Scheduling for heavy-tailed and light-tailed workloads in queueing systems

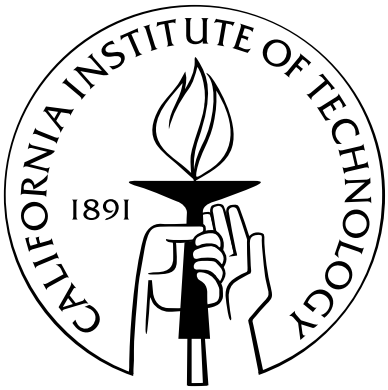
Thesis by

Jayakrishnan U. Nair

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2012

(Defended May 1, 2012)

This thesis is dedicated to

my wife Sushree,

whose companionship makes my life complete,

and my parents,

whose love and support have made everything possible.

Acknowledgements

I have had a very memorable five years as a graduate student at Caltech, and it is with great pleasure that I acknowledge the people who made this possible.

First, I thank my advisor Adam Wierman for his guidance over the years, on research matters, and beyond. No Ph.D. career is without the occasional roadblock, and Adam's encouragement has been instrumental in helping me navigate around mine. Adam's boundless excitement about research, his ability to communicate complex ideas cleanly and effectively, and his knack of always seeing the big picture, without being encumbered by the details, have been a constant source of inspiration for me. It's been an honor working with you, Adam!

I am also very grateful to my advisor Steven Low for his support and advice throughout my graduate study. Steven guided my first research endeavors at Caltech, and got me interested in heavy-tailed phenomena as a theme for my thesis. Thanks, Steven! I would like to thank my thesis committee members: Mani Chandy, Babak Hassibi, and Tracey Ho, for their valuable feedback on this thesis.

I have had the chance to work with some brilliant collaborators as a graduate student, I am grateful to them for the opportunity. Collaborations with Bert Zwart, Sachin Adlakha, Krishna Jagannathan, and Lachlan Andrew have been highly rewarding. I also thank Bert for hosting me at CWI in the Netherlands on two occasions. I am thankful to Prof. Borkar for hosting me at the Tata Institute of Fundamental Research in India over the summer of 2010.

I have benefited immensely from the research environment in the RSRG group. I thank the group members — working alongside you guys has been very inspiring. I would also like to thank the exceptionally helpful administrative staff in Annenberg, especially Sydney Garstang.

Moving to life outside academia, I would like to acknowledge the desi gang at Caltech for some wonderful times. I specially wish to thank Rangoli, Krishna, Prabha, Varun, Uday, Shweta, and Mayank.

Finally, I would like to express my heartfelt gratitude to my family. I would not be here without the love and support of my parents. My wife Sushree has simultaneously been my support system, and the prime source of joy and spice in my life during these Ph.D. years. I look forward to seeing what life holds ahead for us.

Abstract

In much of classical queueing theory, workloads are assumed to be light-tailed, with job sizes being described using exponential or phase type distributions. However, over the past two decades, studies have shown that several real-world workloads exhibit heavy-tailed characteristics. As a result, there has been a strong interest in studying queues with heavy-tailed workloads. So at this stage, there is a large body of literature on queues with light-tailed workloads, and a large body of literature on queues with heavy-tailed workloads. However, heavy-tailed workloads and light-tailed workloads differ considerably in their behavior, and these two types of workloads are rarely studied jointly.

In this thesis, we design scheduling policies for queueing systems, considering both heavy-tailed as well as light-tailed workloads. The motivation for this line of work is twofold. First, since real world workloads can be heavy-tailed or light-tailed, it is desirable to design schedulers that are robust in their performance to distributional assumptions on the workload. Second, there might be scenarios where a heavy-tailed and a light-tailed workload interact in a queueing system. In such cases, it is desirable to design schedulers that guarantee fairness in resource allocation for both workload types.

In this thesis, we study three models involving the design of scheduling disciplines for both heavy-tailed as well as light-tailed workloads. In Chapters 3 and 4, we design schedulers that guarantee robust performance across heavy-tailed and light-tailed workloads. In Chapter 5, we consider a setting in which a heavy-tailed and a light-tailed workload complete for service. In this setting, we design scheduling policies that guarantee good response time tail performance for both workloads, while also maintaining throughput optimality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In queueing systems, it is well known that variability in the incoming workload has a significant impact on the congestion experienced by incoming jobs. As a result, modeling workload variability is a key aspect of performance evaluation in queueing systems.

In classical queueing theory, variability in the workload is typically modeled via the class of phase-type distributions; most commonly, using the exponential distribution [17, 37]. The benefit of this modeling approach is that it enables the analysis of the queue using Markov chains. However, phase-type distributions are limited in the extent of variability they can capture. Specifically, all phase-type distributions are *light-tailed*.

However, over the past two decades, studies have shown that several real-life workloads exhibit extremely high variability, and are better modeled using *heavy-tailed* distributions [4, 19, 29]. Empirical studies have also shown that internet traffic exhibits long range dependence and self similarity [8, 39]. The prevailing explanation for these properties is that they are caused by the heavy-tailed distribution of internet file sizes [18,30]. Consequently, there has been a considerable interest over the past two decades in analyzing queueing models with heavy-tailed workloads.

In general, one expects that certain real-world workloads are best described via light-tailed distributions, while others are best described using heavy-tailed distributions. However, there are fundamental differences in the behavior of heavy-tailed and light-tailed workloads. Specifically, this difference lies in the manner in which collections of heavy-tailed and light-tailed random variables cause rare events.

Collections of heavy-tailed random variables tend to obey the *catastrophe principle*, which states that rare events occur most likely due to the smallest possible number of contributing factors. In queueing systems with heavy-tailed workloads, this principle manifests itself in the manner in which rare events such as large delays or backlogs occur: they occur most likely due to the arrival of one (or a few) large jobs into the system [11, 13, 68].

In contrast, collections of light-tailed random variables tend to obey the *conspiracy principle*, which states that rare events occur most likely due to a combination of a large number of contributing factors. Accordingly, in queueing systems with light-tailed workloads, rare events such as large delays or backlogs are most likely

caused by a conspiracy involving a large number of jobs that are stochastically larger than usual, arriving at a stochastically faster rate than usual [2, 13].

The above dichotomy, which we elaborate on in greater detail in Chapter 2, creates a tension between scheduling for heavy-tailed and light-tailed workloads. In general, scheduling disciplines that work well under heavy-tailed workloads do not perform well under light-tailed workloads, and vice versa. This is particularly true when the performance metric under consideration involves the probability of rare events (see [13, 68]).

So at this stage, the literature understands how to schedule for good performance with light-tailed workloads, as well as how to schedule for good performance with heavy-tailed workloads. However, these contrasting workload types are rarely studied jointly. *In this thesis, we design scheduling policies jointly for heavy-tailed and light-tailed workloads.*

One key motivation for this line of work is robustness. Since real world workloads might be heavy-tailed or light-tailed, it is desirable to design scheduling policies that work well for either class. In other words, it is desirable to design scheduling policies that are robust in their performance to distributional assumptions on the workload. This motivates the work presented in Chapters 3 and 4 in this thesis.

Another motivation for studying heavy-tailed and light-tailed workloads jointly is that there might be settings where workloads of both types interact. For example, consider a communication network in which a highly bursty traffic flow and a less bursty traffic flow co-exist. In such a setting, it is desirable to schedule network resources such that each type of workload receives a fair share, without either type throttling the other. This motivates the work presented in Chapter 5 in this thesis.

In the remainder of this introductory chapter, we outline the technical contributions of this thesis.

Outline of the thesis

Throughout this thesis, we design scheduling policies seeking to 'lighten' the response time tail. In other words, we seek to minimize the probability of very large response times. Analytically, this corresponds to maximizing the asymptotic rate at which the response time tail distribution function decays to zero.

The main motivation for this metric is that in many applications, users are disproportionately sensitive to large response times [28, 40]. Indeed, quality of service guarantees for web applications typically involve guarantees on the response time tail; e.g., 95% of jobs will complete in less that s seconds. The response time tail is a well studied performance metric in the queueing literature, for heavy-tailed workloads (see, for example, [11, 50, 52, 74]) and light-tailed workloads (see, for example, [41, 53, 54, 57]).

Given the theme of this thesis, a technical motivation for considering the response time tail as the performance metric is that it is associated with the probability of rare events. Since heavy-tailed and light-tailed workloads cause large response times via contrasting mechanisms (catastrophe versus conspiracy), the problem of scheduling for good response time tail behavior while jointly considering heavy-tailed as well as

light-tailed workloads becomes particularly challenging [13, 68].

The remainder of this thesis is composed of four chapters. Each chapter is self-contained, and can be read independently.

In Chapter 2, we provide some background on heavy-tailed and light-tailed distributions. The purpose of this chapter is to give relevant definitions, and to give the reader a concrete illustration of the catastrophe principle and the conspiracy principle.

The technical contributions of this thesis are contained in the following three chapters. In each of these chapters, we consider a different problem formulation within the running theme of scheduling for heavy-tailed as well as light-tailed workloads. Chapters 3 and 4 deal with the issue of robust scheduling across heavy-tailed and light-tailed workloads. Chapter 3 addresses this issue at the intra-job level, focusing on robust recovery mechanisms to ensure timely completion of a single job in an unreliable service environment. Chapter 4 addresses the issue of robust scheduling at the inter-job level, focusing on scheduling across waiting jobs in a single server queue. Chapter 5 deals with the problem of scheduling when a heavy-tailed and a light-tailed workload compete for service in a queueing system. We now describe briefly the contributions of Chapters 3, 4, and 5.

Chapter 3: Server Failures and Recovery Mechanisms

In Chapter 3, we study the effect of server failure and recovery mechanisms on job completion times. This work is motivated by the recent discovery that heavy-tailed job completion times can result from recovery mechanisms even when job sizes are light-tailed [5, 33, 34, 63]. A key to this phenomenon is the RESTART feature, where if a job is interrupted before it is completed, it needs to restart from the beginning.

However, the above mentioned line of work does not account for the fact that most recovery mechanisms operating in uncertain service environments implement job fragmentation. For example, when a file is to be transmitted over an unreliable channel, it is fragmented into packets. Similarly, in a computing environment, checkpointing is implemented when the server is prone to failure.

In this chapter, we show that recovery mechanisms that implement reasonable fragmentation strategies cannot produce heavy-tailed completion times from light-tailed job sizes. Specifically, we prove that recovery mechanisms that fragment the job into independent or bounded chunks produce light-tailed file completion times if the job size distribution is light-tailed. In other words, heavy-tailed file completion times can only originate from heavy-tailed file sizes. When the job size is heavy-tailed (with a power-law tail), we show that with independent or bounded fragmentation, the completion time tail distribution function is asymptotically upper bounded by that of the original file size stretched by a constant factor. In other words, the completion time tail is optimal in the degree sense. The above results imply that a large class of reasonable fragmentation policies are tail-robust, i.e., they provide good completion time tail behavior for heavy-tailed and light-tailed job sizes.

Intuitively, the reason we can prove such a strong robustness guarantee for a large class of fragmentation

policies is that we focus on intra-job scheduling, i.e., we restrict attention to the completion time of a single job with size sampled from either a heavy-tailed or light-tailed distribution. Since we do not deal with collections of heavy-tailed and light-tailed random variables, we do not face the conspiracy versus catastrophe contrast with respect to rare events.

Additionally, in Chapter 3, we characterize the fragmentation policy that minimizes the average completion time, and also a simple policy that is blind to the job size, but is asymptotically optimal for the average completion time. Both these policies optimal create bounded fragment sizes, and therefore also provide good completion time tail behavior.

The work presented in Chapter 3 is based on the publications [45, 46].

Chapter 4: Tail-robust scheduling via Limited Processor Sharing

In Chapter 4, we focus on tail-robust inter-job scheduling in a single server queue. In a $GI/GI/1$ queue, there is a well known tension between scheduling for heavy-tailed and light-tailed workloads when seeking to optimize the response time tail. It has been observed that scheduling disciplines that are optimal under light-tailed workloads produce the worst possible response time tail under heavy-tailed workloads, and vice versa. This dichotomy was recently formalized by Wierman & Zwart (see [68]), who proved that no scheduling policy can be optimal for the response time tail for both heavy-tailed and light-tailed workloads. These results imply that there are fundamental limitations in designing schedulers that are robust to distributional assumptions on the workload.

In Chapter 4, *we show how to exploit partial workload information (system load) to design a scheduler that provides robust performance across heavy-tailed and light-tailed workloads.* Specifically, we derive new asymptotics for the tail of the stationary sojourn time under Limited Processor Sharing (LPS) scheduling for both heavy-tailed and light-tailed job size distributions, and show that LPS can be robust to the tail of the job size distribution if the multiprogramming level is chosen carefully as a function of the system load. Our design guarantees strictly better than worst-case response time tail performance for heavy-tailed and light-tailed workloads, and optimal performance across large subsets of heavy-tailed and light-tailed workloads. Moreover, this design is robust to estimation errors in the system load

The work presented in Chapter 4 is based on the publications [47, 48].

Chapter 5: When heavy-tailed and light-tailed workloads compete

While Chapters 3 and 4 address the issue of robust scheduling in queueing systems that might see either a heavy-tailed or a light-tailed workload, Chapter 5 deals with the issue of scheduling in a queueing system that sees both.

In Chapter 5, we consider a setting in which a light-tailed and a heavy-tailed workload compete for service from a single server. Our model captures a wireless uplink/downlink scenario with two nodes communicating

with a single access point. One of the nodes generates heavy-tailed traffic, while the other generates light-tailed traffic. In this setting, our scheduling design goal is that each traffic flow must experience good response time tail behavior. Additionally, we seek scheduling policies that are throughput optimal, i.e., the policy must stabilize the queueing system over the largest possible set of arrival rates.

In the context of wireless networks, the most well studied throughput optimal scheduling policy is the celebrated max-weight policy [66, 67]. Our first result is to show that in our setting, the max-weight policy causes the light-tailed workload to experience heavy-tailed response times. In other works, the max-weight policy severely throttles the light-tailed workload. Intuitively, this is because under max-weight scheduling, a large burst generated by the heavy-tailed workload (the catastrophe) causes the light-tailed workload to be denied service for a long time. One way of avoiding this throttling of the light-tailed workload is of course to schedule it with strict priority over the heavy-tailed workload. However, the main drawback of this scheme is that it is not throughput optimal. This suggests a tradeoff between throughput optimality and good response time tail performance for the light-tailed workload.

The main contribution of this chapter is to show that it is indeed possible to design a throughput optimal scheduling policy that guarantees light-tailed response times for the light-tailed workload, without affecting the response time tail for the heavy-tailed workload. Our design entails a careful choice of inter-queue scheduling policy (from the class of generalized max-weight policies) that gives a relative priority to the light-tailed workload, and intra-queue scheduling policies that complement this priority.

Chapter 2

Heavy-tailed and light-tailed distributions

In this chapter, we give a brief introduction of heavy-tailed and light-tailed distributions. The purpose of this chapter is to give definitions, examples, and illustrations of the catastrophe principle and the conspiracy principle.

The catastrophe principle and the conspiracy principle highlight the contrast in the behavior of heavy-tailed and light-tailed phenomena: they state that collections of heavy-tailed and light-tailed random variables cause rare events in fundamentally different ways. This distinction informs our joint designs of scheduling policies for heavy-tailed and light-tailed workloads in this thesis.

2.1 Heavy-tailed distributions

In this section, we cover heavy-tailed distributions. The material presented here is based on [16, 25, 55, 64]. We first define the class of heavy-tailed distributions and give examples. We then introduce three important subclasses of heavy-tailed distributions: long-tailed distributions, subexponential distributions, and regularly varying distributions. Finally, we give two illustrations of the catastrophe principle.

2.1.1 Definition and examples

Formally, a non-negative random variable X (or its distribution) is said to be *heavy-tailed* if

$$\limsup_{x\to\infty} \frac{P\left(X > x\right)}{e^{-\phi x}} = \infty \quad \text{for all } \phi > 0.$$

Intuitively, the above condition states that the tail distribution function of X is asymptotically 'heavier' than that of any exponential distribution. In other words, the tail distribution function decays to zero 'slower' than that of any exponential distribution. Heavy-tailed distributions can be equivalently characterized in terms of the moment generating function as follows. A non-negative random variable X (or its distribution) is heavy-

tailed if $\mathbb{E}\left[e^{sX}\right] = \infty$ for all $s > 0$. Intuitively, heavy-tailed distributions take extremely large values with a non-negligible probability. We denote the class of (non-negative) heavy-tailed distributions by $\mathcal{K}$.

Heavy-tailed distributions have been empirically observed in a wide range of settings, including incomes of people, sizes of cities, sizes of firms, and node degrees in the web graph [1, 15, 22, 23]. In the context of workloads in queueing systems, heavy tails have been observed in file size distributions and session size distributions on the internet [4, 19, 29].

We now give some common examples of heavy-tailed distributions. In each case, we describe the distribution via its density function f, or its cumulative tail distribution function $\bar{F}$ ($\bar{F}(x)$ is the probability that a random number sampled from the distribution exceeds x).

1. **Pareto distribution:** The Pareto distribution is defined by two parameters: a scale parameter $x_0 > 0$, and a shape parameter $\alpha > 0$. Its tail distribution function is described by the following power-law.

$$\bar{F}(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq x_0 \\ \left(\frac{x}{x_0}\right)^{-\alpha} & \text{for } x > x_0 \end{cases}$$

   The Pareto distribution is named after Italian economist Vilfredo Pareto, who used it to model the distribution of incomes of individuals. It has since been used to model diverse phenomenon such as the frequencies of words in written language, the populations of cities, the sizes of sand particles, and the value of oil fields. In the context of computer systems, this distribution has been found to be a good model for hard disk error rates, internet file sizes, and UNIX process lifetimes.

2. **Weibull distribution:** The (heavy-tailed) Weibull distribution is defined by two parameters: a scale parameter $\lambda > 0$ and a shape parameter $k \in (0, 1)$.[1] Its tail distribution function is given by

$$\bar{F}(x) = e^{-\left(\frac{x}{\lambda}\right)^k}.$$

   The Weibull distribution, named after Swedish physicist Waloddi Weibull, is used extensively in the areas of reliability engineering and failure analysis (see [61] for an application to failure rates in computer systems).

