chirps*. Chirps induce congestion at bottlenecks along a given path, and the ABW is inferred from one-way delay measurements of the packet probes.

Since ABE methods rely on delay measurements to infer ABW, most of them ignore the impact of packet loss by simply discarding chirps containing lost probe packets [65]. Such packet loss can occur when bursty cross traffic fills queues at bottlenecks along a path. Further, the probe packets themselves may fill the queues when queues are shallow and chirps are long. Discarding a chirp in response to probe packet losses can be wasteful, as the remainder of a chirp may still carry information that can be useful for deriving an ABW estimate. Another source of packet probe loss is modern active queue management (AQM) techniques, which purposefully induce packet drops to mitigate the impact of congestion at network queues. Thus, AQM can also impact the performance of ABE methods by dropping probe packets. Most ABE methods are not formulated to handle packet losses and consequently implementations prevent ABW estimation when packet losses happen in the chirp. Therefore, it is usually not possible to perform ABE on paths that have packet losses, such as paths with small queues or aggressive AQMs.

More recent ABE methods, such as PathCos++ [65], SLDRT [46] and Voyager-D [66] employ chirps that consist of probe packets sent initially at a fast rate to induce congestion and then probes sent with decreasing rate. redAs a result, the first part of the chirp train induces a build-up at the bottleneck queues and the tail part of the chirp, with lower probe rate, allows the queues to deplete and the congestion to diminish. The ABW estimate is obtained from measurements taken during the congestion period. This estimation is done by looking at two points in the chirp with similar experienced congestion, one during the first part of the chirp with high rate and the second one during the tail part of the chirp with low rate, and computing the rate of probe packets between these two points. These so-called *decreasing-rate* methods have been shown to be significantly more accurate than earlier ABE schemes [66]. At the same time, however, the decreasing rate methods are more sensitive to packet drops and their performance advantage can be greatly compromised in the presence of packet loss (see Section 8.4.4).

In this chapter, we study the impact of packet loss on decreasing rate ABE methods such as PathCos++, SLDRT and VOyager-D and propose a set of techniques to address this issue. Our approach involves two steps. First, the formulation of the estimate is changed to use the number of received packets, instead of the number of sent packets, to take into account probe packets lost at the bottleneck. Second, we compute an estimate of the amount of cross traffic dropped at the bottleneck of given path, and use it to adjust the ABW estimate obtained from the decreasing ABE methods. This second step relies on an estimate of the path capacity. We present experimental results obtained on a network testbed, which show a significant improvement in accuracy of the proposed extension to decreasing-rate methods in the presence of packet loss. Moreover, the ABW estimates obtained by our method are relatively insensitive to the accuracy of the path capacity estimate.

8.2 Related Work and Motivation

In this section, we summarize related work most relevant to our proposed ABE approach and motivate our work.

8.2.1 Available bandwidth estimation

The **available bandwidth (ABW)** of an individual link is its unused capacity, i.e., the difference between its capacity and the current amount of traffic using it [81]. The ABW of a network path is the smallest ABW across its links. PathCos++ [65], SLDRT [46] and Voyager-D [66] are examples of a class of methods to estimate ABW, which are referred to as **decreasing-rate methods**. They offer much better accuracy than traditional ABW estimation (ABE) methods [65], especially in the presence of bursty cross traffic and interrupt coalescence [66, 89].

These methods use active probing, i.e., they send dedicated probe packets to measure the network path. They are based on the classic Probe Rate Model, which uses the concept of *self-induced congestion* [86]. If the rate of probe packets is below the path ABW, the probe packets experience no queuing delay, whereas if it exceeds the ABW, congestion is created, and probe packets are queued at the tight link and experience an increase in their one-way delay (OWD). The various ABE methods mostly differ in the construction of the chirp train, the sequence of probe packets that is sent, and how the packet OWD values are processed to obtain an ABW estimate.

PathCos++ [65] was the first method to propose using a chirp train with decreasing probe rates, by increasing the time between packets. The goal of such a chirp train is to first congest and then decongest the path, creating a congestion peak. Congestion manifests itself as a build-up at bottleneck queues. The intuition is that two packets with similar OWDs usually experience similar congestion, and therefore the probe traffic between those packets should be congestion-neutral and representative of the ABW. PathCos++ tries to find the widest spaced pair of packets that are on both sides of the congestion bump and with similar OWD, and then computes the sending rate of probe packets between these two packets as the ABW estimate (see Fig. 8.3).

SLDRT [46] also uses a chirp train with a decreasing rate. It searches the point at which the path becomes decongested, by picking the first packet for which the OWD returns to its minimum, and then uses the rate of the chirp train up to that point as the ABW estimate. The main differences with PathCos++ is the exponentially decreasing rate of the chirp train and the fact that ABE estimates are always done from the first packet of the chirp train.

Voyager-D [66] is derived from PathCos++. Voyager-D introduces a noise threshold based on the measured OWD noise, and modifies the pair selection to prefer probe pairs which are above the noise threshold. Voyager-D also adjusts the leading and trailing probes to find a pair with less difference in OWD. The chirp train of Voyager-D is designed for systems with rate adaptation, with reduced density at the edge of the rate window, and has an exponential decrease like SLDRT, rather than the linear decrease of PathCos++.

8.2.2 TCP and congestion losses

While most ABE methods use packet delay, most TCP congestion algorithms use packet losses [42], which is usually a more reliable measure of congestion. Since all queues in the network devices have finite buffer capacity, with enough congestion a given queue can become full. When a queue is full, *tail-drop losses* occur, i.e., packets that arrive to a full queue are discarded. TCP was designed to take advantage of losses to infer congestion and regulate its sending rate [42].

Relying on tail-drop losses to perform congestion control has the disadvantage that the bottleneck queue must operate in a nearly full condition, which adds latency and degrades TCP performance. This phenomenon is called BufferBloat [40]. Active Queue Management (AQM) [28] solves this by triggering packet losses or ECN (Early Congestion Notification) signals at early onset of congestion, before the queue builds up. When AQM can not be used, the queue size may be reduced [45] to minimize latency.

TCP congestion control depends so much on congestion losses that, for good performance, link layers must present to TCP a nearly lossless service. Transmission losses at the link layer may be interpreted by TCP as congestion loss and may induce it to lower its rate, hindering performance. As a consequence, link layers include various mechanisms to minimize transmission losses, such as powerful and complex retransmission mechanisms (ARQ) [32]. Thanks to those mechanisms, the probability that a packet loss is not related to congestion is very small in actual deployments. We shall assume that nearly all packets losses are due to congestion.

8.2.3 ABW estimation and packet losses

Many ABE methods assume that no probe packet is lost during measurement. If a chirp train contains lost packets, the entire chirp must be discarded and no estimate is generated [84]. If the congestion created by the ABE cause packet losses at the bottleneck, then every chirp train needs to be discarded and ABE can never by done on that network path. PathLoad [48] assumes packet losses are due to congestion, and reacts to losses by reducing the maximum probing rate, although packet losses are not used in the computation of the final ABW estimate. NEXT-V2 [79] assumes losses are due to transmission errors at the link layer, and reacts to losses by interpolating the missing packets, however NEXT-V2 was not designed to handle congestion losses.

Packet losses are especially problematic for decreasing-rate methods. Those methods need to generate enough congestion in every chirp train to create a delay bump, making it more likely that every chirp suffers from congestion losses. Further, they need to measure packets throughout the full congestion period [65], where congestion losses are clustered (see Fig. 8.3), so this measurement process will be impacted by congestion losses (Section 8.4.4). Neither PathCos++ nor SLDRT nor Voyager-D are formulated to handle packet losses [46, 65, 66]. The accuracy of those methods is directly related to the chirp train length [46], increasing the chirp train length provides improved ABE accuracy in most cases and can overcome network noises [66]. However, longer chirp trains are more likely to fill the queues and cause tail-drop or AQM losses, which prevents accurate ABW estimation. This effectively puts a limit on the accuracy that can be achieved via those methods, and motivates the need to handle congestion-induced packet losses, so that this limit on chirp train length can be removed.

8.3 ABE with Lost packets

In this section, we extend the formulation of PathCos++ [65] to account for the potential loss of probe packets. The formulation of SLDRT [46] is effectively a simplified version of

PathCos++. Voyager-D [66] is based on PathCos++ and use the same formulation. All decreasing-rate methods rely on measuring the rate of probe packets over a subset of the chirp train and therefore can be extended to compensate for potential packet losses using our extended formulation. Our algorithm is independent of other features of decreasing methods, such as the construction of the chirp train and how the two points to measure the rate are chosen. Most additional noise processing techniques, such as mitigation of interrupt coalescence [89,98], modify the received chirp train prior to application of the ABE method. Our algorithm operates on the modified chirp train and is therefore also compatible with such noise pre-processing schemes.

8.3.1 Service time for cross traffic

When a chirp is sent, some packets may be dropped due to queue overflow at any hop of the path. Such queue overflow is more likely to happen at a bottleneck link. In existing ABE algorithms, chirps containing lost packets are ignored and not used in the estimation of the available bandwidth. As stated in [65], end-to-end available bandwidth is defined as the minimal residual capacity of the links along a path within a certain time interval. We shall assume only a single bottleneck link on the path that drops the packets due to congestion. As shown in Section 8.4, our algorithm estimates the ABW accurately in scenarios with multiple bottlenecks, among which just one bottleneck drops the packets. Cases involving multiple bottlenecks that drop packets due to congestion at different stages of the chirp train, however, are not considered in this chapter.

Consider an end-to-end path consisting of N links, denoted as $L_1, L_2, \ldots, L_j, \ldots, L_N$, where L_j denotes the single bottleneck link that drops the packets due to congestion. From now on, when we refer to the bottleneck link in our formulation, we mean link L_j. The capacity of the ith link L_i is C_i in bits. Probe packets $1, 2, \ldots, M$ are sent at times $t_1, t_2, \ldots, t_M$ and are received at times $t'_1, t'_2, \ldots, t'_M$. Let q^i_k be the queuing delay of packet k at the ith link, and let d_i be the propagation delay of the ith link, and s be the probe packet size in bits. Then the kth packet is received at time

$$t'_k = t_k + \sum_{i=1, i \neq j}^{N} \left(q_k^i + \frac{s}{C_i} + d_i \right) + \left(q_k^j + \frac{s}{C_j} + d_j \right), \tag{8.1}$$

Let T_k be the time when the kth probe packet arrives at the bottleneck link. Then,

$$T_k = t_k + \sum_{i=1}^{j-1} \left(q_k^i + \frac{s}{C_i} + d_i \right). \tag{8.2}$$

Let $W(T_k, T_{k+1})$ be the service time spent at the bottleneck link L_j on cross traffic during time interval $[T_k, T_{k+1}]$, assuming no cross traffic packet is dropped during this interval. Then,

$$W(T_k, T_{k+1}) = \sum_{i=1}^{V(T_k, T_{k+1})} \frac{cs_i}{C_j}, \tag{8.3}$$

where $V(T_k, T_{k+1})$ is the number of received cross traffic packets during interval $[T_k, T_{k+1}]$, and cs_i is the size of the ith cross traffic packet in bits. A busy period is defined in [65] as an interval of time during which the queue at the bottleneck does not become idle. If probe packets k and $k+1$ are within the same busy period and neither of them is dropped as they traverse the end-to-end path, the queuing delay of the $(k+1)$st packet can be expressed as follows:

$$q_{k+1}^j = q_k^j + \frac{s}{C_j} + W(T_k, T_{k+1}) - (T_{k+1} - T_k). \tag{8.4}$$

On the other hand, if probe packets l and m are two consecutively received packets that have traversed the end-to-end path and all the probe packets $l+1, l+2, \ldots, m-1$ are dropped, then

$$q_m^j = q_l^j + \frac{s}{C_j} + W^*(T_l, T_m) - (T_m - T_l), \tag{8.5}$$

where $W^*(T_l, T_m)$ is the service time spent on the portion of the cross traffic that was not dropped during interval $[T_k, T_{k+1}]$ at link L_j. Therefore,

$$W^*(T_l, T_m) = \sum_{i=1}^{V^*(T_l,T_m)} \frac{cs_i}{C_j}, \tag{8.6}$$

where $V^*(T_l, T_m)$ is the number of received cross traffic packets which are not dropped during interval $[T_l, T_m]$, and cs_i is the size of the ith cross traffic packet.

For two consecutive packets k and $k+1$, that are not dropped as they traverse the end-to-end path, Q_k^b denotes the difference between the corresponding queuing delay in the links after the bottleneck. Therefore,

$$Q_k^b = \sum_{i=j+1}^{N} (q_{k+1}^i - q_k^i). \tag{8.7}$$

Let owd_k denote the one-way delay of the kth probe. From (8.1)-(8.4) and (8.7), we have

$$owd_{k+1} - owd_k = \frac{s}{C_j} + W(T_k, T_{k+1}) + Q_k^b - (t_{k+1} - t_k). \tag{8.8}$$

We extend the definition of Q_l^b to denote the difference between the queuing delay in the links after the bottleneck of the two consecutively received packets l and m, where all the probe packets $l+1, l+2, \ldots, m-1$ are dropped along the path. Therefore,

$$Q_l^b = \sum_{i=j+1}^{N} (q_m^i - q_l^i). \tag{8.9}$$

Similar to (8.8), we have

$$owd_m - owd_l = \frac{s}{C_j} + W^*(T_l, T_m) + Q_l^b - (t_m - t_l). \tag{8.10}$$

In deriving (8.10), we have made the following assumption,

A0 All probe packets that are dropped on the end-to-end path were dropped at the single bottleneck link that drops packets due to congestion.

Assumption A0 is a valid assumption, as packet losses not caused by congestion are rare (Section 8.2.2).

As in [65], we take the 4-tuple $\langle t_k, t'_k, t_{k+1}, t'_{k+1}\rangle$ as one sample of the path. If the probe packets sent at times t_k and t_{k+1} are within the same busy period and both are received at the destination of the end-to-end path, we call the sample *clean*; otherwise, we call it *contaminated*. Assuming M probe packets were sent in the same busy period, out of these M packets some were dropped along the end-to-end path. Let us denote the set of consecutive packet pairs that have been received at the receiver side of the end-to-end path as set $\mathcal{L}$. For example, if both the kth and $(k+1)$th packet are received at the receiver side, then $(k, k+1) \in \mathcal{L}$, as shown in Fig. 8.1. On the other hand, if two packets that are received at the receiver side of the end-to-end path, but all the sent packets between these two packets are dropped along the end-to-end path, they belong to $\mathcal{L}^c$. For example, if packet l and m are received at the receiver side of the end-to-end path, but all the packets $l+1, l+2, \ldots, m-1$ are dropped along the path, then $(l, m) \in \mathcal{L}^c$. Therefore, by summing over (8.8) and (8.10) for all of the received packets we obtain

$$owd_M - owd_1 = (M - 1 - |\mathcal{L}^c|)\frac{s}{C_j} + W_t(T_1, T_M) + \sum_{i=j+1}^{N}(q_M^i - q_1^i) - (t_M - t_1), \quad (8.11)$$

where $W_t(T_1, T_M)$ is defined as the total service time spent on the cross traffic packets that were not dropped during the interval $[T_1, T_M]$, and is defined as

$$W_t(T_1, T_M) := \sum_{(l,m)\in\mathcal{L}^c} W^*(T_l, T_m) + \sum_{(k,k+1)\in\mathcal{L}} W(T_k, T_{k+1}). \quad (8.12)$$

In deriving Eq. (8.11), packets 1 and M are at the onset and end of the busy period. In

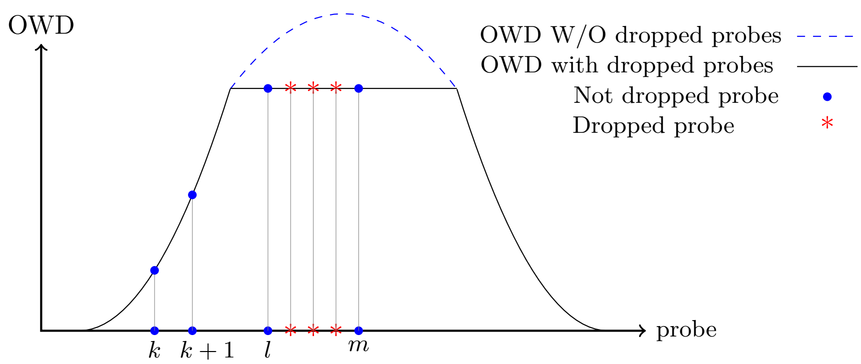


Figure 8.1: One-way-delay with and without dropped probes, $(k, k+1) \in \mathcal{L}$ and $(l, m) \in \mathcal{L}^c$.

decreasing-rate methods, *owd* for a chirp has a bump shape as shown in Fig. 8.1. At the beginning of the chirp, as the probes are being sent at a high rate, congestion will arise at the bottleneck; hence, *owd* will increase, until it reaches its maximum value. However, as the probe rates decrease further and go below the ABW the built-up congestion will decongest and *owd* will decrease back to its minimum value. This increase and decrease in *owd* is called a *bump* [65]. When there are tail-dropped packets in the bump, the middle part of the bump will be flattened. This is explained further using a testbed experiment in Section 8.4.2. We assume that if two consecutive packets k and $k+1$ are both received at the receiver side of the end-to-end path, no cross traffic packets are dropped in the interval $[T_k, T_{k+1}]$ (see Assumption A1 below). As the inter-departure time between two consecutive probes is very small with respect to scale of the events on the bottleneck link this assumption is a valid assumption.

8.3.2 Residual bandwidth at bottleneck

The residual bandwidth of the bottleneck link at interval $[T_1, T_M]$ can be estimated as

$$R_j(T_1, T_M) = C_j \left[1 - \frac{W(T_1, T_M)}{T_M - T_1}\right] = C_j \left[1 - \frac{W(T_1, T_M)}{t_M - t_1 + \sum_{i=1}^{j-1}(q_M^i - q_1^i)}\right], \tag{8.13}$$

where $W(T_1, T_M)$ is defined as the total time that would be spent on cross traffic packets, if no cross traffic packet was dropped at the bottleneck. In the case of no dropped probes at the bottleneck, $W(T_1, T_M) = W_t(T_1, T_M)$, and equations (8.11) and (8.13) simplify to equations (9) and (10) in [65]. However, if some packets, either probes or cross traffics, are dropped at the bottleneck, we need to estimate $W(T_1, T_M)$ from $W_t(T_1, T_M)$.

As we can just measure the amount of cross traffic that went through the bottleneck and was not dropped, to estimate the total cross traffic, we make some simplifying assumptions as follows:

A1 No cross traffic packets are dropped in the interval $[T_k, T_{k+1}]$, if both packets k and $k+1$ have traversed the end-to-end path.

A2 The amount of aggregate cross traffic of any interval is proportional to the interval length.

A3 The probability of packet drop on any interval is equal between probe packets and cross traffic packets.

A4 There is only a single bottleneck that drops packets.

Note, that while these simplifying assumptions are used to obtain an estimate of the amount of cross traffic, they by no means lead to the exact value for the total cross traffic, unless the cross traffic rate is almost constant in the interval $[T_1, T_M]$. In practice, the number of cross traffic packets between two consecutive probe packets is at maximum only a few packets; therefore, meaningful variations of cross traffic and queue occupancy are much longer than an interval between two consecutive packets (for example, for 100 Mb/s, the interval is around 120 μs) and the quantization does not prevent from a good enough estimation of cross traffic. As is shown in our testbed evaluation 8.4.4, in spite of these simplifying assumptions, our approach yields an accurate estimate of the available bandwidth.

One method of obtaining an estimate $W(T_1, T_M)$ from $W_t(T_1, T_M)$, is to simply set $W(T_1, T_M) = W_t(T_1, T_M)$. Obviously, when there are dropped packets this assumption is not correct. However, utilizing this assumption, if we choose probe 1 and M such that $owd_M = owd_1$, then using (8.11) and (8.13), the available bandwidth can be derived as

$$R_j(T_1, T_M) = C_j \left[1 - \frac{W(T_1, T_M)}{T_M - T_1}\right] = \frac{(M - 1 - |\mathcal{L}^c|)}{t_M - t_1} s. \tag{8.14}$$

In deriving (8.14), as in [65] it is assumed that

$$\sum_{i=1}^{j-1} (q_M^i - q_1^i) = \sum_{i=j+1}^{N} (q_M^i - q_1^i) = 0 \tag{8.15}$$

Equation (8.14) can be further improved if we estimate $W(T_1, T_M)$ from $W_t(T_1, T_M)$, instead of simply equating them. Applying assumptions A1 and A2, we have,

$$\sum_{l,m \in \mathcal{L}^c} W^*(T_l, T_m) = \sum_{l,m \in \mathcal{L}^c} \left(1 - P_{\text{drop}_l}\right) \frac{T_m - T_l}{T_M - T_1} W(T_M, T_1), \tag{8.16}$$

where P_{drop_l} is defined as the probability of dropping cross traffic packets during interval $[T_l, T_m]$. Under assumption A3, P_{drop_l} is the same for cross traffic packets and probe packets in the interval $[T_l, T_m]$ and can be derived as

$$P_{\text{drop}_l} = \frac{l - m}{l - m + 1}. \tag{8.17}$$

Using assumption A2 we have,

$$\sum_{(k,k+1) \in \mathcal{L}} W(T_k, T_{k+1}) = \left[1 - \frac{T_{|\mathcal{L}^c|}}{T_M - T_1}\right] W(T_M, T_1), \tag{8.18}$$

where $T_{|\mathcal{L}^c|}$ is defined as the duration of the intervals for which probe packets are dropped, or

$$T_{|\mathcal{L}^c|} := \sum_{(l,m)\in\mathcal{L}^c} |[T_l, T_m]| = \sum_{(l,m)\in\mathcal{L}^c} (T_m - T_l). \tag{8.19}$$

Using (8.12) and (8.16)-(8.19), we obtain

$$W_t(T_1, T_M) = \alpha_{\mathcal{L}} W(T_1, T_M), \tag{8.20}$$

where $\alpha_{\mathcal{L}}$ is the success coefficient, defined as

$$\alpha_{\mathcal{L}} := 1 - \frac{T_{|\mathcal{L}^c|}}{T_M - T_1} + \sum_{l,m\in\mathcal{L}^c} \left(1 - P_{\mathrm{drop}_l}\right) \frac{T_m - T_l}{T_M - T_1}. \tag{8.21}$$

Note that when there are no dropped packets $\alpha_{\mathcal{L}} = 1$. Using (8.20) and (8.21), and choosing probes 1 and M such that $owd_1 = owd_M$, and using (8.15), we have

$$W(T_1, T_M) = \frac{1}{\alpha_{\mathcal{L}}} \left[(t_M - t_1) - (M - 1 - |\mathcal{L}^c|) \frac{s}{C_j} \right]. \tag{8.22}$$

Therefore, the residual bandwidth can be expressed as

$$R_j(T_1, T_M) = C_j \left(1 - \frac{1}{\alpha_{\mathcal{L}}}\right) + \frac{(M - 1 - |\mathcal{L}^c|)}{\alpha_{\mathcal{L}}(t_M - t_1)} s \tag{8.23}$$

Note that Eq. (8.23) simplifies to Eq. (11) in [65] when there are no dropped packets. However, unlike Eq. (11) in [65], Eq. (8.23), depends on the bottleneck capacity. Hence, some method is needed to estimate the bottleneck capacity. One of the popular methods for capacity estimation is *pathrate* [30]. This method requires sending a train of packet pairs/groups to estimate the bottleneck capacity utilizing the dispersion of these packets. Capacity measurement introduces additional overhead, and is typically more intrusive than modern ABE. However, since capacity is a static value, this needs to be done only once per path, as opposed to ABW, which is time-varying. Moreover, numerical results presented in Section 8.4, show that our ABE algorithm is relatively insensitive to the accuracy of the bottleneck link capacity estimate.

Note that, according to (8.16) and (8.17), during the flat part of the bump, violations of assumptions A2 and A3 will effectively cancel out. For example, suppose there is a burst of cross traffic between probe packets l and m. Therefore, the amount of cross traffic in the interval $[T_l, T_m]$ is greater than an estimate directly proportional to the interval length and assumption A2 is violated. In addition, as there is a burst of cross traffic, the probability of packet drop for cross traffic will also be greater than the corresponding probability for probe packets and assumption A3 will be violated. Therefore, in (8.16), these two violations will, in effect, cancel out. A similar argument can be given for the case where the cross traffic in an interval $[T_l, T_m]$ is less than an estimate directly proportional to the interval length.

8.4 Numerical Study

We use a testbed to evaluate our algorithm and compare it with PathCos++, Voyager-D and SLDRT. The evaluation is done on a testbed, as simulations fail to account for real world OWD noise [89].

8.4.1 Testbed configuration

Our testbed consists of 7 Linux-based workstations, as shown in Fig. 8.2. All links L1−L7 are implemented using Ethernet switches, and each link is configured at 1 Gb/s or 10 Gb/s. We refer to these configurations as the 1 Gb/s or 10 Gb/s testbed, respectively. The link L5 is set as the bottleneck that drops packets. Note that, having cross traffic going through the bottleneck limits its available bandwidth and therefore induce it to act as a bottleneck. In our experiments, the bottleneck capacity is set at 100 Mb/s, 1 Gb/s or 10 Gb/s. The Ethernet NICs on the nodes are a mix of Intel e1000e, Intel IGB and Broadcom TG3. For the 10 Gb/s testbed, the NICs are Intel X710. Nodes r2−r4 are configured as

Table 8.1: *Experiment parameters*

Parameters	**Description**
B_capacity	Bottleneck link capacity: 100Mb/s, 1Gb/s, or 10Gb/s
R_capacity	Rest of the links' capacity in testbed: 1Gb/s or 10Gb/s
AQM [parameters]	AQM type: Pfifo [number of packets], CoDel [number of packets, target (ms), interval window (ms)] PI^2 [number of packets, target (ms)]
bn (bn_delay, bn_drop)	Total # bottlenecks (# bottlenecks that just add delay to OWD , # bottlenecks that drop packets)
ct	Type of cross traffic at bottleneck: 5UDP (bursty) or 5TCP
u	Highest train rate in a chirp train
l	Lowest train rate in a chirp train
N	# packets in a chirp train
n	# chirps in the experiment
p	Probe packet size (Bytes)

routers. We use software routers rather than Ethernet switches as they are slower and more unpredictable when processing packets, generating more OWD noise. This makes ABE a more challenging process compared to a testbed with hardware routers. On the router r3 acting as a bottleneck, we configure the output queue towards the bottleneck L5 as a simple FIFO queue, CoDel [74], or PI^2 [28] AQM. Configuration of these bottleneck queues is explained further in Section 8.4.4. Configurable experiment parameters are listed in Table 8.1.

A dedicated tool is used on s1 to send to d9 a chirp train conforming to PathCos++, Voyager-D or SLDRT. As listed in Table 8.1, the number of UDP packets in the chirp train in the experiments is denoted by N, its lowest train rate as l and its highest train rate as u. The *train rate* of a probe is the sending rate from the beginning of the chirp train to that probe, it corresponds to how a decreasing chirp train interacts with a bottleneck. The lowest and highest train rate are chosen manually so as to cover the range of possible available

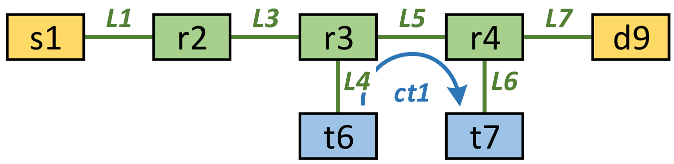


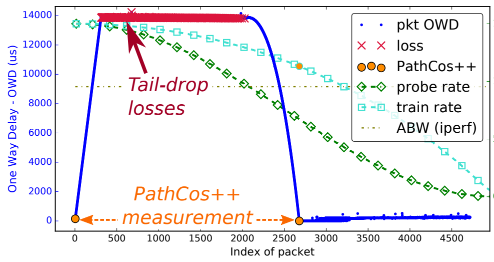
Figure 8.2: Testbed topology for ABE evaluation.

bandwidths. The chirp train is received at d9, and both the relative OWD of each packet and each packet loss are recorded in sequence, and processed to produce ABW estimates. Nodes t6−t7 are used to send and receive cross traffic, the cross traffic is generated using a number of parallel UDP iperf sessions [3] through the bottleneck, which creates traffic with known bandwidth and micro bursts [66][1]. By default, the target cross traffic bandwidth is equally divided between 5 parallel iperf (version 2.0.12). We measure the capacity of a network path by flooding it with UDP chirp train traffic using iperf. We compute the "ground truth" ABW by subtracting from the capacity the amount reported by the iperf cross traffic sessions injected on the bottlenecks.

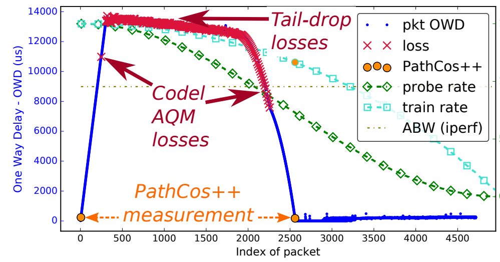
8.4.2 Tail-drop and AQM losses

Fig. 8.3a shows the OWD and packet losses recorded in a chirp train when the bottleneck is a simple FIFO and has a capacity of 95.5 Mb/s. In this example, there is no cross traffic, however, as the highest rate of the chirp train is 150 Mb/s, link L5 acts as a bottleneck. In the first phase, the queue which was empty is filling with probe packets, and as a result the service time of packets increase, which leads to an increase of the OWD. After around 300 packets, the queue becomes full, and the queue operates in tail-drop mode and discards excess packets. At this point, OWD no longer increases, because the queue can not get any longer. After around 2100 packets, the *probe rate* of the chirp train (instantaneous sending rate between consecutive probes) becomes lower than the bottleneck rate, the queue starts to shrink, leading to an OWD decrease, and packet losses stop. ABW estimation algorithms

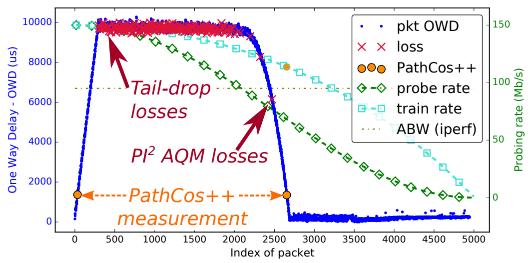
[1]Other cross traffic such as Poisson or Internet traces cause temporal uncertainty and introduce evaluation errors [46, 66], so are not used.