3. **Lognormal distribution:** The lognormal distribution is defined by two parameters: a location parameter $\mu \in \mathbb{R}$, and a shape parameter $\sigma > 0$. Its density function, defined over the positive reals, is given by

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}}.$$

   The lognormal distribution derives its name due to the following property. If the random variable X has a lognormal distribution, then $\log(X)$ is normally distributed with mean μ and variance σ^2. There

[1] The Weibull distribution is also defined for $k \geq 1$, but the distribution is light-tailed over this range.

is a long-standing debate in the scientific community on the question of whether the lognormal or the Pareto distribution should be used to model heavy-tailed real-world phenomenon (see [43] for a survey).

2.1.2 Important subclasses of heavy-tailed distributions

For the purposes of modeling and analysis, the class of heavy-tailed distributions is often too broad to be useful. Accordingly, a typical approach in the literature is to impose additional regularity assumptions on this class for analytical tractability. We now introduce three important subclasses of heavy-tailed distributions that are often used in the literature: long-tailed distributions, subexponential distributions, and regularly varying distributions.

A non-negative random variable X (or its distribution function) is said to be *long-tailed* if

$$\lim_{x\to\infty} \frac{P\left(X > x+y\right)}{P\left(X > x\right)} = 1 \qquad \forall \quad y > 0.$$

The above definition can be interpreted by noting that the quantity in the limit equals $P\left(X > x+y \mid X > x\right)$. Therefore, the above definition states that for any fixed $y > 0$ and large x, if a long tailed random variable X exceeds x, then it also exceeds $x+y$ with high probability. We denote the class of long-tailed distributions by $\mathcal{L}$. It can be shown that $\mathcal{L} \subset \mathcal{K}$, the inclusion being strict [16].

A non-negative random variable X (or its distribution function) is said to be *subexponential* if

$$\lim_{x\to\infty} \frac{P\left(\max\{X_1, X_2\} > x\right)}{P\left(X_1 + X_2 > x\right)} = 1,$$

where X_1 and X_2 are independent random variables distributed as X. This definition may be interpreted by noting that the quantity in the limit equals $P\left(\max\{X_1, X_2\} > x \mid X_1 + X_2 > x\right)$. Therefore, informally, the above definition states that the sum of X_1 and X_2 is large most likely because one of the X_is is large. We denote the class of subexponential distributions by $\mathcal{S}$. The class $\mathcal{S}$ includes most of the common heavy-tailed distributions, including the Pareto, the heavy-tailed Weibull, and the lognormal distributions. It can be shown that $\mathcal{S} \subset \mathcal{L}$, the inclusion being strict.

From the standpoint of modeling heavy-tailed queueing workloads, the most important class of heavy-tailed distributions is the class of regularly varying distributions. Formally, a non-negative random variable (or its distribution function) is said to be regularly varying with index $\theta > 0$ (denoted $X \in \mathcal{RV}(\theta)$) if

$$P\left(X > x\right) = x^{-\theta} L(x),$$

where $L(x)$ is a slowly varying function, i.e., $L(x)$ satisfies $\lim_{x\to\infty} \frac{L(xy)}{L(x)} = 1 \; \forall \; y > 0$. Intuitively, the tail distribution function of a regularly varying distribution decays asymptotically as a power law. Regularly varying distributions are a generalization of the class of Pareto distributions. Note that a smaller value of

index θ implies a heavier tail. Throughout this thesis, we model heavy-tailed workloads using regularly varying distributions. We denote the class of regularly varying distributions by $\mathcal{RV}$. It can be shown that the class of regularly varying distributions is strictly contained in the class of subexponential distributions; therefore, $\mathcal{RV} \subset \mathcal{S} \subset \mathcal{L} \subset \mathcal{K}$, all inclusions being strict.

2.1.3 The catastrophe principle

An important rule of thumb regarding heavy-tailed distributions is the so called 'catastrophe principle', which concerns the manner in which collections of heavy-tailed random variables produce rare events. *The catastrophe principle states that rare events occur most likely due to the smallest possible number of contributing factors.*

We now give two illustrations of the catastrophe principle. Let $\{X_i\}_{i\geq 1}$ denote a sequence of non-negative, independent, and identically distributed random variables. Our first illustration of the catastrophe principle is the following property of subexponential distributions.

Theorem 1. *If $X_1 \in \mathcal{S}$, then for any $n \geq 2$,*

$$\lim_{x\to\infty} \frac{P\left(\{\max_{1\leq i\leq n} X_i\} > x\right)}{P\left(\sum_{i=1}^{n} X_i > x\right)} = 1. \tag{2.1}$$

Note that the quantity in the limit above equals $P\left(\{\max_{1\leq i\leq n} X_i\} > x \mid \sum_{i=1}^{n} X_i > x\right)$. The above property therefore implies if that the sum of the n independent and identically distributed subexponential random variables is large, then it is most likely because of *one* large value (the catastrophe). We point out here that the statement of (2.1) for $n = 2$ is simply the definition of the class of subexponential distributions. That this implies that (2.1) holds for all $n > 2$ was first proved in [16].

Another illustration of the catastrophe principle is the following property of regularly varying distributions. We use the notation $f(n) \sim g(n)$ to mean that $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$.

Theorem 2. *Suppose that $X_i \in \mathcal{RV}(\theta)$, with $\theta > 1$. Let $\mu := \mathbb{E}[X_i]$. Then for $y > 0$,*

$$P\left(\sum_{i=1}^{n} X_1 > (\mu + y)n\right) \sim P\left(\max_{i=1,\cdots,n} X_i > yn\right).$$

The above theorem concerns the probability of the rare event $\{\sum_{i=1}^{n} X_i > (\mu + y)n\}$, i.e., the event that the running sum of the first n random variables exhibits a 'large deviation' of $O(n)$ from its expected value. Theorem 2 implies that for large n, this event occurs most likely with a single random variable accounting for the entire 'large deviation'. Informally, the sum is large most likely due to a single large value (the catastrophe). The proof of Theorem 2 can be found in [58].

In the context of queues fed by a heavy-tailed workload, the catastrophe principle manifests itself in the manner in which rare events, such as large delays or backlogs occur: they occur most likely because of the

arrival of one (or a few) large jobs into the queue.[2] It is therefore not surprising that the catastrophe principle informs the design of scheduling policies that perform well in queueing systems with heavy-tailed workloads.

This concludes our discussion on heavy-tailed distributions in this chapter. We now turn to light-tailed distributions.

2.2 Light-tailed distributions

In this section, we give a brief introduction to light-tailed distributions. We first define the class of light-tailed distributions and give examples. We then give an illustration of the conspiracy principle.

2.2.1 Definition

Formally, a non-negative random variable X is said to be light-tailed if it is not heavy-tailed, i.e., if there exists $\phi > 0$ such that

$$P\left(X > x\right) \leq e^{-\phi x} \text{ for large enough } x.$$

The above condition states that the tail distribution function of X is asymptotically bounded above by that of an exponential distribution. In other words, the tail distribution function decays to zero exponentially or faster. Equivalently, a non-negative random variable X is light-tailed if there exists $s > 0$ such that $\mathbb{E}\left[e^{sX}\right] < \infty$.

The class of light-tailed distributions includes the important class of phase-type distributions. A phase-type distribution is defined as the distribution of the time to absorption of an absorbing Markov chain (see [38] for a detailed characterization of phase-type distributions). An important property of the class of phase-type distributions is that it is dense in the space of all non-negative, continuous distributions [49].[3] This property makes phase-type distributions useful for modeling a wide range of stochastic variability. Examples of phase-type distributions include the exponential distribution, the Erlang distribution, and the hyper-exponential distribution [38].

2.2.2 The conspiracy principle

In stark contrast with the catastrophe principle for heavy-tailed distributions, light-tailed distributions tend to obey a 'conspiracy principle'. *The conspiracy principle states that rare events occur most likely because of a combination of a large number of contributing factors.*

To illustrate the conspiracy principle, we now state and prove the analogue of Theorem 2 for phase-type distributions. To state the result, we need the following notation. As before, let $\{X_i\}_{i\geq 1}$ denote a sequence of non-negative, independent, and identically distributed random variables. Corresponding to the distribution

[2] We will see examples of this in Chapters 4 and 5 in this thesis.

[3] It is important to note that this property does not imply that phase-type distributions can approximate heavy-tailed distributions well. Specifically, phase-type distributions cannot capture well the asymptotic decay of the tail distribution function of a heavy-tailed distribution, since all phase-type distributions have an (asymptotically) exponentially decaying tail.

of X_i, let $\Lambda(\cdot)$ denote its cumulant generating function, i.e., $\Lambda(\theta) := \log \mathbb{E}\left[e^{\theta X_i}\right]$, and let $\Lambda^*(\cdot)$ denote the convex conjugate of $\Lambda(\cdot)$, i.e., $\Lambda^*(z) := \sup_{\theta \geq 0}[\theta z - \Lambda(\theta)]$. We are now ready to state the analogue of Theorem 2 for phase-type distributions.

Theorem 3. *Suppose that X_i is phase-type, with $\mu := \mathbb{E}[X_i] > 0$. Then for $y > 0$,*

$$\lim_{n\to\infty} \frac{1}{n} \log P\left(\sum_{i=1}^{n} X_i > (\mu + y)n\right) = -\Lambda^*(\mu + y).$$

The above theorem is a special case of Cramèr's theorem [20]. Similar to Theorem 2, this theorem concerns the probability of the rare event $\{\sum_{i=1}^{n} X_i > (\mu + y)n\}$, i.e., the event that the running sum of the first n random variables exhibits a 'large deviation' of $O(n)$ from its expected value. The proof, which we present below, highlights *how* this rare event occurs: it occurs via a conspiracy involving all n random variables. The proof presented here is almost identical to that in [24].

Proof of Theorem 3. Before we begin the main proof, we state some useful facts regarding the cumulant generating function and its convex conjugate. Define $\gamma := \lim_{x\to\infty} -\frac{\log P(X_i > x)}{x}$. Since X_i is phase-type, it can be proved that the limit in the definition of γ exists, and $\gamma \in (0, \infty)$. Moreover, $\Lambda(\cdot)$ is strictly convex over $(-\infty, \gamma)$, $\Lambda(\theta) = \infty$ for $\theta \geq \gamma$, and $\lim_{\theta\uparrow\gamma} \Lambda(\theta) = \infty$. As a result of these properties, the supremum in the definition of $\Lambda^*(z)$ is achieved at a unique $\theta^*(z) < \gamma$, such that the first order condition

$$\Lambda'(\theta^*(z)) = \frac{\mathbb{E}\left[X e^{\theta^*(z) X_i}\right]}{\mathbb{E}\left[e^{\theta^*(z) X_i}\right]} = z \tag{2.2}$$

holds. Finally, since $\Lambda'(0) = \mu$, $\theta^*(z) > 0$ for $z > \mu$.

We are now ready to prove the statement of Theorem 3. Let $S_n := \sum_{i=1}^{n} X_i$, $a := \mu + y$. We prove Theorem 3 by establishing asymptotically matching upper and lower bounds on $P(S_n > an)$.

The upper bound follows easily from the Chernoff bound. Indeed, the Chernoff bound implies that

$$\begin{aligned} P(S_n > an) &\leq e^{-n\Lambda^*(a)} \\ \Rightarrow \limsup_{n\to\infty} \frac{1}{n} \log P(S_n > an) &\leq -\Lambda^*(a). \end{aligned}$$

We now turn to the lower bound. Let F denote the distribution function of X_i. We first bound $P(S_n > an)$ from below as follows.

$$\begin{aligned} P(S_n > an) &\geq P\left(an < S_n < an + \sqrt{n}\right) \\ &= \int_{A_n} dF(x_1) \cdots dF(x_n), \end{aligned}$$

where $A_n := \{(x_1, \cdots, x_n) \in \mathbb{R}^n \mid an < \sum_{i=1}^{n} x_i < an + \sqrt{n}\}$.

We define the 'exponentially tilted' distribution $\tilde{F}$ as follows.

$$d\tilde{F}(x) = \frac{e^{\theta^*(a)x}}{\mathbb{E}\left[e^{\theta^*(a)X_i}\right]} dF(x).$$

Since $\theta^*(a) > 0$, the tilted distribution $\tilde{F}$ stochastically dominates F. Moreover, it follows from (2.2) that mean of the distribution $\tilde{F}$ equals a. Let $\{\tilde{X}_i\}_{i\geq 1}$ denote an independent and identically distributed sequence of random variables distributed as $\tilde{F}$, $\tilde{S}_n := \sum_{i=1}^n \tilde{X}_i$. We prove our desired lower bound by relating the probability of the event $\{an < S_n < an + \sqrt{n}\}$ to the probability of the event $\{an < \tilde{S}_n < an + \sqrt{n}\}$. Note that since the tilted distribution $\tilde{F}$ stochastically dominates the original distribution F, with mean $a > \mu$, the latter event is much more likely. Now,

$$\begin{aligned} P\left(an < S_n < an + \sqrt{n}\right) &= \int_{A_n} dF(x_1) \cdots dF(x_n) \\ &= \mathbb{E}\left[e^{\theta^*(a)X}\right]^n \int_{A_n} e^{-\theta^*(a)(x_1+\cdots+x_n)} d\tilde{F}(x_1) \cdots d\tilde{F}(x_n) \\ &\geq \mathbb{E}\left[e^{\theta^*(a)X}\right]^n \int_{A_n} e^{-\theta^*(a)(an+\sqrt{n})} d\tilde{F}(x_1) \cdots d\tilde{F}(x_n). \end{aligned}$$

Noting now that $\mathbb{E}\left[e^{\theta^*(a)X}\right] = e^{\Lambda(\theta^*(a))}$, we have

$$\begin{aligned} P\left(an < S_n < an + \sqrt{n}\right) &\geq e^{-n[a\theta^*(a) - \Lambda(\theta^*(a)]}\, e^{-\theta^*(a)\sqrt{n}}\, P\left(an < \tilde{S}_n < an + \sqrt{n}\right) \\ &= e^{-n\Lambda^*(a)}\, e^{-\theta^*(a)\sqrt{n}}\, P\left(an < \tilde{S}_n < an + \sqrt{n}\right). \end{aligned} \tag{2.3}$$

The central limit theorem implies that

$$\lim_{n\to\infty} P\left(an < \tilde{S}_n < an + \sqrt{n}\right) = P\left(0 < U < 1\right),$$

where U is normally distributed with mean zero, and the same variance as $\tilde{X}_i$. It therefore follows from (2.3) that

$$\liminf_{n\to\infty} \frac{1}{n} \log P\left(an < S_n < an + \sqrt{n}\right) \geq -\Lambda^*(a),$$

which gives us the matching lower bound, since $P\left(S_n > na\right) \geq P\left(an < S_n < an + \sqrt{n}\right)$. This completes the proof. □

The above proof reveals that a typical manner in which the rate event $\{\sum_{i=1}^n X_i > (\mu + y)n\}$ occurs is by conspiracy: all the n random variables are sampled from the titled (and stochastically larger) distribution $\tilde{F}$ instead of F.

In the context of queues fed by a light-tailed workload, the conspiracy principle manifests itself in the manner in which rare events, such as large delays or backlogs occur: they occur most likely because of a conspiracy involving a large number of arriving jobs. For example, in a single server queue fed by a light-

tailed workload, it is known that a large backlog is most likely caused due to a sustained period over which inter-arrival times are stochastically smaller than usual, and job sizes are stochastically larger than usual [2]. The conspiracy principle therefore informs an understanding of which scheduling policies perform well in queueing systems with light-tailed workloads.

Chapter 3

Server Failures and Recovery Mechanisms: The Impact on the Processing Time Tail

3.1 Motivation and summary

It has been recently discovered that heavy-tailed job completion times can result from recovery mechanisms even when the job size is light-tailed [5, 33, 34, 63]. Indeed, the completion time can be heavy-tailed even when the job size has a tail that decays exponentially or superexponentially. A key to this phenomenon is the RESTART feature, where if a job is interrupted in the middle of its processing, the entire job needs to restart from the beginning, i.e., the work that is partially completed is lost. This can model, e.g., a packet that is corrupted by bit errors and needs to be retransmitted. This effect has been shown to be robust to several schemes aimed at alleviating it. The fragmentation scheme of [35], which uses the sizes of the previous $k + m$ server availability periods, lightens the completion time tail by adding k additional moments, but the resulting tail is still heavy. Multipath is explored in [65] to mitigate power-law completion time. It is shown there that redundant routing, where the entire file is sent along multiple paths and the completion time is the time when the first copy arrives at the destination correctly, preserves the power law. Split routing, where disjoint fragments of the file are sent along multiple paths and the completion time is the time when the last fragment arrives, also retains a power-law completion time, though the tail can be lightened with a larger index.

In this chapter, we show that the heavy-tailed completion times can actually be quite fragile and are removed by a large class of fragmentation schemes. In particular, we consider a model for file transfer over an unreliable channel and propose fragmentation policies that guarantee light-tailed completion times for light-tailed file sizes.[1] In the models of [5, 33, 34, 63], heavy-tailed completion time seems to arise from repeated comparison of a sequence of independent, identically distributed (i.i.d.) random variables

[1] In this chapter, we use the words file and job interchangeably, and the words server and channel interchangeably.

(availability periods) with the *same* random variable (original job size) that has an *infinite support*. This motivates fragmentation policies that avoid this character.

Specifically, we consider policies that partition files into fragments with independent, or bounded sizes; note that packet sizes are typically bounded automatically by network hardware. We show that these policies produce a light-tailed completion time as long as the original file size is light-tailed, i.e., in this case, a heavy-tailed file completion time can only originate from a heavy-tailed file size (Section 3.3). If the file size is heavy-tailed, then the file completion time is necessarily heavy-tailed. In this case, we show that if the file size distribution is regularly varying, then under independent or bounded fragmentation, the completion time tail distribution function is asymptotically upper bounded by that of the original file size stretched by a constant factor. This means that in the degree sense, the completion time distribution is only as heavy-tailed as the job size distribution. Our results therefore imply that a broad class of fragmentation policies (that produce independent/bounded fragments) are *tail-robust*, i.e., they guarantee good completion time tail behavior for heavy-tailed and light-tailed job sizes.

Since a broad class of fragmentation policies guarantees good completion time tail performance, it is then natural to seek to minimize the *average* job completion time. We prove that if the failure distribution has a non-decreasing failure rate, it is optimal to divide the file into equal sized fragments, whose size depends on the file size (Section 3.4.1). We also present a simple blind fragmentation policy where the fragment size is constant and independent of the file size and prove that its expected file completion time is asymptotically optimal (Section 3.4.2). Importantly, the optimal policy as well as the suboptimal blind policy create bounded fragments, and therefore produce desirable completion time tail behavior, as described above (Section 3.4.3).

3.2 Model and preliminaries

3.2.1 Model

Consider a file with a possibly random size $L > 0$. The file is fragmented into packets which are then sent over an unreliable channel with unit transmission rate. A packet contains a fragment of the file and a fixed-sized overhead (header, trailer). The larger the packet size, the more likely the transmission is to fail. This will be the case, e.g., if the channel randomly introduces independent bit errors so a packet with more bits has a higher probability of being corrupted and needing a retransmission; see [62, p. 132] for such a failure model for satellite and terrestrial communications. More generally, for the nth transmission attempt, let $x_n + \phi$ be the packet size, where x_n is the size of the file fragment and ϕ is the constant overhead. All sizes are measured in terms of the transmission time over the channel with unit rate. Let $(A_n, n = 1, 2, \dots)$ be i.i.d. non-negative random variables with common distribution F and independent of L, with $P(A_1 > \phi) > 0$. The nth transmission attempt will be successful if and only if $A_n \geq x_n + \phi$.

To formulate the problem precisely, we abuse notation and use $x = (x_n, n = 1, 2, \dots)$ to denote both

the control (fragmentation) policy and the fragment sizes under the policy, depending on the context. Let the state $l_n := l_n^x$ be the remaining file size just after the start of the nth transmission under control policy x. Then the state l_n evolves according to,

$$l_{n+1} = l_n - x_n \, \mathbf{1}(A_n \geq x_n + \phi), \quad n = 1, 2, \ldots \tag{3.1}$$

$$l_1 = L \tag{3.2}$$

where $\mathbf{1}(z) = 1$ if z is true and 0 otherwise. We implicitly restrict ourselves to admissible policies x under which $0 \leq x_n \leq l_n$ for all n. We emphasize that the state sequence $(l_n, n \geq 1)$ depends on the control policy $x = (x_n, n \geq 1)$ though this is not explicit in the notation. The time between the nth and the $n+1$st submission is the cost at the nth stage and is given by:

$$\tau_n := (x_n + \phi) \, \mathbf{1}(l_n > 0). \tag{3.3}$$

Clearly, the transmission time sequence $(\tau_n, n \geq 1)$ also depends on the control x. Let $T(L)$ be the file completion time under control x as a function of the initial file size L;

$$T(L) := T^x(L) := \sum_{n \geq 1} \tau_n. \tag{3.4}$$

In summary, our file fragmentation model is specified by (3.1)–(3.4) with the i.i.d. random sequence $(A_n, n \geq 1)$. In subsequent sections, we will study the impact of the choice of the fragment sizes $(x_n, n = 1, 2, \ldots)$ on the file completion time.

Our model is an adaptation of the model in [5, 33, 34, 63] where a server alternates between availability periods and unavailability periods. There, the server availability periods have durations $(A_n, n \geq 1)$ that are i.i.d. random variables. The unavailability periods have durations $(U_n, n \geq 1)$ that are i.i.d. and independent of $(A_n, n \geq 1)$. Without fragmentation, the entire file is submitted at the beginning of each availability period until it completes successfully, $x_n = L$ till the transmission succeeds. Our model here has $U_n = 0$; furthermore, the one-stage cost is $x_n + \phi$ in our case, but A_n (before successful transmission) in theirs. This models the case where the sender is informed of the failure only after the entire packet has been sent. These differences do not qualitatively change our conclusions (see a parallel set of results in [46] for a *job* fragmentation model that is closer to the model in [5,33,34,63] and the models in the checkpointing literature.

3.2.2 Notation and preliminaries

Throughout this chapter, $\overline{\lim}$ denotes the limit superior, $\underline{\lim}$ the limit inferior, and $\mathbb{E}[\cdot]$ the expectation. For any functions $\gamma(t)$ and $\lambda(t)$,

1. $\gamma(t) \sim \lambda(t)$ means $\lim_{t \to \infty} \gamma(t)/\lambda(t) = 1$,

2. $\gamma(t) \lesssim \lambda(t)$ means $\overline{\lim}_{t\to\infty} \gamma(t)/\lambda(t) \leq 1$,

3. $\gamma(t) = o(\lambda(t))$ means $\lim_{t\to\infty} \gamma(t)/\lambda(t) = 0$.

Consider non-negative random variables X, Y. We will use the notation $X \leq_{\text{a.s.}} Y$ to mean $X \leq Y$ almost surely. The notation $X \leq_{\text{st}} Y$ means X is stochastically dominated by Y, i.e., $P(X > t) \leq P(Y > t)$ for all $t \geq 0$. It is easy to see that $X \leq_{\text{a.s.}} Y$ implies $X \leq_{\text{st}} Y$.

Lemma 1. *If random variables A, B, C satisfy $A \leq_{\text{st}} B \leq_{\text{st}} C$, and $P(A > x) \sim P(C > x)$, then*

$$P(A > x) \sim P(B > x) \sim P(C > x).$$

The elementary proof is omitted. Let $G(x) = P(X \leq x)$ denote the distribution function (d.f.) of non-negative random variable X and $\overline{G}(x) := 1 - G(x)$ denote its tail distribution function.

Definition 1. *The d.f. G (or the random variable X) is said to be* heavy-tailed (HT) *if*

$$\overline{\lim_{x\to\infty}} e^{\theta x}\overline{G}(x) = \infty$$

for all $\theta > 0$. The d.f. G (or the random variable X) is said to be light-tailed (LT) *if it is not heavy-tailed, i.e., if there exists a $\theta > 0$ such that*

$$\lim_{x\to\infty} e^{\theta x}\overline{G}(x) = 0.$$

Intuitively, a distribution is heavy-tailed if its tail d.f. is (asymptotically) heavier than that of any exponential distribution. Conversely, a distribution is light-tailed if its tail d.f. is (asymptotically) dominated by that of some exponential distribution. The following lemma describes some closure properties of the class of light-tailed distributions we will use in this chapter.

Lemma 2. *[Closure properties of light-tailed distributions]*

1. *Let X, Y be non-negative random variables satisfying $X \leq_{\text{st}} Y$. If Y is light-tailed, then X is light-tailed.*

2. *Let X, Y be non-negative random variables. If X, Y are light-tailed, then $X + Y$ is light-tailed.*

3. *Let $(X_i, i \geq 1)$ be a sequence of non-negative i.i.d. light-tailed random variables, and N be an integer random variable. If N is light-tailed, then the random sum $\sum_{i=1}^{N} X_i$ is light-tailed.*

4. *Let L be a non-negative random variable and $\{X_i\}_{i\geq 1}$ a sequence of non-negative i.i.d. random variables independent of L and satisfying $P(X_i > 0) > 0$. If L is light-tailed, so is $\inf\{n | \sum_{i=1}^{n} X_i \geq L\}$.*

We give the proof of this lemma in Appendix 3.A.

An important class of heavy-tailed distributions is the class of regularly varying distributions (see [10], Chapter 2 of [55]).

Definition 2. *A d.f. G is regularly varying with index/degree $\alpha > 0$ (denoted $G \in \mathcal{RV}(\alpha)$) if*

$$\overline{G}(x) = x^{-\alpha}\chi(x)$$

where $\chi(x)$ is a slowly varying function, i.e., $\chi(x)$ satisfies

$$\lim_{x\to\infty} \frac{\chi(xy)}{\chi(x)} = 1 \quad \forall y > 0.$$

We will abuse notation and use $L \in \mathcal{RV}(\alpha)$ to mean the d.f. G_L of a random variable L is in $\mathcal{RV}(\alpha)$. Regularly varying distributions are a generalization of the class of Pareto distributions, also referred to as power-law distributions or Zipf distributions. The closer α is to 0, the 'heavier' the tail d.f. is.

Lemma 3. *Consider non-negative random variables X, Y. If $X \in \mathcal{RV}(\alpha)$ and $P(X > t) \sim P(Y > t)$, then $Y \in \mathcal{RV}(\alpha)$.*

The proof follows easily from the definition.

Lemma 4. *If $X \in \mathcal{RV}(\alpha)$, then $P(X > t) \sim P(X > t + c)$ for all $c \in \mathbb{R}$.*

This lemma is a consequence of the fact that regularly varying distributions are a sub-class of the class of long-tailed distributions; see [64].

Lemma 5. *If $\chi(x)$ is slowly varying, then*

$$\lim_{x\to\infty} x^{\beta}\chi(x) = \begin{cases} \infty & \text{if } \beta > 0 \\ 0 & \text{if } \beta < 0 \end{cases}.$$

See Proposition 2.6 in [55] for a proof.

3.3 Completion time tail asymptotics

In this section, we study the tail behavior of the completion time under a broad class of fragmentation policies. To motivate our results, we first state the following theorem, which considers the case of no fragmentation.

Theorem 4 ([5, 33, 34, 63])**.** *Without fragmentation, i.e., $x_n = L$ until the whole file is transmitted successfully, $T(L)$ is heavy-tailed as long as L has infinite support.*

The proof follows from Lemma 1 in [33]. Theorem 4 implies that without fragmentation, the completion time $T(L)$ can be heavy-tailed even for light-tailed file sizes, e.g., file size distributions with an exponential

or even superexponential tail. Our results in this section (Theorems 5–7) imply that under a broad class of fragmentation policies, the completion time $T(L)$ is light-tailed provided L is light-tailed. Thus, with these policies, *heavy-tailed completion times can only arise from heavy-tailed file sizes.* Moreover, we show if L is heavy-tailed (specifically, regularly varying), then the tail d.f. of $T(L)$ is bounded above by a scaled version of the tail d.f. of L. This means that in the degree sense, the completion time is only as heavy-tailed as the file size.

3.3.1 Results

We now define the three classes of fragmentation policies studied in this section.

- **Independent fragmentation:** $x_n = \min\{X_n, l_n\}, n \geq 1$, where $(X_n, n \geq 1)$ is a sequence of i.i.d., strictly positive, light-tailed random variables independent of L and $(A_n, n \geq 1)$ such that $P(A_1 \geq X_1 + \phi) > 0$.
- **Bounded fragmentation:** x_n satisfies $\min\{b, l_n\} \leq x_n \leq \min\{c, l_n\}$, $n \geq 1$, for some constants $0 < b \leq c$ such that $P(A_1 \geq c + \phi) > 0$.
- **Constant fragmentation:** $x_n = \min\{b, l_n\}$ for some constant $b > 0$ satisfying $P(A_1 \geq b + \phi) > 0$. This is a special case of independent fragmentation and of bounded fragmentation.

We now state our results for each of these classes.

Theorem 5 (Independent fragmentation)**.** *Under the independent fragmentation policy*

1. *If L is light-tailed, then $T(L)$ is light-tailed.*
2. *If $L \in \mathcal{RV}(\alpha)$, then $P(T(L) > t) \lesssim P\left(L > \frac{t}{\sigma}\right)$ where*

$$\sigma = \frac{\mathbb{E}[X_1] + \phi}{P(X_1 + \phi \leq A_1)\,\mathbb{E}[X_1 | X_1 + \phi \leq A_1]}.$$

The next result says that any policy that does not choose arbitrarily large or arbitrarily small fragment sizes produces light-tailed completion time provided L is light-tailed.

Theorem 6 (Bounded fragmentation)**.** *Under the bounded fragmentation policy*

1. *If L is light-tailed, then $T(L)$ is light-tailed.*
2. *If $L \in \mathcal{RV}(\alpha)$, then $P(T(L) > t) \lesssim P\left(L > \frac{t}{\sigma}\right)$ where*

$$\sigma = \frac{c + \phi}{bP(A_1 \geq c + \phi)}.$$

Intuitively, if packet size is too small, the overhead can dominate the transmission, reducing efficiency. If the packet is too large, the failure probability can be too high. Hence it is reasonable to choose packet sizes that are neither too small nor too large. Theorem 6 then guarantees that any reasonable fragmentation policy 'lightens' the completion time tail.

Since constant fragmentation is a special case of independent and bounded fragmentation, Theorems 5 and 6 imply that under constant fragmentation, $T(L)$ is light-tailed if L is light-tailed. When L is regularly varying, we have a sharper characterization of the asymptotics: $T(L)$ is regularly varying with the same degree.

Theorem 7 (Constant fragmentation)**.** *Under the constant fragmentation policy*

1. *If L is light-tailed, then $T(L)$ is light-tailed.*
2. *If $L \in \mathcal{RV}(\alpha)$, then $P(T(L) > t) \sim P\left(L > \frac{t}{g(b)}\right)$ where*

$$g(x) = \frac{x + \phi}{xP(A_1 \geq x + \phi)}.$$

Theorem 7 motivates choosing the constant fragment size $a := \arg\min_{x>0} g(x)$. Within the class of constant fragmentation policies, this choice produces in some sense the lightest possible completion time tail asymptotics. We will prove in Section 3.4 that this policy also almost minimizes the expected completion time; see Theorem 11.

3.3.2 Proofs of Theorems 5–7

Proofs of Theorems 5–7 rely on Lemma 6, which we state and prove first.

Lemma 6. *Let L be a random variable, and $(X_n, n \geq 1)$ be a sequence of i.i.d. strictly positive light-tailed random variables independent of L and $(A_n, n \geq 1)$ such that $P(A_1 > X_1 + \phi) > 0$. Let*

$$\begin{aligned} Y_n &:= X_n \mathbf{1}(X_n + \phi \leq A_n), \\ M &:= \inf\left\{m : \sum_{n=1}^{m} Y_n \geq L\right\}, \qquad (3.5) \\ \tilde{T}(L) &:= \sum_{n=1}^{M} (X_n + \phi). \qquad (3.6) \end{aligned}$$

1. *If L is light-tailed, then $\tilde{T}(L)$ is light-tailed.*
2. *If $L \in \mathcal{RV}(\alpha)$, then $P\left(\tilde{T}(L) > t\right) \sim P(L > t/\sigma)$ where*

$$\sigma = \frac{\mathbb{E}[X_1] + \phi}{P(X_1 + \phi \leq A_1)\,\mathbb{E}[X_1 | X_1 + \phi \leq A_1]}.$$

The proof of this lemma for the case of regularly varying L is based on the following theorem, proved in [26].

Theorem 8 ([26]). *Let $L \in \mathcal{RV}(\alpha)$. For $t \geq 0$, let $R(t)$ be a non-negative, almost surely non-decreasing stochastic process independent of L satisfying the following conditions:*

1. *For some $\gamma \in (0,1), \lim_{t\to\infty} R(t)/t = \gamma$ almost surely*
2. *For some positive finite constant K, $P(R(t)/t < K) = o(P(L > t))$.*

Then $P(L > R(t)) \sim P(L > \gamma t)$.

Proof of Lemma 6. We consider the cases of light-tailed and regularly varying L separately.

Case 1: L is light-tailed. Under the assumptions of the lemma, $(Y_n, n \geq 1)$ is an i.i.d. sequence satisfying $P(Y_1 > 0) > 0$. Invoking Lemma 2(4), we conclude from (3.5) that M is light-tailed. It follows that $\tilde{T}(L)$ is light-tailed from (3.6) invoking Lemma 2(3).

Case 2: $L \in \mathcal{RV}(\alpha)$. Let $N(t) := \sup\{n : \sum_{i=1}^{n}(X_i + \phi) \leq t\}$, $R(t) := \sum_{i=1}^{N(t)} Y_i$. Note that $P\left(\tilde{T}(L) > t\right) = P(R(t) < L)$. To complete the proof, it suffices to show that the process $R(t)$ satisfies conditions (1) and (2) of Theorem 8 with $\gamma = 1/\sigma$.

Condition (1) of Theorem 8 is verified using the renewal reward theorem.

$$\lim_{t\to\infty} \frac{R(t)}{t} = \frac{\mathbb{E}[Y_1]}{\mathbb{E}[X_1 + \phi]} = \frac{1}{\sigma}$$

almost surely. Note that $\sigma > 1$ since $\phi > 0$. To verify Condition (2), pick $K \in (0, 1/\sigma)$. Since $K < 1/\sigma$, we can find $\eta, \nu > 0$ such that $K = \eta\nu$, $\eta < \mathbb{E}[Y_1]$ and $\nu < 1/\mathbb{E}[X_1 + \phi]$. Then

$$\begin{aligned}
P(R(t) < Kt) &= P\left(\sum_{i=1}^{N(t)} Y_i < Kt\right) \\
&= P(N(t) < t\nu) - P\left(\sum_{i=1}^{N(t)} Y_i \geq Kt \wedge N(t) < t\nu\right) + P\left(\sum_{i=1}^{N(t)} Y_i < Kt \wedge N(t) \geq t\nu\right) \\
&\leq P(N(t) < t\nu) + P\left(\sum_{i=1}^{N(t)} Y_i < Kt \wedge N(t) \geq t\nu\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
P(R(t) < Kt) &\leq P\left(\sum_{i=1}^{\lfloor t\nu \rfloor}(X_i + \phi) \geq t\right) + P\left(\sum_{i=1}^{\lceil t\nu \rceil} Y_i < Kt\right) \\
&\leq P\left(\sum_{i=1}^{\lfloor t\nu \rfloor}(X_i + \phi) \geq \frac{\lfloor t\nu \rfloor}{\nu}\right) + P\left(\sum_{i=1}^{\lceil t\nu \rceil} Y_i < \eta \lceil t\nu \rceil\right).
\end{aligned}$$

Noting that $1/\nu > \mathbb{E}[X_1 + \phi]$ and $\eta < \mathbb{E}[Y_1]$, and that X_1, Y_1 are light-tailed, we can use the Chernoff bound to argue that there exist positive constants C, λ such that for large enough t,

$$P(R(t) < Kt) \leq Ce^{-\lambda t}.$$

Since $P(L > t) = t^{-\alpha}\chi(t)$ for slowly varying χ, this implies

$$\overline{\lim_{t\to\infty}} \frac{P(R(t) < Kt)}{P(L > t)} \leq \overline{\lim_{t\to\infty}} \frac{Ce^{-\lambda t}}{t^{-\alpha}\chi(t)} = \overline{\lim_{t\to\infty}} \frac{Ct^{\alpha+1}e^{-\lambda t}}{t\chi(t)} = 0.$$

The last step above uses Lemma 5. It follows that $P(R(t) < Kt) = o(P(L > t))$. This completes the proof. □

We are now ready to prove Theorems 5–7.