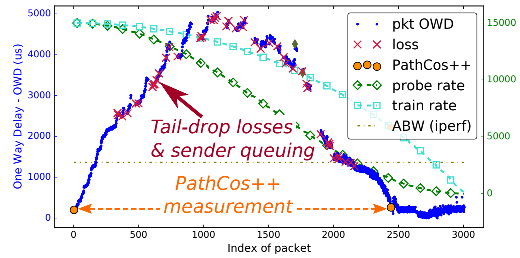
(a) FIFO[100p], bn=1(0,1), B_capacity=100Mb/s, cr=0Mb/s, u=150Mb/s, l=4Mb/s, N=5000, p=1024.



(b) CoDel[5ms, 100p], bn=1(0,1), B_capacity=100Mb/s, cr=0Mb/s, u=150Mb/s, l=4Mb/s, N=5000, p=1024.



(c) PI^2[5ms, 100p], bn=1(0,1), B_capacity=100Mb/s, cr=0Mb/s, u=150Mb/s, l=4Mb/s, N=5000, p=1024.



(d) FIFO[100p], bn=2(1,1), B_capacity=10Gb/s, ct=5UDP, cr=7Gb/s, u=15Gb/s, l=400Mb/s, N=3000, p=8972.

Figure 8.3: One Way Delay (OWD) and packet losses for PathCos++ chirp train with various bottleneck configurations.

like PathCos++, estimate ABW by measuring the rate of packets between two points with equal OWD at start and end of the bump, like the two big orange circles at the bottom of the bump, and this measurement process is impacted by packet losses.

Fig. 8.3b shows the same experiment when the FIFO is replaced by CoDel [74], a modern AQM (Section 8.2.2). CoDel triggers a single packet loss around packet 250. Unlike TCP, PathCos++ does not react to AQM packet losses, so probe packets fill the queue and cause tail-drops losses, like in the FIFO queue. Once the queue is full, CoDel increases packet drops to shrink the queue, thus managing to reduce the queue service time and OWD. When the probe rate is low enough, the queue become small and CoDel stops dropping packets. Fig. 8.3c shows that PI^2 [28] has similar effect on the chirp train.

By design, AQM triggers few packet losses: TCP congestion algorithms react drastically to packet losses, so only a few losses are needed to reduce congestion in general [28]. Keeping the number of packet losses small is advantageous for performance, as the lost packets need to be retransmitted. In our experience, most AQM techniques cause around 1% extra packet loss during the onset of congestion. Furthermore, all AQMs average queue delay over multiple packets and change their probability of drop fairly slowly, this can be seen as CoDel only drops a single packet during the onset of congestion, but continues dropping packets even after congestion is reduced.

In our experience, the AQM losses at the onset of congestion are too few to be reliably measured. In summary, AQM queues produces losses similar to tail-drop losses and for the purpose of ABE behave like a FIFO queue.

8.4.3 Multiple bottlenecks and Interrupt Coalescence

The combined effect of two active bottlenecks along the path can be seen in Fig. 8.3d. The capacity of the network path is 10 Gb/s, and the main bottleneck at L5 has 7 Gb/s of cross traffic. Initially, probes are sent at 15 Gb/s, therefore both the sender s1 and the main bottleneck at r3 are queuing packets. After packet 400, the main bottleneck is full and starts dropping packets, however the sender is still queuing packets, resulting in a slower increase in OWD. From packets 1100 to 2000, the probe rate is below 10 Gb/s, and the sender queue is gradually reduced to zero (at the point the train rate goes below 10 Gb/s). After packet 2200, the probe rate is below 2.7 Gb/s, causing the losses to stop and the queue at the main bottleneck to finally deplete. The combined effect of the two bottlenecks is quite complex and creates a more complex bump than the single bottleneck case. However, in practice, this does not impact much our techniques (see Section 8.4.4).

Fig. 8.3d also highlight the effect of interrupt coalescence, traffic burstiness and other OWD noises. From the bursty variations of the OWD, one can assume that assumptions A1, A2 and A3 are violated (see Section 8.3.2), however this will not impact much our techniques (see Section 8.4.4). On the other hand, such burtiness does impact the underlying ABE method [66], so to combat those effects, we use **MaxIAT** [89] to pre-process the chirp train prior to ABE.

Table 8.2: ABE for FIFO queue [Mb/s]

PathCos++	Cross traffic [Mb/s]				
	0	**20**	**40**	**60**	**80**
Original - Eq. (11)	113.3	102.1	88.6	71	45
Compensation I Eq. (8.14)	95.5	79.6	63.4	48.5	28.9
Compensation II Eq. (8.23)	95.6	76.3	55.1	36.3	14.8
True value	95.6	75.6	55.6	35.6	15.6
Dropped packets (%)	15.7%	22%	28.5%	31.6%	35.7%

Table 8.3: ABE for CoDel AQM [Mb/s]

PathCos++	Cross traffic [Mb/s]				
	0	**20**	**40**	**60**	**80**
Original - Eq. (11)	116.5	105.7	93.2	75.2	48.6
Compensation I Eq. (8.14)	95.4	81.8	65.4	51.5	30
Compensation II Eq. (8.23)	95.5	78.6	56	39.2	13.6
True value	95.6	75.6	55.6	35.6	15.6
Dropped packets (%)	18.1%	22.6%	29.8%	31.5%	38.3%

8.4.4 ABE accuracy

We evaluated two versions of our algorithm. The **original** versions of PathCos++ [65], Voyager-D [66] and SLDRT [46] discard all the chirp trains with packet losses, in our experiment we forced those methods to ignore packet losses and produce an ABE using Eq. (11) in [65]. **Compensation I** is based on (8.14), which only compensates for lost probe packets, does not use path capacity information, and is therefore simpler. **Compensation II** is based on the more accurate formulation in (8.23), which compensates for both lost probe packets and lost cross traffic, but requires the path capacity. Both compensations are implemented as a post-processing step of each original ABE method. In each experiment, the same set of received chirp trains is processed using each technique. In addition, **Max-IAT** [89] pre-processing is used to handle interrupt coalescence. Our compensations are a simple extension to the existing ABE methods, and can improve the ABE measured by each method by providing a framework to compensate for the lost packets during the measurement process.

The results for ABE using a single chirp train conforming to PathCos++ are shown in Tables 8.2 and 8.3. Table 8.2 shows the result of ABE with the different compensations implemented on 1 Gb/s testbed with bottleneck capacity set at 100 Mb/s. In this experiment, the chirp train length, N, is 5000 packets, the queue at the bottleneck is a FIFO with buffer capacity 100 packets, the highest rate of the chirp, u, is 150 Mb/s, the lowest rate, l, is 5 Mb/s, and the rest of the testbed is in its default configuration (Section 8.4.1). As the cross traffic increases, the available queue size becomes smaller and the number of packets dropped across the rate measurement period increases. The original formulation of PathCos++ always overestimates the ABW. When the cross traffic rate is 80 Mb/s, the PathCos++ estimate is 3 times the actual ABW, this explain why chirp train with packet losses are normally discarded. When the path is idle, i.e., there is no cross traffic, Compensation I is sufficient to provide a good ABW estimate. In the presence of cross traffic, Compensation I improves upon the estimate of PathCos++, but still has a high error. Compensation II gives an estimate very close to the true ABW under all conditions.

Table 8.3 shows ABE obtained from a single chirp using a CoDel AQM with buffer capacity 100 packets, target delay 5 ms and interval 10 ms (see [74] for details) at the bottleneck. The chirp train still uses 5000 packets, and the highest and lowest rate of the chirp train are as before, i.e., 150 Mb/s and 5 Mb/s, respectively. The results are almost the same as for the FIFO queue, showing that Compensation II yields accurate estimates even when AQM is used. The percentage of dropped packets during the measurement intervals is shown in Table 8.2 and Table 8.3. As can be observed, even when there are a considerable number of dropped packets over the measurement interval, Compensation II is still very accurate for both FIFO queue and CoDel AQM.

Fig. 8.4 shows ABE accuracy for different cross traffic bandwidth on a single bottleneck of capacity 100 Mb/s on link L5. The network path is 1 Gb/s, so only L5 acts as a bottleneck, and the outgoing interface on router r3 can be configured for tail-drop (simple FIFO), CoDel AQM or PI^2 AQM. We configured SLDRT with a longer chirp train to create a similar amount of congestion and probe packet drops as PathCos++ and Voyager-D. We also configured PI^2 with a larger queue to reduce tail-drop losses and highlight better the effect of AQM losses.

Without any compensation (**original** - Fig. 8.4), the ABE error is quite significant, which explains why most implementations of ABE discard chirp train with packet losses. **Compensation I** offers very low estimation error when there is no cross traffic, however its error increases as cross traffic increases, as expected. **Compensation II** offers very low estimation error in all cross traffic conditions. The type of bottleneck and the type of ABE method used does not change those observations. In all cases, the cross traffic is bursty, generated using 5 parallel UDP iperf sessions. Such bursty cross traffic most likely violates assumptions A1, A2 and A3 (see Section 8.3.2); however, this does not impact the accuracy of the results.

Fig. 8.5 shows ABE accuracy for different cross traffic bandwidth on a 1 Gb/s network path. The link L5 is the main bottleneck due to the cross traffic. However, the start of the chirp train is sent faster (2 Gb/s) than the network path can handle; therefore, the sender s1 is also a bottleneck. This network path is a combination of two bottlenecks at sender s1 and router r3 and is similar to the network path in Section 8.4.3. The bottleneck at sender s1 has capacity 1 Gb/s and only queues packets (no drops). The bottleneck at router r3 has variable cross traffic, can be configured with tail-drop (simple FIFO), CoDel AQM or PI^2 AQM, and it queues and drops packets. In addition, those experiments explore larger queue sizes (1000 packets on r3).

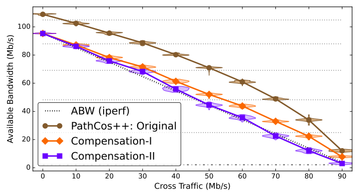
Without cross traffic, the bottleneck at the sender s3 queues all the excess traffic; therefore, the bottleneck at router r3 does not see any congestion and does not need to drop packets. As the result, all techniques perform the same. With increasing levels of cross traffic, more congestion is created at router r3, which increases the amount of packet drops (Fig. 8.5). In those conditions, we see results similar to the experiments at 100 Mb/s: **original** has the highest error and **Compensation II** offers very low estimation error in all cross traffic conditions. This confirms that our techniques work appropriately in the presence of multiple bottlenecks, as long as only a single bottleneck drops packets.

Fig. 8.6 shows ABE accuracy for different cross traffic bandwidth on a 10 Gb/s network path. The setup is similar to the previous experiments at 1 Gb/s, sender s1 is limited to 10 Gb/s and router r3 has cross traffic and can drop packets. Performing ABE in software at 10 Gb/s is challenging [66]; however, our techniques do improve accuracy. **Compensation II** tends to underestimate the ABW. This is caused by violations of assumption A3. In these experiments, there are additional packets drops that are caused by the burden of processing packets at 10 Gb/s. The algorithm assumes that all of these packet drops happen at the bottleneck, and therefore overestimates the probability of drop for the cross traffic. This shows that when there are multiple sources of packet drops, it is not possible to accurately estimate the amount of cross traffic drops. However, the error of **Compensation II** is limited and it still gives the best performance in this case.

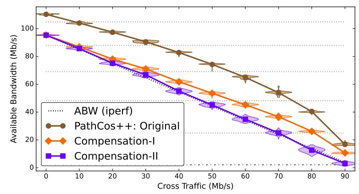
Fig. 8.7 shows that the ABE relative error with respect to the measurement errors in the path capacity using a single chirp on a 100 Mb/s bottleneck capacity. The error relation is linear: a greater amount of packet loss leads to higher error, which confirms the intuition from (8.23). We note that the ABE is not very sensitive to the measurement of the path capacity.

8.5 Conclusion

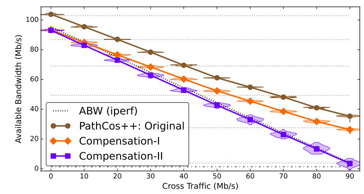
We developed a set of techniques to improve the accuracy of decreasing-rate ABE methods in the presence of packet loss. The technique estimates both the amount of probe traffic and cross traffic lost due to congestion in order to compute a more accurate ABW estimate. These techniques are compatible with all decreasing-rate methods and are also compatible with noise pre-processing of chirp train, such as mitigation of interrupt coalescence. We presented experimental results from a network testbed that confirms the effectiveness of the proposed technique in enhancing ABE accuracy in scenarios involving packet loss due to tail-drop as well as AQM losses. Two variants of the technique were evaluated: Compensation I does not require path capacity information, whereas Compensation II requires an estimate of path capacity to obtain more accurate ABW estimates. The results show that Compensation I significantly improves upon existing decreasing-rate methods under all conditions tested, but its error is still high in the presence of cross traffic. Compensation II provides ABW estimates that are very close to the true values under various levels of cross traffic. Our results also show that Compensation II is not very sensitive to the accuracy of the path capacity estimate. Our proposed ABE method enables the use of larger chirps, which by extension can provide improved accuracy of the ABW estimate even in the presence of short queues that are needed to reduce buffer-bloat.



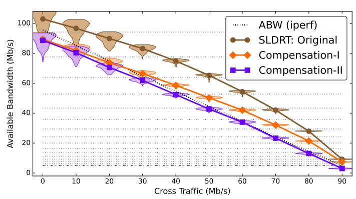
(a) FIFO[100p], bn=1(0,1), ct=5UDP, PathCos++, u=200Mb/s, l=5Mb/s, N=1000, p=1024, n=125.



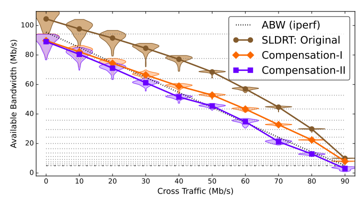
(b) CoDel[5ms, 100p], bn=1(0,1), ct=5 UDP, PathCos++, u=200Mb/s, l=5Mb/s, N=1000, p=1024, n=125.



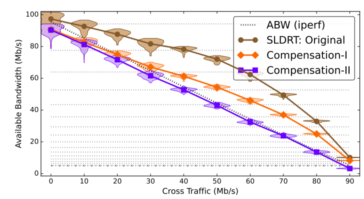
(c) PI^2[5ms, 1000p], bn=1(0,1), ct=5 UDP, PathCos++, u=200Mb/s, l=5Mb/s, N=3000, p=1024, n=125.



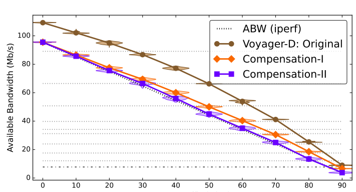
(d) FIFO[100p], bn=1(0,1), ct=5UDP, SLDRT, u=200Mb/s, l=5Mb/s, N=10000, p=1024, n=125.



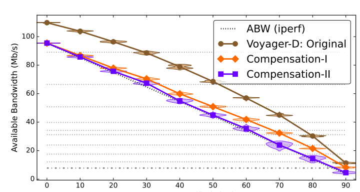
(e) CoDel[5ms, 100p], bn=1(0,1), ct=5UDP, SLDRT, u=200Mb/s, l=5Mb/s, N=10000, p=1024, n=125.



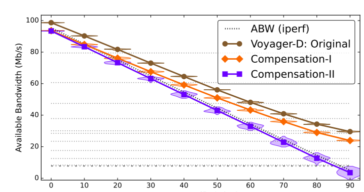
(f) PI^2[5ms, 300p], bn=1(0,1), ct=5UDP, SLDRT, u=200Mb/s, l=5Mb/s, N=10000, p=1024, n=125.



(g) FIFO[100p], bn=1(0,1), ct=5 UDP, Voyager-D, u=200Mb/s, l=5Mb/s, N=1000, p=1024, n=125.

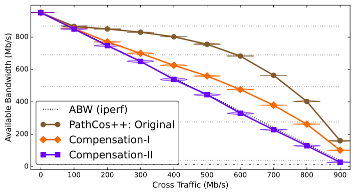


(h) CoDel[5ms, 100p], bn=1(0,1), ct=5 UDP, Voyager-D, u=200Mb/s, l=5Mb/s, N=1000, p=1024, n=125.

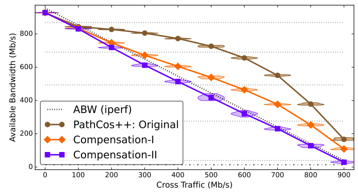


(i) PI^2[5ms, 1000p], bn=1(0,1), ct=5 UDP, Voyager-D, u=200Mb/s, l=5Mb/s, N=3000, p=1024, n=125.

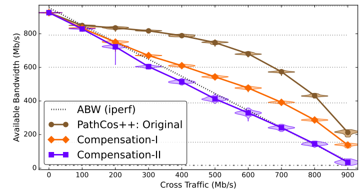
Figure 8.4: ABE for PathCos++, SLDRT and Voyager-D methods with various bottleneck at 100Mb/s vs. cross traffic.



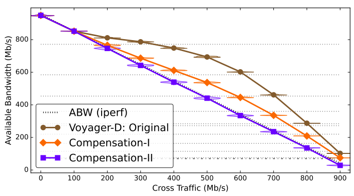
(a) FIFO[1000p], bn=2(1,1), ct=5 UDP, PathCos++, u=2Gb/s, l=50Mb/s, N=15000, p=1024, n=75.



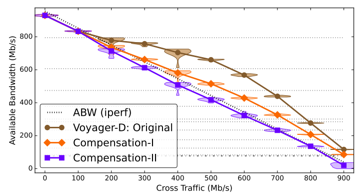
(b) CoDel[5ms, 1000p], bn=2(1,1), ct=5 UDP, PathCos++, u=2Gb/s, l=50Mb/s, N=15000, p=1024, n=75.



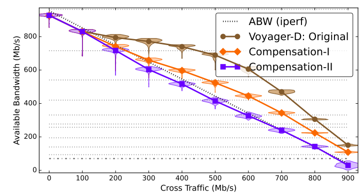
(c) PI^2[5ms, 1000p], bn=2(1,1), ct=5 UDP, PathCos++, u=2Gb/s, l=50Mb/s, N=15000, p=1024, n=125.



(d) FIFO[1000p], bn=2(1,1), ct=5 UDP, Voyager-D, u=2Gb/s, l=50Mb/s, N=15000, p=1024, n=75.



(e) CoDel[5ms, 1000p], bn=2(1,1), ct=5 UDP, Voyager-D, u=2Gb/s, l=50Mb/s, N=15000, p=1024, n=75.



(f) PI^2[5ms, 1000p], bn=2(1,1), ct=5 UDP, Voyager-D, u=2Gb/s, l=50Mb/s, N=15000, p=1024, n=125.

Figure 8.5: ABE for PathCos++ and Voyager-D methods with various bottleneck at 1Gb/s vs. cross traffic.

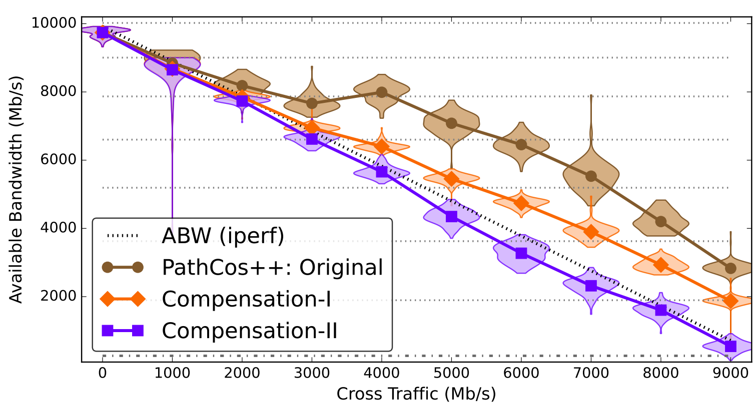


Figure 8.6: ABE for PathCos++ method with various bottleneck at 10Gb/s vs. cross traffic, FIFO[100p], bn=2(1,1), ct= 5 UDP, u=15Gb/s, l=500 Mb/s, N=3000, p=8972, n=125.

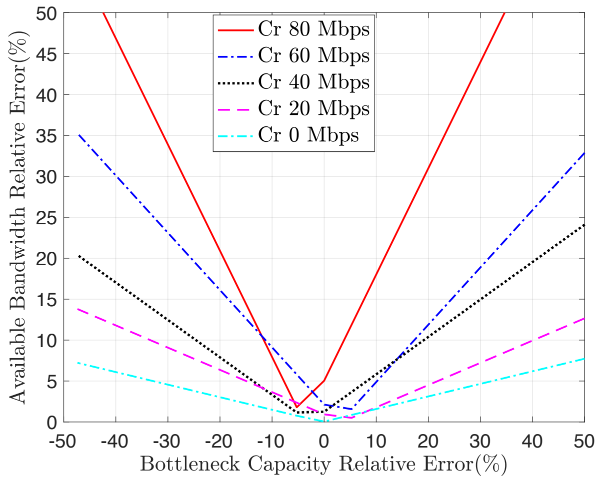


Figure 8.7: Available bandwidth estimation relative error for Compensation-II with respect to capacity value.

Chapter 9: Conclusion

In this dissertation we developed a stochastic traffic envelope based on the phase-type distribution to characterize a traffic flow such that tools from stochastic network calculus can be used to evaluate network performance in terms of probabilistic end-to-end delay bounds. Our model is a particular form of the gSBB traffic envelope developed in [51, 99], which we refer to as phase-type traffic bounds. We showed that the proposed phase-type network calculus, is closed with respect to the family of phase-type functions and if the input traffics of a feedforward network can be characterized using phase-type traffic bounds, all the traffic streams in the network can also be characterized using phase-type traffic bounds. This property allows us to analyze the network in terms of performance measures such as probabilistic end-to-end delay. Further development of the phase-type network calculus for the closed networks with loops and cross-traffic is still left to done.

We also developed two methods of characterizing a given traffic using a phase-type traffic bounds and finding the corresponding parameters. The first method is based on a least squares approach and the second one was based on the EM algorithm for the class of phase-type distributions. Both methods have the potential to be further developed into an online algorithm for traffic flows with slow time-scale statistical fluctuations.

We closed a gap in the literature on stochastic network calculus by developing a stochastic traffic regulator to shape any given traffic stream according to a specified gSBB traffic envelope. Such traffic shaping is essential in order to apply the stochastic network calculus in practical networks.

Motivated by the desire to provide performance guarantees for time-critical applications in future networks, we proposed the W-envelope, a traffic bound on the MGF of the workload process resulting from offering a traffic source to a constant rate server. In contrast to the MGF traffic arrival envelope (A-envelope), the W-envelope is amenable to traffic regulation and traffic fitting via the methods we developed for the gSBB traffic envelope in Chapters 5 and 4, respectively. Our numerical results showed that the W-envelope policy for admission control with independent flows leads to significant statistical multiplexing gain. We showed that the W-envelope framework can provide network performance guarantees in a resource-efficient manner.

We presented a practical framework for providing stochastic delay guarantees based on results from stochastic network calculus. Key elements of our approach are the phase-type traffic bound (Chapter 3), the W-envelope (Chapter 7), a method for fitting a traffic flow to a phase-type bound (Chapter 4), stochastic traffic regulators to enforce compliance of a traffic flow to a negotiated gSBB traffic envelope (Chapter 5) and an admission control scheme (Chapter 6). Each flow characterizes its traffic process by a phase-type traffic envelope, which can be determined using the procedure developed in Chapter 4. Using our result relating the gSBB envelope to the W-envelope (Theorem 7.4.1), an W-envelope can be determined for each traffic flow (Chapter 7). A phase-type envelope, or more generally, a gSBB envelope can be enforced by our proposed stochastic regulator. The associated W-envelope can be used in an admission control scheme.

Finally, we developed a set of techniques to improve the accuracy of decreasing-rate ABE methods in the presence of packet loss (Chapter 8). The technique estimates both the amount of probe traffic and cross traffic lost due to congestion in order to compute a more accurate ABW estimate. These techniques are compatible with all decreasing-rate methods and are also compatible with noise pre-processing of chirp train, such as mitigation of interrupt coalescence. We presented experimental results from a network testbed that confirms the effectiveness of the proposed technique in enhancing ABE accuracy in scenarios involving packet loss due to tail-drop as well as AQM losses. Our ABE method could be used to discover the amount of available bandwidth on an end-to-end path, which could then be used to provide stochastic delay guarantees for time-sensitive traffic via our proposed QoS framework.

In the ongoing work, we are trying to further develop our framework for providing end-to-end stochastic delay guarantees on a multi-hop path with different scheduling methods on the end-to-end path. The work in dissertation can be applied to augment the current Internet with a service to provide end-to-end delay guarantees for time-critical applications. Such a service can benefit a wide range of applications, including delay-sensitive, bandwidth-hungry Internet applications such as multimedia streaming, network gaming, and virtual/augmented reality, and time-sensitive military communications over a network.

Appendix A: Inequivalence of SBB and gSBB

According to the (2.4), a traffic process with instantaneous rate process $R = \{R(t) : t \geq 0\}$ is said to have SBB with upper rate ρ and bounding function $f(\sigma) \in \mathcal{F}$ if, for all $t, s \geq 0$ and all $\sigma \geq 0$,

$$\mathsf{P}\left\{R(s,t) - \rho(t-s) \geq \sigma\right\} \leq f(\sigma), \tag{A.1}$$

On the other hand, a traffic process with instantaneous rate process $R = \{R(t) : t \geq 0\}$ is said to have gSSB with upper rate ρ and bounding function $f(\sigma) \in \mathcal{BF}$ if, for all $t \geq 0$ and all $\sigma \geq 0$,

$$\mathsf{P}\left\{\max_{0 \leq s \leq t}\left\{R(s,t) - \rho(t-s)\right\} \geq \sigma\right\} \leq f(\sigma), \tag{A.2}$$

where for continuous-time process $R(t)$, $R(s,t) = \int_s^t R(\tau)\,\mathrm{d}\tau$ and for discrete-time process $R[n]$, $R(s,t) = \sum_{i=s+1}^{t} R[i]$. In this appendix we argue that being gSBB means being SBB, but not the other way around. In other words, if a traffic process is SBB it might not be gSBB. We provide an example of such a case in here. We define events $A(s;t,\sigma)$, $B(s;t)$, and $Z(t,\sigma)$ as

$$A(s;t,\sigma) := \left\{R(s,t) - \rho(t-s) \geq \sigma\right\} \tag{A.3}$$

$$B(r;t) := \left\{r = \arg\max_{0 \leq s \leq t}\left\{R(s,t) - \rho(t-s)\right\}\right\} \tag{A.4}$$

$$Z(t,\sigma) := \left\{\max_{0 \leq s \leq t}\left\{R(s,t) - \rho(t-s)\right\} \geq \sigma\right\} \tag{A.5}$$

Therefore according to (A.2), a traffic process $R(t)$ is gSBB if we have

$$\mathsf{P}\left\{Z(t,\sigma)\right\} \leq f(\sigma) \tag{A.6}$$

We should note that $B(r;t)$ for $r = 0, 1, \ldots, t$ form a partition of sample space Ω. Therefore we have

$$\mathsf{P}\left\{Z(t,\sigma)\right\} = \mathsf{P}\left\{Z(t,\sigma) \bigcap \left(\bigcup_{i=0}^{t} B(i;t)\right)\right\} = \sum_{i=0}^{t} \mathsf{P}\left\{Z(t,\sigma) \cap B(i;t)\right\}$$

$$= \sum_{i=0}^{t} \mathsf{P}\left\{Z(t,\sigma)|B(i;t)\right\} \mathsf{P}\left\{B(i;t)\right\} \tag{A.7}$$

We should note that,

$$\left\{Z(t,\sigma) \cap B(i;t)\right\} = \left\{A(i;t,\sigma) \cap B(i;t)\right\} \tag{A.8}$$

Therefore a traffic process $R(t)$ is gSBB if we have

$$\mathsf{P}\left\{Z(t,\sigma)\right\} = \sum_{i=0}^{t} \mathsf{P}\left\{A(i;t,\sigma) \cap B(i;t)\right\} \leq f(\sigma) \tag{A.9}$$

for all $t \geq 0$ and for all $\sigma \geq 0$. On the other hand, a traffic process $R(t)$ is SBB if we have

$$\mathsf{P}\left\{A(s;t,\sigma)\right\} \leq f(\sigma) \tag{A.10}$$

for all $t, s \geq 0$ and all $\sigma \geq 0$. But for $\mathsf{P}\left\{A(s;t,\sigma)\right\}$ we can use the same partion of $B(r;t)$ for $r = 0, 1, \ldots, t-1$ and therefore, we will have

$$\mathsf{P}\left\{A(s;t,\sigma)\right\} = \mathsf{P}\left\{A(s;t,\sigma) \bigcap \left(\bigcup_{i=0}^{t} B(i;t)\right)\right\} = \sum_{i=0}^{t} \mathsf{P}\left\{A(s;t,\sigma) \cap B(i;t)\right\} \tag{A.11}$$

We can easily verify that $\{A(s;t,\sigma) \cap B(i;t)\} \subset \{A(i;t,\sigma) \cap B(i;t)\}$ for $s, i = 1, 2, \ldots, t$. Therefore we have

$$\mathsf{P}\{A(s;t,\sigma)\} = \sum_{i=0}^{t} \mathsf{P}\{A(s;t,\sigma) \cap B(i;t)\} \quad \leq \sum_{i=0}^{t} \mathsf{P}\{A(i;t,\sigma) \cap B(i;t)\} = \mathsf{P}\{Z(t,\sigma)\} \tag{A.12}$$

which means being gSBB leads to being SBB. Now as an example of having SBB and not haiving gSBB, consider a discrete-time traffic process $R[n]$, where we have

$$R[1] = \begin{cases} 4\rho & if\ A = 1 \\ 0 & if\ A = 0 \end{cases} \tag{A.13}$$

$$R[2] = \begin{cases} 0 & if\ A = 1 \\ 2\rho & if\ A = 0 \end{cases} \tag{A.14}$$

and $R[n] = 0$ for $n \geq 3$, where A is a binary randon variable and $\mathsf{P}\{A = 0\} = \mathsf{P}\{A = 1\} = 0.5$. In this case we have

$$R(0,2) - 2\rho = \begin{cases} 2\rho & if\ A = 1 \\ 0 & if\ A = 0 \end{cases} \tag{A.15}$$

$$R(1,2) - \rho = \begin{cases} -\rho & if\ A = 1 \\ \rho & if\ A = 0 \end{cases} \tag{A.16}$$

Therefore, if we define $f(\sigma)$ as,

$$f(\sigma) = \begin{cases} 1 & \sigma = 0 \\ 0.5 & 0 < \sigma \leq 4\rho \\ 0 & 4\rho < \sigma \end{cases} \tag{A.17}$$

With this choise of $f(\sigma)$, $R[n]$ is clearly SBB, but this process is not gSBB as,

$$\mathsf{P}\left\{B(0;2)\right\} = \mathsf{P}\left\{A = 1\right\} = 0.5 \tag{A.18}$$

$$\mathsf{P}\left\{B(1;2)\right\} = \mathsf{P}\left\{A = 0\right\} = 0.5 \tag{A.19}$$

and $\mathsf{P}\left\{Z(2,\rho)\right\} = 1 > f(\sigma)$, which shows $R(n)$ is not gSBB.