Proof of Theorem 5. Consider the completion time $\tilde{T}(L)$ under the policy $\tilde{x}_n := X_n$. Clearly $T(L) \leq_{\text{a.s.}} \tilde{T}(L)$.

If L is light-tailed, then from Lemma 6, we conclude that $\tilde{T}(L)$ is light-tailed, which implies $T(L)$ is light-tailed (Lemma 2(1)).

If $L \in \mathcal{RV}(\alpha)$, then from Lemma 6, we conclude that $P\left(\tilde{T}(L) > t\right) \sim P\left(L > \frac{t}{\sigma}\right)$. Since $T(L) \leq_{\text{a.s.}} \tilde{T}(L)$, it follows that $P(T(L) > t) \lesssim P\left(L > \frac{t}{\sigma}\right)$. □

Proof of Theorem 6. Define $\tilde{L} := cL/b$. With file size $\tilde{L}$, consider the policy $\tilde{x}_n = \min\{c, \tilde{l}_n\}$, $n \geq 1$, where $\tilde{l}_1 = \tilde{L}$, $\tilde{l}_n$ denotes the remaining file size just after the nth submission. Note that this policy satisfies the conditions of Theorem 5. Denote the completion time under this scheme by $T^c(\tilde{L})$.

We will now argue that $T(L) \leq_{\text{a.s.}} T^c(\tilde{L})$. Consider a sample path, determined by the realization of L, $(A_n, n \geq 1)$ and the fragment sizes $(x_n, n \geq 1)$. Noting that for any n, if fragment submission $\tilde{x}_n$ succeeds, then submission x_n succeeds, it can be seen that $l_n \leq b\tilde{l}_n/c$ for all $n \geq 1$. This implies $T(L) \leq T^c(\tilde{L})$.

If L is light-tailed, so is $\tilde{L}$. Theorem 5 then implies that $T^c(\tilde{L})$ is light-tailed, which implies $T(L)$ is light-tailed (Lemma 2(1)).

If $L \in \mathcal{RV}(\alpha)$, it is easy to see that $\tilde{L} \in \mathcal{RV}(\alpha)$. Theorem 5 implies that

$$\begin{aligned} P\left(T^c(\tilde{L}) > t\right) &\lesssim P\left(\tilde{L} > \frac{tcP(A_1 \geq c + \phi)}{c + \phi}\right) \\ &= P\left(L > \frac{tbP(A_1 \geq c + \phi)}{c + \phi}\right) \\ &= P\left(L > \frac{t}{\sigma}\right). \end{aligned}$$

Since $T(L) \leq_{\text{a.s.}} T^c(\tilde{L})$, we conclude that $P(T(L) > t) \lesssim P\left(L > \frac{t}{\sigma}\right)$. □

Proof of Theorem 7. Since constant fragmentation is a special case of independent and bounded fragmentation, the proof for the case of light-tailed L follows directly from Theorems 5 or 6.

Assume then that $L \in \mathcal{RV}(\alpha)$. We will invoke Lemma 6 with $X_n := b$, $n \geq 1$. Define

$$\hat{L} := b \left\lfloor \frac{L}{b} \right\rfloor, \quad \tilde{L} := b \left\lceil \frac{L}{b} \right\rceil.$$

It is easy to see that

$$\tilde{T}(\hat{L}) \leq_{\text{a.s.}} T(L) \leq_{\text{a.s.}} \tilde{T}(\tilde{L}).$$

We will now argue that $\hat{L}, \tilde{L} \in \mathcal{RV}(\alpha)$. Clearly,

$$\max\{L - b, 0\} \leq_{\text{a.s.}} \hat{L} \leq_{\text{a.s.}} L \leq_{\text{a.s.}} \tilde{L} \leq_{\text{a.s.}} L + b.$$

Using Lemma 4, we see that $P\left(\max\{L - b, 0\} > t\right) \sim P\left(L + b > t\right)$. This implies, using Lemma 1, that

$$P\left(\hat{L} > t\right) \sim P\left(L > t\right) \sim P\left(\tilde{L} > t\right),$$

which in turn implies $\hat{L}, \tilde{L} \in \mathcal{RV}(\alpha)$ (see Lemma 3). By Lemma 6, we see that

$$P\left(\tilde{T}(\hat{L}) > t\right) \sim P\left(\tilde{T}(\tilde{L}) > t\right) \sim P\left(L > \frac{t}{g(b)}\right).$$

This implies $P\left(T(L) > t\right) \sim P\left(L > \frac{t}{g(b)}\right)$ by Lemma 1. □

3.4 Fragmentation to minimize the average completion time

In the previous section, we studied the tail asymptotics of the completion time; in this section, we turn our attention to its mean. Specifically, under the assumption that F has a non-decreasing failure rate, we derive the fragmentation policy that minimizes the expected completion time. We show that this policy divides the file into equal sized fragments, whose size depends on the file size. We also present a fragmentation policy that is blind to the file size, but is asymptotically optimal. We show that under both these policies, the completion time is light-tailed so long as L is light-tailed. If L is regularly varying, then the completion time is regularly varying with the same index.

Consider

$$\min_x \mathbb{E}\left[T^x(L)\right] := \min_x \left(\lim_{N \to \infty} \mathbb{E}\left[\sum_{n=1}^{N} \tau_n \mid l_1 = L \right] \right) \tag{3.7}$$

An *optimal policy* is one that achieves the minimum of (3.7). We will restrict ourselves to the class of stationary Markov policies where the decision at time n depends only on the state l_n and not on the time n nor on past states. Since any optimal policy will never choose fragment sizes x_n with $P(A_1 \geq x_n + \phi) = 0$,

we will assume without loss of generality that $P(A_1 \geq x_n + \phi) > 0$ for the class of policies that we consider. Our discussion in this section (except in 3.4.3, which deals with completion time tail asymptotics) will be for any realization of the initial file size $L > 0$.

3.4.1 Optimal policy

A stationary Markov policy is a function $x(l)$ of the remaining file size l with the following interpretation. Given l, a packet of size $x(l) + \phi$ is formed. If the packet is successfully transmitted, the remaining file size will be $l - x(l)$. If the transmission fails, the file size remains unchanged and therefore the next fragment remains $x(l)$, until the packet is successfully transmitted. Recall that F is the d.f. of A_i. The expected time it takes to successfully transmit a fragment is $(x(l) + \phi)/\overline{F}(x(l) + \phi)$, the cost per trial multiplied by the expectation of the number of trials, which is geometrically distributed with parameter $F(x(l) + \phi)$. This implies that if we let $J(l) := \mathbb{E}[T(l)]$ denote the expected completion time when the file size is l under a generic Markov policy $x(l)$, then

$$J(l) \quad = \quad J(l - x(l)) + \frac{x(l) + \phi}{\overline{F}(x(l) + \phi)}.$$

Given any Markov policy $x(l)$, consider the sequence of fragments $x_1, x_2, \ldots$, generated from an initial file size L, defined recursively as:

$$x_1 := x(L);\ x_{i+1} := x(L - x_i), i \geq 1$$

such that $\sum_k x_k = L$. Define the expected time to successfully transmit a segment of size x as

$$h(x) \quad = \quad \frac{x + \phi}{\overline{F}(x + \phi)}. \tag{3.8}$$

The expected completion time is thus

$$J(L) \quad = \quad \sum_k h(x_k).$$

Since $h(x) \geq h(0) > \phi > 0$ for all $x \geq 0$, an optimal policy must only have finitely many terms in $J(L)$. Let $J^*(L)$ denote the (minimum) expected completion time under an optimal policy x^*.

Consider the following optimization problem:

$$H^* := \min_K \min_{y_1,\ldots,y_K} \quad \sum_{k=1}^{K} h(y_k) \tag{3.9a}$$

$$\text{subject to} \quad \sum_{k=1}^{K} y_k = L \tag{3.9b}$$

$$y_k > 0, \;\; k = 1, \ldots, K \tag{3.9c}$$

$$K = 1, 2, \ldots. \tag{3.9d}$$

We now argue that, given $L > 0$, the sequence of fragment sizes $x^* := (x_1^*, x_2^*, \ldots, x_{K^*}^*)$ generated by a Markov policy $x^*(l)$ minimizes the expected completion time $\mathbb{E}[T(L)]$ if and only if (K^*, x^*) is a minimizer of (3.9a)–(3.9d). We can thus focus on solving (3.9a)–(3.9d). Indeed, we will show that under Assumption A1, (3.9a)–(3.9d) has a unique solution with $x_i^* = x^*$ for all i, implying that the optimal policy divides the file into equal sized fragments.[2]

Now, any finite sequence $(x_1, x_2, \ldots, x_K)$ with $\sum_k x_k = L$, $x_k > 0$ is a feasible solution of (3.9a)–(3.9d). Hence, $H^* \leq J^*(L)$. Conversely, given any minimizer (K^*, y^*) of (3.9a)–(3.9d), we will exhibit a Markov policy $x(l)$ that generates the sequence of fragment sizes that coincide with the given $y^* = (y_1^*, \ldots, y_{K^*}^*)$. This implies the minimum expected completion time satisfies $J^*(L) \leq H^*$. Hence, $J^*(L) = H^*$.

Parametrize the optimization problem (3.9a)–(3.9d) by the file size in (3.9b), and write any minimizer as $(K^*(l), y^*(l))$ when the file size is l. Consider the Markov policy $x(l)$ that solves (3.9a)–(3.9d) with file size l and selects the segment size $x(l) = y_1^*(l)$, i.e., the policy uses the first element of the solution $y^*(l)$ as the segment size when the remaining file size is l. The next segment size under policy $x(l)$ therefore comes from the solution of (3.9a)–(3.9d) with file size $l - x(l))$, i.e., $x(l - x(l)) = y_1^*(l - y_1^*(l))$. But $y_1^*(l - y_1^*(l))$ must be (equal to) the second element in the original solution, i.e., $y_1^*(l - y_1^*(l)) = y_2^*(l)$, for otherwise, $y^*(l)$ could not have been a minimizer. This implies by induction that the Markov policy $x(l)$ generates the sequence of fragment sizes from L that coincides with (K^*, y^*).

The main result of this section is the following theorem that says that the optimal policy creates equal sized fragments. The optimal fragment size depends on the file size.

$$g(x) = \frac{x + \phi}{x\overline{F}(x + \phi)} \tag{3.10}$$

and

$$a = \arg\min_x g(x), \qquad x \in \mathbb{R}_+ \tag{3.11}$$

Note that $g(x) = h(x)/x$ where $h(x)$ is the expected cost (time) to successfully transmit a segment of size

[2]We abuse notation and use x to denote a fragmentation policy, a vector of fragment sizes, or a scalar representing a constant fragment size, depending on the context; x^* denotes these quantities under an optimal policy.

x defined in (3.8). Hence we can interpret $g(x)$ as the per-bit cost for a fragment of size x, and a as the fragment size that minimizes the per-bit cost. It will become clear below that the optimal fragment size x^* is close to a and the minimum cost $J^*(L)$ is close to $Lg(a)$, under the following assumption:

A1: The density function $F' =: f$ exists. Moreover, the failure rate $\lambda(x) := f(x)/\overline{F}(x)$ is continuous and non-decreasing.[3]

Theorem 9 (Optimal fragmentation)**.** *Under assumption A1, for any $L > 0$, minimizers (K^*, x^*) of (3.9) is given by:*

1. *K^* equals $\lfloor L/a \rfloor$ or $\lceil L/a \rceil$ whichever produces a smaller value of $g(L/K^*)$.*
2. *$x_k^* = L/K^*$ for $k = 1, \ldots, K^*$.*

Therefore, the optimal policy divides the file into K^* fragments of equal size. Each fragment is (re)submitted to the channel until the transmission is successful.

Proof of Theorem 9. We will first prove that, given any K, the minimizer x^* of the inner minimization exists, is unique, and $x_k^* = L/K$ for all k. We then prove that the optimal K^* is as stated in the theorem.

Given any integer $K > 0$, by (3.8), the KKT condition [14] for the inner optimization problem in (3.9a) implies that the optimum $x^* = (x_1^*, \ldots, x_K^*)$ satisfies, for all $k = 1, \ldots, K$,

$$\frac{dh(y_k)}{dy_k} = \frac{1}{\overline{F}(y_k + \phi)} + (y_k + \phi)\frac{f(y_k + \phi)}{\big(\overline{F}(y_k + \phi)\big)^2} = \lambda. \tag{3.12}$$

By assumption A1, $\lambda(x) = f(x)/\overline{F}(x)$ is non-decreasing. Moreover $1/\overline{F}(x)$ is non-decreasing, and $x/\overline{F}(x)$ is strictly increasing. Therefore $h'(x)$ is strictly increasing, which is equivalent to $h(x)$ being strictly convex. Thus the inner minimization problem is strictly convex and the KKT condition is also sufficient. A unique solution $x^* = (x_1^*, \ldots, x_K^*)$ exists. Moreover, since all x_k^* are uniquely determined by (3.12), they are the same and hence $x_k^* = L/K$ for all k.

This reduces the minimization (3.9) to:

$$\min_K \qquad K\frac{L/K + \phi}{\overline{F}(L/K + \phi)} \;=\; L\frac{L/K + \phi}{L/K\overline{F}(L/K + \phi)}.$$

Since L is constant, this is equivalent to solving

$$x^* = \arg\min_x g(x), \qquad x = \left\{L, \frac{L}{2}, \frac{L}{3}, \ldots\right\}, \tag{3.13}$$

where g is defined in (3.10). The derivative of $g(x)$ is

$$\frac{dg(x)}{dx} = \frac{(x^2 + \phi x)f(x + \phi) - \phi\overline{F}(x + \phi)}{\big(x\overline{F}(x + \phi)\big)^2}.$$

[3] If $f(x) = \overline{F}(x) = 0$, define $\lambda(x) = \infty$.

Since $\lambda(x) = f(x)/\overline{F}(x)$ is continuous by assumption, and since $\lim_{x\to 0} g(x) = \infty$ and $\lim_{x\to\infty} g(x) = \infty$, an optimal $x^* \in \{L, L/2, L/3, \dots\}$ and hence optimal K^* exists. Moreover, any unconstrained minimum a of $g(x)$ must also be an extremum. Thus, setting $g'(x) = 0$ yields

$$\xi(x) \;:=\; \frac{f(x+\phi)}{\overline{F}(x+\phi)} \cdot \frac{x(x+\phi)}{\phi} \;=\; 1.$$

Since $f(x+\phi)/\overline{F}(x+\phi)$ is non-decreasing, $x(x+\phi)/\phi$ is strictly increasing, $\xi(0) = 0$, $\lim_{x\to\infty} \xi(x) = \infty$, and $f(x)$ is continuous, it follows that the equation $\xi(x) = 1$ will have a unique solution, which is the unique minimizer a of $g(x)$ defined in (3.11). Moreover, it implies that $g(x)$ is unimodal. This means that x^* is equal to $\lfloor L/a \rfloor$ or $\lceil L/a \rceil$, whichever produces a smaller $g(x)$ value. □

Note that since $g(0) = \infty$, the theorem implies that $K^* = 1$ if $L \le a$. [7] provides a useful sufficient condition for Assumption A1: if f is log-concave, so is F. Since F is log-concave if and only if its failure rate is non-decreasing, a log-concave f satisfies A1. This is useful when F is hard to determine, e.g., for the Gaussian distribution.

The result of Theorem 9 applies to two failure models described in [62, pp. 131] — a model for satellite communication wherein A_i is exponentially distributed, and a model for terrestrial communication, wherein A_i has a uniform distribution.

We now show that, when L is large, the unique optimal fragment size x^* is close to a; indeed, x^* approaches a as L increases.

Theorem 10. *Suppose $L > a$. Under assumption A1, the optimal fragment size $x^*(L)$ satisfies:*

1. $a/2 < x^*(L) \le 2a$.
2. $a/(1 + a/L) < x^*(L) \le a/(1 - a/L)$.

Proof. We know that for some integer K:

$$\frac{L}{K+1} \le a < \frac{L}{K}, \tag{3.14}$$

and

$$x^* \;=\; \frac{L}{K} \quad \text{or} \quad x^* \;=\; \frac{L}{K+1}.$$

In the first case, $x^*K/(K+1) \le a < x^*$ implying $x^*/2 \le a < x^*$, i.e., $a < x^* \le 2a$. In the second case, $x^* \le a < x^*(K+1)/K \le 2x^*$ implying $a/2 < x^* \le a$. Combining yields $a/2 < x^* \le 2a$.

From (3.14) we get

$$\frac{L}{a} - 1 \le K < \frac{L}{a},$$

implying

$$a < \frac{L}{K} \leq \frac{a}{1-a/L} \qquad \text{and} \qquad \frac{a}{1+a/L} < \frac{L}{K+1} \leq a.$$

Hence

$$\frac{a}{1+a/L} < x^* \leq \frac{a}{1-a/L}.$$

□

This admits the following useful corollary.

Corollary 1.

$$\lim_{L\to\infty} x^*(L) = a.$$

3.4.2 Simple blind policy $x(l) = \min\{a, l\}$

The optimal fragmentation policy in Theorem 9 depends on the file size L. Consider the L-independent blind policy $x(l) = \min\{a, l\}$ where the fragment size a, given by (3.11), is always used until the remaining file size drops below a when it is transmitted in a single packet. We will abuse notation and use a to denote both this blind policy and the fragment size under this policy. Let $J^a(L)$ denote the expected file completion time under policy a when the file size is L. Recall that $J^*(L)$ denotes the minimum expected completion time. From Corollary 1, we know that policy a is asymptotically optimal, i.e., $x^*(L) \to a$. Hence we would expect $J^a(L)$ and $J^*(L)$ to be close for large L. The following result bounds their distance by an L-independent constant for any L.

Theorem 11. *Under Assumption A1, for any $L > 0$,*

$$\begin{aligned} 0 \;\leq\; J^*(L) - Lg^* \;&\leq\; h(a) \\ J^a(L) - J^*(L) \;&\leq\; h(a) \end{aligned}$$

where $h(x)$ is defined in (3.8) and $g^* := g(a)$ is defined by (3.10) *and* (3.11)*.*

Proof. If $L = ka$ for some integer k, the proof of Theorem 9 shows that the policy a is optimal, in which case $J^a(L) = J^*(L)$. Suppose then that $ka < L < (k+1)a$ for some integer k. Clearly, $J^a(L) = kh(a) + h(L - ka)$. Since h is monotone, we have

$$kh(a) \leq J^a(L) \leq (k+1)h(a). \tag{3.15}$$

Since $J^*(L)$ is monotone in L, we have

$$kh(a) = J^*(ka) \leq J^*(L) \leq J^*((k+1)a) = (k+1)h(a). \tag{3.16}$$

Combining (3.15) and (3.16), we get that $J^a(L) - J^*(L) \leq h(a)$. This proves the sub-optimality bound. Moreover, (3.16) also implies $Lg^* \leq J^*(L) \leq Lg^* + h(a)$, as desired. □

We make the following remarks:

1. Under both the optimal policy x^* and the blind policy a, the expected completion time grows (roughly) linearly in the file size, the approximating proportionality constant being the minimum per-bit cost $g(a)$.

2. The sub-optimality in expected completion time under the blind policy a is bounded by a constant independent of the file size.

3.4.3 Tail asymptotics under policies x^* and a

Denote by $T^*(L)$ and $T^a(L)$ respectively the completion times under the policies x^* and a.

Theorem 12. *1. If L is light-tailed, then $T^*(L)$ and $T^a(L)$ are light-tailed.*

2. If $L \in \mathcal{RV}(\alpha)$, then

$$P\left(T^*(L) > t\right) \sim P\left(T^a(L) > t\right) \sim P\left(L > \frac{t}{g(a)}\right)$$

Since the blind policy a belongs to the class of constant fragmentation policies (see Section 3.3), the tail behavior of $T^a(L)$ stated in the theorem follows from Theorem 7. Theorem 10 implies that the optimal policy x^* is a bounded fragmentation policy (see Section 3.3). It follows then from Theorem 6 that $T^*(L)$ is light-tailed if L is light-tailed. However, the exact tail asymptotics of $T^*(L)$ when $L \in \mathcal{RV}(\alpha)$ claimed above requires a separate proof, which we give in Appendix 3.B.

Theorem 12 implies that the policies x^* and a yield good completion time tail behavior. Therefore, these policies have the desirable property of providing good performance with respect to the mean as well as the tail of the completion time distribution.

3.5 Conclusion

It has been discovered that file completion time on an unreliable channel can be heavy-tailed even when file size distribution is not. To mitigate this, we show that independent or bounded file fragmentation guarantees light-tailed file completion time as long as the file size is light-tailed. When the file size is heavy-tailed (specifically, regularly varying), the completion time is as heavy-tailed as the file size (in the degree sense). Finally, seeking to minimize the expected completion time, we derive the optimal fragmentation policy as well as a simple suboptimal blind fragmentation policy. Importantly, both these policies also provide good completion time tail performance.

3.A Proof of Lemma 2

Proof of Lemma 2. Proofs are Statements (1) and (2) are elementary and are omitted.

Proof of Statement (3) Let $Z = \sum_{i=1}^{N} X_i$. Pick $\beta \in (0, 1/\mathbb{E}[X_1])$.

$$\begin{aligned} P(Z > t) &= P(Z > t;\ N \leq \beta t) + P(Z > t; N > \beta t) \\ &\leq P\left(\sum_{i=1}^{\lfloor \beta t \rfloor} X_i > t\right) + P(N > \beta t). \\ &\leq P\left(\sum_{i=1}^{\lfloor \beta t \rfloor} X_i > \frac{\lfloor \beta t \rfloor}{\beta}\right) + P(N > \beta t) \\ &=: I + II. \end{aligned}$$

Using the Chernoff bound, we conclude that there exists $\alpha_1 > 0$ such that $I \leq e^{-\alpha_1 \lfloor \beta t \rfloor}$. Also, since N is light-tailed, there exists $\alpha_2 > 0$ such that $II \leq e^{-\alpha_2 \beta t}$ for large enough t. Since we have an exponentially decaying upper bound on the tail of Z, it follows that Z is light-tailed. This completes the proof of Statement (3).

Proof of Statement (4) Define $N := \min\{n \in \mathbb{N} | \sum_{i=1}^{n} X_i \geq L\}$. It suffices to prove the proposition assuming the X_i are light-tailed. Indeed, if the X_i are heavy-tailed, then we may define $Y_i = X_i \mathbf{1}(X_i \leq y)$ for some $y > 0$ such that $P(Y_i > 0) > 0$. We would then be able to claim that $\tilde{N} := \inf\{n \in \mathbb{N} | \sum_{i=1}^{n} Y_i \geq L\}$ is light-tailed. However, since $N \leq_{\text{a.s.}} \tilde{N}$, this would imply N is light-tailed (from Statement (1) of this lemma). Let us assume therefore that the X_i are light-tailed for the remainder of the proof. Pick $\beta \in (0, \mathbb{E}[X_1])$.

$$\begin{aligned} P(N > n) &= P\left(\sum_{i=1}^{n} X_i < L\right) \\ &\leq P\left(\sum_{i=1}^{n} X_i < L;\ L > \beta n\right) + P\left(\sum_{i=1}^{n} X_i < L;\ L \leq \beta n\right) \\ &\leq P(L > \beta n) + P\left(\sum_{i=1}^{n} X_i < \beta n\right) \\ &=: I + II. \end{aligned}$$

Since L is light-tailed, there exists $\alpha_1 >$ such that $I \leq e^{-\alpha_1 n}$ for large enough n. Also, using the Chernoff bound, we conclude that there exists $\alpha_2 > 0$ such that $II \leq e^{-\alpha_2 n}$. Since we now have an exponentially decaying upper bound on the tail d.f. of N, it follows that N is light-tailed. This completes the proof of Statement (4). □

3.B Proof of Theorem 12: Tail asymptotics of $T^*(L)$

This section is devoted to proving the tail behavior of $T^*(L)$ claimed in the statement of Theorem 12, i.e., we prove that

$$P\left(T^*(L) > t\right) \sim P\left(L > \frac{t}{g(a)}\right). \tag{3.17}$$

The proof of (3.17) is based on stochastically bounding the optimal completion time $T^*(l)$ from both sides. We need the following notation. Let $W^{(z)}$ denote a random variable distributed as the time to successfully transmit a fragment of size $z > 0$. Note that $h(z) = \mathbb{E}\left[W^{(z)}\right]$, and that $W^{(z)}$ is stochastically increasing in z. Since $W^{(z)} \stackrel{d}{=} \sum_{i=1}^{N}(z+\phi)$, where N is a geometric random variable with mean $1/P\left(A_1 \geq z+\phi\right)$, we infer from Lemma 2 that $W^{(z)}$ is light-tailed. Let $(W_i^{(z)}, i \geq 1)$ denote a sequence of i.i.d. random variables independent of L distributed as $W^{(z)}$.

Now, pick $\epsilon \in (0, a)$. Since $x^*(l) \stackrel{l\uparrow\infty}{\to} a$ by Corollary 1, there exists an $l_0 > 0$ such that $x^*(l) \in (a-\epsilon, a+\epsilon)$ for all $l \geq l_0$. Note that for $l \geq l_0$,

$$\left\lfloor \frac{l}{a-\epsilon} \right\rfloor \geq K^*(l) \geq \left\lceil \frac{l}{a+\epsilon} \right\rceil.$$

Define

$$\hat{T}(l) := \begin{cases} 0 & \text{for } 0 \leq l < l_0 \\ \sum_{i=1}^{\lceil \frac{l}{a+\epsilon} \rceil} W_i^{(a-\epsilon)} & \text{for } l \geq l_0 \end{cases} \quad ; \quad \tilde{T}(l) := \begin{cases} T^*(l) & \text{for } 0 \leq l < l_0 \\ \sum_{i=1}^{\lfloor \frac{l}{a-\epsilon} \rfloor} W_i^{(a+\epsilon)} & \text{for } l \geq l_0 \end{cases}.$$

It is easy to check that $\hat{T}(l) \leq_{\mathrm{st}} T^*(l) \leq_{\mathrm{st}} \tilde{T}(l)$ for all l, which implies that

$$\hat{T}(L) \leq_{\mathrm{st}} T^*(L) \leq_{\mathrm{st}} \tilde{T}(l). \tag{3.18}$$

The following lemmas characterize the tail asymptotics of $\hat{T}(L)$ and $\tilde{T}(l)$.

Lemma 7.

$$P\left(\hat{T}(L) > t\right) \sim P\left(L > \frac{t(a+\epsilon)}{h(a-\epsilon)}\right).$$

Lemma 8.

$$P\left(\tilde{T}(L) > t\right) \sim P\left(L > \frac{t(a-\epsilon)}{h(a+\epsilon)}\right).$$

The above lemmas follow easily from Lemma 6; we omit the proofs. Note that the definitions of $\hat{T}(l)$ and $\tilde{T}(l)$ over $l \leq l_0$ do not contribute to the tails of $\hat{T}(L)$ and $\tilde{T}(L)$.

(3.17) now follows easily by combining (3.18) and Lemmas 7 and 8, and letting $\epsilon \downarrow 0$. This completes the proof.

Chapter 4

Tail-Robust Scheduling via Limited Processor Sharing

4.1 Introduction

In the study of scheduling policies, much of the focus has traditionally been on designing policies that have good performance in expectation. For example, in order to minimize the expected sojourn time (a.k.a. response time, flow time) in a single server queue it is well known that the scheduler should give priority to jobs with small remaining sizes via Shortest Remaining Processing Time (SRPT) [60], which is optimal regardless of the job size distribution and arrival process.

However, providing good performance in expectation is not sufficient. It is also important for a scheduler to provide good *distributional* performance. For example, quality of service guarantees in web applications often rely on specifying guarantees about the tail of the sojourn time distribution, e.g., that 95% of requests will have sojourn time $< s$ seconds.

Resultantly, there has been a substantial amount of work in recent years studying the sojourn time distribution, $P(V > x)$ of scheduling policies in a GI/GI/1 setting. Due to the difficulty of an exact distributional analysis, much of this work focuses on understanding the sojourn time tail asymptotics, i.e., the behavior of $P(V > x)$ as $x \to \infty$, which provides a characterization of the likelihood of large delays. From this work, which we survey briefly in Section 4.2, has emerged an understanding of how to optimally schedule for the sojourn time tail. Interestingly, unlike when optimally scheduling for the expected sojourn time, prior work shows that there are two distinct regimes: when the job size distribution is light-tailed, First Come First Served (FCFS) scheduling minimizes the sojourn time tail [54], while if the job size distribution is heavy-tailed, SRPT, Processor Sharing (PS), and many other policies (e.g., all SMART policies [52]) minimize (up to a constant) the sojourn time tail [13].

Interestingly, among the prior work, there are no policies that are optimal across both light-tailed and heavy-tailed job size distributions. In fact, Wierman & Zwart have recently proved an impossibility result [68], which states that no work-conserving policy that is non-learning (i.e., does not learn information

about the workload) can optimize the sojourn time tail across both light-tailed and heavy-tailed job size distributions. Further, among the prior work, the policies that produce the best possible sojourn time tail behavior under heavy-tailed job size distributions produce the worst possible sojourn time tail behavior under light-tailed job size distributions, and vice-versa. Indeed, there are no policies that have been shown to maintain even better than worst-case sojourn time tail performance across both light-tailed and heavy-tailed job size distributions.

So, at this stage, the literature understands how to design a scheduling policy to be optimal for the sojourn time tail given a particular workload, but cannot design a scheduling policy that is robust, even minimally so, across both light-tailed and heavy-tailed job size distributions. This is in stark contrast to the case of scheduling for expected sojourn time, where SRPT is optimal and robust.

The lack of robustness when scheduling for the sojourn time tail is relevant from a practical perspective because determining whether a particular real-world workload is light-tailed or heavy-tailed is a difficult task. For example, there is an unending debate over whether to model web file sizes as an unbounded heavy-tailed distribution or as a bounded distribution with a power-law body. Ideally, a scheduler design should be robust to such assumptions. *The goal of this chapter is to present a scheduling policy that is 'tail-robust', i.e., provides robust performance (in terms of the sojourn time tail) across both heavy-tailed and light-tailed job size distributions.*

The main contribution of this work is to prove that Limited Processor Sharing (LPS-c) can be designed to be tail-robust. Under LPS-c, there is a limited multiprogramming level c, which determines the maximum number of jobs that the service rate is shared among. Specifically, jobs are queued according to the order of arrival and if there are n jobs in the system then the $\min(n, c)$ jobs which arrived earliest each receive a service rate of $1/\min(n, c)$. LPS-c is a natural candidate for our goal because, as c grows from 1 to ∞, LPS-c transitions from FCFS, which is optimal under light-tailed job sizes, to PS, which is optimal under heavy-tailed job sizes. Our goal will be to determine how to choose an intermediate c such that LPS-c is tail-robust. It turns out that to achieve tail-robustness, the choice of c must incorporate some information about the workload. We will prove that this c can be chosen in such a way that only information about the system load ρ is necessary, which is not an unreasonable assumption as this information is also necessary to achieve system stability.

It is important to point out that LPS-c is not a policy that we artificially constructed to fit the goals of this chapter. LPS-c is a practical policy that is actually a more realistic version of both FCFS and PS in the case of many computer systems, where it is unrealistic to share the server among unboundedly many jobs or to devote the server entirely to a single job. Given its practical importance, there have been a number of prior studies of LPS-c: Avi-Itzhak & Halfin [6] propose an approximation for the mean response time assuming Poisson arrivals. A computational analysis based on matrix geometric methods is performed in Zhang & Lipsky [69, 70]. Some stochastic ordering results are derived in Nuyens & van der Weij [51]. Zhang, Dai & Zwart [71–73] develop fluid, diffusion, and heavy traffic approximations. Finally, Gupta & Harchol-

Balter [27] consider approximation methods and Markov decision techniques to determine the optimal level c when the system is not work-conserving. However, none of the prior work has focused on the sojourn time tail of LPS-c.

In order to understand how to design LPS-c so that it is tail-robust, we first need to analyze the sojourn time tail asymptotics in both the case of heavy-tailed and light-tailed job size distributions. We do this in Sections 4.3 and 4.4, respectively. In both cases our analysis reveals interesting insights. For example, for heavy-tailed job sizes we find that the behavior of LPS-c is similar to that of the analogous GI/GI/c queue, where each server works at rate $1/c$. However, this is not the case for light tails, where quite a few qualitatively different scenarios may lead to large sojourn times. In particular, a large sojourn time may occur through a combined effect of a large backlog in the system upon arrival, a large service time, and a higher than usual input during the sojourn of the customer under consideration. Interestingly, this is in contrast to policies that have been analyzed up to this point, under which one of these phenomena typically dominates.

The sojourn time tail asymptotics of LPS-c that we derive in Sections 4.3 and 4.4 also highlight a tension that must be resolved when attempting to design LPS-c robustly. In particular, when the job size distribution is light-tailed, reducing c lightens the sojourn time tail; however, when the job size distribution is heavy-tailed, increasing c lightens the sojourn time tail. This highlights the trade-off necessary between optimality and robustness.

In Section 4.5, we show that despite the conflicting demands on c placed by the light-tailed and heavy-tailed regimes, it is indeed possible to choose c so that LPS-c is tail robust. In particular, we prove that with $c = \lfloor 1/(1-\rho) \rfloor + 1$, the sojourn time tail under LPS-c is better than worst-case across a large class of heavy-tailed (regularly varying) job size distributions and light-tailed (phase-type) job size distributions. Further, this choice of c ensures that for large subclasses of heavy-tailed and light-tailed job size distributions the sojourn time tail is optimal (see Corollary 2). Additionally, this design is robust to estimation errors in ρ — as long as the estimate of ρ that is used is an upper bound on the true ρ, this c will still be tail-robust.

Importantly, there is some freedom among the class of tail-robust designs possible using LPS-c. In particular, Corollary 3 presents a parameterized design for c that allows the designer to vary the importance placed on optimality in the heavy-tailed and light-tailed regimes while still guaranteeing tail-robustness. However, in order to ensure that LPS-c is tail robust, it is necessary to maintain $c \geq \lfloor 1/(1-\rho) \rfloor + 1$ to handle heavy-tailed job size distributions.

The remainder of the chapter is organized as follows. In Section 4.2, we introduce the model and notation for the chapter, and discuss prior work studying the sojourn time asymptotics of scheduling policies. In Sections 4.3 and 4.4 we present our new results characterizing the sojourn time asymptotics of LPS-c. Then, in Section 4.5 we present the main results of the chapter showing how to design the multiprogramming level c for LPS-c to ensure tail-robust performance. Finally, we conclude in Section 4.6.

4.2 Preliminaries

4.2.1 Model and notation

Throughout this chapter, our focus will be on the GI/GI/1 queue. Jobs arrive according to a renewal process; let A denote a generic interarrival time. Each job has an independent, identically distributed service requirement (size); let B denote a generic job size. The server speed is taken to be unity. We make the following standard assumptions: (i) load $\rho := \frac{\mathbb{E}[B]}{\mathbb{E}[A]} \in (0,1)$, (ii) $P(B > A) > 0$ (otherwise there would be no queueing).