Appendix B: Some Theorems about Alg-Er and Alg-Ga

B.1 Concavity of $g_i(\alpha, \lambda)$

We derive some useful results related to the gamma distribution and gamma mixture model.

Lemma B.1.1. The function $\log f(x; \alpha, \lambda)$ is strictly concave with respect to the shape α.

Proof. Taking partial derivatives with respect to α,

$$\frac{\partial^2}{\partial \alpha^2} \log f(x; \alpha, \lambda) = -\left[\frac{\Gamma(\alpha)\Gamma''(\alpha) - (\Gamma'(\alpha))^2}{\Gamma^2(\alpha)}\right] = -\psi^{(1)}(\alpha), \tag{B.1}$$

where $\psi^{(1)}(z)$ is the polygamma function of order 1. The polygamma function of order m is defined for complex z with positive real part by

$$\psi^{(m)}(z) := \frac{\mathrm{d}^m}{\mathrm{d}z^m}\psi(z) = \frac{\mathrm{d}^{m+1}}{\mathrm{d}z^{m+1}} \log \Gamma(z), \tag{B.2}$$

for $m = 1, 2, \ldots$, and

$$\psi^{(0)}(z) := \psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}. \tag{B.3}$$

The polygamma function of order m can be written in the form of a series as follows [4, Eq. (6.4.10)]:

$$\psi^{(m)}(z) = (-1)^{m+1} m! \sum_{k=0}^{\infty} \frac{1}{(z+k)^{m+1}}. \tag{B.4}$$

In particular, for $\alpha > 0$,

$$\psi^{(1)}(\alpha) = \sum_{k=0}^{\infty} \frac{1}{(\alpha+k)^2} > \sum_{k=0}^{\infty} \frac{1}{(\alpha+k)(\alpha+k+1)}$$
$$= \left(\frac{1}{\alpha} - \frac{1}{\alpha+1}\right) + \left(\frac{1}{\alpha+1} - \frac{1}{\alpha+2}\right) + \cdots = \frac{1}{\alpha}. \tag{B.5}$$

Therefore, the right-hand side of (B.1) is negative, from which the result follows immediately. □

Lemma B.1.2. The function $\log f(x;\alpha,\lambda)$ is strictly concave with respect to the rate λ.

Proof. Differentiating with respect to λ,

$$\frac{\partial}{\partial\lambda} \log f(x;\alpha,\lambda) = \frac{\alpha}{\lambda} - x. \tag{B.6}$$

Differentiating once more with respect to λ yields

$$\frac{\partial^2}{\partial\lambda^2} \log f(x;\alpha,\lambda) = -\frac{\alpha}{\lambda^2}, \tag{B.7}$$

which is negative. □

Lemma B.1.3. The function $\log f(x;\alpha,\lambda)$ is strictly concave with respect to (α,λ).

Proof. Let

$$q(x;\alpha,\lambda) := \log f(x;\alpha,\lambda). \tag{B.8}$$

It is sufficient to show that the Hessian matrix of $q(x;\alpha,\lambda)$ with respect to (α,λ) is negative definite, or equivalently, that

$$-\nabla^2 q(x;\alpha,\lambda) := \begin{bmatrix} a & b \\ b & d \end{bmatrix}, \tag{B.9}$$

is positive definite. From Lemmas B.1.1 and B.1.2,

$$a = \frac{\alpha}{\lambda^2}, \quad d = \psi^{(1)}(\alpha), \tag{B.10}$$

and are both positive. To compute b, differentiate (B.6) with respect to α to obtain

$$b = -\frac{\partial}{\partial\alpha\,\partial\lambda} \log f(x;\alpha,\lambda) = -\frac{1}{\lambda}. \tag{B.11}$$

Let z_1 and z_2 be real numbers, not both equal to zero, and let $\mathbf{z} = \mathrm{col}(z_1, z_2)$. The quadratic form $\mathbf{z}^T[-\nabla^2 q(x;\alpha,\lambda)]\mathbf{z}$ can be written in the form

$$\left(\sqrt{a}z_1 + \frac{b}{\sqrt{a}}z_2\right)^2 + \left(d - \frac{b^2}{a}\right) z_2^2. \tag{B.12}$$

From the second term,

$$d - \frac{b^2}{a} = \psi^{(1)}(\alpha) - \frac{1}{\alpha}, \tag{B.13}$$

which is positive, from (B.5). Thus, both terms in (B.12) are non-negative for all $\mathbf{z}$. For non-zero $\mathbf{z}$, if $z_2 = 0$, the first term in (B.12) will be positive. Otherwise, if $z_2 \neq 0$, the second term will be positive. Therefore, the right-hand side of (B.12) is positive for all non-zero $\mathbf{z}$, which establishes that $q(x;\alpha,\lambda)$ is strictly concave with respect to (α,λ). $\square$

The function $g_i(\alpha,\lambda)$ defined in (4.32) with $\theta = (\alpha,\lambda)$ is a non-negatively weighted sum of log-likelihood functions and thus is itself strictly concave by virtue of Lemma B.1.3.

Proposition B.1.1. The function $g_i(\alpha, \lambda)$ is strictly concave with respect to (α, λ).

The auxiliary function $Q(\phi, \hat{\phi})$ given in (4.31) depends on $(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ only through the second term $\sum_{i=1}^{M} g_i(\alpha_i, \lambda_i)$, which is a separable, strictly concave function with respect to $\{(\alpha_i, \lambda_i)\}_{i=1}^{M}$, in view of (B.1.1).

Proposition B.1.2. The auxiliary function $Q(\phi, \hat{\phi})$ is a separable, strictly concave function with respect to $\{(\alpha_i, \lambda_i)\}_{i=1}^{M}$.

B.2 Inequality of $\psi(\alpha)$

To prove (4.49), we first establish the following results.

Lemma B.2.1. For $t > 0$ and any $\epsilon \geq 1/6$,

$$\frac{1}{2} - \frac{1}{t} + \frac{1}{e^t - 1} \leq \frac{e^{\epsilon t}}{2} - \frac{1}{2}. \tag{B.14}$$

Proof. Let

$$\beta(t; \epsilon) := 1 - \frac{1}{t} + \frac{1}{e^t - 1} - \frac{e^{\epsilon t}}{2} \tag{B.15}$$

Clearly, $\beta(t; \epsilon)$ is continuous and differentiable for $t > 0$. By using L'Hôpital's rule two times we can easily verify $\lim_{t \to 0} \beta(t; \epsilon) = 0$. Taking derivatives, we have

$$\beta'(t; \epsilon) = \frac{1}{t^2} - \frac{e^t}{(e^t - 1)^2} - \frac{\epsilon e^{\epsilon t}}{2} \tag{B.16}$$

$$\beta''(t; \epsilon) = -\frac{2}{t^3} + \frac{2e^{2t}}{(e^t - 1)^3} - \frac{e^t}{(e^t - 1)^2} - \frac{\epsilon^2 e^{\epsilon t}}{2} \tag{B.17}$$

Applying L'Hôpital's rule four times we can verify that

$$\lim_{t \to 0} \beta'(t; \epsilon) = \frac{1}{12} - \frac{\epsilon}{2}. \tag{B.18}$$

Therefore, $\beta'(0^+;\epsilon) \le 0$ for any $\epsilon \ge 1/6$. On the other hand, clearly $e^t - 1 > t$ for $t > 0$. Hence,

$$\frac{1}{(e^t-1)^3} < \frac{1}{t^3} \tag{B.19}$$

and the sum of the first two terms in (B.17) is negative. We then conclude that $\beta(t;\epsilon)$ is a concave function of t for $t > 0$ and any $\epsilon \ge 1/6$. Eq. (B.14) then follows. □

Lemma B.2.2. For $t > 0$,

$$\eta(t) := \frac{1}{2} - \frac{1}{t} + \frac{1}{e^t - 1} \ge 0. \tag{B.20}$$

Proof. Using a similar approach as in Lemma B.2.1, we can show that $\eta(0^+) = 0$ and $\eta(t)$ is increasing at $t = 0$ and convex for $t > 0$. Eq. (B.20) then follows. □

From Binet's first formula (cf. [9, p. 21, Eq. (4)]), for $\alpha > 0$,

$$\log\Gamma(\alpha) = \left(\alpha - \frac{1}{2}\right)\log\alpha - \alpha + \frac{1}{2}\log(2\pi) + \kappa(\alpha) \tag{B.21}$$

where

$$\kappa(\alpha) := -\int_0^\infty \left(\frac{1}{2} - \frac{1}{t} + \frac{1}{e^t-1}\right)\frac{e^{-t\alpha}}{t}\,\mathrm{d}t. \tag{B.22}$$

Taking derivatives on both sides of (B.22),

$$\psi(\alpha) = \log\alpha - \frac{1}{2\alpha} + \kappa'(\alpha), \tag{B.23}$$

or

$$-\kappa'(\alpha) = [\log\alpha - \psi(\alpha)] - \frac{1}{2\alpha}, \tag{B.24}$$

where

$$-\kappa'(\alpha) = \int_0^\infty \left(\frac{1}{2} - \frac{1}{t} + \frac{1}{e^t - 1}\right) e^{-t\alpha}\ \mathrm{d}t. \tag{B.25}$$

Lemma B.2.2 and (B.25) imply that $-\kappa'(\alpha) \geq 0$. Applying this inequality with (B.24) shows the first inequality in (4.49). Next, we apply Lemma B.2.1 with $\epsilon = 1/6$ to (B.25) to obtain

$$-\kappa'(\alpha) \leq \int_0^\infty \left(\frac{e^{\epsilon t}}{2} - \frac{1}{2}\right) e^{-t\alpha}\mathrm{d}t = \frac{1}{2\left(\alpha - \frac{1}{6}\right)} - \frac{1}{2\alpha}. \tag{B.26}$$

Using (B.26) with (B.24) shows the second inequality in (4.49).

B.3 Deriving r_i as in (4.53)

Consider α_i as given by (4.39). In Section 4.2.1, it was shown that $\alpha_i \in (b_i, b_i + 1/6)$, where b_i is given by (4.50) (see (4.49)). In (4.52), it was stated that r_i could take on one of three values: $\lfloor b_i \rfloor \vee 1$, $\lceil b_i \rceil$, or $\lceil b_i \rceil + 1$. We shall show that actually $r_i \neq \lceil b_i \rceil + 1$, thereby justifying the simpler expression in (4.53). We first prove the following lemma.

Lemma B.3.1. For $\alpha_i \geq 1$,

$$h_i(\alpha_i - \epsilon) \geq h_i(\alpha_i - \epsilon + 1) \ \text{ for any } \epsilon \leq \frac{5 - \sqrt{13}}{6}. \tag{B.27}$$

Proof. Using (4.41) and (4.47) and the following relationships involving $\Gamma(\alpha)$ and $\psi(\alpha)$ (see (1) and (8), respectively, in [9])

$$\Gamma(\alpha + 1) = \alpha\Gamma(\alpha), \tag{B.28}$$

$$\psi(\alpha + 1) = \psi(\alpha) + \frac{1}{\alpha}, \tag{B.29}$$

we can derive

$$h_i(\alpha_i - \epsilon + 1) - h_i(\alpha_i - \epsilon) = (\alpha_i - \epsilon + 1) \log\left(1 + \frac{1}{\alpha_i - \epsilon}\right) + \psi(\alpha_i) - \log \alpha_i - 1. \tag{B.30}$$

Next, we apply (4.48) and use the following inequality (see [88, (22)]):

$$\log\left(1 + \frac{1}{x}\right) \leq \frac{x(x+6)}{2(3+2x)}, \quad x \geq 0, \tag{B.31}$$

to obtain

$$\begin{aligned} h_i(\alpha_i - \epsilon + 1) - h_i(\alpha_i - \epsilon) &\leq \frac{1}{4(\alpha_i - \epsilon + 2/3)} - \frac{1}{4(\alpha_i - \epsilon)} - \frac{1}{2\alpha_i} \\ &= -\frac{\left(\frac{1}{3} - \epsilon\right)\alpha_i + \left(\epsilon^2 - \frac{2}{3}\epsilon\right)}{2\alpha_i(\alpha_i - \epsilon)\left(\alpha_i - \epsilon + \frac{2}{3}\right)}. \end{aligned} \tag{B.32}$$

The left-hand side of the inequality (B.32) is less than or equal to zero if the following conditions hold:

$$\alpha_i \geq \epsilon, \ \epsilon < \frac{1}{3}, \ \alpha_i \geq \frac{2\epsilon - 3\epsilon^2}{1 - 3\epsilon}. \tag{B.33}$$

From these conditions and the assumption that $\alpha_i \geq 1$, we have that the left-hand side of (B.32) is less than or equal to zero if

$$\epsilon \leq \frac{5 - \sqrt{13}}{6} \approx 0.232. \tag{B.34}$$

□

If $\alpha_i < 1$, then clearly $r_i = 1$, since the shape parameter of the Erlang distribution must be at least 1. Therefore, we may consider the case $\alpha_i \geq 1$ as in Lemma B.3.1. In this case, if

$$\lfloor b_i \rfloor \leq \alpha_i \leq \lceil b_i \rceil, \tag{B.35}$$

then either $r_i = \lfloor b_i \rfloor$ or $r_i = \lceil b_i \rceil$. Thus, (4.53) holds when either $\alpha_i < 1$ or $\alpha_i \geq 1$ and satisfies (B.35). Next, suppose that $\alpha_i > \lceil b_i \rceil \geq 1$. In this case, by (4.49), we know that $\alpha_i \leq \lceil b_i \rceil + \frac{1}{6}$. Applying Lemma B.3.1, we conclude that $h\left(\lceil b_i \rceil\right) \geq h_i\left(\lceil b_i \rceil + 1\right)$. Thus, we have established the validity of (4.53).

B.4 Existence of the Stationary Point α_i

In this section we prove a root to the equation (4.47) always exists. By taking derivative of $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ with respect to α_i we can show the existence of the stationary point or the root of (4.47). For the sake of convenience we have repeated $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ here.

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}) = \sum_{i=1}^{M} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\pi_i) \cdot q(i|x_k, \hat{\mathbf{\Theta}}) + \sum_{i=1}^{M} g_i(\theta_i) + N_0 \log(\pi_{M+1}), \tag{B.36}$$

where

$$g_i(\theta_i) = \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(p_i(x_k|\lambda_i)) \cdot q(i|x_k, \hat{\mathbf{\Theta}}), \tag{B.37}$$

and $\theta_i = (\alpha_i, \lambda_i)$, $i = 1, \ldots, M$. For any fixed value of λ_i, by taking partial derivative of the (B.36) with respect to α_i and using (4.21), we will have

$$\frac{\partial Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(p_i(x_k|\lambda_i)) \cdot q(i|x_k, \hat{\Theta}) = \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\lambda_i x_k) q(i|x_k, \hat{\Theta})$$

$$- \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \frac{\Gamma'(\alpha_i)}{\Gamma(\alpha_i)} q(i|x_k, \hat{\Theta}) = D_i - \frac{\Gamma'(\alpha_i)}{\Gamma(\alpha_i)} A_i \tag{B.38}$$

where A_i is defined in (4.42), and D_i is defined as

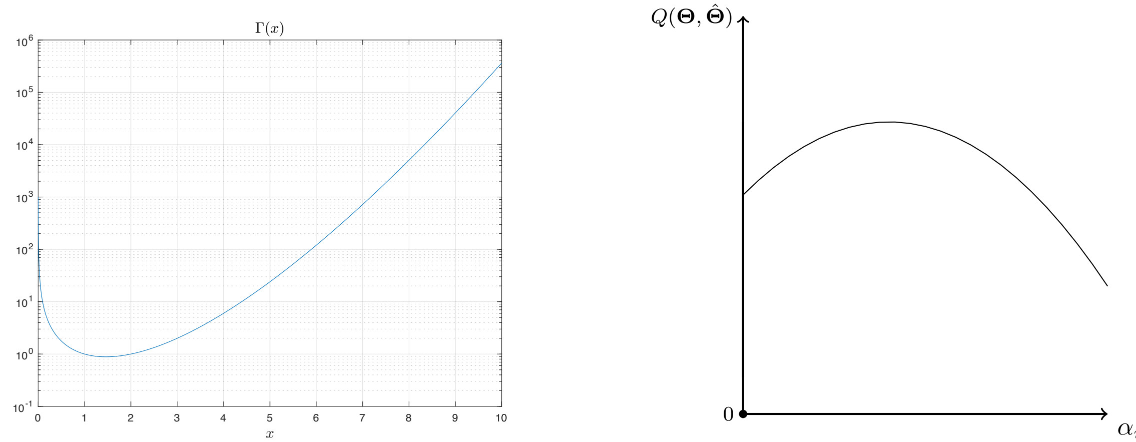
$$D_i := \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\lambda_i x_k) q(i|x_k, \hat{\Theta}). \tag{B.39}$$

In (B.38), A_i and D_i do not depend on α_i and therefore in maximization of $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ with respect to α_i, can be considered as constant. Clearly A_i is always greater or equal to zero. It is shown in Appendix B.1, $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ is concave with respect to r_i.

Gamma function is a convex positive function for $x \in (0, \infty)$. We have $\lim_{x\to 0} \Gamma(x) = \infty$ and $\lim_{x\to\infty} \Gamma(x) = \infty$. Gamma function is shown in fig. B.1a for $x \in [1e{-}3, 10]$. As we can see in the figure it decreases to a value less than 1 at about 1.5 and then it is monotonically increasing afterwards. Therefore as $A_i \geq 0$, and D_i being constant with respect to α_i the second term in (B.38) will become eventually negative. Therefore, we have

$$\lim_{\alpha_i \to \infty} \frac{\partial Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})}{\partial \alpha_i} = -\infty \tag{B.40}$$

On the other hand, we have $\lim_{\alpha_i\to\infty} \psi(\alpha_i) = \lim_{\alpha_i\to\infty} \frac{\Gamma'(\alpha_i)}{\Gamma(\alpha_i)} = -\infty$. Therefore, $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ will be in a form as shown in fig. B.1b. Therefore, for every fixed value λ_i, $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ will be maximized at a positive α_i. In (4.38), however, λ_i is not a fixed value and it is linearly



(a) $\Gamma(x)$ for $x \in [10^{-3}, 10]$

(b) $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$

related to α_i as in (4.40). This however, does not change the sign of α_i, as in this case the stationary point for each i lies on the intersection of two-dimensional $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ surface and the line (4.38), which because of the form of $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$ still will be positive. Therefore, there is always a positive root to the equation (4.47) and the stationary point is always a valid positive value for α_i for each $i = 1, 2, \ldots, M$.

Appendix C: Numerical Results for Alg-Er

In this section we present numerical results comparing the performance of three fitting algorithms for phase-type distributions:

1. Alg-Er with a fixed number M of Erlang mixture components;

2. HErD (Hyper-Erlang Distribution): The algorithm in [87] in which the shape vector $\mathbf{r}$ ranges over the set of partitions, $\mathcal{R}_n$, of a fixed number of phases n;

3. GPHD (General Phase-type Distribution): The general phase-type EM fitting algorithm of [8] with n phases.

Similar to [87], we have obtained results for both synthetically generated data from a set of well-known probability distributions and from real data traces from a call center. We have also compared Alg-Er and HErD in estimating the waiting time distribution of a heavy-tailed M/G/1 queue. In addition, we have compared the performance of Alg-Er and Alg-Ga.

C.1 Synthetically Generated Data

Following [87], we have used synthetic data generated from distributions given in Table C.1, which include two heavy-tailed distributions: a Pareto-like distribution denoted here as Pareto2 and a 2-parameter Weibull distribution. In particular, we have used the Pareto2 distribution with parameters $a = 1.5$ and $b = 2$ and two instances of the Weibull distribution: Weibull with scale parameter $\alpha = 1$ and shape parameter $\beta = 0.5$ and Weibull with $\alpha = 1$ and $\beta = 5$. The Weibull$(1, 5)$ distribution was used in [43] as an example of a heavy-tailed distribution with a non-monotonically decreasing pdf. Similarly, Pareto2$(1.5, 2)$ is heavy-tailed with non-monotonically decreasing pdf, whereas Weibull$(1, 0.5)$ is an example of a heavy-tailed distribution with monotonically decreasing pdf. The uniform distribution is an example of a distribution with finite support. The shifted exponential and matrix exponential distributions were among those used as benchmarks for phase-type fitting in [13]. Each of the synthetic data traces consisted of 10^4 samples generated according to the distributions shown in Table C.1. To generate random samples from a given distribution, we applied the function $g(y) = F^{-1}(y)$ to uniform random samples of $y \sim \text{Uniform}(0, 1)$, where $F(t)$ denotes the cdf. The cdfs for each distribution are also provided in Table C.1. In Table C.1, $\Gamma^u_{\text{inc}}(x, a)$ denotes the upper incomplete Gamma function and is defined as [4, Eq. (6.5.1)]

$$\Gamma^u_{\text{inc}}(x, a) = \int_x^\infty t^{a-1} e^{-t} \mathrm{d}t \tag{C.1}$$

For the Pareto2 and matrix exponential distributions, for which closed forms for $F^{-1}(y)$ do not exist, the function $g(y) = F^{-1}(y)$ was computed numerically.

For all three methods, we set $\epsilon = 10^{-6}$ for the stopping criterion. A limit of $N = 100$ iterations of the EM algorithm was imposed for Alg-Er and HErD. For GPHD, the stopping criterion consisted of a limit of $N = 50,000$ EM iterations and a maximum run time of 10 hours. The Alg-Er was programmed in MATLAB, while the MATLAB code for HErD from [1] was used. An off-the-shelf C language implementation from [76] was used to run

Table C.1: Probability distributions defined on $[0, \infty)$

name	pdf	cdf
Weibull	$f(t; \beta, \alpha) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} e^{(\frac{t}{\alpha})^\beta}$	$F(t; \beta, \alpha) = 1 - e^{(\frac{t}{\alpha})^\beta}$
Pareto2	$f(t; a, b) = \frac{b^a e^{-b/t}}{\Gamma(a)} t^{-a-1}$	$F(t; a, b) = \frac{1}{\Gamma(a)} \Gamma^u_{\text{inc}}\left(\frac{b}{x}, a\right)$
Shifted Exp.	$f(t) = \begin{cases} \frac{1}{2}e^{-t} & 0 \leq t \leq 1 \\ \frac{1}{2}e^{-t} + \frac{1}{2}e^{-(t-1)} & t \geq 1 \end{cases}$	$F(t) = \begin{cases} \frac{1}{2} - \frac{1}{2}e^{-t} & 0 \leq t < 1 \\ 1 - \frac{1}{2}e^{-t} - \frac{1}{2}e^{-(t-1)} & t \geq 1 \end{cases}$
Matrix Exp.	$f(t) = \left[1 - \frac{1}{(2\pi)^2}\right](1 - \cos(2\pi t))e^{-t}$	$F(t) = 1 - \frac{4\pi^2 + 1}{\pi^2} e^{-t} - \frac{1}{2\pi} \sin(2\pi t) e^{-t} + \frac{1}{4\pi^2} \cos(2\pi t) e^{-t}$
Uniform	$f(t) = \frac{1}{b-a}, \ a \leq t \leq b$	$F(t) = \frac{t-a}{b-a}$ for $a \leq t \leq b$

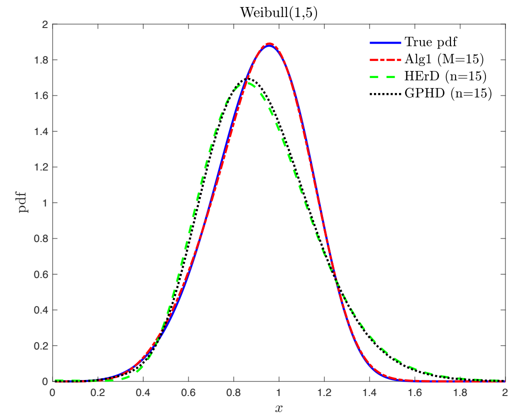


Figure C.1: Weibull$(1, 5)$ distribution with order $= 15$.

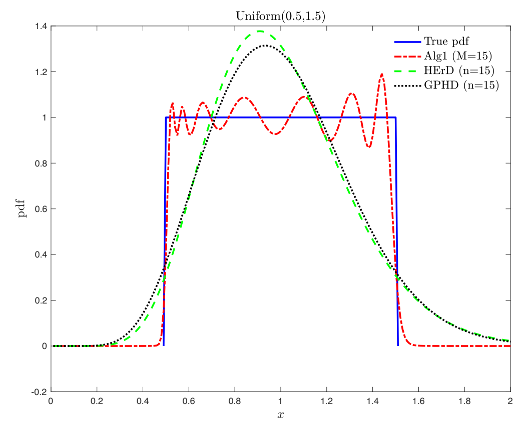


Figure C.2: Uniform$(0.5, 1.5)$ distribution with order $= 15$.

the GPHD method. In spite of the greater efficiency of C vs. MATLAB, when the number of phases of the fitting distribution exceeds 10, the run times for GPHD were far longer than those for Alg-Er and HErD. All of the codes were run on a MacBook Pro with a 2.9 GHz Intel Core i7 processor and 16 GB 2133 MHz LPDDR3 memory. In our numerical experiments, Alg1-Er usually ran for less than 6 seconds, whereas the GPHD implementation took up to the full 10 hours imposed as the maximum run time.

We have compared HErD and GPHD using a fixed number of phases n. For Alg-Er, the number of mixture components M is fixed, while n is allowed to vary. To compare Alg-Er

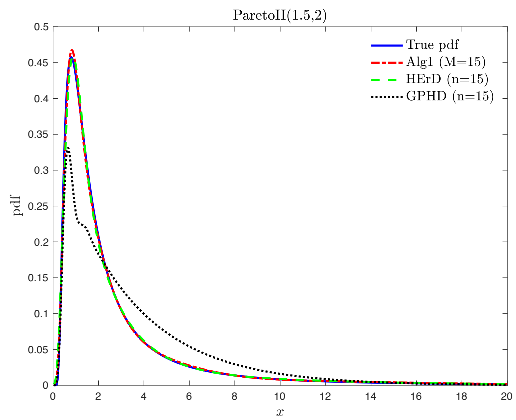


Figure C.3: Pareto2$(1.5, 2)$ distribution with order $= 15$.

with the other two methods, we set M equal to the value of n used for HErD and GPHD. For convenience, we will refer to the value of n used for HErD or GPHD or the value of M used for Alg-Er simply as the *order* of the method. Figures C.1, C.2, and C.3 provide qualitative comparisons of the performance of the three fitting methods applied to samples generated from Weibull, uniform, and Pareto distributions, respectively. More detailed, quantitative comparisons of the three methods are given in Tables C.2 and C.3.

Table C.2 shows results comparing the three phase-type fitting algorithms with data samples generated using the distributions given in Table C.1 where the order for the three methods is 5 and 15. A similar table for the case an order of 25 is given in Table C.3. For brevity, Table C.3 only shows the results for Weibull$(1, 5)$, Uniform$(0.5, 1.5)$, and Pareto2$(1.5, 2)$. In general, the number of phases obtained with Alg1-Er will be much larger than the number of mixture components. On the other hand, the run-time, denoted by t_c, in units of seconds, for Alg-Er is comparable to that of HErD and GPHD when the order is relatively small, i.e., 5. As can be seen from Tables C.2 and C.3, when the order is 15 or 25, the run-time for Alg is much smaller than that of the other two methods and the gap increases with increasing order.

We have computed the first three moments of the synthetic traces, denoted by μ_1, μ_2, and μ_3, and the corresponding moments of the three fitting distributions for Alg-Er, HErD, and GPHD. All three methods were initialized with parameters such that the first moment of the fitting distribution matches the first moment of the trace data (cf. (4.55)). The first moment matching property is maintained by each iteration of the corresponding EM algorithm and as can be seen in Tables C.2 and C.3, the first moment is always matched perfectly using all three methods. We have also computed the squared coefficient of variation defined by $c^2 = \mu_2/\mu_1^2 - 1$. The log-likelihood ℓ (see Algorithm 4 lines 2 and 12) of each data trace has been computed for the three methods. For Alg1-Er and HErD, the Erlang shape vector $\mathbf{r}$ is also displayed.

As shown in the first row of Tables C.2 and C.3, for Weibull$(1, 0.5)$, both Alg-Er and HErD, the fitting distributions are mixtures of exponentials. This result confirms that monotonically decreasing pdfs can be well-approximated by the hyper-exponential pdf (see [34]). It is interesting that for the Weibull$(1, 0.5)$ fitting Alg-Er and HErD yield the same parameter estimates. In general, as the order of the method is increased from 5 to 15 and 20, better matching of the moments and higher log-likelihood values is obtained. Interestingly, for Weibull$(1, 0.5)$, GPHD outperforms the other two methods with respect to the log-likelihood value when the order is 5. However, when the order is increased to 15 or 25, Alg and HErD outperform GPHD in terms of the log-likelihood value for all of the fitting distributions, due to the overparameterization of the general phase-type representation. In almost all cases shown in Tables C.2 and C.3, Alg-Er outperforms HErD and GPHD with respect to fitting accuracy, as well as computation time, especially when the order is 15 or 25.