Denote $\alpha := \mathbb{E}[A]$, $\beta := \mathbb{E}[B]$. Let B_e denote a random variable distributed as the excess/residual lifetime of B, i.e., $P(B_e > x) = \frac{1}{\beta}\int_x^\infty P(B > t)\,dt$ for $x \geq 0$. For functions $\varphi(x)$ and $\xi(x)$, the notation $\varphi(x) \sim \xi(x)$ means $\lim_{x\to\infty} \frac{\varphi(x)}{\xi(x)} = 1$, $\varphi(x) \gtrsim \xi(x)$ means $\liminf_{x\to\infty} \frac{\varphi(x)}{\xi(x)} \geq 1$.

The *sojourn time* (response time) of a job refers to the time between its arrival and its departure. The *waiting time* (delay) of a job refers to the time between its arrival and the instant it first receives service. V_π and D_π denote respectively random variables distributed as per the sojourn time and waiting time of a job in the stationary GI/GI/1 queue operating under scheduling discipline (policy) π. In this chapter, our interest is centered around the asymptotic behavior of the sojourn time tail, i.e., the behavior of $P(V_\pi > x)$ as $x \to \infty$.

In our analysis of the tail behavior of the stationary sojourn time, we focus on the sojourn time of a 'tagged' job, assumed to arrive into the stationary queue at time 0, with size B_0. W denotes the total work (backlog) in the system just before the arrival of the tagged job. B_i denotes the size of the i-th arrival after time 0. For $i \geq 1$, A_i denotes the time between the $(i-1)$-st and i-th arrival. For $x > 0$, $N(x) := \max\{n \in \mathbb{N} : \sum_{i=1}^n A_i \leq x\}$ is the number of arrivals into the system in time interval $(0, x]$. $A(x) := \sum_{i=1}^{N(x)} B_i$ is the total work entering the system in the interval $(0, x]$.

4.2.2 Heavy-tailed and light-tailed distributions

For any non-negative random variable X, $F_X(\cdot)$ denotes the distribution function (d.f.) of X, i.e., $F_X(x) = P(X \leq x)$, and $\bar{F}_X(x) := 1 - F_X(x)$ denotes the tail distribution function of X. $\Phi_X(\cdot)$ denotes the moment generating function of X, i.e., $\Phi_X(s) = \mathbb{E}\left[e^{sX}\right]$. The random variable X (or its d.f. F_X) is defined to be *heavy-tailed* if $\Phi_X(s) = \infty$ for all $s > 0$. X (or its d.f. F_X) is defined to be *light-tailed* if it is not heavy-tailed, i.e., if $\Phi_X(s) < \infty$ for some $s > 0$.

The following subsets of the class of heavy-tailed distributions will be of interest to us. X (or its d.f. F_X) is said to be *long-tailed* (denoted $X \in \mathcal{L}$) if $\lim_{x\to\infty} \frac{P(X>x+y)}{P(X>x)} = 1$ for all $y > 0$. The class of long-tailed distributions includes most of the common heavy-tailed distributions, including the Pareto, the lognormal and the heavy-tailed Weibull distribution [64]. X (or its d.f. F_X) is said to be *regularly varying* with index $\theta > 1$ (denoted $X \in \mathcal{RV}(\theta)$) if $P(X > x) = x^{-\theta}L(x)$, where $L(x)$ is a slowly varying function, i.e., $L(x)$ satisfies $\lim_{x\to\infty} \frac{L(xy)}{L(x)} = 1\ \forall\ y > 0$. Note that all Pareto distributions are included in this class. The class

of regularly varying distributions is a strict subset of the class of long-tailed distributions, which in turn is a strict subset of the class of heavy-tailed distributions [64].

We describe the (logarithmic) asymptotic tail behavior of a heavy-tailed random variable X using its *tail index*, defined as

$$\Gamma(X) := \lim_{x\to\infty} -\frac{\log P(X > x)}{\log(x)},$$

when the limit exists. Note that if $\Gamma(X) \in (0,\infty)$, then for arbitrarily small $\epsilon > 0$, $x^{-(\Gamma(X)+\epsilon)} \leq P(X > x) \leq x^{-(\Gamma(X)-\epsilon)}$ for large enough x. This means the tail index is useful for describing the asymptotic tail behavior of distributions that exhibit a roughly 'power-law' tail. Note that a smaller value of tail index implies a 'heavier' tail. Section 4.3 is devoted to the analysis of $\Gamma(V_{LPS-c})$ when $B \in \mathcal{RV}(\theta)$.

Similarly, we describe the (logarithmic) asymptotic tail behavior of a light-tailed random variable X using its *decay rate*, defined as

$$\gamma(X) := \lim_{x\to\infty} -\frac{\log P(X > x)}{x},$$

when the limit exists. If $\gamma(X) \in (0,\infty)$, then for arbitrarily small $\epsilon > 0$, $e^{-(\Gamma(X)+\epsilon)x} \leq P(X > x) \leq e^{-(\Gamma(X)-\epsilon)x}$ for large enough x. This means the decay rate is useful for describing the asymptotic tail behavior of distributions that have a roughly 'exponential' tail. Again, note that a smaller value of $\gamma(X)$ implies a 'heavier' tail. It is easy to prove that

(i) $\Phi_X(s) < \infty$ for all $s < \gamma(X)$,

(ii) if $\gamma(X) < \infty$, then $\Phi_X(s) = \infty$ for all $s > \gamma(X)$.

This implies that if we assume that the decay rate of X exists, then X is light-tailed if and only if $\gamma(X) \in (0,\infty]$. Section 4.4 is devoted to the analysis of $\gamma(V_{LPS-c})$ for the case $\gamma(B) \in (0,\infty]$.

4.2.3 Related literature

We now review the sojourn time tail asymptotics for the following well known scheduling policies: First Come First Served (FCFS), Preemptive Last Come First Served (PLCFS), Processor Sharing (PS), and Shortest Remaining Processing Time (SRPT). Note first, that for any work conserving scheduling policy π, V_π may be stochastically bounded as follows: $B \leq_{\rm st} V_\pi \leq_{\rm st} Z^*$, where Z^* denotes the total time to emptiness of the queue in steady state, just after an arrival. We consider separately the case of heavy-tailed and light-tailed job sizes.

Heavy-tailed job sizes: For the GI/GI/1 queue with regularly varying job sizes, it is known that

$$\Gamma(V_{PS}) = \Gamma(V_{SRPT}) = \Gamma(V_{PLCFS}) = \Gamma(B) = \theta,$$
$$\Gamma(V_{FCFS}) = \Gamma(Z^*) = \theta - 1.$$

See the survey by Boxma & Zwart [13] for details. Based on the bounds on V_π described above, it is clear that PS, SRPT, and PLCFS produce the optimal sojourn time tail index. The sojourn time tail under FCFS is

one degree heavier; moreover, FCFS produces the worst possible sojourn time tail index. In fact, it turns out that all non-preemptive scheduling policies produce this worst possible sojourn time tail index (see the paper by Anantharam [3]).

Light-tailed job sizes: In describing the sojourn time asymptotics under light-tailed job sizes, the following function plays a key role. For $s \geq 0$, $\Psi(s) := -\Phi_A^{-1}\left(\frac{1}{\Phi_B(s)}\right)$. The following lemma gives an interpretation to this function.

Lemma 9 (Mandjes-Zwart [41])**.** *For $s \geq 0$, $\lim_{x\to\infty} \frac{\log \mathbb{E}\left[e^{sA(x)}\right]}{x} = \Psi(s)$. Further, $\Psi(s)$ is strictly convex and lower semi-continuous.*

For the GI/GI/1 queue with light-tailed job sizes,

$$\begin{aligned}\gamma(V_{FCFS}) &= \gamma_F := \sup\{s \geq 0 \,:\, \Psi(s) - s \leq 0\},\\ \gamma(V_{PLCFS}) &= \gamma_L := \sup_{s\geq 0}\{s - \Psi(s)\};\end{aligned}$$

see Nuyens & Zwart [53]. From the strict concavity of $s - \Psi(s)$, and since $\Psi'(0) = \rho$, it is easy to show that $\gamma_L < (1-\rho)\gamma_F$. This implies the stationary sojourn time tail under FCFS is 'lighter' than that under PLCFS. It has been proved by Ramanan and Stolyar [54] that the sojourn time decay rate under FCFS is actually optimal. Moreover, it has been proved by Nuyens et al. [52] that $\gamma(Z^*) = \gamma_L$, implying that PLCFS produces the worst possible sojourn time decay rate under light-tailed job sizes. Under mild regularity conditions, we additionally have

$$\gamma(V_{PS}) = \gamma(V_{SRPT}) = \gamma_L.$$

The above discussion highlights the dichotomy described before; the policies that produce the best possible sojourn time tail behavior under heavy-tailed job size distributions produce the worst possible sojourn time tail behavior under light-tailed job size distributions, and vice versa.

4.2.4 Busy period decay rate as a function of server speed

We conclude this section with a brief discussion on the dependence of the busy period decay rate of a GI/GI/1 queue on the server speed. This discussion will play a role in our analysis of the sojourn time decay rate under LPS-c in Section 4.4.

Define, for $r \geq 0$, $g(r) := \sup_{s\geq 0}[rs - \Psi(s)]$. $g(r)$ is the busy period decay rate, if the server speed equals r. It is easy to see that $g(\cdot)$ is convex increasing; $g(r) = 0$ for $r \leq \rho$, $g(1) = \gamma_L$. Suppose that $\gamma(B) \in (0,\infty)$. In this case, for all $s > \gamma(B)$, $\Phi_B(s) = \infty$, implying $\Psi(s) = \infty$. This means $g(r) = \sup_{s\in[0,\gamma(B)]}[rs - \Psi(s)]$. Now, since $\Psi(\cdot)$ is strictly convex and lower semi-continuous, the supremum in the definition of $g(\cdot)$ is uniquely achieved; define $\hat{s}(r) = \arg\max_{s\geq 0}[rs - \Psi(s)]$.

Lemma 10. *If $\gamma(B) \in (0,\infty)$, then $g(r)$ is continuously differentiable over $r \geq 0$. Moreover, $g'(r) = \hat{s}(r)$.*

Proof. That $g'(r) = \hat{s}(r)$ follows by invoking an envelope theorem like Danskin's theorem; see Proposition B.25 in [9]. Since $g(\cdot)$ is convex and differentiable, its derivative must be continuous; see Theorem 25.5 in [56]. □

4.3 Tail asymptotics under heavy-tailed job sizes

We start our analysis by focusing on heavy-tailed job size distributions. In this section, we describe tail asymptotics for the sojourn time under LPS-c for the case of regularly varying job sizes. As we discussed in Section 4.2, there is a significant amount of prior work deriving the sojourn time tail asymptotics for scheduling policies in the heavy-tailed regime. This prior work has shown that: (i) non-preemptive policies (e.g., FCFS) have a sojourn time tail that is one degree heavier than the job size distribution, which is (up to a constant) as bad as possible; and (ii) many preemptive policies (e.g., SRPT, PS) have sojourn time tails that are proportional to the tail of the job size distribution, which is optimal (up to a constant). Interestingly, almost all policies that have been studied have a sojourn time tail that falls into either case (i) or case (ii). Only recently was a policy constructed that has an intermediate sojourn time tail [12]. Our analysis shows that, in many settings, LPS-c also has an intermediate sojourn time tail.

For technical reasons, we must make the following assumption in our analysis.

Assumption 1. *$c\rho$ is not an integer, i.e., $\lfloor c\rho \rfloor < c\rho$.*

Under this assumption, we can state the sojourn time asymptotics of LPS-c as follows.

Theorem 13. *Consider the GI/GI/1 queue. Under Assumption 1, if $B \in \mathcal{RV}(\theta)$ for $\theta > 1$, then*

$$\Gamma(D_{LPS-c}) = \lim_{x\to\infty} -\frac{\log P\left(D_{LPS-c} > x\right)}{\log(x)} = (\theta - 1)(c - \lfloor c\rho \rfloor), \tag{4.1}$$

$$\Gamma(V_{LPS-c}) = \lim_{x\to\infty} -\frac{\log P\left(V_{LPS-c} > x\right)}{\log(x)} = \min\left\{\theta, (\theta - 1)(c - \lfloor c\rho \rfloor)\right\}. \tag{4.2}$$

One natural way to view an LPS-c queue is as a work-conserving version of a GI/GI/c queue, where each server has speed $1/c$. In this view, this theorem can be interpreted as stating that LPS-c has the same sojourn time tail as the GI/GI/c queue in the heavy-tailed regime. In fact, the proof relies on the parallels between these two queues. We prove an upper bound on the tail of delay in Appendix 4.A.1, a lower bound on the tail of delay in Appendix 4.A.2, and then combine them to complete the proof of Theorem 13 in Appendix 4.A.3. The upper bound follows immediately from bounding the LPS-c queue by the GI/GI/c queue, while the lower bound proof uses a probabilistic argument that offers insight into *how* large waiting times occur.

The parallel between the GI/GI/c and LPS-c also motivates us to define $k = k_c := c - \lfloor c\rho \rfloor$. We refer to k as the number of 'spare slots'. The name 'spare slots' refers to the fact that k is the minimum number of infinite-sized jobs that must be added to the LPS-c queue before the it becomes unstable. This definition of k parallels the notion of 'spare servers' which is used in the context of the GI/GI/c queue. In the GI/GI/c setting, the number of spare servers, k, has been shown to determine the moment conditions for delay (see [59]), thus

it is perhaps not surprising that k determines the weight of the sojourn time tail in the LPS-c queue. However, we will see that the parallel between the LPS-c queue and the GI/GI/c queue does not hold in light-tailed regime (Section 4.4).

Another natural view of LPS-c is as a hybrid version of FCFS ($c = 1$) and PS ($c \to \infty$). In this view, Theorem 13 highlights that the sojourn time tail transitions between the sojourn time tails of FCFS and PS as c increases. Specifically, recall that the sojourn time tail index for any work-conserving scheduling policy (if it exists) lies between $\Gamma(V_{FCFS}) = \theta - 1$ and $\Gamma(V_{PS}) = \Gamma(B) = \theta$. When $c = 1$ we have $\Gamma(V_{LPS-1}) = \Gamma(V_{FCFS})$, which is the heaviest possible tail index. However, the weight of the sojourn time tail lightens monotonically as c increases and, as $c \to \infty$, the sojourn time tail index matches that of the job size distribution, which is optimal. Specifically, for all c large enough that $k_c > \theta/(\theta - 1)$ we have that $\Gamma(V_{LPS-c}) = \Gamma(V_{PS}) = \theta$. Thus, in the heavy-tailed setting, LPS-c should be designed with 'large enough' c.

Unfortunately, we will see the that the opposite is true in the light-tailed regime — when job sizes are light-tailed, LPS-c should be designed so that c is 'small enough'. This highlights the tension of designing LPS-c so that it is tail-robust. Understanding this tension is the goal of Section 4.5.

4.4 Tail asymptotics under light-tailed job sizes

We now move to the light-tailed regime and again analyze the sojourn time asymptotics of the LPS-c queue. As we discussed in Section 4.2, there is a significant amount of prior work devoted to the sojourn time asymptotics of scheduling policies in the light-tailed regime. From this prior work has evolved an understanding of what 'bad' events lead to large delays under most common scheduling policies. In particular, a long delay will occur because of one (or more) of the following three effects:

(i) a large backlog is at the server when the tagged job arrives,

(ii) the tagged job has a large size,

(iii) a large number of jobs enter the system during the tagged job's sojourn.

Prior work has provided an understanding of which combination of these three effects is most likely to lead to a large delay under most common scheduling policies. For example, under FCFS a long delay is most likely caused by (i), while under SRPT and PS, a long delay is most likely caused by the combination of (ii) and (iii). As discussed in Section 4.2, FCFS produces the optimal (largest possible) sojourn time decay rate whereas SRPT and PS produce the worst (smallest) possible sojourn time decay rate.

As in the heavy-tailed setting, there is a dichotomy in the previous analyses: all policies that have been analyzed (to the best of our knowledge) have sojourn time tail asymptotics that fall into two categories: long delays are most likely caused by either (i) or a combination of (ii) and (iii). Like in the heavy-tailed setting,

in some cases, LPS turns out to have intermediate tail-asymptotics where the most likely way a 'bad' event can occur is a (workload-dependent) combination of (i), (ii), and (iii).

Let us now state the main result for this section. Throughout, we will assume that the decay rate of the job size distribution exists.

Theorem 14. *Consider the GI/GI/1 queue. If* $\gamma(B) \in (0, \infty)$,

$$\gamma(V_{LPS-c}) = \lim_{x\to\infty} -\frac{\log P\left(V_{LPS-c} > x\right)}{x} = \min_{a\in[0,1]} f_c(a), \tag{4.3}$$

$$\textit{where} \quad f_c(a) := a\gamma_F + \frac{(1-a)\gamma(B)}{c} + \sup_{s\geq 0}\left[(1-a)s\left(1-\frac{1}{c}\right) - \Psi(s)\right]. \tag{4.4}$$

Otherwise, if $\gamma(B) = \infty$, *then* $\gamma(V_{LPS-c}) = \gamma(V_{FCFS})$.

We prove (4.3) by providing matching asymptotic lower and upper bounds on the tail of V_{LPS-c}. The lower bound is proved in 4.B.1, the upper bound is proved in 4.B.2. We then prove the result for the case of $\gamma(B) = \infty$ in 4.B.3.[1]

Given the complexity of the decay rate in Theorem 14, it is important to provide some interpretation of the theorem. To begin, note that, unlike in the heavy-tailed regime, the decay rate of LPS-c does not parallel the decay rate of the GI/GI/c queue where servers have speed $1/c$ (see [57] for a derivation of the decay rate for the GI/GI/c queue). However, the decay rate does still highlight the fact that LPS-c can be viewed as a hybrid of FCFS and PS. In particular, in the case of $\gamma(B) \in (0, \infty)$, which includes all phase-type distributions, the tail asymptotics of LPS-c can vary between the asymptotics of FCFS for small enough c, which is optimal, and the asymptotics of PS as $c \to \infty$, which is pessimal. But, the complexity of (4.3) hides much of the behavior of the decay rate; thus we spend some time in the following sections interpreting and deriving important properties of the decay rate.