C.2 Call Center Data Traces

As was done in [87], we have also used some call center data traces from [70] to compare Alg-Er, HErD, and GPHD. The data archives calls handled by a bank over a period of 12 months from January 1999 to December 1999. For every month there are about 20,000 to 30,000 entries in this data set. Each entry in this data set includes several attributes of the

Table C.2: Comparison of HErD, Alg-Er and GPHD with synthetically generated data

		Trace	order = 5 HErD	 Alg-Er	 GPHD	order = 15 HErD	 Alg-Er	 GPHD
μ_1	Weibull(1, 0.5)	1.99	1.99 (0.0%)	1.99 (0.0%)	1.99 (0.0%)	1.99 (0.0%)	1.99 (0.0%)	1.99 (0.0%)
μ_2		22.1	20.7 (6.0 %)	20.7 (6.0%)	20.2 (8.6%)	21.7 (1.8%)	21.7 (1.8%)	13.4 (3.9%)
μ_3		549.3	421.8 (23.2%)	421.8 (23.2%)	381.8 (30.5%)	500.6 (8.9%)	500.6 (8.9%)	139.0 (74.7%)
c^2		4.59	4.26 (7.3%)	4.26 (7.3%)	4.11(10.5%)	4.49 (2.2%)	4.49 (2.2%)	2.39 (48.0%)
ℓ			-11345.1	-11345.1	-11283.1	-11298.3	-11298.3	-12605.4
t_c			0.4	0.3	seconds	23.8	0.8	minutes
r			[1,1,1,1,1]	[1,1,1,1,1]	5	[1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1]	[1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1]	15
μ_1	Weibull(1, 5)	0.92	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)
μ_2		0.89	1.05 (18.7%)	0.89 (0.0%)	1.01 (14.0%)	0.90 (1.8%)	0.89 (0.0%)	0.90 (1.7%)
μ_3		0.90	1.45 (62.2%)	0.90 (0.1%)	1.30 (45.3%)	0.95 (6.2%)	0.90 (0.0%)	0.95 (5.6%)
c^2		0.05	0.25 (371.6%)	0.05 (0.7%)	0.2 (277.3%)	0.07 (36.8%)	0.05 (0.4%)	0.07 (33.7%)
ℓ			-2520.3	1414.7	-1665.2	881.5	1423.8	976.8
t_c			0.2	2.7	seconds	16.6	8.9	minutes
r			[1,4]	[32,51,21,16,11]	-	[1,14]	[35,127,68,47,31, 25,21,19,17,18, 17,16,14,13,9]	-
μ_1	Uniform(0.5, 1.5)	1.00	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)
μ_2		1.08	1.25 (15.4%)	1.08 (0.0%)	1.20 (10.1%)	1.09 (0.7%)	1.08 (0.0%)	1.09 (1.0%)
μ_3		1.25	1.87 (50.2%)	1.25 (0.0%)	1.68 (34.6%)	1.29 (3.3%)	1.25 (0.0%)	1.29 (3.6%)
c^2		0.083	0.25 (202.1%)	0.083 (0.0%)	0.20 (141.6%)	0.091 (9.8%)	0.08 (0.1%)	0.093 (12.7%)
ℓ			-3804.7	-294.5	-3096.2	-1755.3	-179.4	-1688.4
t_c			0.16	14.2	seconds	47.5	96.9	minutes
r			[1,4]	[130,555,43,97,449]	-	[1,1,1,1,11]	[49,1815,376,141,36, 35,72,63,154,223, 218,610,1083,911,911]	-
μ_1	Pareto2(1.5, 2)	3.609	3.61 (0.0%)	3.61 (0.0%)	3.61 (0.0%)	3.61 (0.0%)	3.61 (0.0%)	3.61 (0.0%)
μ_2		69.276	76.0 (9.7%)	64.2 (7.3%)	53.5 (22.7%)	73.3 (5.8%)	71.4 (3.1%)	23.7 (65.7%)
μ_3		4002.8	5890.7 (47.1%)	3272.4 (18.2%)	1874.8 (53.2%)	5288.3 (32.1%)	4774.8 (19.3%)	233.3 (94.2%)
c^2		4.32	4.84 (11.9%)	3.93 (9.0%)	3.11 (28%)	4.63 (7.1%)	4.48 (3.8%)	0.82 (80.9%)
ℓ			-20305.8	-20014.0	-20125.3	-19981.5	-19965.5	-21981.7
t_c			0.22	0.65	seconds	19.3	2.54	minutes
r			[1,1,3]	[1,7,6,5,5]	-	[1,2,3,4,5]	[1,8,10,10,10,10,9, 8,7,7,6,6,6,6,6]	-
μ_1	Shifted Exponential	1.493	1.49 (0.0%)	1.49 (0.0%)	1.49 (0.0%)	1.49 (0.0%)	1.49 (0.0%)	1.49 (0.0%)
μ_2		3.465	3.44 (0.6%)	3.52 (1.5%)	3.46 (0.2%)	3.47 (0.2%)	3.45 (0.5%)	3.46 (0.2%)
μ_3		10.803	10.5 (2.4%)	11.6 (7.4%)	10.6 (1.5%)	10.8 (0.05%)	10.6 (2.1%)	10.6 (1.9%)
c^2		0.555	0.55 (1.8%)	0.58 (4.1%)	0.55 (0.5%)	0.56 (0.5%)	0.55 (1.5%)	0.55 (0.6%)
ℓ			-13263.9	-12998.4	-13144.0	-13088.3	-12927.4	-13187.6
t_c			1.17	1.4	seconds	92.0	5.1	minutes
r			[1,4]	[1,9,53,5,2]	-	[1,4,10]	[1,9,12,14,17,52,117, 12,7,4,3,2,2,2,3]	-
μ_1	Matrix Exponential	1.064	1.06 (0.0%)	1.06 (0.0%)	1.06 (0.0%)	1.06 (0.0%)	1.06 (0.0%)	1.06 (0.0%)
μ_2		2.134	2.17 (1.8%)	2.27 (6.5%)	2.22 (4.1%)	2.25 (5.4%)	2.15 (1.0%)	2.17 (1.5%)
μ_3		6.659	6.71 (0.9%)	7.95 (19.5%)	7.36 (11%)	8.33 (25%)	7.08 (6.3%)	6.83 (2.5%)
c^2		0.884	0.92 (4.0%)	1.00 (12.8%)	0.96 (8.8%)	0.99 (11%)	0.90 (1.7%)	0.91 (3.2%)
ℓ			-9319.5	-8179.7	-9036.6	-8832.9	-7685.6	-8399.7
t_c			0.27	2.4	seconds	19.55	6.6	minutes
r			[2,3]	[2,98,5,20,6]	-	[1,7,7]	[1,213,76,64,21,6,44, 20,16,13,9,8,7,5,5]	-

Table C.3: Comparison of HErD, Alg-Er and GPHD for synthetically generated data

		order = 25		
		HErD	Alg-Er	GPHD
μ_1	Weibull(1, 5)	0.92 (0.0%)	0.92 (0.0%)	0.92 (0.0%)
μ_2		0.89 (0.8 %)	0.89 (0.0%)	0.89 (0.4%)
μ_3		0.92 (2.6%)	0.90 (0.0%)	0.91 (1.4%)
c^2		0.06 (15.6%)	0.05 (0.4%)	0.06 (8.2%)
ℓ		1159.8	1433.5	1311.9
t_c		661.6	17.8	hours
$\mathbf{r}$		[5,20]	[37,611,141,85,63,43, 32,28,24,22,20,19,18, 18,19,18,18,17,17, 15,15,14,11,10,10]	-
μ_1	Uniform(0.5, 1.5)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)
μ_2		1.09 (1.2%)	1.08 (0.0%)	1.09 (0.6%)
μ_3		1.30 (4.5%)	1.25 (0.0%)	1.28 (2.9%)
c^2		0.10 (15.8%)	0.08 (0.1%)	0.09 (8.0%)
ℓ		-1716.6	-166.3	-1754.4
t_c		1731.1	190.2	hours
$\mathbf{r}$		[12,13]	[54,1811,944,387,191,65, 38,34,35,48,56,67,100,226 214,209,332,670,954, 810,622,909,909,909,909]	-
μ_1	Pareto2(1.5, 2)	3.61 (0.0%)	3.61 (0.0%)	3.61 (0.0%)
μ_2		71.5 (3.2%)	74.6 (7.8%)	23.7 (65.8%)
μ_3		4792.8 (19.7%)	5668.4 (41.6%)	232.6 (94.2%)
c^2		4.49 (3.9%)	4.73 (9.5%)	0.82 (81.0%)
ℓ		-19952.9	-19957.4	-22041.2
t_c		672.7	4.74	hours
$\mathbf{r}$		[2,2,3,4,6,8]	[1,8,9,10,11,11, 11,10,10,10, 9,9,8,8,7,7,7,7, 6,6,6,6,6,6]	-

handled calls, like, service time, waiting time, etc. We have used the service time attribute and tried to fit a distribution to the empirical one. We have scaled the data to have a sample mean of 1.

As can be seen from the results in Table C.4. For these two data traces, Alg-Er and HErD perform much better than GPHD, both with respect to fitting accuracy and computation time. The case of January data is the only case in which HErD outperforms Alg-Er in terms of log-likelihood value. Even in this case, the performance of Alg-Er is very close to that of HErD. However, in terms of CPU-time, Alg-Er is much faster than HErD. By increasing M to 40, Alg-Er achieved better performance than HErD in terms of log-likelihood value with a run-time of only 16 seconds.

Table C.4: Comparison of HErD, Alg-Er, and GPHD for two Call Center Traces

		Trace	order = 15 HErD	Alg-Er	GPHD	order = 25 HErD	Alg-Er	GPHD
μ_1	January 1999	1.00	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)
μ_2		2.57	2.47 (3.58%)	2.48 (3.29%)	2.57 (0.0%)	2.49 (2.95%)	2.50 (2.41%)	2.01 (21.9%)
μ_3		14.78	11.90 (19.5%)	12.12 (18.0%)	14.82 (2.8%)	12.27 (17.0%)	12.71 (14.0%)	6.03 (59.2%)
c^2		1.57	1.47 (5.9%)	1.48 (5.4%)	1.57 (0.0%)	1.49 (4.8%)	1.50 (3.9%)	1.01 (35.8%)
ℓ			-25551.8	-25721.4	-25519.5	-25521.2	-25709.2	-26996.3
t_C			52.3	6.3	minutes	1169.9	11.1	hours
$\mathbf{r}$			[2,2,2,4,5]	[1,7,10,11,11,11, 10,9,8,7,7,5,7,3,1]	-	[2,2,2,3,3,3,3,7]	[1,6,10,11,11,11,11,11, 11,11,11,10,10,9,8, 8,7,8,7,7,7,3,1,1,1]	-
μ_1	December 1999	1.00	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)	1.00 (0.0%)
μ_2		2.73	2.47 (9.5%)	2.58 (5.6%)	2.52 (7.7%)	2.49 (8.9 %)	2.62 (4.2%)	1.83 (3.3%)
μ_3		23.51	11.90 (49.4%)	14.49 (38.3%)	13.04 (38.3%)	12.27 (47.8%)	15.83 (32.7%)	4.99 (32.7%)
c^2		1.73	1.47 (14.9%)	1.58 (8.8%)	1.52 (12.2%)	1.49 (14.0%)	1.62 (6.6%)	0.83 (52.0%)
ℓ			-32476.6	-31880.9	-31801.3	-32437.8	-31871.1	-33292.2
t_C			51.9	6.9	minutes	1203.6	11.7	hours
$\mathbf{r}$			[2,2,2,4,5]	[1,8,10,11,11,10,8, 6,5,5,5,4,4,2,19]	-	[2,2,2,3,3,3,3,7]	[1,8,10,11,11,11,11, 11,10,9,8,6,5,5,5,5, 5,4,4,4,3,3,2,2,1]	-

C.3 Workload fitting in a M/G/1 heavy-tailed queue

In this section we try to fit the workload samples in a M/G/1 heavy-tailed queue using our algorithm. In this example we have generated 10^6 samples based on the heavy-tailed distribution for the workload in a M/G/1 queue, which is presented in [61, eq. (25)]. What is particular about this example is that a considerable portion of the workload samples are 0. This comes from the fact that in a M/G/1 queue, queue is empty with probability of $1 - \rho$, where ρ is the utilization factor of the queue. Therefore, in this case we are trying to fit samples with a considerable portion of 0 ones with hyper-Erlang model. Therefore, we will have a mass at absorbing state of the Markov model and we can see the advantage of the likelihood with respect to the measure function of Lebesgue and Dirac function in here. As our algorithm is designed such a way to accommodate for the case with mass at absorbing state, we can see the satisfactory resulting hyper-Erlang fit to the samples in figure C.4. In this example we have compared our algorithm Alg-Er with HErD. Actually, in this case that we have a mass at absorbing state, likelihood in HErD algorithm is not a suitable measure. In HErD algorithm, we have the order of the phase-type as n, and the Erlang orders as $(r_1, r_2, \ldots, r_M)$, where M is the number of Erlang branches. Therefore, for the case of having samples with a considerable portion of 0 ones, the incomplete data log-likelihood will be as follows

$$\log \ell(\boldsymbol{\Theta}|\mathbf{x}) = \log \prod_{k=1}^{K} p(x_k|\boldsymbol{\Theta}) = \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\sum_{i=1}^{M} \pi_i p_i(x_k|\lambda_i))$$

$$+ \sum_{\substack{k=1 \\ x_k = 0}}^{K} \log(\sum_{i=1}^{M} \pi_i p_i(x_k|\lambda_i)) = \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\sum_{i=1}^{M} \pi_i p_i(x_k|\lambda_i)) + K_0 \log(\sum_{i=1}^{M} \pi_i p_i(0|\lambda_i)) \quad \text{(C.2)}$$

, where K_0 is the number of 0 samples. But, for $p_i(0|\lambda_i)$ for $i = 1, 2, \ldots, M$ we have

$$p_i(0|\lambda_i) = \begin{cases} \lambda_i & if \quad r_i = 1 \\ 0 & if \quad r_i > 1 \end{cases} \quad \text{(C.3)}$$

Therefore, the incomplete data log-likelihood will simplify to

$$\log L(\boldsymbol{\Theta}|\mathbf{x}) = \sum_{\substack{k=1 \\ x_k \neq 0}}^{K} \log(\sum_{i=1}^{M} \pi_i p_i(x_k|\lambda_i)) + K_0 \log(\sum_{\substack{i=1 \\ r_i = 1}}^{M} \pi_i \lambda_i) \quad \text{(C.4)}$$

Because we have a considerable portion of samples as 0, at least one of the branches will be of order 1 with $p_i(x|\lambda_i) = \lambda_i e^{-\lambda_i x}$, with $\lambda_i \to \infty$, which is an exponential pdf going toward a $\delta(x)$ function, and models the absorbing state. Therefore, the incomplete data log-likelihood will be summation of a $\delta(x)$ and other Erlang pdfs, or according to (C.4), $\lim_{\lambda_i \to \infty} \log L(\boldsymbol{\Theta}|\mathbf{x}) = \infty$. Therefore, any effort in maximization of such a function will result in $\lambda_i = \infty$. Whereas, in (4.26), because of using measure of summation of Lebesgue and Dirac function, we will not have this case and log-likelihood never goes to infity. The better performance of our algorithm in compare to HErD can be seen in figure C.4. In our simulations we also tried GPHD method, but as it took a long time and the results were not as good as the other two methods, we omitted the fitting derived by GPHD. Also in HErD method, because of getting very large value for one of the λ (in order of 10^{180}), it is

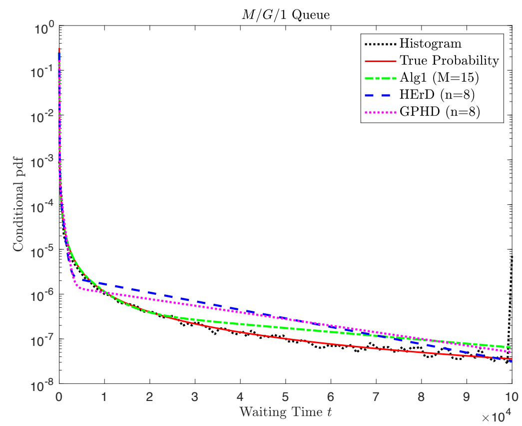


Figure C.4: Densities of fitted distributions using Alg-Er and HErD for M/G/1 queue workload samples

impossible to try phase-type orders of higher than 6, whereas in our algorithm as can be seen in the figure, very close fitting distributions can be achieved in order of 2 seconds with 15 branches.

Appendix D: Proof of theorems about stochastic traffic regulator

D.1 Proof of Proposition 5.3.1

We first establish the following lemma[1].

Lemma D.1.1.

$$W(\tilde{s}_j; A_1) = W(s_j; A_\mathrm{i}) - (\tilde{s}_j - s_j)\rho \tag{D.1.1}$$

Proof. We prove (D.1.1) using induction. For $j = 1$, i.e., the first packet arrival, $\tilde{s}_1 = s_1$ and $W(\tilde{s}_j; A_1) = W(s_j; A_\mathrm{i}) = 0$, so (D.1.1) holds in this case. Assuming (D.1.1) is valid for the jth packet, we now verify that it holds for the $(j+1)$st packet. Note that in the interval $(\tilde{s}_j, \tilde{a}_j)$, the workload function $W(t; A_1)$ increases linearly with slope $C - \rho$ by an amount δ_j (see (5.11)) and then decreases linearly with slope $-\rho$ in the interval $(\tilde{a}_j, \tilde{s}_{j+1})$ (see (5.20) and (5.21)). Hence,

$$W(\tilde{s}_{j+1}, A_1) = W(\tilde{s}_j, A_1) + \delta_j - \rho(\tilde{s}_{j+1} - \tilde{a}_j). \tag{D.1.2}$$

By a similar argument (see (5.12)-(5.13)),

$$W(s_{j+1}, A_\mathrm{i}) = W(s_j, A_\mathrm{i}) + \delta_j - \rho(s_{j+1} - a_j). \tag{D.1.3}$$

Next, we apply first (D.1.1) and then (D.1.3) into (D.1.2) and re-arrange terms to obtain

$$W(\tilde{s}_{j+1}, A_1) = W(s_{j+1}, A_\mathrm{i}) - \rho(\tilde{s}_{j+1} - s_{j+1}) + [(\tilde{a}_j - \tilde{s}_j) - (a_j - s_j)]\rho. \tag{D.1.4}$$

The last term in (D.1.4) vanishes, since $\tilde{a}_j - \tilde{s}_j = a_j - s_j = L_j/C$. Thus, we have established

[1] For notational convenience we drop the subscript ρ when referring to workload functions $W(\cdot;\cdot)$.

(D.1.1) using mathematical induction. $\square$

Proof of Proposition 5.3.1. First, suppose $W(\tilde{s}_j; A_1) \le \sigma$. Since $W(\tilde{s}_j; A_1) = W(\tilde{s}_j; A_\mathrm{o})$ (see (5.22)), we have $W(\tilde{s}_j; A_\mathrm{o}) \le \sigma$, i.e., the (σ, ρ) constraint is satisfied by the output process at time $\tilde{s}_j$. This implies that the jth packet departs the regulator starting at time $t_j = \tilde{s}_j$, which confirms (5.18) in this case.

Next, suppose $W(\tilde{s}_j; A_1) > \sigma$. Then $W(s_j; A_\mathrm{i}) \ge W(s_j; A_1) \ge W(\tilde{s}_j; A_1) > \sigma$. Thus, in this case, we can remove the $[\cdot]^+$ operator in both (2.12) and (5.18). Applying Lemma D.1.1 to the right-hand side of (5.18), we have

$$\begin{aligned}
[W(\tilde{s}_j; A_1) - \sigma]/\rho + \tilde{s}_j &= [W(s_j; A_\mathrm{i}) - (\tilde{s}_j - s_j)\rho - \sigma]/\rho + \tilde{s}_j \\
&= [W(s_j; A_\mathrm{i}) - \sigma]/\rho + s_j = t_j.
\end{aligned}$$

This completes the proof of Proposition 5.3.1. $\square$

D.2 Proof of Theorem 5.4.2

The proof of Theorem 5.4.2 is based on the following two lemmas.

Lemma D.2.1. Let $\mathcal{B}_j$ be as defined in (5.46), let $k = \min \mathcal{B}_j$, and $\mathcal{I}_j$ is defined as

$$\mathcal{I}_j = \big\{2 \le \ell \le k : \; o_{T_i}(b_j(\sigma_\ell)) \le \bar{f}(T_i) - \epsilon_{i,j}(\sigma_\ell), \quad \forall i = 1, \ldots, \ell - 1\big\}, \tag{D.2.1}$$

where

$$\epsilon_{i,j}(\sigma_\ell) = \begin{cases} \frac{W(b_j(\sigma_\ell); A_\mathrm{o}) - T_i}{\rho b_j(\sigma_\ell)}(1 - \bar{f}(T_i)) & i \in \{1, \ldots, \ell - 2\} \\ \bar{f}(T_{\ell-1}) - \bar{f}(T_\ell) & i = \ell - 1, \end{cases} \tag{D.2.2}$$

where $t_j(\sigma_\ell)$ and $b_j(\sigma_\ell)$ are given by (5.18) and (5.9), respectively. Set

$$i^* = \begin{cases} \max \mathcal{I}_j, & \mathcal{I}_j \neq \emptyset, \\ 1, & \text{otherwise.} \end{cases} \tag{D.2.3}$$

If the burst parameter $\sigma^*(j)$ is set as follows:

$$\sigma^*(j) = \sigma_{i^*}, \tag{D.2.4}$$

then

$$o_{T_i}(t) \leq \bar{f}(T_i), \qquad t \in [b_{j-1}, b_j(\sigma_{i^*})], \quad \forall\ i \in \{1, 2, \ldots, M\}. \tag{D.2.5}$$

Proof. We proof (D.2.5) using induction. Note that, for $j = 1$ we have $\mathcal{I}_j = \emptyset$ and according to (5.23)

$$O_{T_i}(t; A_\text{o}) = 0 \quad \forall\ i \in \{1, 2, \ldots, M\}. \tag{D.2.6}$$

Therefore (D.2.5) holds for $j = 1$. Lets assume (D.2.5) is valid for the jth packet, we now verify it holds for the $(j+1)$th packet. We assume $\sigma^*(j) = \sigma_m$, where $m \in \mathcal{I}_j$. Therefore $b_j = b_j(\sigma_m)$ and according to (D.2.1) and assumption (D.2.5)

$$o_{T_i}(b_j) \leq \bar{f}(T_i), \quad \forall\ i \in \{1, 2, \ldots, M\}, \tag{D.2.7}$$

$$o_{T_i}(b_j) \leq \bar{f}(T_i) - \epsilon_{i,j}(\sigma_m), \quad \forall i \in \{1, 2, \ldots, m-1\}, \tag{D.2.8}$$

where $\epsilon_{i,j}(\sigma_m)$ is defined in (D.2.2). As $\sigma_{m+1} > W(b_j; A_\text{o}) \geq W(t_{j+1}; A_\text{o})$, therefore $\sigma^*(j+1)$ can be chosen from $\{\sigma_1, \ldots, \sigma_m, \sigma_{m+1}\}$. We define $\hat{W}(t; A_\text{o})$ as the decreasing workload with slope ρ from $W(b_j; A_\text{o})$ as shown in Fig. D.1. Also, if $\sigma^*(j+1)$ is set as $\sigma^*(j+1) = \sigma_\ell$ for $\ell \in \{1, 2, \ldots, m+1\}$ as in Fig. D.1, we have

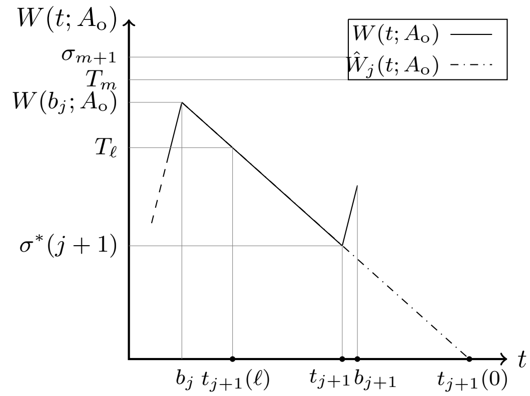


Figure D.1: $W(t; A_o)$, when $\sigma^*(j) = \sigma_m > \sigma_1$ and $\sigma^*(j+1) \in \{\sigma_1, \ldots, \sigma_m\}$.

$$W(t; A_o) = \hat{W}(t; A_o), \qquad t \in [b_j, t_{j+1}(\sigma_\ell)]. \tag{D.2.9}$$

We define $t_{j+1}(i)$ as the time that

$$\hat{W}(t_{j+1}(i); A_o) = T_i, \quad \text{for} \quad i \in \{0, 1, \ldots, m\}, \tag{D.2.10}$$

with $T_0 = 0$. Therefore, according to (5.19), we have

$$t_{j+1}(i) = b_j + \frac{W(b_j; A_o) - T_i}{\rho}. \tag{D.2.11}$$

According to Proposition 5.4.1 and Fig. D.1, we have

$$O_{T_i}(t; A_o) = O_{T_i}(b_j; A_o), \quad t \in [b_j, b_{j+1}], \quad i \in \{m, m+1, \ldots, M\}. \tag{D.2.12}$$

Therefore, according to (D.2.7)

$$o_{T_i}(t) \leq \bar{f}(T_i), \quad t \in [b_j, b_{j+1}], \quad i \in \{m, m+1, \ldots, M\}. \tag{D.2.13}$$

On the other hand, as $\sigma^*(j+1)$ is chosen according to (D.2.3) we can have two following subcases

Case 1: $\sigma^*(j+1) = \sigma_n \in \{\sigma_1, \ldots, \sigma_m\}$.

In this case, we have

$$O_{T_i}(t; A_o) = \begin{cases} O_{T_i}(b_j; A_o) + (t - b_j) & t \in [b_j, t_{j+1}(i)], \\ O_{T_i}(t_{j+1}(i); A_o) & t \in (t_{j+1}(i), b_{j+1}], \end{cases} \tag{D.2.14}$$

for $i \in \{n, n+1, \ldots, m-1\}$. Therefore, according to (D.2.11) and (D.2.14), for $i \in \{n, n+1, \ldots, m-1\}$

$$\max_{t \in [b_j, b_{j+1}]} o_{T_i}(t) = o_{T_i}(t_{j+1}(i)).$$

But as we have (D.2.8) therefore,

$$o_{T_i}(t) \leq \bar{f}(T_i), \qquad t \in [b_j, b_{j+1}], \quad i \in \{n, n+1, \ldots, m-1\}. \tag{D.2.15}$$

It can be easily verified according to Proposition 5.4.1, for $i \in \{1, 2, \ldots, n-1\}$ we have

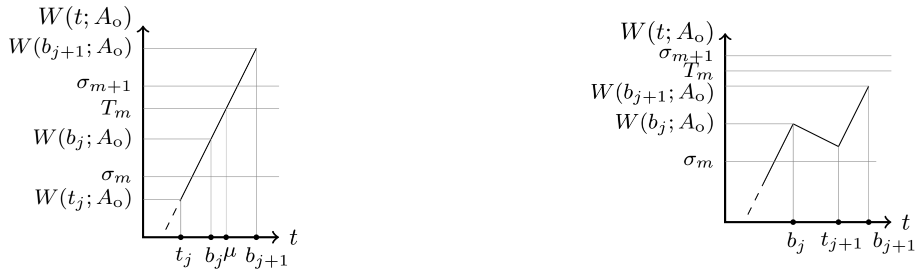
$$O_{T_i}(t; A_o) = O_{T_i}(b_j; A_o) + (t - b_j), \qquad t \in [b_j, b_{j+1}]. \tag{D.2.16}$$

Therefore, for $i \in \{1, 2, \ldots, n-1\}$

$$\max_{t \in [b_j, b_{j+1}]} o_{T_i}(t) = o_{T_i}(b_{j+1}) \tag{D.2.17}$$

But as $\sigma^*(j+1) = \sigma_n$ is chosen using (D.2.3) we have

$$o_{T_i}(b_{j+1}) \leq \bar{f}(T_i) - \epsilon_{i,j+1}(\sigma_n), \quad \forall i \in \{1, 2, \ldots, n-1\}, \tag{D.2.18}$$



(a) $W(b_{j+1}; A_o) > T_m$

(b) $W(b_{j+1}; A_o) \leq T_m$

Figure D.2: Two subcases I, II of $W_\rho(t; A_o)$, when $\sigma^*(j) = \sigma_m$ and $\sigma^*(j+1) = \sigma_{m+1}$.

Therefore according to (D.2.17) and (D.2.18)

$$o_{T_i}(t) < \bar{f}(T_i), \qquad t \in [b_j, b_{j+1}], \quad i \in \{1, 2, \ldots, n-1\}. \tag{D.2.19}$$

Therefore, for this case using (D.2.13), (D.2.15) and (D.2.19)

$$o_{T_i}(t) < \bar{f}(T_i), \qquad t \in [b_j, b_{j+1}], \; i \in \{1, 2, \ldots, M\}. \tag{D.2.20}$$

Case 2: $\sigma^*(j+1) = \sigma_{m+1}$.