4.4.1 Interpreting the decay rate under LPS-c

To build an understanding of (4.3) it is useful to begin by interpreting it in the context of effects (i), (ii), and (iii) described above that could lead to a long delay. Intuitively, effects (i), (ii), and (iii) correspond respectively to the first, second and third term in (4.4). Further, the variable $a \in [0, 1]$ captures the relative contribution of these effects to a large sojourn time. If a is close to 1, then effect (i) dominates; if a is close to 0, then effects (ii) and (iii) dominate; and intermediate values of a represent different combinations of all three effects. The minimization operation in (4.3) indicates that the most dominant combination of effects (i), (ii), and (iii) determines the decay rate, and which combination is dominant depends on c, A, and B. Thus, one should interpret the value of $a_c^* := \arg\min_{a\in[0,1]} f_c(a)$ as providing a description of *how* large sojourn times are caused. Informally, for large x, if the tagged job experiences a sojourn time $V > x$, it is most likely due to (a) a backlog of the order of $a_c^* x$ being present in the system when the job arrives, (b) the tagged job

[1] The condition $\gamma(B) = \infty$ characterizes *very light-tailed* job-size distributions (that have a tail that decays faster than exponentially) and includes all distributions with bounded support.

having a size of the order of $\frac{(1-a_c^*)x}{c}$, and (c) work of the order of $(1-a_c^*)\left(1-\frac{1}{c}\right)x$ entering the queue in the interval $(0,x)$.

4.4.2 Properties of the decay rate under LPS-c

In this section, we focus on the case $\gamma(B) \in (0,\infty)$, and try to provide insight into two questions: *How does the decay rate of LPS-c vary with c? Can we provide a more explicit characterization of a_c^* and, thus, $\gamma(V_{LPS-c})$?* Additionally, we present some numeric examples to illustrate the points in our discussion.

We start by studying the behavior of $\gamma(V_{LPS-c})$ as a function of c. Given the view that LPS-c is a hybrid of FCFS and PS, one expects that the decay rate of LPS-c will transition monotonically between $\gamma(V_{FCFS})$, the optimal decay rate, and $\gamma(V_{PS})$, the pessimal decay rate, as c grows from 1 to ∞. This is indeed what happens; the following lemma establishes the monotonicity of the sojourn time decay rate with respect to c.

Lemma 11. *Consider the GI/GI/1 queue. Assuming $\gamma(B) \in (0,\infty)$, $\gamma(V_{LPS-c})$ is monotonically decreasing in c. Moreover, $\lim_{c\to\infty}\gamma(V_{LPS-c}) = \gamma(V_{PS}) = \gamma_L$.*

Lemma 11 implies that for light-tailed job sizes, the sojourn time tail under LPC-c gets 'heavier' with increasing c. In contrast, for heavy-tailed job sizes, we proved in Section 4.3 that the sojourn time tail gets 'lighter' with increasing c. We prove Lemma 11 in 4.B.4.

Next, we provide a more explicit characterization of a_c^*, and thus $\gamma(V_{LPS-c})$. To accomplish this, we must consider two classes of light-tailed workloads separately: $\gamma_F < \gamma(B)$ and $\gamma_F = \gamma(B)$. Recall the background provided in Section 4.2.4 on the decay rate of the busy period. In light of that discussion, we may rewrite $f_c(\cdot)$ as follows.

$$f_c(a) = a\gamma_F + \frac{(1-a)\gamma(B)}{c} + g\left((1-a)(1-\frac{1}{c})\right).$$

Moreover, $f_c(\cdot)$ is continuously differentiable and convex. Let $f_c^* := \min_{a\in[0,1]} f_c(a)$.

Case 1: $\gamma_F < \gamma(B)$.

Note that this case includes most common light-tailed job size distributions, e.g., all phase-type distributions.[2] To get a more explicit representation of a_c^*, begin by noting that

$$f_c(0) = \frac{\gamma(B)}{c} + g\left(1-\frac{1}{c}\right), \quad f_c(1) = \gamma_F.$$

Next, Lemma 10 allows us to capture the derivative of $f_c(a)$ with respect to a.

$$f_c'(a) = \gamma_F - \frac{\gamma(B)}{c} - \left(1-\frac{1}{c}\right)\hat{s}\left((1-\frac{1}{c})(1-a)\right)$$

$$\Rightarrow \quad f_c'(0) = \gamma_F - \frac{\gamma(B)}{c} - \left(1-\frac{1}{c}\right)\hat{s}\left(1-\frac{1}{c}\right), \quad f_c'(1) = \gamma_F - \frac{\gamma(B)}{c}.$$

So, for $c \le \frac{\gamma(B)}{\gamma_F}$, $f_c'(1) \le 0$, implying $a_c^* = 1$ (recall that $f_c(\cdot)$ is convex) and $\gamma(V_{LPS-c}) = \gamma_F$. Therefore, for small enough c (specifically, $c \le \frac{\gamma(B)}{\gamma_F}$), the decay rate of LPC-c matches that of FCFS. Moreover, long delays are most likely caused by effect (i).

[2] If B is phase-type, then $\gamma(B) \in (0,\infty)$ and $\lim_{s\uparrow\gamma(B)}\Phi_B(s) = \infty$. This implies that $\lim_{s\uparrow\gamma(B)}\Psi(s) = \infty$, which is sufficient to guarantee that $\gamma_F < \gamma(B)$.

Consider now the case $c > \frac{\gamma(B)}{\gamma_F}$. In this case, the function $f_c(a)$ is increasing in a for $a \geq 1$. This means $\gamma(V_{LPS-c}) = \min_{a\geq 0} f_c(a)$. This observation allows us to express the decay rate differently:

$$\begin{aligned}\gamma(V_{LPS-c}) &= \min_{a\in[0,1]}\left[a\gamma_F + \frac{(1-a)\gamma(B)}{c} + \max_{s\geq 0}\left[(1-a)s\left(1-\frac{1}{c}\right) - \Psi(s)\right]\right] \\ &= \min_{a\geq 0}\max_{s\geq 0}\left[a\gamma_F + \frac{(1-a)\gamma(B)}{c} + (1-a)s\left(1-\frac{1}{c}\right) - \Psi(s)\right] \\ &= \frac{\gamma(B)}{c} + \min_{a\geq 0}\max_{s\geq 0}\left[s\left(1-\frac{1}{c}\right) - \Psi(s) - a\left(s\left(1-\frac{1}{c}\right) - \left(\gamma_F - \frac{\gamma(B)}{c}\right)\right)\right].\end{aligned}$$

We may interpret the second term above to be the dual of the convex optimization problem

$$\max_{s\in[0,\kappa_c]}\left[s\left(1-\frac{1}{c}\right) - \Psi(s)\right],$$

where $\kappa_c := \frac{\gamma_F - \frac{\gamma(B)}{c}}{1-\frac{1}{c}} = \gamma_F - \frac{\gamma(B)-\gamma_F}{c-1}$. Since this optimization problem has zero duality gap (see Proposition 5.2.1 in [9]), we can rewrite the sojourn time decay rate as follows.

$$\gamma(V_{LPS-c}) = \frac{\gamma(B)}{c} + \max_{s\in[0,\kappa_c]}\left[s\left(1-\frac{1}{c}\right) - \Psi(s)\right]. \tag{4.5}$$

Note that the above form for the decay rate is more computationally convenient than that in the statement of Theorem 14. Additionally, it allows us to characterize the value of a_c^*; this is summarized in the following lemma.

Lemma 12. *Consider the GI/GI/1 queue. If $\gamma_F < \gamma(B)$, then for $c > \frac{\gamma(B)}{\gamma_F}$, a_c^* is monotonically decreasing in c. Moreover, there exists $\hat{c} > \frac{\gamma(B)}{\gamma_F}$ such that for $c > \hat{c}$,*
(i) $a_c^* = 0$, (ii) $\gamma(V_{LPS-c}) = \frac{\gamma(B)}{c} + g\left(1-\frac{1}{c}\right) > \gamma(V_{PS})$.

From the standpoint of tail-robust scheduling using LPS, which is the focus of Section 4.5, the above lemma has the following important implication: For the class of workload distributions that satisfy $\gamma_F < \gamma(B)$, for all c, *the sojourn time decay rate under LPS-c is strictly better than worst-case* (recall that PS has the smallest possible decay rate). The monotonicity of a_c^* with respect to c implies that as c increases, the contribution of effects (ii) and (iii) to a large sojourn time increases relative to (i). Moreover, for large enough c, large sojourn times are most likely caused by effects (ii) and (iii). Interestingly, for intermediate values of c, it is possible that $a_c^* \in (0,1)$. In this case, the 'bad' event is a combination of all three effects (i)–(iii); see Example 2 below. We give the proof of Lemma 12 in 4.B.5.

To illustrate the properties described above, we consider a couple of examples.

Example 1: Consider first the M/M/1 case; $A \sim Exp(\lambda)$, $B \sim Exp(\mu)$.[3] In this case, $\frac{\gamma(B)}{\gamma_F} = \frac{1}{1-\rho}$. Interestingly, it can be proved that for $c > \frac{\gamma(B)}{\gamma_F}$, $a_c^* = 0$. Figure 4.1 shows $\gamma(V_{LPS-c})$ and a_c^* as a function c for the case $\mu = 1$, $\rho = 0.9$.

Example 2: Next, we consider an M/GI/1 example, where $A \sim Exp(\lambda)$, and B has an Erlang-2 distribution, i.e., $\Phi_B(s) = \left(\frac{\mu}{\mu-s}\right)^2$. Figure 4.2 shows $\gamma(V_{LPS-c})$ and a_c^* as a function c for the case $\mu = 1$, $\rho = 0.9$. Note that for some intermediate values of c, $a_c^* \in (0,1)$.

[3] $A \sim Exp(\lambda)$ means A is exponentially distributed with mean $1/\lambda$.

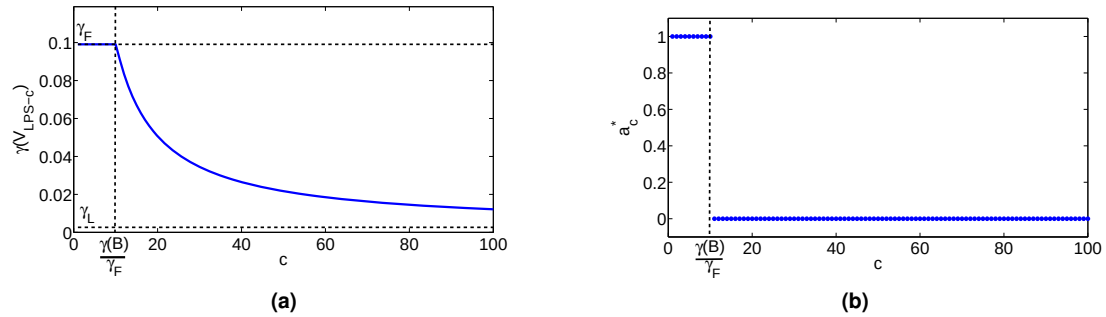


Figure 4.1: *M/M/1 example:* $B \sim Exp(1)$, $\rho = 0.9$

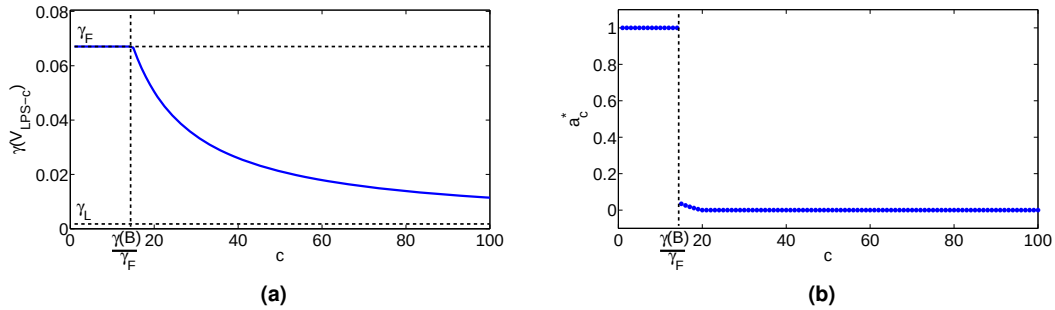


Figure 4.2: *M/Erlang-2/1 example*

Case 2: $\gamma_F = \gamma(B)$**.**

This case behaves fundamentally differently than Case 1 above, however it is easier to characterize. For $c = 1$, it is obvious that $\gamma(V_{LPS-c}) = \gamma_F$ and $a_c^* = 1$. On the other hand, for $c > 1$, $f_c'(0) = \left(1 - \frac{1}{c}\right)\left(\gamma(B) - \hat{s}\left(1 - \frac{1}{c}\right)\right) \geq 0$. This means $a_c^* = 0$ and

$$\gamma(V_{LPS-c}) = f_c(0) = \frac{\gamma(B)}{c} + g\left(1 - \frac{1}{c}\right).$$

Therefore, if $\gamma_F = \gamma(B)$, for $c > 1$, a large sojourn time is most likely caused by a combination of effects (ii) and (iii).

4.5 Designing LPS robustly

Now that we have derived the sojourn time asymptotics in both the light-tailed and heavy-tailed regimes, we can return to the question of designing a scheduling policy that has robust performance across both heavy-tailed and light-tailed job size distributions.

Recall from our discussion in Section 4.2 that there is a dichotomy in prior results showing that scheduling policies that perform optimally in the heavy-tailed regime (e.g., SRPT and PS) have the worst-case sojourn time tail in the light-tailed regime and policies that perform optimally in the light-tailed regime (e.g., FCFS) have the worst-case sojourn time tails in the heavy-tailed regime. In fact, a recent result by Wierman and

Zwart [68] shows that this is a fundamental limit on all work-conserving policies that do not learn the job size distribution. Specifically, no work-conserving, non-learning policy can be optimal under heavy-tailed (light-tailed) job sizes and better than worst-case under light-tailed (heavy-tailed) job sizes.

Further, prior work provides no schedulers that are 'tail-robust', i.e., provide robust performance (even a better than worst-case sojourn time tail) across both light-tailed and heavy-tailed job size distributions. This is problematic because determining whether a workload is heavy-tailed or light-tailed is extremely difficult (if not impossible) and thus designing such an assumption into a scheduler is undesirable. Ideally, a scheduler should provide performance that is robust to such an assumption, and designing such a scheduler is the goal of this section.

We show in this section that by choosing the multiprogramming level c carefully, it is possible to design LPS-c so that it provides 'tail-robust' performance. Our results in Section 4.3 and 4.4 highlight the tension in designing LPS-c robustly. Recall that as c grows the sojourn time tail gets heavier in the light-tailed regime while the sojourn time tail gets lighter in the heavy-tailed regime. Thus, an intermediate value of c must be carefully chosen to provide robustness. Note that c cannot be chosen to be workload independent; indeed, Theorem 13 implies that with regularly varying job sizes, for any fixed c, the sojourn time tail index matches that under FCFS (the worst-case) as load ρ approaches 1. Our designs choose c as a function of ρ. Thus, these policies must learn some information about the workload, but they only need to learn expectations, which can be accomplished quickly and is certainly much easier than learning the tail.

In this section we propose two possible choices for c that both provide tail-robust performance, but balance differently performance in the heavy-tailed and light-tailed regimes.

Design 1. Our first proposed design guarantees better than worst-case performance for a broad class of light-tailed distributions (phase type distributions) and heavy-tailed distributions (regularly varying distributions). Further, it guarantees optimality under large subclasses of both heavy-tailed and light-tailed distributions.

Corollary 2. *Consider the GI/GI/1 queue. Using LPS-c with* $c = \left\lfloor \frac{1}{1-\rho} \right\rfloor + 1$ *ensures that the (logarithmic) asymptotic tail behavior of the stationary sojourn time is*

(i) better than worst-case when the job size distribution is regularly varying with index $\theta > 1$, *i.e.,* $\Gamma(V_{LPS-c}) > \Gamma(V_{FCFS})$.

(ii) better than worst-case when the job size distribution is phase-type, i.e., $\gamma(V_{LPS-c}) > \gamma(V_{PS})$.[4]

(iii) optimal when the job size distribution is regularly varying with index $\theta \geq 2$, *i.e.,* $\Gamma(V_{LPS-c}) = \Gamma(V_{PS})$.

(iv) optimal when the job size distribution is light-tailed and satisfies $\frac{\gamma(B)}{\gamma_F} \geq \left\lfloor \frac{1}{1-\rho} \right\rfloor + 1$ *or* $\gamma(B) = \infty$, *i.e.,* $\gamma(V_{LPS-c}) = \gamma(V_{FCFS})$.

[4]Note that the sojourn time tail is actually better than worst-case for a larger class of light-tailed workloads, including workloads satisfying $\gamma_F < \gamma(B)$.

This corollary follows immediately from the discussion in Sections 4.3 and 4.4 and Lemma 22 (see 4.C).

One important point about Design 1 is that it is 'light-tailed centric', by which we mean that the c is chosen as the smallest possible c that guarantees better than worst-case performance under regularly varying job size distributions. Thus, sojourn time tail is the lightest possible in the light-tailed regime while still maintaining tail-robust performance.

Additionally, a practical remark about Design 1 is that, though it requires learning the ρ, it does not actually require the exact ρ to be learned. If an upper bound on ρ is learned, then points (i) and (ii) remain true, and only the optimality regions change. Thus, providing tail-robustness is possible even with quite inexact estimates of the load.

Finally, it is worth discussing the subclasses of job size distributions where Design 1 provides the optimal sojourn time tail. In the heavy-tailed regime, the sub-class includes all regularly varying distributions with finite variance. In the light-tailed regime, it is more difficult to explicitly describe the subclass. However, it is important to note that the exponential distribution is on the boundary. In particular, for the M/M/1 queue, $\gamma(B)/\gamma_F = \frac{1}{1-\rho}$. Thus, it is impossible for LPS-c to be designed with an optimal sojourn time tail for exponential job size distributions and a better than worst-case sojourn time tail for all regularly varying job size distributions.