In this case workload can be as in Fig. D.2 and can have subcases I, II, and III. In all subcases as

$$W(t; A_o) < T_i, \qquad t \in [b_j, b_{j+1}], \quad i \in \{m+1, \ldots, M\}.$$

Therefore, according to Proposition 5.4.1 and Fig. D.2, we have

$$O_{T_i}(t; A_o) = O_{T_i}(b_j; A_o), \qquad t \in [b_j, b_{j+1}], \quad i \in \{m+1, \ldots, M\}.$$

Therefore, according to (D.2.7)

$$o_{T_i}(t) \leq \bar{f}(T_i), \quad t \in [b_j, b_{j+1}], \quad i \in \{m+1, \ldots, M\}. \tag{D.2.21}$$

On the other hand, for all subcases as

$$W(t; A_{\mathrm{o}}) > T_i, \quad t \in [b_j, b_{j+1}], \quad i \in \{1, 2, \ldots, m-1\}.$$

Therefore, according to Proposition 5.4.1 and Fig. D.2, we have

$$O_{T_i}(t; A_{\mathrm{o}}) = O_{T_i}(b_j; A_{\mathrm{o}}) + (t - b_j), \quad t \in [b_j, b_{j+1}], \quad i \in \{1, 2, \ldots, m-1\}.$$

Therefore, for $i \in \{1, 2, \ldots, m-1\}$

$$\max_{t \in [b_j, b_{j+1}]} o_{T_i}(t) = o_{T_i}(b_{j+1}) \tag{D.2.22}$$

But as $\sigma^*(j+1) = \sigma_{m+1}$ is chosen using (D.2.3) we have

$$o_{T_i}(b_{j+1}) \leq \bar{f}(T_i) - \epsilon_{i,j+1}(\sigma_{m+1}), \quad \forall i \in \{1, 2, \ldots, m-1\}, \tag{D.2.23}$$

Therefore according to (D.2.22) and (D.2.23)

$$o_{T_i}(t) < \bar{f}(T_i), \quad t \in [b_j, b_{j+1}], \quad i \in \{1, 2, \ldots, m-1\}. \tag{D.2.24}$$

In subcases I and II , as

$$W(t; A_{\mathrm{o}}) \geq T_m, \qquad t \in [\mu, b_{j+1}],$$

where μ is defined as follows:

$$W(\mu; A_{\mathrm{o}}) = T_m, \quad \mu \in [b_j, b_{j+1}].$$

Therefore,

$$O_{T_m}(t; A_o) = \begin{cases} O_{T_m}(b_j; A_o) & t \in [b_j, \mu] \\ O_{T_m}(\mu; A_o) + (t - \mu) & t \in (\mu, b_{j+1}] \end{cases}$$

Hence,

$$\arg\max_{t \in [b_j, b_{j+1}]} o_{T_m}(t) \in \{b_j, b_{j+1}\}.$$

But as $\sigma^*(j+1) = \sigma_{m+1}$ is chosen using (D.2.3), therefore

$$o_{T_m}(b_{j+1}) \leq \bar{f}(T_m) - \epsilon_{m,j+1}(\sigma_{m+1}) = \bar{f}(T_{m+1}) < \bar{f}(T_m)$$

Also as we have (D.2.7) therefore,

$$o_{T_m}(t) \leq \bar{f}(T_m), \qquad t \in [b_j, b_{j+1}]. \tag{D.2.25}$$

For subcase III, on the other hand

$$W(t; A_o) < T_m, \qquad t \in [b_j, b_{j+1}].$$

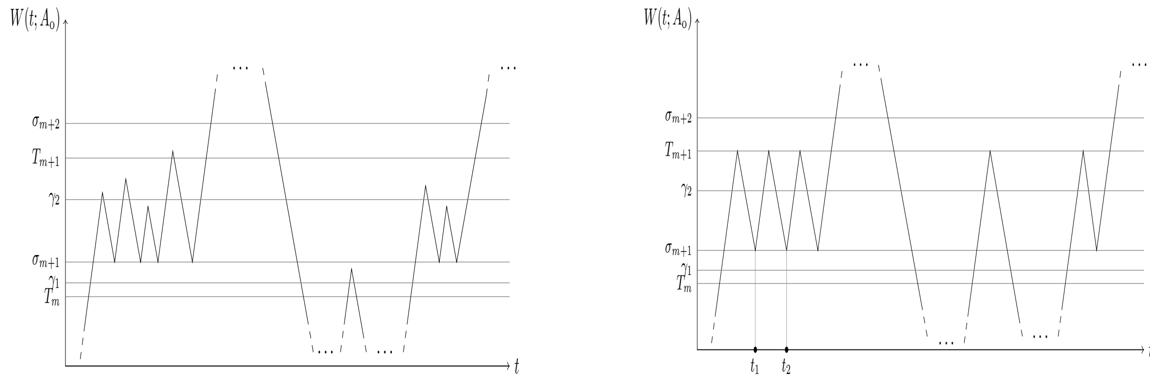
Therefore, it can easily be shown

$$o_{T_m}(t) \leq \bar{f}(T_m), \qquad t \in [b_j, b_{j+1}]. \tag{D.2.26}$$

Therefore, using (D.2.21), (D.2.24), (D.2.25) and (D.2.26) we have

$$o_{T_m}(t) \leq \bar{f}(T_m), \qquad t \in [b_j, b_{j+1}], \quad i \in \{1, \dots, M\}. \tag{D.2.27}$$

□



(a) $O_{\gamma_1,T_{m+1}}(t)$ is not at its maximum value for a constant $O_{T_m,T_{m+1}}(t) = c$

(b) $O_{\gamma_1,T_{m+1}}(t)$ is at its maximum value for a constant $O_{T_m,T_{m+1}}(t) = c$

Figure D.3: Fluctuation of the $W(t; A_\mathrm{o})$ between T_m and T_{m+1}.

Lemma D.2.2. If

$$o_{T_m}(t) \leq c_1; \quad o_{T_{m+1}}(t) \leq c_2, \tag{D.2.28}$$

for $m \in \{1, 2, \dots, M-2\}$, and $t \in [b_{j-1}, b_j(\sigma_\ell)]$. Then

$$o_\gamma(t) \leq \bar{f}(T_m) - (\gamma - T_m)\frac{c_1 - c_2}{T_{m+1} - \sigma_{m+1}}, \tag{D.2.29}$$

for $\forall \gamma \in [\sigma_{m+1}, T_{m+1})$ and

$$o_\gamma(t) < c_1, \qquad \forall \gamma \in [T_m, \sigma_{m+1}). \tag{D.2.30}$$

Proof. We prove this Lemma for two following cases

Case 1: $\gamma \in [\sigma_{m+1}, T_{m+1})$.

We know according to (D.2.28)

$$O_{T_m}(t; A_\mathrm{o}) \leq tc_1; \quad O_{T_{m+1}}(t; A_\mathrm{o}) \leq tc_2. \tag{D.2.31}$$

For the simplification of the proof we extend the concept of the overshoot to the overshoot duration with respect to two threshold values.

Definition D.2.1. Given two threshold values $\zeta_2 > \zeta_1 > 0$ and a traffic process A, a *limited overshoot interval* with respect to A, ζ_1 and ζ_2 is a maximal interval of time κ such that $\zeta_2 > W(\tau; R) \geq \zeta_1$ for all $\tau \in \kappa$. Let $|\kappa|$ denote the length of interval κ. Let $\mathcal{O}_{[\zeta_1,\zeta_2)}(t)$ denote the set of limited overshoot intervals contained in $[0, t]$. Then the *limited overshoot duration* up to time t is defined as

$$O_{\zeta_1,\zeta_2}(t; A) = \sum_{\kappa \in \mathcal{O}_{[\zeta_1,\zeta_2)}(t)} |\kappa|. \tag{D.2.32}$$

According to the Definition 5.4.1 and D.2.1 it is obvious that

$$O_{\zeta_2}(t; A) = O_{\zeta_1,\zeta_2}(t; A) + O_{\zeta_1}(t; A)$$

On the other hand, for a fixed $O_{T_m}(t; A) - O_{T_{m+1}}(t; A) = O_{T_m,T_{m+1}}(t; A) = c$, for any $\gamma \in [T_M, T_{M+1}]$, $O_\gamma(t; A_\text{o})$ is maximized when $O_{\gamma,T_{m+1}}(t; A)$ is at its maximum value. But we should note that as shown in Fig. 5.3 and according to equations (5.22)-(5.24) the workload $W(t; A_\text{o})$ can fluctuate between T_M and T_{M+1} as shown in Fig. D.3. It can be seen by comparing Fig. D.3a and D.3b that $O_{\gamma,T_{m+1}}(t; A)$ is greater in Fig. D.3b in compare to Fig. D.3a. In other words, we should have the fluctuation of $W(t; A_\text{o})$ between T_m and T_{m+1} in units of the complete fluctuation as shown in Fig. D.4. By considering the increasing slope of $W(t; A_\text{o})$ as $C - \rho$ and the decreasing slope of $-\rho$, it can easily be shown that

$$\Delta t = t_2 - t_1 = (T_{m+1} - \sigma_{m+1}) \left(\frac{1}{\rho} + \frac{1}{C - \rho} \right) \tag{D.2.33}$$

$$\Delta \tau = \tau_2 - \tau_1 = (T_{m+1} - \gamma_2) \left(\frac{1}{\rho} + \frac{1}{C - \rho} \right) \tag{D.2.34}$$

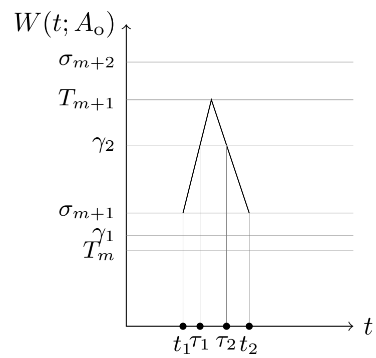


Figure D.4: One unit of complete fluctuation of $W(t; A_o)$ between T_m and T_{m+1}

Therefore, if $O_{T_m,T_{m+1}}(t; A) = c$, in order to maximize $O_{\gamma,T_{m+1}}(t; A)$ we should have n_1 complete fluctuation interval, where n_1 is

$$n_1 = \left\lfloor \frac{c}{\Delta t} \right\rfloor . \tag{D.2.35}$$

For example in Fig. D.3b, $n_1 = 6$. Therefore,

$$O_{\gamma,T_{m+1}}(t; A) \leq \frac{c\Delta\tau}{\Delta t} \tag{D.2.36}$$

This upper bound is tight and can happen for M being at its maximum value, such that $\sigma_{m+1} \approx T_m$, and for the fluctuation of $W(t; A_o)$ as in Fig. D.3b. Hence,

$$\begin{aligned} o_\gamma(t) &= \frac{O_{\gamma,T_{m+1}}(t; A) + O_{T_{m+1}}(t; A)}{t} \leq \frac{c\Delta\tau}{t\Delta t} + o_{T_{m+1}}(t) \\ &= o_{T_m}(t) - (\gamma - \sigma_{m+1})\frac{o_{T_m}(t) - o_{T_{m+1}}(t)}{T_{m+1} - \sigma_{m+1}} \end{aligned} \tag{D.2.37}$$

It can easily be shown with the constraint of (D.2.28), we have

$$\frac{O_\gamma(t; A)}{t} \leq c_1 - (\gamma - \sigma_{m+1})\frac{c_1 - c_2}{T_{m+1} - \sigma_{m+1}} \tag{D.2.38}$$

Case 2: $\gamma \in [T_m, \sigma_{m+1})$.

It can be seen in Fig. D.3a and Fig. D.3b, that

$$O_\gamma(t; A) < O_{T_m}(t; A) \tag{D.2.39}$$

Therefore,

$$o_\gamma(t) < o_{T_m}(t) \leq f(T_m) \tag{D.2.40}$$

If we choose M at its maximum level the bounding function can be a linear function between any two points T_m and T_{m+1}. □

Corollary D.2.1. Let

$$o_{T_m}(t) \leq \bar{f}(T_m); \quad o_{T_{m+1}}(t) \leq \bar{f}(T_{m+1}), \tag{D.2.41}$$

for $m \in \{1, 2, \ldots, M-2\}$, and $t \in [b_{j-1}, b_j(\sigma_\ell)]$ and M is chosen as the maximum value such that $T_i \approx \sigma_{i+1}$ for $i \in \{1, 2, \ldots, M-1\}$. Then

$$o_\gamma(t) \leq \bar{f}(T_m) - (\gamma - T_m)\frac{\bar{f}(T_m) - \bar{f}(T_{m+1})}{T_{m+1} - T_m}, \tag{D.2.42}$$

for $\forall \gamma \in [T_m, T_{m+1})$.

Remark. For the case that M is not chosen as the maximum possible value and $T_i < \sigma_{i+1}$ for some $i \in \{1, 2, \ldots, M-1\}$, by slightly modifying the definition of the $\bar{f}(\gamma)$, we can get a result similar to Corollary D.2.1. In this modification, in the interval $[\sigma_{i+1}, T_{i+1})$ let

$$l_i(\gamma) := f(T_{i+1}) + \omega_i(\gamma - T_{i+1}) \tag{D.2.43}$$

represent the line connecting the points $(\sigma_{i+1}, f(\sigma_{i+1}))$ and $(T_{i+1}, f(T_{i+1}))$ with slope

$$\hat{\omega}_i := \frac{f(T_{i+1}) - f(\sigma_{i+1})}{T_{i+1} - \sigma_{i+1}} \tag{D.2.44}$$

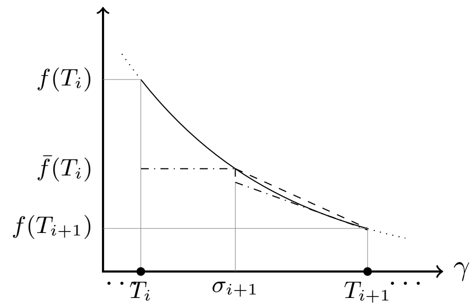


Figure D.5: Modified definition of $\bar{f}(\gamma)$ when $M < M_{\max}$.

for $i = 1, \ldots, M-2$. If $f(\gamma) \geq l_i(\gamma)$ for all $\gamma \in [\sigma_{i+1}, T_{i+1})$ we set $\bar{f} = l_i$ in this interval. Otherwise, we set $\bar{f} = h_i$ on $[\sigma_{i+1}, T_{i+1})$, where

$$h_i(\gamma) = f(T_{i+1}) + f'(T_{i+1})(\gamma - T_{i+1}). \tag{D.2.45}$$

On the other hand, in the interval $[T_i, \sigma_{i+1})$ we set $\bar{f} = f(\sigma_{i+1})$. Similarly, we set $\bar{f}(\gamma) = f(T)$ for $\gamma \in [T_{M-1}, T_M]$ and $\bar{f}(\gamma) = 0$ for $\gamma > T_M$. To summarize, we define

$$\bar{f}(\gamma) := \begin{cases} 1, & \gamma \in [0, T_1), \\ f(\sigma_{i+1}), & \gamma \in [T_i, \sigma_{i+1}), \\ f(T_{i+1}) + \hat{m}_i(\gamma - T_{i+1}), & \gamma \in [\sigma_{i+1}, T_{i+1}), \\ f(T), & \gamma \in [T_{M-1}, T_M], \\ 0, & \gamma > T_M, \end{cases} \tag{D.2.46}$$

where the slopes $\hat{m}_i$ are defined by

$$\hat{m}_i = \begin{cases} \hat{\omega}_i, & \text{if } f \geq h_i \text{ on } [\sigma_{i+1}, T_{i+1}), \\ f'(T_{i+1}), & \text{otherwise}, \end{cases} \tag{D.2.47}$$

for $i = 1, \ldots, M-2$. This modified $\bar{f}(\gamma)$ is shown in Fig. D.5.

Corollary D.2.2. Let

$$o_{T_m}(t) \leq \bar{f}(T_m); \quad o_{T_{m+1}}(t) \leq \bar{f}(T_{m+1}), \tag{D.2.48}$$

for $m \in \{1, 2, \ldots, M-2\}$, and $t \in [b_{j-1}, b_j(\sigma_\ell)]$ and $\bar{f}(\gamma)$ is defined as (D.2.46). Then

$$o_\gamma(t) \leq \bar{f}(\sigma_{m+1}) - (\gamma - T_m)\frac{\bar{f}(\sigma_{m+1}) - \bar{f}(T_{m+1})}{T_{m+1} - \sigma_{m+1}}, \tag{D.2.49}$$

for $\forall \gamma \in [\sigma_{m+1}, T_{m+1})$ and

$$o_\gamma(t) \leq \bar{f}(\sigma_{m+1}), \tag{D.2.50}$$

for $\forall \gamma \in [T_m, \sigma_{m+1})$.

Proof of Theorem 5.4.2. In Lemma D.2.1 we showed if $\sigma^*(j)$ is chosen using (D.2.4) then

$$o_{T_i}(t) \leq \bar{f}(T_i), \quad t \in [b_{j-1}, b_j(\sigma^*(j))], \quad i \in \{1, 2, \ldots, M\}. \tag{D.2.51}$$

On the other hand, we showed in Corollary D.2.1 that if we have (D.2.51) and $M = M_{\max}$, defined in (5.36), then

$$o_\gamma(t) \leq \bar{f}(T_i) - (\gamma - T_i)\frac{\bar{f}(T_i) - \bar{f}(T_{i+1})}{T_{i+1} - T_i}, \tag{D.2.52}$$

for $\forall \gamma \in [T_i, T_{i+1})$ and $\forall i \in \{1, \ldots, M\}$. On the other hand, in Corollary D.2.2 we showed, if if we have (D.2.51) and $M < M_{\max}$, then with the modified definition of $\bar{f}(\gamma)$ in (D.2.46),

$$o_\gamma(t) \leq \bar{f}(\sigma_{i+1}) - (\gamma - T_i)\frac{\bar{f}(\sigma_{i+1}) - \bar{f}(T_{i+1})}{T_{i+1} - \sigma_{i+1}}, \tag{D.2.53}$$

for $\forall\gamma \in [\sigma_{i+1}, T_{i+1})$ and

$$o_\gamma(t) \le \bar{f}(\sigma_{i+1}), \tag{D.2.54}$$

for $\forall\gamma \in [T_i, \sigma_{i+1})$ and $\forall i \in \{1, \dots, M\}$. Therefore, for $M = M_{\max}$ if for all $i \in \{1, 2, \dots, M-1\}$ and all $\gamma \in [T_i, T_{i+1}]$,

$$f(\gamma) \ge \bar{f}(T_i) - (\gamma - T_i)\frac{\bar{f}(T_i) - \bar{f}(T_{i+1})}{T_{i+1} - T_i}, \tag{D.2.55}$$

with $\bar{f}(\gamma)$ defined in (5.41), and for $M < M_{\max}$ if for all $i \in \{1, 2, \dots, M-1\}$ and all $\gamma \in [\sigma_{i+1}, T_{i+1})$

$$f(\gamma) \ge \bar{f}(\sigma_{i+1}) - (\gamma - T_i)\frac{\bar{f}(\sigma_{i+1}) - \bar{f}(T_{i+1})}{T_{i+1} - \sigma_{i+1}}, \tag{D.2.56}$$

and if for all $\gamma \in [T_i, \sigma_{i+1})$

$$f(\gamma) \ge \bar{f}(\sigma_{i+1}), \tag{D.2.57}$$

with $\bar{f}(\gamma)$ defined in (D.2.46), then

$$o_\gamma(t) \le f(\gamma), \qquad t \in [b_{j-1}, b_j(\sigma^*(j))], \quad \forall\gamma \in [T_1, T]. \tag{D.2.58}$$

But definition of $\bar{f}(\gamma)$ assures inequalities in (D.2.55)–(D.2.57). □

D.3 Proof of Theorem 5.4.3

In order to prove Theorem 5.4.3 we first establish the following lemma:

Lemma D.3.1. Let $\mathcal{B}_j$ be as define in (5.46), let $k = \min \mathcal{B}_j$ and $\mathcal{J}_j$ be as defined in (5.52).

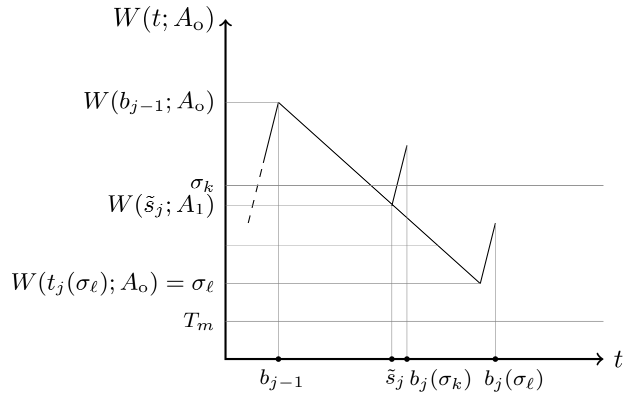


Figure D.6: $W(t; A_o)$, when $k = \min \mathcal{B}_j$, $m \in \mathcal{J}_j$ and $m < k-1$ for $\forall \ell \in \{m+1, \ldots, k\}$.

If $m \in \mathcal{J}_j$ and $m < k-1$ then

$$\frac{O_{T_m}(b_j(\sigma_\ell); A_o)}{b_j(\sigma_\ell)} + \frac{W(b_j(\sigma_\ell); A_o) - T_m}{\rho b_j(\sigma_\ell)}(1 - \bar{f}(T_m)) \leq \bar{f}(T_m), \tag{D.3.1}$$

for $\forall \ell \in \{m+1, \ldots, k\}$.

Proof. Note that,

$$\frac{W(b_j(\sigma_\ell); A_o) - T_m}{\rho b_j(\sigma_\ell)}(1 - f(T_m)) = \epsilon_{m,j}(\sigma_\ell), \tag{D.3.2}$$

for $\forall \ell \in \{m+2, \ldots, k\}$. According to (2.12), and as it is shown in Fig. D.6, it can easily be verified that

$$b_j(\sigma_\ell) = b_j(\sigma_k) + \frac{W(\tilde{s}_j; A_1) - \sigma_\ell}{\rho} \tag{D.3.3}$$

Therefore, using Proposition 5.4.1, we have

$$O_{T_m}(b_j(\sigma_\ell); A_o) = O_{T_m}(b_j(\sigma_k); A_o) + \frac{W(\tilde{s}_j; A_1) - \sigma_\ell}{\rho} \tag{D.3.4}$$

Also we have

$$W(b_j(\sigma_\ell); A_o) = W(b_j(\sigma_k); A_o) - (W(\tilde{s}_j; A_1) - \sigma_\ell) \tag{D.3.5}$$

Therefore, using (D.3.3), (D.3.4) and (D.3.5) and $m \in \mathcal{J}_j$ after some manipulations we can show (D.3.1) holds. □

Proof of Theorem 5.4.3. Based on m, defined in (5.53), we have two following cases:

Case 1: $m = k - 1$.

In this case, as $m \in \mathcal{J}_j$, therefore according to (5.52)

$$\frac{O_{T_\ell}(b_j(\sigma_k); A_o)}{b_j(\sigma_k)} \leq \bar{f}(T_\ell) - \epsilon_{\ell,j}(\sigma_k), \tag{D.3.6}$$

for $\ell = 1, 2, \ldots, k-1$. Therefore, according to (5.50)

$$\frac{O_{T_m}(b_j(\sigma_k); A_o)}{b_j(\sigma_k)} \leq \bar{f}(T_m) - \epsilon_{m,j}(\sigma_k) = \bar{f}(T_k) \tag{D.3.7}$$

Hence, according to (5.54), $m + 1 \in \mathcal{K}_j$. Hence, $\sigma^*(j)$ derived using Theorem 5.4.3 is $\sigma^*(j) = \sigma_k$. On the other hand, according to (D.3.6) and (5.49), $\sigma^*(j)$ derived using Theorem 5.4.2 will be also $\sigma^*(j) = \sigma_k$.

Case 2: $m < k - 1$.

Lets assume $\sigma^*(j)$ derived using Theorem 5.4.3 and $\sigma^*(j) = \sigma_n$. We will show $\sigma^*(j)$ derived using Theorem 5.4.2 will be also $\sigma^*(j) = \sigma_n$. In this case according to (5.54) and (5.57), if $\ell \in \{1, 2, \ldots, n-1\}$ then $\ell \in \mathcal{J}_j$ and $\ell < k - 1$. Therefore, according to Lemma D.3.1

$$\frac{O_{T_\ell}(b_j(\sigma_n); A_o)}{b_j(\sigma_n)} \leq \bar{f}(T_\ell) - \frac{W(b_j(\sigma_n); A_o) - T_\ell}{\rho b_j(\sigma_n)}(1 - \bar{f}(T_\ell)), \tag{D.3.8}$$

for $l = 1, 2, \ldots, n-1$. On the other hand, as $n \in \mathcal{K}_j$, according to (5.54) and (5.50)

$$\frac{O_{T_{n-1}}(b_j(\sigma_n); A_{\mathrm{o}})}{b_j(\sigma_n)} \leq \bar{f}(T_{n-1}) = \bar{f}(T_n) - \epsilon_{n-1,j}(\sigma_n) \tag{D.3.9}$$

Therefore, according to (D.3.8), (D.3.9) and (D.3.8), $\sigma^*(j)$ derived using Theorem 5.4.2 will be also $\sigma^*(j) = \sigma_n$. □

D.4 Proof of Theorem 5.4.1

D.4.1 Proof of Theorem 5.4.1, Part I

In this section we prove the following lemma, which is a preliminary version of Theorem 5.4.1. Then using the results in this appendix, we prove Theorem 5.4.1 in the next section. We also provide some details about the practical implementation of Algorithm 6 in the next section.

Lemma D.4.1. Assume that T_M is chosen sufficiently large such that for every packet j the set

$$\mathcal{B}_j = \left\{1 \leq \ell \leq M : \sigma_\ell \geq W(\tilde{s}_j; A_1)\right\}, \tag{D.4.1}$$

is non-empty. Set

$$\mathcal{I}_j = \left\{2 \leq \ell \leq \min \mathcal{B}_j : o_{T_{\ell-1}}(b_j(\sigma_\ell)) \leq \bar{f}(T_\ell)\right\} \tag{D.4.2}$$

where $t_j(\sigma_\ell)$ and $b_j(\sigma_\ell)$ are given by (5.18) and (5.9), respectively. Set

$$i^* = \begin{cases} \max \mathcal{I}_j, & \mathcal{I}_j \neq \emptyset, \\ 1, & \text{otherwise.} \end{cases} \tag{D.4.3}$$

Let

$$\sigma^*(j) = \sigma_{i^*}. \tag{D.4.4}$$

If

$$b_j \geq \frac{L}{\epsilon\rho} + \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}, \tag{D.4.5}$$

where $\epsilon > 0$ is given by

$$\epsilon = \min_{2\leq k\leq M}[f(T_{k-1}) - f(T_k)], \tag{D.4.6}$$

then

$$o_{T_{i^*-1}}(t) \leq \bar{f}(T_{i^*-1}), \qquad t \in [b_{j-1}, b_j]. \tag{D.4.7}$$

By comparing (D.4.7) and (5.43), we can see Lemma D.4.1 guarantees satisfying the constraint in (5.43) only for one specific value $\gamma = T_{i^*-1}$ rather than $\forall\gamma \in [0, T]$. Proof of Lemma D.4.1 is based on the following three lemmas.

Lemma D.4.2. Let $\mathcal{B}_j$ be as defined in (5.46) and let $k = \min \mathcal{B}_j$. Let assume $k > 1$. Set $b_j = b_j(\sigma_k)$. Then

$$\operatorname*{arg\,max}_{t\in[b_{j-1},b_j]} o_{T_{k-1}}(t) \in \{b_{j-1}, b_j, \eta, \nu\} \tag{D.4.8}$$

where $\eta \in [b_{j-1}, t_j]$ and $\nu \in [t_j, b_j]$ are determined by

$$W(\eta; A_\mathrm{o}) = W(\nu; A_\mathrm{o}) = T_{k-1}. \tag{D.4.9}$$

Proof. According to Proposition 5.4.1, $O_{T_{k-1}}(t; A_\mathrm{o})$ is related to $O_{T_{k-1}}(b_{j-1}; A_\mathrm{o})$ over the interval $t \in [b_{j-1}, b_j]$ as follows,

$$O_{T_{k-1}}(t; A_\mathrm{o}) = \begin{cases} O_{T_{k-1}}(b_{j-1}; A_\mathrm{o}) + \beta(b_{j-1}, t, T_{k-1}), & t \in [b_{j-1}, t_j] \\ O_{T_{k-1}}(t_j; A_\mathrm{o}) + \alpha(t_j, t, T_{k-1}), & t \in [t_j, b_j] \end{cases} \tag{D.4.10}$$

We can have one of the two following cases based on $\tilde{s}_j$

Case 3: $\tilde{s}_j = s_j$.

In this case $s_j > b_{j-1}$. According to (5.18) and (5.20), $t_j = \tilde{s}_j$ and $W(t_j; A_\mathrm{o}) = W(\tilde{s}_j; A_1)$. In this case using (5.19)-(5.23) we have

$$W(t; A_\mathrm{o}) = W(t; A_1), \quad \forall\ t \in [b_{j-1}, b_j] \tag{D.4.11}$$

If $T_k - T_{k-1} > \delta$, for $W(t; A_\mathrm{o})$ on the interval $t \in [b_{j-1}, b_j]$ we can have one the five subcases shown depicted Fig. D.7. On the other hand, If $T_k - T_{k-1} = \delta$, then $T_{k-1} = \sigma_k$ and $W(t; A_\mathrm{o})$ on the interval $t \in [b_{j-1}, b_j]$ will be like the four subcases shown in Figs. D.7b-D.7e.

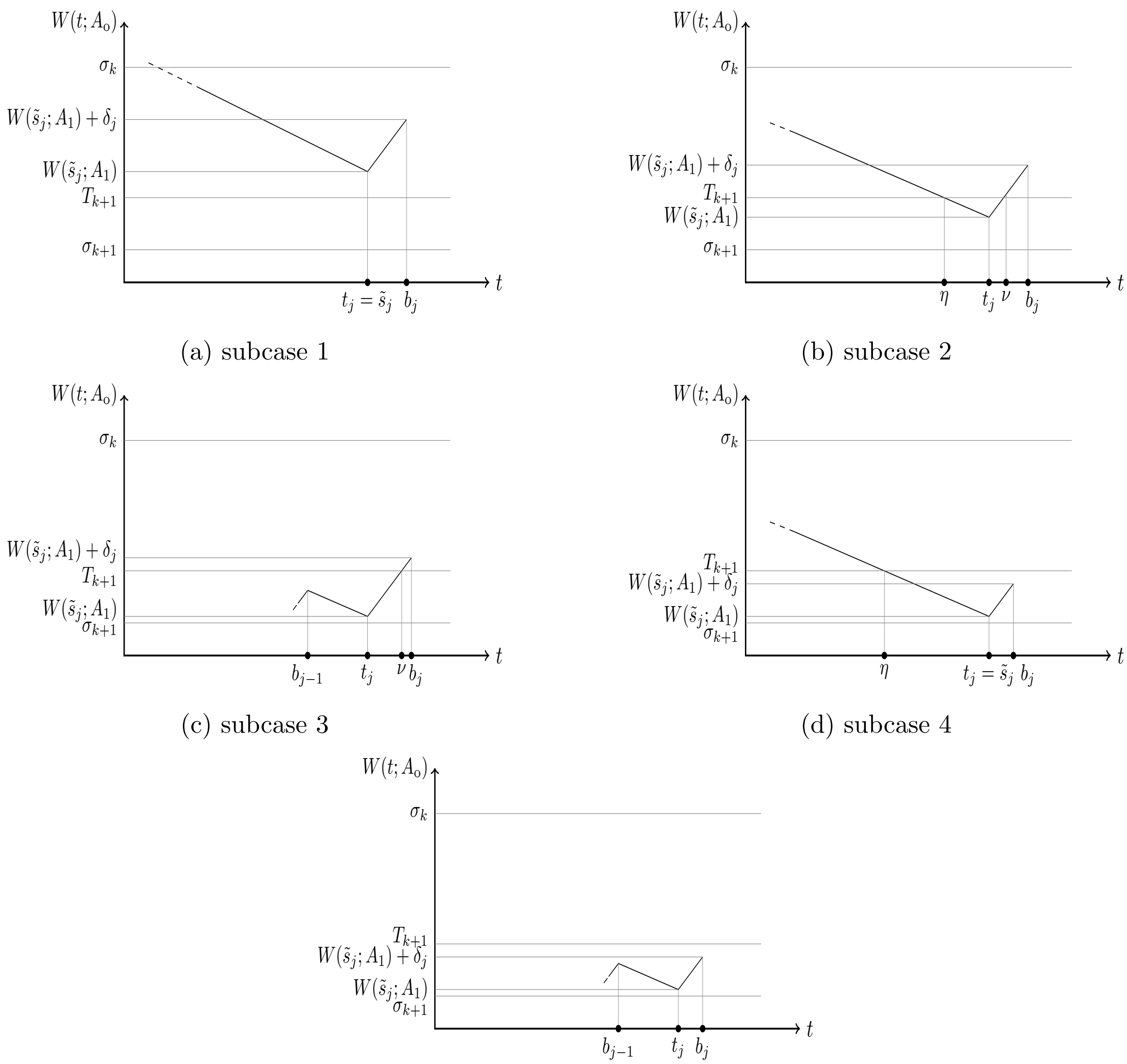
According to (D.4.11), in subcase D.7a, $W(t; A_\mathrm{o}) > T_{k-1}$ for $\forall t \in [b_{j-1}, b_j]$. Hence, using (D.4.10), (5.30), and (5.31) we have

$$o_{T_{k-1}}(t) = \frac{O_{T_{k-1}}(b_{j-1}; A_\mathrm{o}) + (t - b_{j-1})}{b_{j-1} + (t - b_{j-1})}, \tag{D.4.12}$$

for $\forall\ t \in [b_{j-1}, b_j]$. As $O_\gamma(t; A_\mathrm{o}) < t$ for $\forall \gamma \in [0, T]$, it can be easily verified that in this case

$$\underset{t \in [b_{j-1}, b_j]}{\arg\max}\ o_{T_{k-1}}(t) = b_j. \tag{D.4.13}$$

On the other hand, for subcase D.7b, as $W(t; A_\mathrm{o}) > T_{k-1}$ for $\forall t \in \{[b_{j-1}, \eta] \cup [\nu, b_j]\}$, where



(a) subcase 1

(b) subcase 2

(c) subcase 3

(d) subcase 4

(e) subcase 5

Figure D.7: Different cases of $W(t; A_o)$ on the interval $[b_{j-1}, b_j]$ with $\sigma = \sigma_k$, $\zeta(\sigma) = T_{k+1}$, $t_j = \tilde{s}_j$ and $T_k - T_{k+1} > \delta$.

η and ν are defined in (D.4.9), we have

$$o_{T_{k-1}}(t) = \begin{cases} \frac{O_{T_{k-1}}(b_{j-1};A_{\mathrm{o}})+(t-b_{j-1})}{b_{j-1}+(t-b_{j-1})} & t \in [b_{j-1}, \eta] \\ \frac{O_{T_{k-1}}(\eta;A_{\mathrm{o}})}{\eta+(t-\eta)} & t \in [\eta, \nu] \\ \frac{O_{T_{k-1}}(\eta;A_{\mathrm{o}})+(t-\nu)}{\nu+(t-\nu)} & t \in [\nu, b_j] \end{cases} \tag{D.4.14}$$

Therefore, it can be easily verified that in this case

$$\arg\max_{t\in[b_{j-1},b_j]} o_{T_{k-1}}(t) \in \{\eta, b_j\}. \tag{D.4.15}$$

For subcase D.7c, as $W(t; A_{\mathrm{o}}) > T_{k-1}$ for $\forall t \in [\nu, b_j]$, we have

$$o_{T_{k-1}}(t) = \begin{cases} \frac{O_{T_{k-1}}(b_{j-1};A_{\mathrm{o}})}{b_{j-1}+(t-b_{j-1})} & t \in [b_{j-1}, \nu] \\ \frac{O_{T_{k-1}}(b_{j-1};A_{\mathrm{o}})+(t-\nu)}{\nu+(t-\nu)} & t \in [\nu, b_j] \end{cases} \tag{D.4.16}$$

Therefore, it can be easily verified that in this case

$$\arg\max_{t\in[b_{j-1},b_j]} o_{T_{k-1}}(t) \in \{b_{j-1}, b_j\}. \tag{D.4.17}$$

For subcase D.7d, as $W(t; A_{\mathrm{o}}) > T_{k-1}$ for $\forall t \in [b_{j-1}, \eta]$, we have

$$o_{T_{k-1}}(t) = \begin{cases} \frac{O_{T_{k-1}}(b_{j-1};A_{\mathrm{o}})+(t-b_{j-1})}{b_{j-1}+(t-b_{j-1})} & t \in [b_{j-1}, \eta] \\ \frac{O_{T_{k-1}}(\eta;A_{\mathrm{o}})}{\eta+(t-\eta)} & t \in [\eta, b_j] \end{cases} \tag{D.4.18}$$

Therefore, it can be easily verified that in this case

$$\arg\max_{t\in[b_{j-1},b_j]} o_{T_{k-1}}(t) = \eta. \tag{D.4.19}$$

For subcase D.7e, as $W(t;A_\mathrm{o}) < T_{k-1}$ for $\forall t \in [b_{j-1}, b_j]$, we have

$$o_{T_{k-1}}(t) = \frac{O_{T_{k-1}}(b_{j-1};A_\mathrm{o})}{b_{j-1} + (t - b_{j-1})}, \tag{D.4.20}$$

for $\forall t \in [b_{j-1}, b_j]$. Therefore, it can be easily verified that in this case

$$\operatorname*{arg\,max}_{t \in [b_{j-1}, b_j]} \frac{O_{T_{k-1}}(t;A_\mathrm{o})}{t} = b_{j-1}. \tag{D.4.21}$$

On the other hand , when $T_k - T_{k-1} = \delta$, we have the subcases similar to the subcases D.7b-D.7e. Therefore, we will have the same relations as (D.4.15)-(D.4.21).

Case 4: $\tilde{s}_j = b_{j-1}$

In this case $s_j < b_{j-1}$. According to (5.18) and (5.20), $t_j = \tilde{s}_j = b_{j-1}$ and $W(t_j;A_\mathrm{o}) = W(\tilde{s}_j;A_1)$. In this case using (5.20)-(5.23) we have

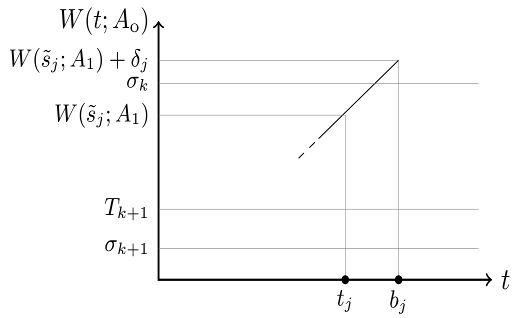
$$W(t;A_\mathrm{o}) = W(t;A_1), \qquad \forall\ t \in [b_{j-1}, b_j] \tag{D.4.22}$$

If $T_k - T_{k-1} > \delta$, for $W(t;A_\mathrm{o})$ on the interval $t \in [b_{j-1}, b_j]$ we can have one the four subcases shown depicted Fig. D.8. On the other hand, If $T_k - T_{k-1} = \delta$, then $T_{k-1} = \sigma_k$ and we can have one the two subcases shown in Fig. D.8c and D.8d. As in subcase D.8a and D.8b, $W(t;A_\mathrm{o}) > T_{k-1}$ for $\forall t \in [b_{j-1}, b_j]$, we have

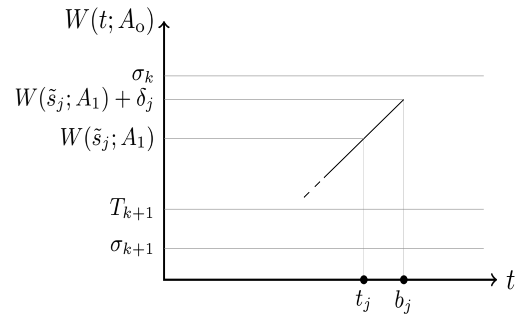
$$o_{T_{k-1}}(t) = \frac{O_{T_{k-1}}(b_{j-1};A_\mathrm{o}) + (t - b_{j-1})}{b_{j-1} + (t - b_{j-1})} \tag{D.4.23}$$

for $\forall t \in [b_{j-1}, b_j]$. Therefore, it can be easily verified that in this case

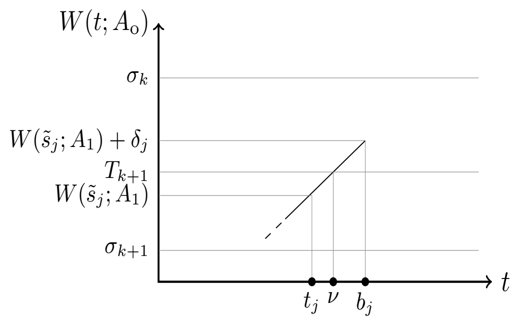
$$\operatorname*{arg\,max}_{t \in [b_{j-1}, b_j]} o_{T_{k-1}}(t) = b_j. \tag{D.4.24}$$



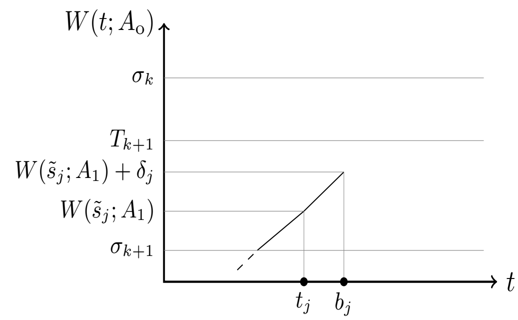
(a) subcase 1



(b) subcase 2



(c) subcase 3



(d) subcase 4

Figure D.8: Different cases of $W(t; A_o)$ on the interval $[b_{j-1}, b_j]$ with $\sigma = \sigma_k$, $\zeta(\sigma) = T_{k+1}$, $t_j = \tilde{s}_j = b_{j-1}$ and $T_k - T_{k+1} > \delta$.

For subcase D.8c, as $W(t; A_\text{o}) > T_{k-1}$ for $\forall t \in [\nu, b_j]$, we have

$$o_{T_{k-1}}(t) = \begin{cases} \frac{O_{T_{k-1}}(b_{j-1};A_\text{o})}{b_{j-1}+(t-b_{j-1})} & t \in [b_{j-1}, \nu] \\ \frac{O_{T_{k-1}}(b_{j-1};A_\text{o})+(t-\nu)}{\nu+(t-\nu)} & t \in [\nu, b_j] \end{cases} \tag{D.4.25}$$

Therefore, it can be easily verified that in this case

$$\arg\max_{t\in[b_{j-1},b_j]} o_{T_{k-1}}(t) \in \{b_{j-1}, b_j\}. \tag{D.4.26}$$

For subcase D.8d, as $W(t; A_\text{o}) < T_{k-1}$ for $\forall t \in [b_{j-1}, b_j]$, we have

$$o_{T_{k-1}}(t) = \frac{O_{T_{k-1}}(b_{j-1}; A_\text{o})}{b_{j-1} + (t - b_{j-1})} \tag{D.4.27}$$

Therefore, it can be easily verified that in this case

$$\arg\max_{t\in[b_{j-1},b_j]} o_{T_{k-1}}(t) = b_{j-1}. \tag{D.4.28}$$

On the other hand , when $T_k - T_{k-1} = \delta$, we have the subcases similar to the subcases D.8c-D.8d. Therefore, we will have the same relations as (D.4.26)-(D.4.28). $\square$

Lemma D.4.3. Let $\mathcal{B}_j$ be as defined in (5.46) and $k = \min \mathcal{B}_j$. Assume $k > 1$. Let $\ell \in \{2, \ldots, k-1\}$ and set $b_j = b_j(\sigma_\ell)$. Then

$$\arg\max_{t\in[b_{j-1},b_j]} o_{T_{\ell-1}}(t) = b_j. \tag{D.4.29}$$

Proof. As it was mentioned before, $O_{T_{\ell-1}}(t; A_\text{o})$ can be determined using $O_{T_{\ell-1}}(b_{j-1}; A_\text{o})$ over the interval $t \in [b_{j-1}, b_j]$ according to (D.4.10). Similarly, we can have one of the two following cases based on $\tilde{s}_j$:

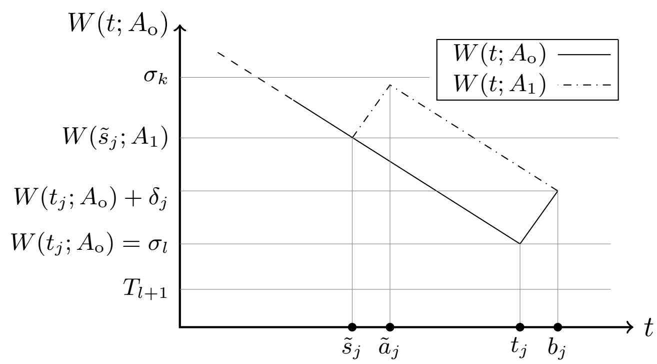


Figure D.9: $W(t; A_o)$, when $\sigma = \sigma_l$, $\zeta(\sigma) = T_{l+1}$, $\tilde{s}_j = b_{j-1}$ and $T_k - T_{k+1} > \delta$.

Case 1: $\tilde{s}_j = s_j$

In this case $s_j > b_{j-1}$. With $\sigma = \sigma_\ell$, t_j is derived using (5.18). According to (5.20), $W(t_j; A_o) = \sigma_\ell$. Hence, in this case, when $T_k - T_{k-1} > \delta$, $W(t; A_o)$ is as shown in Fig. D.9. On the other hand, if $T_k - T_{k-1} = \delta$, $W(t; A_o)$ will be same as in Fig. D.9, except $T_{\ell-1} = \sigma_\ell$. In this case, as $W(t; A_o) \geq T_{\ell-1}$ for $\forall t \in [b_{j-1}, b_j]$, we have

$$o_{T_{\ell-1}}(t) = \frac{O_{T_{\ell-1}}(b_{j-1}; A_o) + (t - b_{j-1})}{b_{j-1} + (t - b_{j-1})}, \tag{D.4.30}$$

for $\forall t \in [b_{j-1}, b_j]$. Therefore, it can be easily verified that in this case

$$\underset{t \in [b_{j-1}, b_j]}{\arg\max}\, o_{T_{\ell-1}}(t) = b_j. \tag{D.4.31}$$

Case 2: $\tilde{s}_j = b_{j-1}$

In this case $s_j < b_{j-1}$. With $\sigma = \sigma_\ell$, t_j is derived using (5.18). According to (5.20), $W(t_j; A_o) = \sigma_\ell$. Hence, in this case, when $T_k - T_{k-1} > \delta$, $W(t; A_o)$ is as shown in Fig. D.10. On the other hand, if $T_k - T_{k-1} = \delta$, $W(t; A_o)$ will be same as in Fig. D.10, except $T_{\ell-1} = \sigma_\ell$.

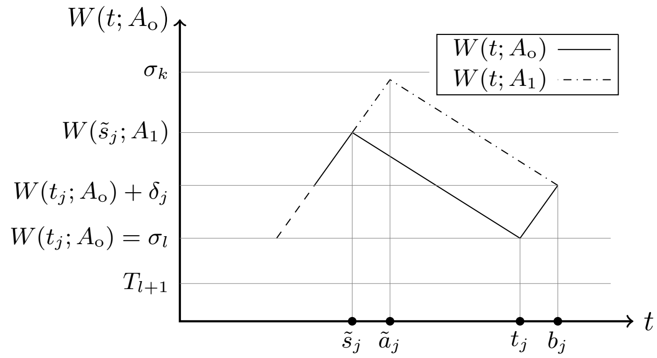


Figure D.10: $W(t; A_o)$, when $\sigma = \sigma_l$, $\zeta(\sigma) = T_{l+1}$, $\tilde{s}_j = b_{j-1}$ and $T_k - T_{k+1} > \delta$.

In this case, as $W(t; A_o) \geq T_{\ell-1}$ for $\forall t \in [b_{j-1}, b_j]$, we have

$$o_{T_{\ell-1}}(t) = \frac{O_{T_{\ell-1}}(b_{j-1}; A_o) + (t - b_{j-1})}{b_{j-1} + (t - b_{j-1})}, \tag{D.4.32}$$

for $\forall t \in [b_{j-1}, b_j]$. Therefore, it can be easily verified that in this case

$$\arg\max_{t \in [b_{j-1}, b_j]} o_{T_{\ell-1}}(t) = b_j. \tag{D.4.33}$$

$\square$

Lemma D.4.4. Let $\mathcal{B}_j$ be as defined in (5.46) and let $k = \min \mathcal{B}_j$. Let assume $k > 1$ and assume that b_j satisfies the following lower bound

$$b_j \geq \frac{L}{\epsilon\rho} + \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}, \tag{D.4.34}$$

where $\epsilon > 0$ is given in (D.4.6). Let $\ell \in \{2, \ldots, k\}$ and set $b_j = b_j(\sigma_\ell)$. Then

$$o_{T_{\ell-1}}(b_j) \leq \bar{f}(T_\ell), \tag{D.4.35}$$

implies

$$o_{T_{\ell-1}}(t) \leq \bar{f}(T_{\ell-1}), \qquad \forall t \in [b_{j-1}, b_j]. \tag{D.4.36}$$

Proof. According to the definition (5.41),

$$\bar{f}(T_{\ell-1}) \geq \bar{f}(T_\ell) \;\text{ for }\; \ell \in \{2, \ldots, M\}. \tag{D.4.37}$$

Therefore, for $\ell \in \{2, \ldots, k\}$ and $b_j = b_j(\sigma_\ell)$, then if

$$\operatorname*{arg\,max}_{t \in [b_{j-1}, b_j]} o_{T_{\ell-1}}(t) = b_j, \tag{D.4.38}$$

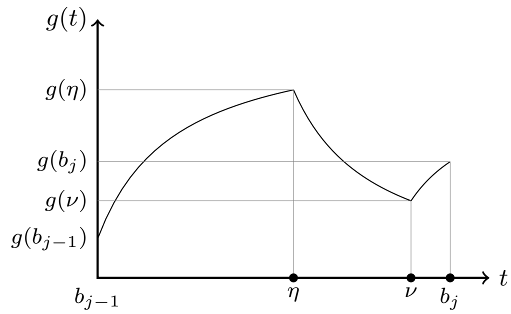
then, based on (D.4.37), having (D.4.35) yields (D.4.36) and no lower bound on b_j is required. On the other hand, we will show for the cases that (D.4.38) does not hold or

$$\operatorname*{arg\,max}_{t \in [b_{j-1}, b_j]} o_{T_{\ell-1}}(t) \neq b_j, \tag{D.4.39}$$

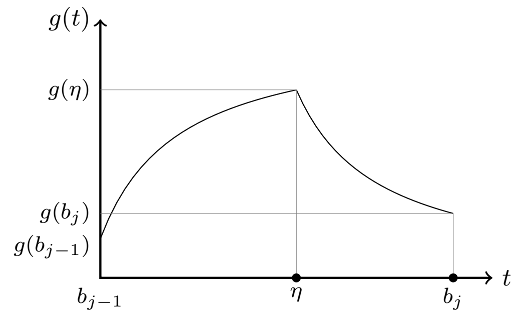
if b_j is greater than the lower bound in (D.4.34), then having (D.4.35) yields (D.4.36). As it was shown previously in Lemmas D.4.2 and D.4.3, the only cases of having (D.4.39) is when $b_j = b_j(\sigma_k)$. When $b_j = b_j(\sigma_k)$ based on $\tilde{s}_j$ we can have two cases:

Case 1: $\tilde{s}_j = s_j$

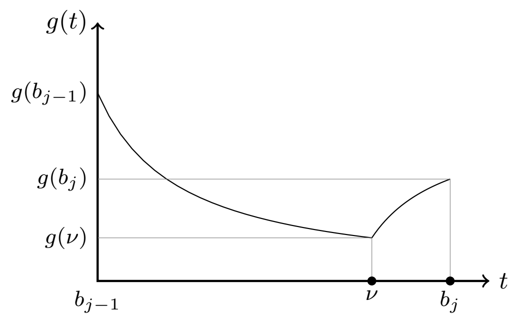
This case is shown in Fig. D.7. As it is explained in Lemma D.4.2, in the four subcases D.7b-D.7e we can have cases of having the maximum of the overshoot ratio function over the interval $[b_{j-1}, b_j]$ at some $t \neq b_j$. The overshoot ratio functions for these case, $o_{T_{k-1}}(t)$, are depicted in Fig. D.11 for $t \in [b_{j-1}, b_j]$. Theses figures are derived using (D.4.14), (D.4.16), (D.4.18), and (D.4.20). Note that, as it was mentioned in Lemma D.4.2, in subcases Fig. D.7b and D.7c we can have the maximum of the overshoot function happening at b_j, these cases are however not considered in Fig. D.11a and D.11c, as if the



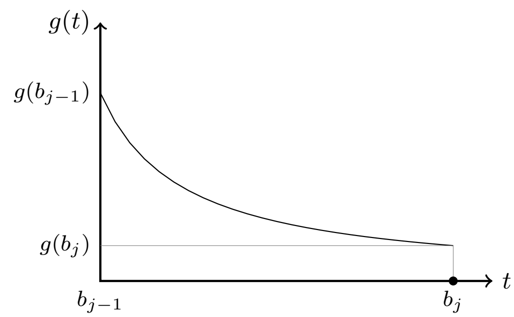
(a) Overshoot ratio function for subcase 2 in Fig. D.7b



(b) Overshoot ratio function for subcase 4 in Fig. D.7d



(c) Overshoot ratio function for subcase 3 in Fig. D.7c



(d) Overshoot ratio function for subcase 5 in Fig. D.7e

Figure D.11: $g(\sigma)$ for $\sigma = \sigma_k$ and $\tilde{s}_j = s_j$ for the cases of (D.4.39)

overshoot ratio function is maximized at b_j, then (D.4.36) holds for $\forall t \in [b_{j-1}, b_j]$ and no lower bound in needed on b_j.

For the subcase Fig. D.11a according to Fig. D.7b we have

$$T_{k-1} - W(\tilde{s}_j; A_{\mathrm{o}}) \leq T_{k-1} - \sigma_{k-1} = \delta.$$

Therefore,

$$t_j - \eta \leq \frac{\delta}{\rho}.$$

Similarly,

$$\nu - t_j \leq \frac{\delta}{C - \rho}.$$

Therefore,

$$\nu - \eta \leq \frac{\delta}{\rho} + \frac{\delta}{C - \rho} = \frac{L}{\rho}. \tag{D.4.40}$$

On the other hand, according to (D.4.35), (D.4.14), and Fig. D.11a

$$o_{T_{k-1}}(\nu) = \frac{O_{T_{k-1}}(\eta; A_{\mathrm{o}})}{\nu} \leq o_{T_{k-1}}(b_j) \leq \bar{f}(T_k).$$

Therefore,

$$\frac{O_{T_{k-1}}(\eta; A_{\mathrm{o}})}{\eta + (\nu - \eta)} \leq \bar{f}(T_k) \to o_{T_{k-1}}(\eta) \leq \frac{\eta + (\nu - \eta)}{\eta} \bar{f}(T_k) \tag{D.4.41}$$

In this subcase

$$\underset{t \in [b_{j-1}, b_j]}{\arg\max}\, o_{T_{k-1}}(t) = \eta.$$

Therefore, we need to find a lower bound on b_j such that $o_{T_{k-1}}(\eta) < \bar{f}(T_{k-1})$. We know,

$$\frac{\bar{f}(T_k)}{\eta} \leq \frac{\bar{f}(T_k)}{b_{j-1}}.$$

Therefore, according to (D.4.41), (D.4.40), and (D.4.6), if

$$\frac{\bar{f}(T_k)}{b_{j-1}} < \frac{\epsilon\rho}{L}$$

then $o_{T_{k-1}}(\eta) < \bar{f}(T_{k-1})$. Therefore, if

$$b_{j-1} > \frac{L}{\rho\epsilon} > \frac{L}{\rho\epsilon}\bar{f}(T_k) \tag{D.4.42}$$

then $o_{T_{k-1}}(\eta) < \bar{f}(T_{k-1})$. Since T_M is chosen large enough such that $\mathcal{B}_j \neq \emptyset$ for all j, we can assert that $W(b_{j-1}; A_\text{o}) \leq T_M = \sigma_M + \delta$. On the other hand, as $k = \min \mathcal{B}_j > 1$, we have that $W(t_j; A_\text{o}) \geq \sigma_1$. Using (5.19), we have

$$t_j - b_{j-1} = \frac{W(b_{j-1}; A_\text{o}) - W(t_j; A_\text{o})}{\rho} \leq \frac{T_M - \sigma_1}{\rho}. \tag{D.4.43}$$

Therefore,

$$b_j - b_{j-1} = \frac{L_j}{C} + t_j - b_{j-1} \leq \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}. \tag{D.4.44}$$

Therefore, if

$$b_j > \frac{L}{\rho\epsilon} + \frac{T_M - \sigma_1}{\rho} + \frac{L}{C} \tag{D.4.45}$$

then $o_{T_{k-1}}(\eta) < \bar{f}(T_{k-1})$. Following the same arguments for the subcase Fig. D.11b, we can have the same lower bound for b_j as (D.4.45).

For the subcase Fig. D.11c, following the same arguments we can show

$$t_j - b_{j-1} \leq \frac{\delta}{\rho}, \qquad \nu - t_j \leq \frac{\delta}{C - \rho}.$$

Therefore,

$$\nu - b_{j-1} \leq \frac{\delta}{\rho} + \frac{\delta}{C - \rho} = \frac{L}{\rho}. \tag{D.4.46}$$

On the other hand, according to (D.4.35), (D.4.16), and Fig. D.11c

$$o_{T_{k-1}}(\nu) = \frac{O_{T_{k-1}}(b_{j-1}; A_{\mathrm{o}})}{\nu} \leq o_{T_{k-1}}(b_j) \leq \bar{f}(T_k).$$

Therefore,

$$\frac{O_{T_{k-1}}(b_{j-1}; A_{\mathrm{o}})}{b_{j-1} + (\nu - b_{j-1})} \leq \bar{f}(T_k) \rightarrow o_{T_{k-1}}(b_{j-1}) \leq \frac{b_{j-1} + (\nu - b_{j-1})}{b_{j-1}} \bar{f}(T_k) \tag{D.4.47}$$

In this subcase

$$\arg\max_{t \in [b_{j-1}, b_j]} o_{T_{k-1}}(t) = b_{j-1}.$$

Similarly, according to (D.4.47), (D.4.46), and (D.4.6), if

$$\frac{\bar{f}(T_k)}{b_{j-1}} < \frac{\epsilon\rho}{L}$$

then $g(b_{j-1}) < \bar{f}(T_{k-1})$. Therefore, if

$$b_{j-1} > \frac{L}{\rho\epsilon} > \frac{L}{\rho\epsilon}\bar{f}(T_k) \tag{D.4.48}$$

then $g(b_{j-1}) < \bar{f}(T_{k-1})$. Therefore the same lower bound on b_j as (D.4.45) will be achieved for this subcase. Following the same arguments for the subcase Fig. D.11d, we can have the same lower bound on b_j.

Case 2: $\tilde{s}_j = b_{j-1}$

This case is shown in Fig. D.8. As it is explained in Lemma D.4.3, in the two subcases D.8c and D.8d we can have cases of having the maximum of the overshoot ratio function, $o_{T_{k-1}}(t)$, over the interval $[b_{j-1}, b_j]$ at $t = b_{j-1}$. The overshoot ratio function for these two subcases is similar to Fig. D.11c and D.11d for $t \in [b_{j-1}, b_j]$. Theses figures are derived using (D.4.25) and (D.4.27). Note that, as it was mentioned in Lemma D.4.3, in subcase Fig. D.8c we can have the maximum of the overshoot function happening at b_j, following the same argument as before, this case however is not considered in here.

For the subcase Fig. D.8c, which its overshoot ratio function is depicted in Fig. D.11c, by following the same argument as before, we have

$$\nu - b_{j-1} \leq \frac{\delta}{C - \rho}. \tag{D.4.49}$$

On the other hand, according to (D.4.35), (D.4.27), and Fig. D.11c

$$o_{T_{k-1}}(\nu) = \frac{O_{T_{k-1}}(b_{j-1}; A_{\mathrm{o}})}{\nu} \leq o_{T_{k-1}}(b_j) \leq \bar{f}(T_k).$$

Therefore,

$$\frac{O_{T_{k-1}}(b_{j-1}; A_{\mathrm{o}})}{b_{j-1} + (\nu - b_{j-1})} \leq \bar{f}(T_k) \rightarrow o_{T_{k-1}}(b_{j-1}) \leq \frac{b_{j-1} + (\nu - b_{j-1})}{b_{j-1}} \bar{f}(T_k) \tag{D.4.50}$$

In this subcase

$$\arg\max_{t \in [b_{j-1}, b_j]} o_{T_{k-1}}(t) = b_{j-1}.$$

Similarly, according to (D.4.50), (D.4.46), and (D.4.6), if

$$\frac{\bar{f}(T_k)}{b_{j-1}} < \frac{\epsilon(C - \rho)}{\delta}$$

then $o_{T_{k-1}}(b_{j-1}) < \bar{f}(T_{k-1})$. Therefore, if

$$b_{j-1} > \frac{\delta}{\epsilon(C - \rho)} = \frac{L}{C\epsilon} > \frac{\delta}{\epsilon(C - \rho)} \bar{f}(T_k) \tag{D.4.51}$$

then $o_{T_{k-1}}(b_{j-1}) < f(T_{k-1})$. As in this case $b_{j-1} = t_j$, the lower bound on b_j in this case will be

$$b_j > \frac{L}{C\epsilon} + \frac{L}{C} \tag{D.4.52}$$

Following the same arguments, we can have the same lower bound for b_j as (D.4.52) for the subcase Fig. D.8d. Therefore, using (D.4.45) and (D.4.52) the lower bound for b_j to ensure (D.4.36) will be

$$b_j > \max\left\{\frac{L}{\rho\epsilon} + \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}, \frac{L}{C\epsilon} + \frac{L}{C}\right\} = \frac{L}{\rho\epsilon} + \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}. \tag{D.4.53}$$

In our case study with $M = 56$, the lower bound in (D.4.53) will be $b_j \geq 2.35\mathtt{e}3$. Therefore, for $j \geq 220$, inequality (D.4.36) holds. $\square$

Proof of Lemma D.4.1. Let $\mathcal{B}_j$ be as defined in Lemma D.4.1 and let $k = \min \mathcal{B}_j$. In lemma D.4.4 we showed if $k > 1$, $\mathcal{I}_j \neq \emptyset$ and

$$b_j > \frac{L}{\rho\epsilon} + \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}. \tag{D.4.54}$$

Then

$$\forall \ell \in \mathcal{I}_j : \quad o_{T_{\ell-1}}(t) \leq \bar{f}(T_{\ell-1}), \qquad t \in [b_{j-1}, b_j(\sigma_\ell)]. \tag{D.4.55}$$

On the other hand, if $\mathcal{I}_j = \emptyset$ or $k = 1$, which in turn means $\mathcal{I}_j = \emptyset$, then $\sigma^*(j) = \sigma_1$. But as $\bar{f}(T_0) = 1$, therefore

$$o_{T_0}(t) \leq \bar{f}(T_0), \qquad t \in [b_{j-1}, b_j(\sigma_1)]. \tag{D.4.56}$$

$\square$

In the next section we show if t is sufficiently large, then the limited constraint in (D.4.7) can be extended to the desired constraint in (5.43).

Lemma D.4.5. Let $k = \min \mathcal{B}_j$, $k > 1$ and assume that b_j satisfies the following lower bound

$$b_j \geq \frac{1}{\epsilon}\left(\frac{L}{C} + \frac{T_M - \sigma_1}{\rho}\right), \tag{D.4.57}$$

where $\epsilon > 0$ is given by

$$\epsilon = \min_{2 \leq k \leq M} [f(T_{k-1}) - f(T_k)]. \tag{D.4.58}$$

Let $\ell \in \{2, \ldots, k\}$ and set $b_j = b_j(\sigma_\ell)$. Then

$$o_{T_{\ell-1}}(b_j) \leq \bar{f}(T_\ell), \tag{D.4.59}$$

implies

$$o_{T_{\ell-1}}(t) < f(T_{\ell-1}), \qquad \forall t \in [b_{j-1}, b_j]. \tag{D.4.60}$$

Proof. Let $t \in [b_{j-1}, b_j]$. We have

$$o_{T_{\ell-1}}(t) \leq \frac{O_{T_{\ell-1}}(b_{j-1}; A_\mathrm{o}) + (t - b_{j-1})}{b_{j-1} + (t - b_{j-1})} =: h(t; b_{j-1}), \tag{D.4.61}$$

for $t \in [b_{j-1}, b_j]$. It can be seen that $h(t; b_{j-1})$ is an increasing function of t in the interval $[b_{j-1}, b_j]$. Thus, we have

$$o_{T_{\ell-1}}(t) \leq h(b_j; b_{j-1}), \quad \forall t \in [b_{j-1}, b_j]. \tag{D.4.62}$$

On the other hand,

$$o_{T_{\ell-1}}(b_j) = \frac{O_{T_{\ell-1}}(b_{j-1}; A_\mathrm{o}) + (b_j - b_{j-1}) - |\mathcal{G}|}{b_{j-1} + (b_j - b_{j-1})} = h(b_j; b_{j-1}) - \frac{|\mathcal{G}|}{b_j} \tag{D.4.63}$$

where $\mathcal{G} := \{t \in [b_{j-1}, b_j] : 0 < W(t; A_\mathrm{o}) < T_{\ell-1}\}$ and $|\mathcal{G}|$ denotes the Lebesgue measure of

$\mathcal{G}$. Using (D.4.62), (D.4.63), and (D.4.59) we have

$$o_{T_{\ell-1}}(t) \leq o_{T_{\ell-1}}(b_j) + \frac{|\mathcal{G}|}{b_j} \leq \bar{f}(T_\ell) + \frac{b_j - b_{j-1}}{b_j} \leq f(T_\ell) + \frac{b_j - b_{j-1}}{b_j}, \tag{D.4.64}$$

where we have used the fact that $|\mathcal{G}| \leq (b_j - b_{j-1})$, and $\bar{f}(T_\ell) \leq f(T_\ell)$ for $\forall\ T_\ell \in \{T_1, \ldots, T_{M-1}\}$. Since T_M is chosen large enough such that $\mathcal{B}_j \neq \emptyset$ for all j, we can assert that $W(b_{j-1}; A_\mathrm{o}) \leq T_M = \sigma_M + \delta$. On the other hand, as $k = \min \mathcal{B}_j > 1$, we have that $W(t_j; A_\mathrm{o}) \geq \sigma_1$. Using (5.19), we have

$$t_j - b_{j-1} = \frac{W(b_{j-1}; A_\mathrm{o}) - W(t_j; A_\mathrm{o})}{\rho} \leq \frac{T_M - \sigma_1}{\rho}. \tag{D.4.65}$$

Therefore,

$$b_j - b_{j-1} = \frac{L_j}{C} + t_j - b_{j-1} \leq \frac{T_M - \sigma_1}{\rho} + \frac{L}{C}. \tag{D.4.66}$$

Since the right-hand side of (D.4.66) is a constant we can choose b_j sufficiently large such that

$$\frac{b_j - b_{j-1}}{b_j} < \epsilon, \tag{D.4.67}$$

where $\epsilon > 0$ is given by (D.4.58). When b_j satisfies (D.4.57), equations (D.4.64), (D.4.67), and (D.4.58) yield the inequality (D.4.60). In our case study with $M = 56$, which has the smallest value for $\epsilon = 0.0089$, with $T_M = 400$, $\sigma_M = 0.1$, $\rho = 0.654$, $L = 10$, and $C = 1$, the lower bound in (D.4.57) will be $b_j \geq 7 \times 10^4$. Therefore, for $j \geq 6125$, inequality (D.4.60) holds. $\square$

D.4.2 Proof of Theorem 5.4.1, Part II

In this section we show in order to achieve the desired constraint in (5.43) rather than the preliminary one in (D.4.7), we need to increase b_j from the lower in bound in (D.4.5) to sufficiently large values. The proof of Theorem 5.4.1 is based on the next two lemmas.

Lemma D.4.6. The (σ^*, ρ) regulator defined by (5.46)–(5.48) produces an output traffic stream that satisfies

$$o_{T_i}(t) \leq \bar{f}(T_i), \quad \text{for } \forall i \in \{1, \ldots, M\}, \tag{D.4.68}$$

for sufficiently large t.

Proof. In order to prove this lemma, we use Fig. D.12 which shows the input workload, $W(t; A_{\mathrm{i}})$, and output workload, $W(t; A_{\mathrm{o}})$, for [2] $t \in [2.5\mathtt{e}4, 3\mathtt{e}4]$ for the numerical example in Section 5.5 with $M = 56$. The corresponding overshoot ratios, $o_{T_i}(t)$ for two values of T_{16} and T_{21} on the interval $[2.5\mathtt{e}4, 3\mathtt{e}4]$ are shown in Fig. D.13. As can be seen in Fig. D.13, over the interval of $[2.5\mathtt{e}4, 3\mathtt{e}4]$ there is violation of the constraints in (D.4.68) for some t, as

$$o_{T_{17}}(t) > \bar{f}(T_{16}), \qquad t \in [t_3, t_6],$$

where $t_3 = 2.73\mathtt{e}4$ and $t_6 = 2.87\mathtt{e}4$ are shown in Fig. D.13. By explaining what happens on the interval $[2.5\mathtt{e}4, 3\mathtt{e}4]$ we can explain why this violation happens and how these violations are avoided when t is sufficiently large. Note that, although this is just one specific example, it can act as a guideline and does not limit the scope of this proof.

As it can be seen in Fig. D.12, at $t = t_1$ output workload increases above T_{16}, and after $t = t_5$ it decreases again to a level below T_{16}. According to Algorithm 6 and (5.47), and as can be seen in Fig. D.13, $o_{T_{16}}(t_1) \leq \bar{f}(T_{17})$. Therefore, at $t = t_1$, $\sigma^*(j)$ will be set to $\sigma^*(j) = \sigma_{17}$ and output workload will increase. As long as $o_{T_{16}}(b_j) \leq \bar{f}(T_{17})$ for $b_j > t_1$ and $W(\tilde{s}_j; A_1) < \sigma_{17}$, this process will continue and $\sigma^*(j)$ will be set to $\sigma^*(j) = \sigma_{17}$ till $W(\tilde{s}_j; A_1)$ is increased to $W(\tilde{s}_j; A_1) > \sigma_{17}$. At this point according to Algorithm 6 and (5.47), $o_{T_{17}}(b_j)$

[2]For notational convenience, we use the E-notation $a\mathtt{e}b := a \times 10^b$.

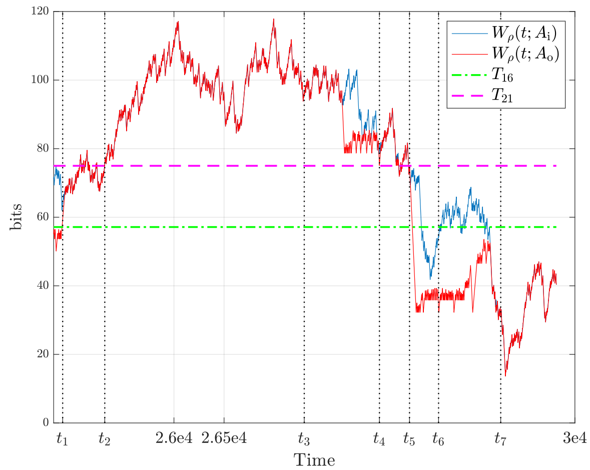


Figure D.12: Input and output workload for $t \in [2.5\mathtt{e}4, 3\mathtt{e}4]$ for the example in Section 5.5 for Algorithm 6 with $M = 56$.

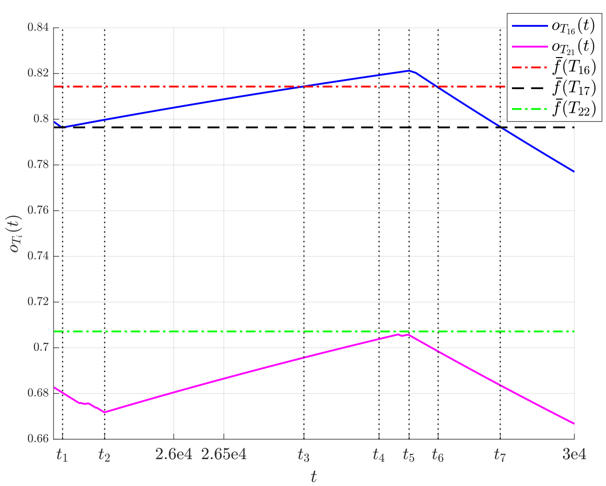


Figure D.13: Overshoot ratio, $o_{T_i}(t)$ for $t \in [2.5\mathtt{e}4, 3\mathtt{e}4]$ and $T_i \in \{T_{16}, T_{21}\}$ for the example in Section 5.5 for Algorithm 6 with $M = 56$.

will be compared against $\bar{f}(T_{18})$ and if $o_{T_{17}}(t) \leq \bar{f}(T_{18})$, as in this example, $\sigma^*(j)$ will be set to $\sigma^*(j) = \sigma_{18}$. On the other hand, at $t = t_5$, $\min \mathcal{B}_j = \sigma_{22}$. Therefore, $o_{T_{21}}(b_j)$ will be compared against $\bar{f}(T_{22})$ and if $o_{T_{21}}(b_j) > \bar{f}(T_{22})$, as in this example, $\sigma^*(j)$ will be set to a value less than σ_{22}. In this example $\sigma^*(j)$ is set to a $\sigma^*(j) = \sigma_{10}$ as $o_{T_{i-1}}(b_j(\sigma_i)) > \bar{f}(T_i)$ for $i = 10, 11, \ldots, 22$.

From the discussion above it can be understood when the output workload at the complete departure time, $W(b_j; A_\mathrm{o})$, increases above σ_i for the jth packet, the overshoot ratio, $o_{T_\ell}(b_m(\sigma_{\ell+1}))$, will be compared against $\bar{f}(T_{\ell+1})$, for $\ell \in \{i, \ldots, M-1\}$ and $m > j$, as long as the output workload stays above σ_i. Therefore, during the interval that the workload is above σ_i, for all $m > j$ such that $W(b_m; A_\mathrm{o}) > \sigma_i$, there is at least one $k \in \{i, \ldots, M-1\}$, such that $o_{T_k}(b_m(\sigma_{k+1})) \leq \bar{f}(T_{k+1})$. Based on this concepts we define threshold violation distance with respect to a threshold value, a bounding value and a traffic stream.

Definition D.4.1. Given a threshold value $\zeta > 0$, a bounding value $\alpha > 0$ and a traffic process A, *threshold violation distance* with respect to R, ζ and α, is defined as the minimum time it takes such that the overshoot ratio reaches the bounding value α. In other words,

$$\mathrm{Dist}_{\zeta,\alpha}(t; A) := \begin{cases} \hat{t}(\zeta) - t + \min_R dt, & \text{s.t: } \; o_\zeta(\hat{t}(\zeta) + dt) = \alpha, \;\text{ if } o_\zeta(\hat{t}(\zeta)) \leq \alpha, \\ 0, & \text{otherwise}, \end{cases} \tag{D.4.69}$$

where $\hat{t}(\zeta)$ is defined as

$$\hat{t}(\zeta) = t + [\zeta - W(t; A)]^+ / (C - \rho) \tag{D.4.70}$$

Note that, in Definition D.4.1, if the output workload is less than the threshold ζ then $\hat{t}(\zeta)$ will be the earliest time the output workload can increase to the threshold level ζ. On the other hand, $\min_R dt$ is the minimum extra time the output workload needs to stay above ζ such that overshoot ratio with respect to ζ reaches the bounding value α. Threshold violation distance for output traffic can be calculated using the following proposition.

Proposition D.4.1. For the output traffic if $o_\zeta(\hat{t}(\zeta)) \leq \alpha$, then

$$\mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o}) = \frac{\hat{t}(\zeta) - t}{1-\alpha} + \frac{\alpha t - O_\zeta(t; A_\mathrm{o})}{1-\alpha} = \frac{\hat{t}(\zeta) - t}{1-\alpha} + \frac{\alpha t - t o_\zeta(t)}{1-\alpha},$$

where $\hat{t}(\zeta)$ can be derived according to (D.4.70) with R replaced by A_o.

Proof. According to (D.4.69) and (5.32), if $o_\zeta(\hat{t}(\zeta)) \leq \alpha$, then

$$\frac{O_\zeta(t + \mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o}); A_\mathrm{o})}{t + \mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o})} = \alpha \tag{D.4.71}$$

But clearly if $o_\zeta(\hat{t}(\zeta)) \leq \alpha$, for the workload we need to have

$$W(t; A_\mathrm{o}) > \zeta, \qquad t \in [\hat{t}(\zeta), t + \mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o})].$$

Therefore,

$$\begin{aligned} O_\zeta(t + \mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o}); A_\mathrm{o}) &= O_\zeta(\hat{t}(\zeta); A_\mathrm{o}) + \mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o}) - (\hat{t}(\zeta) - t) \\ &= O_\zeta(t; A_\mathrm{o}) + \mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o}) - (\hat{t}(\zeta) - t). \end{aligned} \tag{D.4.72}$$

Hence, using equations (D.4.70), (D.4.71) and (D.4.72), Proposition D.4.1 can be derived. □

For the jth packet, we define $\mathrm{dist}_j(T_i)$, for $1 \leq i \leq M-1$, using definition of $\mathrm{Dist}_{\zeta,\alpha}(t; A_\mathrm{o})$ for special values of ζ, α, and t as follow:

$$\mathrm{dist}_j(T_i) := \begin{cases} \mathrm{Dist}_{T_i, \bar{f}(T_{i+1})}(b_j; A_\mathrm{o}), & \text{for } \ i \in \mathcal{L}_j, \\ \mathrm{Dist}_{T_i, \bar{f}(T_i)}(b_j; A_\mathrm{o}), & \text{for } \ i \in \mathcal{M}_j, \end{cases} \tag{D.4.73}$$

where

$$\mathcal{L}_j = \{1 \leq \ell \leq M-1:\ T_\ell > \sigma^*(j)\}, \tag{D.4.74}$$

$$\mathcal{M}_j = \{1 \leq m \leq M-1:\ T_m \leq \sigma^*(j)\}. \tag{D.4.75}$$

When the output workload is $W(b_j; A_\mathrm{o})$, $\mathrm{dist}_j(T_i)$ for $i \in \mathcal{M}_j$, means the extra time the workload can be greater than T_i, such that the desired bound in (D.4.68) is violated. On the other hand, $\mathrm{dist}_j(T_i)$ for $i \in \mathcal{L}_j$, means the time the workload can be greater than T_i, such that the constraint in (5.47) is violated. Note that, as T_M is chosen large enough such that $W(t_j; A_\mathrm{o})$ is always less than T_M, therefore, $\mathrm{dist}_j(T_M)$ can not be defined as the workload never goes beyond T_M.

For the example in Figs. D.12 and D.13, $\mathrm{dist}_j(T_i)$ for $j = 2191$ and $i = 1, 2 \ldots, M-1$, is shown in Fig. D.14. For $j = 2191$, b_j is slightly less than t_2 In Fig. D.12. In Fig. D.14, $\mathrm{dist}_{2191}(T_i)$ is shown in red if $T_i \leq \sigma^*(j)$ and is shown in blue if $T_i > \sigma^*(j)$. In other words, blue bars show how long the workload can stay above the corresponding T_i according to Algorithm 6. Red bars, however, show the longest time the workload can stay above the corresponding T_i such that the desired upper bound at that T_i is violated. Note that, if the blue bars are greater than the red bars for some T_i's, then we can have the cases of the violations of the desired bound at the corresponding T_i's for the red bars. This is actually the case in Fig. D.14. In this case,

$$\mathrm{dist}_{2191}(T_{21}) = 0.301e4 > \mathrm{dist}_{2191}(T_{16}) = 0.201\mathtt{e}4$$

Therefore, as can be seen the workload is allowed to stay above T_{21} according to Algorithm 6 on the interval $t \in [t_2, t_4]$, with $t_2 = 2.531\mathtt{e}4$ and $t_4 = 2.805\mathtt{e}4$. The length of this interval is $t_4 - t_2 = 0.274\mathtt{e}4$, which is greater $\mathrm{dist}(t_1, T_{16})$. Therefore, although according to the Algorithm 6, the output workload is allowed to stay above T_{21}, and no violation of (5.47) happens, the desired bound for T_{16}, however, as can be in seen in Fig. D.13, is violated.

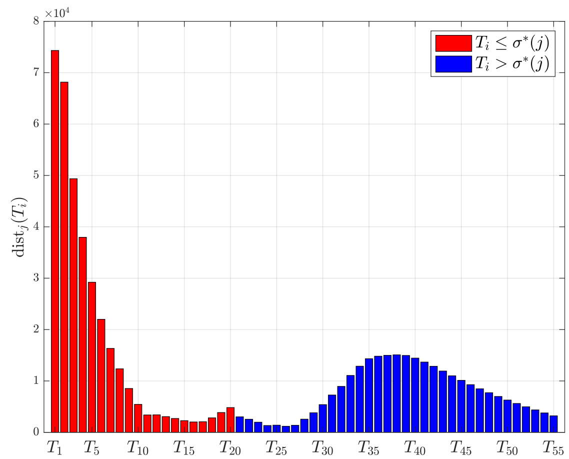


Figure D.14: $\mathrm{dist}_j(T_i)$ for $i = 1, 2 \ldots, M$ for the example in Section 5.5 for Algorithm 6 with $M = 56$, $j = 2191$, $b_j = 2.53\mathtt{e}4$ and $\sigma^*(j) = 71.54$.

On the other hand, for $j = 8950$ and $b_j = 11e4$, when t is sufficiently large, $\mathrm{dist}_{8950}(T_i)$ for $i = 1, 2 \ldots, M$ is shown in Fig. D.15. As we can see in Fig. D.15, all the blue bars are less the red bars in this case.

Based on the discussion for the specific example in Figs. D.14 and D.15, we can generalize these cases and present a sufficient condition on $\mathrm{dist}_j(T_i)$, for $i \in \{1, 2, \ldots, M-1\}$, such that the desired bound in (D.4.68) is satisfied. If output workload is $W(b_j; A_o)$ and b_j is sufficiently large enough, the sufficient condition to satisfy the desired constraint (D.4.68) is,

$$\mathrm{dist}_j(T_\ell) \leq \mathrm{dist}_j(T_m), \quad \forall\ \ell \in \mathcal{L}_j,\ \forall m \in \mathcal{M}_j, \tag{D.4.76}$$

if $\mathcal{M}_j \neq \emptyset$. Note that, if $\mathcal{M}_j = \emptyset$, then $\sigma^*(j) = \sigma_1$. In this case $\bar{f}(\gamma) = 1$ for $\gamma \in [T_0, T_1]$. Therefore, the desired bound of

$$o_{T_0}(t) \leq \bar{f}(\gamma),$$

will never be violated, independent of the duration of the interval that the workload stays above T_0. In the definition of $\mathrm{dist}_j(T_i)$ and in the sufficient condition in (D.4.76), we are

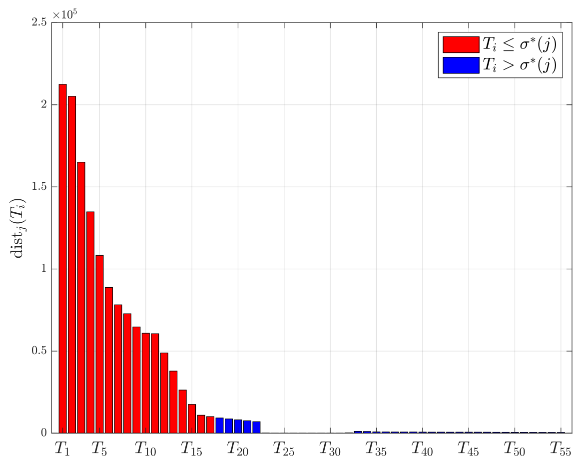


Figure D.15: $\mathrm{dist}_j(T_i)$ for $i = 1, 2 \ldots, M$ for the example in Section 5.5 for Algorithm 6 with $M = 56$, $j = 8950$, $b_j = 11e4$ and $\sigma^*(j) = 60.82$.

just considering the complete departure times and we verify the sufficient condition at those moments. Ascertaining the sufficient condition at those complete departure times moment, however, can guarantee the desired condition in (D.4.68) is satisfied for all sufficiently large t. Because if after the departure of every packet we can assure the duration of the time that the workload stays above the T_m for $\forall T_m \in \mathcal{M}_j$, is less than the time to violate the desired condition in (D.4.68), then the desired condition in (D.4.68) is not only satisfied at the complete departure times, but also it is satisfied at all sufficiently large t.

Note that, if $\mathcal{M}_j \neq \emptyset$, according to (D.4.75) and (5.47),

$$\sigma^*(j) = \sigma_{\max \mathcal{M}_j + 1}. \tag{D.4.77}$$

On the other hand, in the sufficient condition in (D.4.76), if instead of all $m \in \mathcal{M}_j$, just $m = \max \mathcal{M}_j$ is considered the sufficient condition will be simplified as,

$$\mathrm{dist}_j(T_\ell) \leq \mathrm{dist}_j(T_m), \quad \forall\ \ell \in \mathcal{L}_j,\ m = \max \mathcal{M}_j, \tag{D.4.78}$$

if $\mathcal{M}_j \neq \emptyset$ and b_j is sufficiently large enough. This simplified sufficient condition can be explained as follows, when $\sigma^*(j)$ is set according to Algorithm 6 and according to (D.4.77) and (5.47), overshoot ratio at b_j with respect to $T_{\max \mathcal{M}_j}$ is checked against $\bar{f}(T_{\max \mathcal{M}_j+1})$ or

$$o_{T_{\max \mathcal{M}_j}}(b_j) \leq \bar{f}(T_{\max \mathcal{M}_j+1}). \tag{D.4.79}$$

If we make sure the duration of the time that the workload stays above $T_{\max \mathcal{M}_j}$ is less than the time to violate the desired upper bound or

$$o_{T_{\max \mathcal{M}_j}}(j) \leq \bar{f}(T_{\max \mathcal{M}_j+1}), \quad \text{for } \forall t > b_j, \tag{D.4.80}$$

for all packets that b_j is sufficiently large. Then the desired bound in (D.4.68) is never violated for sufficiently large t. It can easily be shown the sufficient condition in (D.4.76) and (D.4.78) are equivalent.

Using Proposition D.4.1, we can simply $\text{dist}_j(T_i)$ in the two following cases:

Case 1: $i \in \mathcal{L}_j$

$$\text{dist}_j(T_i) = \frac{b_j(\bar{f}(T_{i+1}) - o_{T_i}(b_j))}{1 - \bar{f}(T_{i+1})} + \frac{\hat{t}(T_i) - b_j}{1 - \bar{f}(T_{i+1})}, \tag{D.4.81}$$

, if

$$b_j(\bar{f}(T_{i+1}) - o_{T_i}(b_j)) > -(\hat{t}(T_i) - b_j)\bar{f}(T_{i+1}). \tag{D.4.82}$$

Otherwise, $\text{dist}_j(T_i) = 0$. In (D.4.81), $\hat{t}(T_i)$ can be derived from (D.4.70), with $\zeta = T_i$ and $t = b_j$.

Case 2: $i \in \mathcal{M}_j$

$$\text{dist}_j(T_i) = \frac{b_j(\bar{f}(T_i) - o_{T_i}(t))}{1 - \bar{f}(T_i)} + \frac{\hat{t}(T_i) - b_j}{1 - \bar{f}(T_i)}, \tag{D.4.83}$$

, if

$$b_j(\bar{f}(T_i) - o_{T_i}(b_j)) > -(\hat{t}(T_i) - b_j)\bar{f}(T_i). \tag{D.4.84}$$

Otherwise, $\text{dist}_j(T_i) = 0$. In (D.4.83), $\hat{t}(T_i)$ can be derived from (D.4.70), with $\zeta = T_i$ and $t = b_j$.

In order to show the desired condition in (D.4.68) is satisfied for sufficiently large values of t if $\sigma^*(j)$ is chosen according to Algorithm 6, we assume b_j is sufficiently large and we show the sufficient condition in (D.4.78) is satisfied.

According to (D.4.81) and (D.4.83), if we have the following inequality, then the sufficient condition in (D.4.78) is also satisfied,

$$\begin{aligned}&\frac{b_j(\bar{f}(T_{\ell+1}) - o_{T_\ell}(b_j))}{1-\bar{f}(T_{\ell+1})} + \frac{\hat{t}(T_\ell) - b_j}{1-\bar{f}(T_{\ell+1})} \leq \frac{b_j(\bar{f}(T_m) - o_{T_m}(b_j))}{1-\bar{f}(T_m)},\\ &\forall \ell \in \mathcal{L}_j,\ m = \max \mathcal{M}_j,\ \mathcal{M}_j \neq \emptyset.\end{aligned} \tag{D.4.85}$$

The second term in the LHS can be bounded according to (D.4.70) as follows,

$$\begin{aligned}\frac{\hat{t}(T_\ell) - b_j}{1-\bar{f}(T_{\ell+1})} &= \frac{[T_\ell - W(b_j; A_\text{o})]^+}{(1-\bar{f}(T_{\ell+1}))(C-\rho)} \leq \frac{T_{M-1} - W(b_j; A_\text{o})}{(1-\bar{f}(T_M))(C-\rho)}\\ &\leq \frac{T_{M-1} - \sigma_1}{(1-\bar{f}(T_M))(C-\rho)} := c_0 > 0,\end{aligned} \tag{D.4.86}$$

where we have used $W(t; A_\text{o}) > \sigma_1$ as $\mathcal{M}_j \neq \emptyset$. Therefore, we can simply the inequality in (D.4.85) into a more conservative simplified inequality as follows,

$$\begin{aligned}&\frac{b_j(\bar{f}(T_{\ell+1}) - o_{T_\ell}(b_j))}{1-\bar{f}(T_{\ell+1})} + c_0 \leq \frac{b_j(\bar{f}(T_m) - to_{T_m}(b_j))}{1-\bar{f}(T_m)},\\ &\forall \ell \in \mathcal{L}_j,\ m = \max \mathcal{M}_j,\ \mathcal{M}_j \neq \emptyset.\end{aligned} \tag{D.4.87}$$

Note that, if the inequality (D.4.87) is satisfied, then inequality (D.4.85) is also satisfied.

By doing some manipulations we can reach the following inequality,

$$\bar{f}(T_{\ell+1}) + \frac{c_2}{b_j c_1} - \frac{\bar{f}(T_m) - o_{T_m}(b_j)}{c_1} \leq o_{T_\ell}(b_j), \tag{D.4.88}$$

where

$$c_1 = \frac{1 - \bar{f}(T_m)}{1 - \bar{f}(T_{\ell+1})}, \tag{D.4.89}$$

$$c_2 = c_0(1 - \bar{f}(T_m)). \tag{D.4.90}$$

In other words, for the sufficient condition in (D.4.76) to hold, the overshoot ratio, $o_{T_\ell}(b_j)$ for $\forall \ell \in \mathcal{L}_j$ should be higher than the lower bound specified in (D.4.88). By considering the upper bound on $o_{T_m}(b_j)$ in (D.4.79), the lower bound in (D.4.88) can be simplified into a more conservative inequality as follows,

$$\bar{f}(T_{\ell+1}) + \frac{c_2}{b_j c_1} - \frac{\epsilon_m}{c_1} \leq o_{T_\ell}(b_j), \tag{D.4.91}$$

where

$$\epsilon_m := \bar{f}(T_m) - \bar{f}(T_{m+1}). \tag{D.4.92}$$

Note that, if the lower bound in (D.4.91) is satisfied, then the lower bound in (D.4.88) is also satisfied.

For the numerical example in Figs. D.13-D.15, the overshoot ratio, $o_{T_{16}}(t)$ and the lower bounds in (D.4.88) and (D.4.91) are shown in Fig. D.16. As it was mentioned before, these lower bounds are sufficient conditions for the desired constraint in (D.4.68) to hold. The intervals on which the desired constraint is violated or,

$$o_{T_{16}}(t) > \bar{f}(T_{16}), \tag{D.4.93}$$

are shown in shaded blue areas. Therefore, as it can be seen in Fig. D.16, there are some

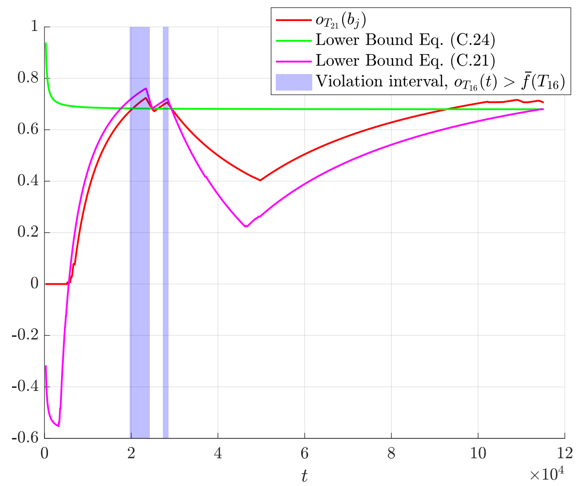


Figure D.16: $o_{T_{16}}(t)$ and the corresponding lower bounds in (D.4.88) and (D.4.91), for the example in Section 5.5 for Algorithm 6 with $M = 56$.

parts that these lower bounds are violated but the desired constraint in (D.4.68) is not violated. On the other hand, on the intervals that the desired constraint in (D.4.68) is violated, as it is shown the corresponding lower bounds are also violated. As it can be seen in Fig. D.16, we do not need a very large b_j to satisfy the lower bound in (D.4.88). However, for the more conservative lower bound in (D.4.91), a larger b_j is necessary.

Now we show when b_j is sufficiently large, the lower bound in (D.4.91) holds. Let define the event $\xi_j(T_\ell)$, for $\ell \in \{1, 2, \ldots, M-1\}$ and the kth packet as,

$$\xi_k(T_\ell) := \{W(\tilde{s}_k; A_1) \geq T_\ell \cap W(t_k; A_{\mathrm{o}}) < T_\ell\} \tag{D.4.94}$$

In other words, when the event $\xi_k(T_\ell)$ occurs, the kth packet is delayed enough such that the output workload becomes less than T_ℓ. The event $\xi_k(T_\ell)$ occurs when

$$o_{T_\ell}(b_k(\sigma_{\ell+1})) > f(T_{\ell+1}). \tag{D.4.95}$$

It can be easily shown between input traffic overshoot ratio, the internal traffic overshoot ration and the output traffic overshoot ratio we have the following relation

$$o_{T_\ell}(t) \leq \frac{O_{T_\ell}(t; A_1)}{t} \leq \frac{O_{T_\ell}(t; A_{\mathrm{i}})}{t} \tag{D.4.96}$$

Note that, due to ergodicity and stationarity of the input and internal traffic, we have

$$\frac{O_{T_\ell}(t; A_{\mathrm{i}})}{t} \sim \mathsf{P}\{W(t; A_{\mathrm{i}}) \geq T_\ell\},$$

$$\frac{O_{T_\ell}(t; A_1)}{t} \sim \mathsf{P}\{W(t; A_1) \geq T_\ell\}$$

Therefore, it can be shown if the probability of the input traffic workload being greater or equal to than T_ℓ is greater or equal to than $\bar{f}(T_\ell)$, then the probability of the event $\xi_j(T_\ell)$ is greater than zero. In other words,

$$\mathsf{P}\{W(t; A_{\mathrm{i}}) \geq T_\ell\} \geq \bar{f}(T_\ell) \implies \mathsf{P}\{\xi_j(T_\ell)\} > 0. \tag{D.4.97}$$

Define $\tilde{t}_{T_\ell}(t)$ for $\ell \in \{1, 2, \ldots, M-1\}$ as the last time before t that event $\xi_k(T_\ell)$ occur, i.e.,

$$\tilde{t}_{T_\ell}(t) = \max\{b_k \leq t : \mathbf{1}_{\xi_k(T_\ell)} = 1\}, \tag{D.4.98}$$

for $\ell \in \{1, 2, \ldots, M-1\}$, where

$$\mathbf{1}_A = \begin{cases} 1 & \text{if event } A \text{ occurs,} \\ 0 & \text{if event } A \text{ does not occur.} \end{cases} \tag{D.4.99}$$

As the output workload is stationary and ergodic and the probability of the event is greater than zero, therefore, the interval between consecutive occurs of the events $\xi_k(T_\ell)$ is bounded. Next we show for the lower bound in (D.4.91) to hold, $b_j - \tilde{t}_{T_\ell}(b_j)$ should have an upper bound. In other words, we find the minimum value of $b_j - \tilde{t}_{T_\ell}(b_j)$, such that the lower bound in (D.4.91) is violated and then we verify that when b_j is sufficiently large, $b_j - \tilde{t}_{T_\ell}(b_j)$ will be always less than this minimum value.

Note that,

$$W(t; A_{\rm o}) \leq T_\ell, \quad \forall t \in [\tilde{t}_{T_\ell}(b_j), b_j]. \tag{D.4.100}$$

Therefore, according to (D.4.96), in order to find the minimum value of $b_j - \tilde{t}_{T_\ell}(b_j)$, such that the lower bound in (D.4.91) is violated we consider

$$o_{T_\ell}(b_j) = \bar{f}(T_{\ell+1}) + \frac{c_2}{b_j c_1} - \frac{\epsilon_m}{c_1}, \tag{D.4.101}$$

$$o_{T_\ell}(\tilde{t}_{T_\ell}(b_j)) = \bar{f}(T_{\ell+1}). \tag{D.4.102}$$

Therefore, according to (5.32) and (D.4.100),

$$b_j - \tilde{t}_{T_\ell}(b_j) = b_j \frac{\epsilon_m - c_2/b_j}{c_1 \bar{f}(T_{\ell+1})} = O(b_j). \tag{D.4.103}$$

Therefore, the minimum time interval that needs to pass between $\tilde{t}_{T_\ell}(b_j)$ and b_j, such that the lower bound in (D.4.91) is violated is linearly proportional to b_j. But as b_j increases the time interval between consecutive occurrences of the event $\xi_k(T_\ell)$, will be less than $O(b_j)$ with probability 1. Note that, $b_j - \tilde{t}_{T_\ell}(b_j)$ is less the time interval between consecutive occurrences of the event $\xi_k(T_\ell)$. Therefore, $b_j - \tilde{t}_{T_\ell}(b_j)$ will be always less then the upper bound derived in (D.4.103). Hence, the lower bound in (D.4.91) is always met for sufficiently large values of b_j.

For the numerical example in Figs. D.13-D.15, $b_j - \tilde{t}_{T_{21}}(b_j)$ and the higher bound in (D.4.103) are shown in Fig. D.17. As it can be seen, when b_j is sufficiently large, $b_j - \tilde{t}_{T_{21}}(b_j)$ will be bounded by the higher bound in (D.4.103).

As mentioned in Appendix D.4.1, for this numerical example the lower bound on b_j, in

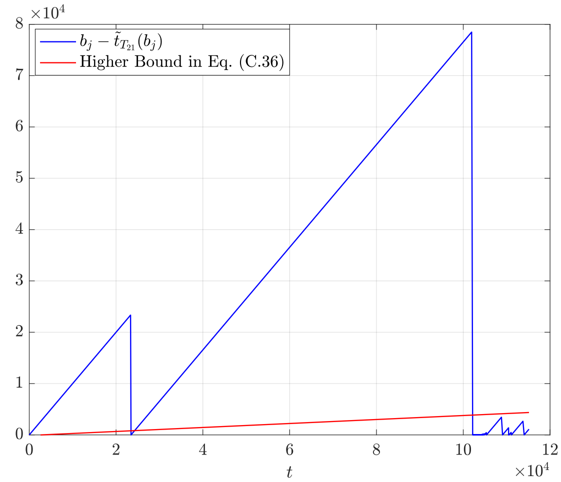


Figure D.17: $b_j - \tilde{t}_{T_{21}}(b_j)$ and the higher bound in (D.4.103) for the example in Section 5.5 for Algorithm 6 with $M = 56$.

order to satisfy the preliminary constraint in (D.4.7), is $b_j \geq 2.35 \times 10^3$. In this example, in order to satisfy the desired bound in (5.43), however, the lower bound is increased to $b_j \geq 10^5$. □

Proof of Theorem 5.4.1. In Lemma D.4.6, we showed if t is sufficiently large, then in a (σ^*, ρ) traffic regulator defined by (5.46)–(5.48),

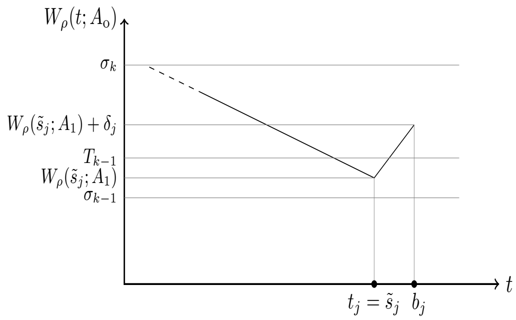
$$o_{T_i}(t) \leq \bar{f}(T_i) \quad \text{for } \forall i \in \{1, \ldots, M\}. \tag{D.4.104}$$

Therefore, using the same argument as in Appendix D.2, and using Corollary D.2.1 and the definition of $\bar{f}(\gamma)$ as (5.41) for the case $M = M_{\max}$, or using Corollary D.2.2 and the definition of $\bar{f}(\gamma)$ as (D.2.46) for the case $M < M_{\max}$ we can show

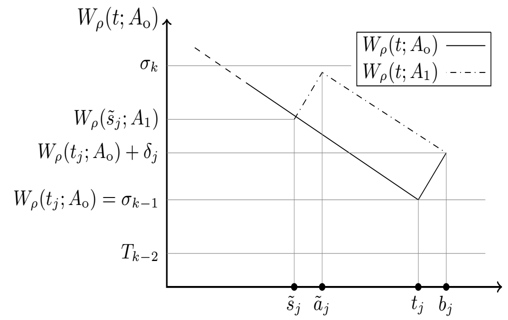
$$o_\gamma(t) \leq f(\gamma), \qquad t \in [b_{j-1}, b_j(\sigma^*(j))], \quad \forall \gamma \in [T_1, T]. \tag{D.4.105}$$

□

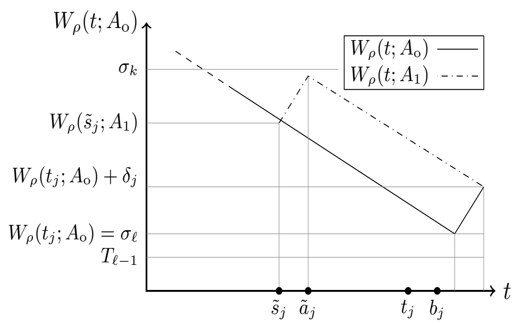
In practice the sufficiently large t constraint for. Algorithm 6 is reasonable as we are



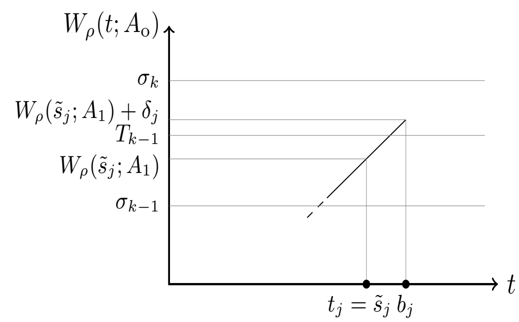
(a) $\tilde{s}_j = s_j$: $\ell = k$



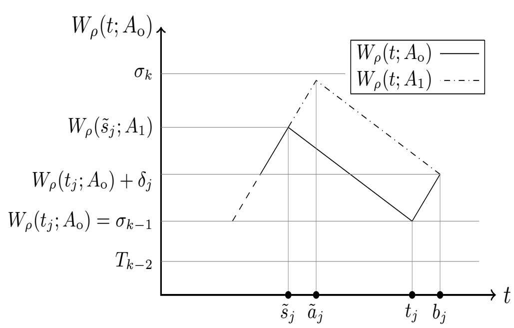
(b) $\tilde{s}_j = s_j$: $\ell = k-1$



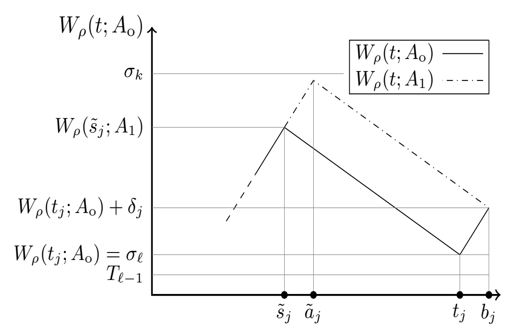
(c) $\tilde{s}_j = s_j$: $\ell < k-1$



(d) $\tilde{s}_j = b_{j-1}$: $\ell = k$



(e) $\tilde{s}_j = b_{j-1}$: $\ell = k-1$



(f) $\tilde{s}_j = b_{j-1}$: $\ell < k-1$

Figure D.18: The process of determining $\sigma^*(j)$ according to Theorem 5.4.1.

approximating the overshoot probability with the overshoot ratio in (5.33), and this approximation is asymptotically valid. In Algorithm 6 we need to compute the index set $\mathcal{I}_j$ in (5.47). The process for computing set $\mathcal{I}_j$ is depicted in Fig. D.18 for two possible cases: 1) $\tilde{s}_j = s_j$ in Figs. D.18a–D.18c and $\tilde{s}_j = b_{j-1}$ in Figs. D.18d–D.18f. In Fig. D.18, $k = \min \mathcal{B}_j$. In the first step of computing $\mathcal{I}_j$ according to (5.47), ℓ is set to k as in Figs. D.18a and D.18d. In these cases according to (5.18), $t_j = \tilde{s}_j$. Then $o_{T_{\ell-1}}(b_j)$ is determined using Proposition 5.4.1. If the condition in (5.47) holds for $\ell = k$, then $k \in \mathcal{I}_j$. Therefore, σ^* will be set as $\sigma^* = \sigma_k$ and the algorithm will terminate at this step. Otherwise, in the next step we set $\ell = k - 1$ as in Figs. D.18b and D.18e. In these cases, t_j will be determined according to (5.18). Again $O_{T_{\ell-1}}(b_j; A_{\mathrm{o}})$ will be determined using Proposition 5.4.1 and the condition in (5.47) is checked for $\ell = k - 1$. If $k - 1 \in \mathcal{I}_j$, then using the same argument as before we set $\sigma^* = \sigma_{k-1}$ and the algorithm will terminate at this step. Otherwise these steps are continued as shown in Figs. D.18c and D.18f and the same process is repeated. If $\mathcal{I}_j$ is determined to be empty, then we set $\sigma^* = \sigma_1$.

Bibliography

Bibliography

[1] "BuTools," http://webspn.hit.bme.hu/ telek/tools/butools/index.php, accessed: 2019-03-29.

[2] "MATLAB," https://www.mathworks.com/help/matlab/ref/fzero.html, accessed: 2019-07-23.

[3] "iPerf source code," 2020. [Online]. Available: https://iperf.fr/

[4] M. Abramowitz and I. Stegan, *Handbook of mathematical functions.* New York: Dover Publications, 1964.

[5] G. E. Andrews, *Theory of Partitions.* Cambridge University Press, 1976.

[6] D. Anick, D. Mitra, and M. M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *The Bell System Technical Journal*, vol. 61, no. 58, pp. 1871–1894, Oct. 1982.

[7] S. Asmussen, *Applied Probability and Queues*, 2nd ed., ser. Stochastic Modelling and Applied Probability. New York: Springer-Verlag, 2003, vol. 51.

[8] S. Asmussen, O. Nerman, and M. Olsson, "Fitting phase-type distributions via the EM algorithm," *Scandin. J. Statist.*, vol. 23, no. 4, pp. 419–441, Dec. 1996.

[9] R. Bateman, *Higher Transcendental Functions*, A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi, Eds. New York: McGraw-Hill, 1953, vol. 1.

[10] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization tech-nique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, pp. 164–171, 1970.

[11] M. Bladt and B. Nielsen, *Matrix-Exponential Distributions in Applied Probability.* New York, NY: Springer, 2017.

[12] A. Bobbio and A. Cumani, "ML estimation of the parameters of a PH distribution in triangular canonical form," *Comput. Perform. Eval.*, pp. 33–46, Jan. 1992.

[13] A. Bobbio and M. Telek, "A benchmark for ph estimation algorithms: results for acyclic-ph," *Stoch. Models*, vol. 10, no. 3, pp. 661–677, 1994.

[14] O. J. Boxma and J. W. Cohen, "The M/G/1 queue with heavy-tailed service time distribution," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 749–763, Jun. 1998.

[15] O. Boxma and V. Dumas, "Fluid queues with long-tailed activity period distributions," *Comput. Commun.*, vol. 21, no. 17, pp. 1509 – 1529, 1998.

[16] L. Brancik, "Techniques of matrix exponential function derivative for electrical engineering simulations," in *Proc. IEEE Int. Ind. Technol.(ICIT)*, Dec. 2006, pp. 2608–2613.

[17] S. Bregni, P. Giacomazzi, and G. Saddemi, "Properties of the traffic output by a leaky-bucket policer with long-range dependent input traffic," in *Proc. IEEE Int. Conf. Communications*, Glasgow, UK, Jun. 2007, pp. 603–609.

[18] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the 'leaky bucket' policing mechanism in ATM networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 3, pp. 335–342, April 1991.

[19] R. Byrd, M. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM J. Optim.*, vol. 9, no. 4, pp. 877–900, 1999.

[20] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Math. Program.*, vol. 89, no. 1, pp. 149–185, Nov. 2000.

[21] E. Çinlar, *Probability and Stochastics.* New York: Springer, 2011.

[22] C. S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Trans. Autom. Control*, vol. 39, no. 5, pp. 913–931, May 1994.

[23] C.-S. Chang, *Performance Guarantees in Communication Networks*, ser. Telecommunication Networks and Computer Systems. London: Springer-Verlag, 2000.

[24] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[25] ——, "A calculus for network delay. II. Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132–141, Jan. 1991.

[26] A. Cumani, "On the canonical representation of homogeneous markov processes modelling failure - time distributions," *Microelectron. Reliab.*, vol. 22, no. 3, pp. 583 – 602, 1982.

[27] G. Dán, V. Fodor, and G. Karlsson, "Packet size distribution: An aside?" in *Quality of Service in Multiservice IP Networks*, M. Ajmone Marsan, G. Bianchi, M. Listanti, and M. Meo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 75–87.

[28] K. De Schepper, O. Bondarenko, I.-J. Tsang, and B. Briscoe, "PI2: A Linearized AQM for Both Classic and Scalable TCP," in *Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, 2016.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. R. Stat. Soc. Series B Stat. Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[30] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 963–977, 2004.

[31] Y. Ephraim and N. Merhav, "Hidden Markov Processes," *IEEE Transactions on Information Theory*, vol. 48, no. 2, pp. 1518–1569, June 2002.

[32] G. Fairhurst and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)," *RFC 3366*, Aug. 2002.

[33] Y. Fang and I. Chlamtac, "Teletraffic analysis and mobility modeling of PCS networks," *IEEE Transactions on Communications*, vol. 47, no. 7, pp. 1062–1072, Jul. 1999.

[34] A. Feldmann and W. Whitt, "Fitting mixtures of exponentials to long-tail distributions to analyze network performance models," in *Proc. IEEE INFOCOM*, vol. 3, Apr. 1997, pp. 1096–1104.

[35] M. Fidler and A. Rizk, "A guide to stochastic network calculus," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 59–86, 2015.

[36] W. Fischer and K. Meier-Hellstern, "The markov-modulated poisson process (MMPP) cookbook," *Perform. Eval.*, vol. 18, no. 2, pp. 149 – 171, 1993.

[37] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. Diot, "Packet-level traffic measurements from the Sprint IP backbone," *IEEE Network*, vol. 17, no. 6, pp. 6–16, 2003.

[38] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Nav. Res. Logist.*, vol. 3, no. 1-2, pp. 95–110, 1956.

[39] T. C. Fung, "Computation of the matrix exponential and its derivatives by scaling and squaring," *Int. J. Numer. Methods Eng.*, vol. 59, no. 10, pp. 1273–1286, Feb. 2004.

[40] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *Communications of the ACM*, vol. 55, no. 1, pp. 57–65, 2012.

[41] C. D. Guerrero and M. A. Labrador, "On the applicability of available bandwidth estimation techniques and tools," *Computer Communications*, vol. 33, no. 1, pp. 11–22, Jan. 2010.

[42] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64—-74, 2008.

[43] A. Horváth and M. Telek, "Markovian modeling of real data traffic: Heuristic phase type and MAP fitting of heavy tailed and fractal like samples," in *Performance Evaluation of Complex Systems: Techniques and Tools*, ser. LNCS, vol. 2459, Berlin, Heidelberg, Sep. 2002, pp. 405–434.

[44] ——, "PhFit: A general phase-type fitting tool," in *Proc. 12th Perform. TOOLS*, Apr. 2002, pp. 82–91.

[45] T. Hrubý, "Byte queue limits," in *Linux Plumbers Conference*, 2012. [Online]. Available: https://blog.linuxplumbersconf.org/2012/resources-slides-and-videos/index.html

[46] Z. Hu, D. Zhang, A. Zhu, Z. Chen, and H. Zhou, "SLDRT: A measurement technique for available bandwidth on multi-hop path with bursty cross traffic," *Computer Networks*, vol. 56, no. 14, pp. 3247–3260, 2012.

[47] D. M. J. R. Gallardo and L. Orozco-Barbosa, "Use of α-stable self-similar stochastic processes for modeling traffic in broadband networks," *Perform. Eval.*, vol. 40, no. 1, pp. 71 – 98, 2000.

[48] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *Passive and Active Network Measurement (PAM) Conference*, 2002.

[49] P. R. Jelenković, "Long-tailed loss rates in a single server queue," in *Proc. IEEE INFOCOM '98*, vol. 3, San Francisco, March 1998, pp. 1462–1469 vol.3.

[50] Y. Jiang and Y. Liu, *Stochastic Network Calculus.* Springer-Verlag, 2008.

[51] Y. Jiang, Q. Yin, Y. Liu, and S. Jiang, "Fundamental calculus on generalized stochastically bounded bursty traffic for communication networks," *Comput. Netw.*, vol. 53, no. 12, pp. 2011 – 2021, Aug. 2009.

[52] F. P. Kelly, *Reversibility and Stochastic Networks.* Cambridge University Press, 1979.

[53] R. E. A. Khayari, R. Sadre, and B. R. Haverkort, "Fitting world-wide web request traces with the EM-algorithm," *Perform. Eval.*, vol. 52, no. 2, pp. 175 – 191, Apr. 2003.

[54] H. Kobayashi and B. L. Mark, *System Modeling and Analysis: Foundations of System Performance Evaluation.* Pearson Education, Inc., 2009.

[55] H. Kobayashi, B. L. Mark, and W. Turin, *Probability, Random Processes, and Statistical Analysis.* Cambridge University Press, 2012.

[56] M. Kordi Boroujeny, Y. Ephraim, and B. L. Mark, "Phase-type bounds on network performance," in *Proc. Inf. Sci. Sys. (CISS)*, Mar. 2018, pp. 1–6.

[57] M. Kordi Boroujeny and B. L. Mark, "Design of a stochastic traffic regulator for end-to-end network delay guarantees," arXiv:2008.07721 [cs.IT], Aug. 2020.

[58] ——, "A framework for providing stochastic delay guarantees in communication networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[59] ——, "Traffic workload envelope for network performance guarantees with multiplexing gain," in *IEEE Global Communications Conference (GLOBECOM)*, accepted for publication, July 2022.

[60] ——, "Design of a stochastic traffic regulator for end-to-end network delay guarantees," *IEEE/ACM Transactions on Networking*, accepted for publication, May 2022.

[61] M. Kordi Boroujeny, B. L. Mark, and Y. Ephraim, "Tail-limited phase-type burstiness bounds for network traffic," in *Proc. Inf. Sci. Sys. (CISS)*, Mar. 2019, pp. 1–6.

[62] ——, "Stochastic traffic regulator for end-to-end network delay guarantees," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[63] A. Kosta, N. Pappas, and V. Angelakis, "Age of Information: A New Concept, Metric, and Tool," *Foundations and Trends in Networking*, vol. 12, no. 3, pp. 162–259, 2017.

[64] H. H. G. Leonard Kleinrock, *Queueing systems. Volume II, Computer applications.* New York: John Wiley, 1976, vol. II.

[65] H. Lin, M. Liu, A. Zhou, H. Liu, and Z. Li, "A novel hybrid probing technique for end-to-end available bandwidth estimation," in *Local Computer Networks (LCN)*, 2010.

[66] C. Liu, J. Tourrilhes, C.-N. Chuah, and P. Sharma, "Voyager: Revisiting available bandwidth estimation with a new class of methods—decreasing- chirp-train methods," *IEEE/ACM Transactions on Networking*, 2022.

[67] R. M. Loynes, "The stability of a queue with non-independent inter-arrival and service times," *Math. Proc. Cambridge*, vol. 58, no. 3, p. 497–520, 1962.

[68] D. M. Lucantoni, "New results on the single server queue with a batch markovian arrival process," *Stoch. Models*, vol. 7, no. 1, pp. 1–46, 1991.

[69] D. Lucantoni, "New results on the single server queue with a batch Markovian arrival process," *Communications in Statistics. Stochastic Models*, vol. 7, no. 1, pp. 1–46, 1991.

[70] A. Mandelbaum, A. Sakov, and S. Zeltyn, "Empirical analysis of a call center," Technion, Israel Institute of Technology, Tech. Rep., 2000.

[71] B. L. Mark and G. Ramamurthy, "Real-time estimation and dynamic renegotiation of UPC parameters for arbitrary traffic sources in ATM networks," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 811–827, Dec. 1998.

[72] *Optimization Toolbox User's Guide*, R2018b, The MathWorks Inc., Natick, MA, 2018.

[73] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, 2nd ed. Hoboken, NJ: Wiley, 2008.

[74] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, "Controlled Delay Active Queue Management," *RFC 8289*, vol. 1, pp. 1–25, Jan. 2018.

[75] H. Okamura, T. Dohi, and K. S. Trivedi, "A refined EM algorithm for PH distributions," *Perform. Eval.*, vol. 68, no. 10, pp. 938 – 954, Oct. 2011.

[76] M. Olsson, "The EMpht-programme," Department of Mathematics, Chalmers University of Technology, and Göteborg University, Tech. Rep., jun 1998.

[77] C. A. O'Cinneide, "On non-uniqueness of representations of phase-type distributions," *Stoch. Models*, vol. 5, no. 2, pp. 247–259, 1989.

[78] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[79] A. K. Paul, A. Tachibana, and T. Hasegawa, "An enhanced available bandwidth estimation technique for an end-to-end network path," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 768–781, 2016.

[80] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, Jun. 1995.

[81] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, no. 6, pp. 27–35, 2003.

[82] V. Ramaswami, "The N/G/1 queue and its detailed analysis," *Advances in Applied Probability*, vol. 12, no. 1, pp. 222–261, 1980.

[83] P. Reinecke, T. Krauß, and K. Wolter, "Cluster-based fitting of phase-type distributions to empirical data," *Comput. Math. Appl.*, vol. 64, no. 12, pp. 3840 – 3851, Dec. 2012.

[84] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Passive and Active Network Measurement (PAM) Conference*, 4 2003.

[85] D. Starobinski and M. Sidi, "Stochastically bounded burstiness for communication networks," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 206–212, Jan. 2000.

[86] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *ACM Internet Measurement Conference (IMC)*, 2003.

[87] A. Thummler, P. Buchholz, and M. Telek, "A novel approach for phase-type fitting with the EM algorithm," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 3, pp. 245–258, Jul. 2006.

[88] F. Topsœ, "Some bounds for the logarithmic function," in *Inequality Theory and Applications*, Y. Cho, J. K. Kim, and S. S. Dragomir, Eds. New York: Nova Science Publishers, 2007, vol. 4, pp. 137–151.

[89] V. Tran, J. Tourrilhes, K. K. Ramakrishnan, and P. Sharma, "Accurate available bandwidth measurement with packet batching mitigation for high speed networks," in *IEEE Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2021.

[90] J. Turner, "New directions in communications (or which way to the information age?)," *IEEE Comm. Mag.*, vol. 24, no. 10, pp. 8–15, Oct. 1986.

[91] S. Vamvakos and V. Anantharam, "On the departure process of a leaky bucket system with long-range dependent input traffic," *Queueing Systems*, vol. 28, pp. 191–214, May 1998.

[92] R. Waltz, J. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Math. Program.*, vol. 107, no. 3, pp. 391–408, Jul. 2006.

[93] J. Wang, J. Liu, and C. She, "Segment-based adaptive hyper-Erlang model for long-tailed network traffic approximation," *J. Supercomput.*, vol. 45, no. 3, pp. 296–312, Sep. 2008.

[94] W. Willinger, M. S. Taqqu, W. E. Leland, and D. V. Wilson, "Self-similarity in high-speed packet traffic: Analysis and modeling of Ethernet traffic measurements," *Stat. Sci.*, vol. 10, no. 1, pp. 67–85, Feb. 1995.

[95] R. W. Wolff, *Stochastic Modeling and the Theory of Queues.* New Jersey: Prentice-Hall, 1989.

[96] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 372–385, Jun. 1993.

[97] ——, "Generalized processor sharing networks with exponentially bounded burstiness arrivals," in *IEEE INFOCOM Proc.*, vol. 2, Jun. 1994, pp. 628–634.

[98] Q. Yin, J. Kaur, and F. D. Smith, "Can bandwidth estimation tackle noise at ultra-high speeds?" in *2014 IEEE 22nd International Conference on Network Protocols*, 2014, pp. 107–118.

[99] Q. Yin, Y. Jiang, S. Jiang, and P. Y. Kong, "Analysis on generalized stochastically bounded bursty traffic for communication networks," in *Proc. IEEE Local Comput. Netw. (LCN)*, Nov. 2002, pp. 141–149.

Curriculum Vitae

Massieh Kordi Boroujeny received the B.Sc degree in electrical engineering from the Shahid Bahonar University of Kerman in 2009, and the M.Sc. degree in electrical engineering from the Isfahan University of Technology in 2012. He worked as a research intern for HP labs, Palo Alto, CA from Sep. 2020 to Aug. 2022. His research interests include network calculus, queuing theory, network resource allocation, and network measurement and analysis.