Design 2. Our second proposed design for c provides a contrast to Design 1 in that it is 'heavy-tailed centric' instead of 'light-tailed centric'. Specifically, compared to Design 1, Design 2 allows the class of heavy-tailed job size distributions where the sojourn time tail index of LPS-c is optimal to be enlarged, while still maintaining better than worst-case performance under light-tailed job size distributions, but shrinking the class of light-tailed distributions where the sojourn time tail index is optimal.

Corollary 3. *Consider the GI/GI/1 queue. Let $\Theta \in (1, 2]$. Using LPS-c with $c = \left\lfloor \frac{\lceil \frac{\Theta}{\Theta-1} \rceil - 1}{1-\rho} \right\rfloor + 1$ ensures that the (logarithmic) asymptotic tail behavior of the stationary sojourn time is*

(i) better than worst-case when the job size distribution is regularly varying with index $\theta > 1$, i.e., $\Gamma(V_{LPS-c}) > \Gamma(V_{FCFS})$.

(ii) better than worst-case when the job size distribution is phase-type, i.e., $\gamma(V_{LPS-c}) > \gamma(V_{PS})$.[4]

(iii) optimal when the job size distribution is regularly varying with index $\theta \geq \Theta > 1$, i.e., $\Gamma(V_{LPS-c}) = \Gamma(V_{PS})$.

(iv) optimal when the job size distribution is light-tailed and satisfies $\frac{\gamma(B)}{\gamma_F} \geq \left\lfloor \frac{\lceil \frac{\Theta}{\Theta-1} \rceil - 1}{1-\rho} \right\rfloor + 1$ or $\gamma(B) = \infty$, i.e., $\gamma(V_{LPS-c}) = \gamma(V_{FCFS})$.

This corollary follows immediately from the discussion in Sections 4.3 and 4.4 and Lemma 22 (see 4.C).

Design 2 provides a parameter, Θ, that allows the scheduling designer to trade-off between the optimality guarantees in the light-tailed and heavy-tailed regimes, while at all times guaranteeing tail-robustness. Additionally, note that like Design 1, Design 2 is also robust against inexact estimates of ρ: as long as an upper

bound on ρ is used, Design 2 still guarantees properties (i) and (ii), though the subclasses of distributions defined in (iii) and (iv) change depending on the estimation accuracy.

4.6 Concluding remarks

The contributions in this chapter can be viewed along two axes. Firstly, we have derived GI/GI/1 sojourn time tail asymptotics for the LPS-c queue for both heavy-tailed and light-tailed job size distributions. These are the first results characterizing the tail asymptotics of LPS-c, which is an important and practical policy that has received increasing attention in recent years. Secondly, the results about LPS-c illustrate that it is possible to design LPS-c so that it is 'tail-robust', i.e., so that it provides a sojourn time tail that is robust across heavy-tailed and light-tailed job size distributions. Prior to this work, there were no known policies that had better than worst-case sojourn time tails under both heavy-tailed and light-tailed job size distributions. Our results show that by choosing $c = \lfloor 1/(1-\rho) \rfloor + 1$, LPS-$c$ is better than worst-case across large classes of heavy-tailed and light-tailed job size distributions and even is optimal across large subclasses of both heavy-tailed and light-tailed job size distributions.

There are many interesting further research questions that this work motivates along both of the directions described above. First, with respect to the analysis of LPS-c, it would be interesting to extend the asymptotic results presented to the non-work-conserving case where the service rate is a function of the number of jobs in service, as studied in [27]. This case is important because computer systems have overheads that vary as a function of the multiprogramming level, usually in a unimodal fashion, and these variations can have a significant impact in the design of c. We believe the analysis in the current chapter should extend to this case naturally. Second, it would be quite interesting to study the performance of the suggested tail-robust designs for c with respect to other performance metrics, e.g., the expected sojourn time. Finally, with respect to the design of tail-robust schedulers, it should be noted that our results provide the first example of a tail-robust policy, and it would be interesting to understand if there are alternative designs that achieve even better trade-offs between robustness and optimality.

4.A Proofs for results in Section 4.3

The goal of this appendix is to prove Theorem 13, which describes the logarithmic tail asymptotics of V_{LPS-c} with regularly varying job sizes. We prove Theorem 13 by proving matching (asymptotic) lower and upper bound on the tail of D_{LPS-c}. The upper bound is established in Section 4.A.1, and the lower bound is established in Section 4.A.2. Finally, we use these bounds to complete the proof of Theorem 13 in Section 4.A.3.

4.A.1 Upper bound

The upper bound follows immediately from comparing LPS-c with a FCFS GI/GI/c queue, where each server has speed $1/c$. Specifically, let $D_{GI/GI/c}$ denote the stationary delay in the GI/GI/c system. It is easy to see that

$$D_{LPS-c} \leq_{\text{st}} D_{GI/GI/c}. \tag{4.6}$$

Therefore, we can obtain an asymptotic upper bound on the tail of D_{LPS-c} from moment conditions for $D_{GI/GI/c}$, derived in [59]. The following lemma follows directly from (4.6) and Theorem 2.1 in [59].

Lemma 13. *Under Assumption 1, for* $\eta > 1$, $\mathbb{E}\left[B^{\eta}\right] < \infty \Rightarrow \mathbb{E}\left[D_{LPS-c}^{k_c(\eta-1)}\right] < \infty$.

4.A.2 Lower bound

We now prove our asymptotic lower bound on the waiting time tail in a GI/GI/1 LPS-c queue.

Theorem 15. *Under Assumption 1, if* $B_e \in \mathcal{L}$ [5] *and* $\mathbb{E}\left[B^{\eta}\right] < \infty$ *for some* $\eta > 1$, *then*

$$P\left(D_{LPS-c} > x\right) \gtrsim \tau_1 P\left(B_e > \tau_2\, x\right)^k$$

for positive constants τ_1 *and* τ_2.

To prove Theorem 15, we construct a 'bad' event in which a tagged job experiences a large waiting time. Intuitively, the 'bad' event corresponds to k spare slots being filled by 'large' jobs, which causes the queue to become overloaded and build a large backlog before the tagged job arrives.

Assume the tagged job enters the system at time 0. Let D denote the waiting time of the tagged job and let S_{-i}, B_{-i} denote respectively the arrival instant and size of the ith job to enter the system before the tagged job. Before beginning the proof we need a bit of notation. For $z > 0$, denote $B^{(z)} = B\mathbf{1}(B \leq z)$, $\beta^{(z)} = \mathbb{E}\left[B^{(z)}\right]$, $\rho^{(z)} = \frac{\beta^{(z)}}{\alpha}$. Since $\rho > \frac{\lfloor c\rho \rfloor}{c}$, we can find a large enough $y > 0$ and small enough $\epsilon \in (0, \beta^{(y)})$ such that

$$\rho > \rho^{(y)} = \frac{\beta^{(y)}}{\alpha} > \frac{\beta^{(y)} - \epsilon}{\alpha + \epsilon} > \frac{\lfloor c\rho \rfloor}{c}.$$

Further, define, for $x > 0$,

$$m(x) = \left\lceil \frac{\frac{\lfloor c\rho \rfloor}{c} x + y\left[\lfloor c\rho \rfloor - 1\right]_+}{(\beta^{(y)} - \epsilon) - (\alpha + \epsilon)\frac{\lfloor c\rho \rfloor}{c}} \right\rceil =: \lceil \nu_1 x + \nu_2 \rceil.$$

Finally, define the following subsets of $\mathbb{N}^k$. For $m \in \mathbb{N}$,

$$\begin{aligned} \mathcal{N}_1(m) &= \left\{n = (n_1, n_2, \cdots, n_k) \in \mathbb{N}^k \,:\, m < n_1 < n_2 < \cdots < n_k\right\}, \\ \mathcal{N}_2(m) &= \left\{n = (n_1, n_2, \cdots, n_k) \in \mathbb{N}^k \,:\, m \leq n_1 \leq n_2 \leq \cdots \leq n_k\right\}. \end{aligned}$$

Now, we are ready to build up the components of the 'bad' event described above. First, define

$$G(x) = \left(S_{-m(x)} > -m(x)(\alpha + \epsilon)\right) \bigcap \left(\sum_{i=1}^{m(x)} B_{-i}^{(y)} > m(x)(\beta^{(y)} - \epsilon)\right) =: G_1(x) \bigcap G_2(x),$$

[5] $B \in \mathcal{L} \Rightarrow B_e \in \mathcal{L}$, but the converse is not true; see Section 3 of [64].

where $B_{-i}^{(y)} = B_{-i}\mathbf{1}(B_{-i} \leq y)$. Next, for $n \in \mathcal{N}_1(m(x))$, define the event $H_n(x)$ as follows.

$$\begin{aligned} H_n(x) &= \left(S_{-n_i} \in (-n_i(\alpha+\epsilon), -n_i(\alpha-\epsilon)),\ i = 1, 2, \cdots, k\right) \bigcap \\ &\quad \left(B_{-n_i} > x + n_i(\alpha+\epsilon),\ i = 1, 2, \cdots, k\right) \bigcap \\ &\quad \left(B_{-p} \leq x + p(\alpha+\epsilon)\ \forall p \in \{m(x)+1, m(x)+2, \cdots\} \setminus \{n_1, n_2, \cdots, n_k\}\right) \\ &=: H_{n,1}(x) \bigcap H_{n,2}(x) \bigcap H_{n,3}(x). \end{aligned}$$

Note that $H_n(x)$ corresponds to k 'large' jobs entering the system before the tagged job, with indices $-n_i$, $i = 1, 2, \cdots, k$. The job with index $-n_i$ arrives in the interval $(-n_i(\alpha+\epsilon), -n_i(\alpha-\epsilon))$ and has size that exceeds $x + n_i(\alpha+\epsilon)$. This implies that this job must remain in the system till time x. $H_n(x)$ also implies that no other 'large' arrivals occur; this means for $n, n' \in \mathcal{N}_1(m(x))$, the events H_n and $H_{n'}$ are mutually exclusive.

Finally, our 'bad' event is defined as follows: $I(x) = G(x) \bigcap \left(\bigcup_{n \in \mathcal{N}_1(m(x))} H_n(x)\right)$. The following lemma shows that the 'bad' event does indeed cause the tagged job to experience a large delay.

Lemma 14. $I(x) \Rightarrow D > x$.

Proof. Since $I(x) \Rightarrow H_n(x)$ for some $n \in \mathcal{N}_1(m(x))$, $I(x)$ implies k large jobs with indices strictly less than $-m(x)$ remain in the system till time x. This means the $m(x)$ arrivals before the tagged job can receive service at a rate no greater than $\frac{\lfloor c\rho \rfloor}{c}$ till time x. This means that at time x, the work remaining in the system corresponding to the $m(x)$ arrivals before the tagged job from jobs with size $\leq y$ strictly exceeds

$$\begin{aligned} & m(x)(\beta^{(y)} - \epsilon) - \frac{\lfloor c\rho \rfloor}{c}\left(x + m(x)(\alpha+\epsilon)\right) \\ = \quad & m(x)\left((\beta^{(y)} - \epsilon) - \frac{\lfloor c\rho \rfloor}{c}(\alpha+\epsilon)\right) - x\frac{\lfloor c\rho \rfloor}{c} \\ \geq \quad & y\left[\lfloor c\rho \rfloor - 1\right]_+. \end{aligned}$$

This implies that at least $\lfloor c\rho \rfloor$ jobs from among the $m(x)$ arrivals before the tagged job remain in the system at time x, which ensures that tagged job receives no service until time x. □

All that remains is to bound $P\left(I(x)\right)$, and thus $P\left(D > x\right)$:

$$\begin{aligned} P\left(D > x\right) &\geq P\left(I(x)\right) = P\left(G(x) \bigcap \left(\bigcup_{n \in \mathcal{N}_1(m(x))} H_n(x)\right)\right) \\ &= P\left(\bigcup_{n \in \mathcal{N}_1(m(x))} \left(G(x) \bigcap H_n(x)\right)\right) = \sum_{n \in \mathcal{N}_1(m(x))} P\left(G(x) \bigcap H_n(x)\right) \\ &= \sum_{n \in \mathcal{N}_1(m(x))} P\left(G_1(x) \cap G_2(x) \cap H_{n,1}(x)\right) P\left(H_{n,2}(x)\right) P\left(H_{n,3}(x)\right) \\ &\geq \sum_{n \in \mathcal{N}_1(m(x))} P\left(G_1(x) \cap G_2(x) \cap H_{n,1}(x)\right) P\left(H_{n,2}(x)\right) P\left(B_{-p} \leq x + p(\alpha+\epsilon)\ \forall\, p \in \mathbb{N}\right). \end{aligned}$$

Using the weak law of large numbers, we see that the the probability of the events $G_1(x), G_2(x)$, and

$H_{n,1}(x)$ approaches 1 as $x \uparrow \infty$. Therefore, fixing $\delta \in (0,1)$, for large enough x,

$$P\left(G_1(x) \cap G_2(x) \cap H_{n,1}(x)\right) \geq 1-\delta.$$

Also, invoking Lemma 15 (stated and proved below), $P\left(B_{-p} \leq x + p(\alpha+\epsilon)\ \forall\, p \in \mathbb{N}\right) \geq \phi > 0$ for large enough x. Therefore, for large enough x,

$$P\left(D > x\right) \geq \phi(1-\delta) \sum_{n \in \mathcal{N}_1(m(x))} \prod_{i=1}^{k} P\left(B > x + n_i(\alpha+\epsilon)\right). \tag{4.7}$$

Define the bijection $\xi : \mathcal{N}_1(m(x)) \to \mathcal{N}_2(m(x)+k)$ as follows.

$$\xi((n_1, n_2, \cdots, n_k)) = (n_1 + k - 1, n_2 + k - 2, \cdots, n_k).$$

From (4.7),

$$\begin{aligned}
\mathbb{P}(D > x) &\geq \phi(1-\delta) \sum_{n \in \mathcal{N}_2(m(x)+k)} \prod_{i=1}^{k} P\left(B > x + n_i(\alpha+\epsilon)\right) \\
&\geq \frac{\phi(1-\delta)}{k!} \sum_{n \in \mathbb{N}^k : n_i \geq m(x)+k} \prod_{i=1}^{k} P\left(B > x + n_i(\alpha+\epsilon)\right) \\
&= \frac{\phi(1-\delta)}{k!} \left(\sum_{n_1 \geq m(x)+k} P\left(B > x + n_1(\alpha+\epsilon)\right) \right)^k \\
&\geq \frac{\phi(1-\delta)\beta^k}{(\alpha+\epsilon)^k k!} \left(P\left(B_e > x + (\alpha+\epsilon)(m(x)+k)\right)\right)^k.
\end{aligned}$$

The last inequality above uses the fact that for $\tilde{\alpha}, x > 0$, $\frac{\tilde{\alpha}}{\beta} \sum_{i=0}^{\infty} P\left(B > x + i\tilde{\alpha}\right) \geq P\left(B_e > x\right)$. Finally, since $B_e \in \mathcal{L}$, it is easy to see that

$$\begin{aligned}
&P\left(B_e > x + (\alpha+\epsilon)(m(x)+k)\right) \sim P\left(B_e > (1+\nu_1(\alpha+\epsilon))x\right) \\
&\Rightarrow P\left(D > x\right) \gtrsim \tfrac{\phi(1-\delta)\beta^k}{(\alpha+\epsilon)^k k!} \left(P\left(B_e > (1+\nu_1(\alpha+\epsilon))x\right)\right)^k.
\end{aligned}$$

This completes the proof.

Lemma 15. *Assume B is a non-negative random variable satisfying $\mathbb{E}\left[B^{1+\delta}\right] < \infty$ for some $\delta > 0$. Then, for $\tilde{b} > 0$ satisfying $F_B(\tilde{b}) > 0$ and $\tilde{a} > 0$,*

$$\prod_{i=1}^{\infty} F_B(\tilde{b} + i\tilde{a}) = \phi(\tilde{b}, \tilde{a}) > 0.$$

Proof. Define $\tilde{B} = \max\{\frac{B-\tilde{b}}{\tilde{a}}, 0\}$. Clearly, $\mathbb{E}\left[\tilde{B}^{1+\delta}\right] < \infty$ and for $x > 0$, $F_{\tilde{B}}(x) = F_B(\tilde{b} + x\tilde{a})$.

Pick $\delta_1 \in (0, \delta)$, $\epsilon > 0$. Since $\mathbb{E}\left[\tilde{B}^{1+\delta}\right] < \infty$, for large enough x,

$$F_{\tilde{B}}(x) \geq 1 - \frac{1}{x^{1+\delta_1}} \geq \exp\left\{-\frac{(1+\epsilon)}{x^{1+\delta_1}}\right\}$$

The last inequality holds since, for small enough $y > 0$, $1 - y \geq e^{-(1+\epsilon)y}$. Let us say this inequality holds

for $x \geq x_0$.

$$\prod_{i=1}^{\infty} F_B(\tilde{b}+i\tilde{a}) = \prod_{i=1}^{\infty} F_{\tilde{B}}(i) \geq \prod_{i<\lceil x_0 \rceil} F_{\tilde{B}}(i) \exp\left\{-(1+\epsilon)\sum_{i=\lceil x_0 \rceil}^{\infty} \frac{1}{i^{1+\delta_1}}\right\} > 0.$$

□

4.A.3 Proof of Theorem 13

We can now complete the proof of Theorem 13 by combining the upper and lower bounds derived in the preceding sections. Assume that $B \in \mathcal{RV}(\theta)$, $\theta > 1$. Fix $\delta \in (0,1)$. Invoking Theorem 15, for large enough x,

$$\begin{aligned} P\left(D_{LPS-c} > x\right) &\geq (1-\delta)\tau_1 P\left(B_e > \tau_2 x\right)^k \\ \Rightarrow \limsup_{x\to\infty} -\frac{\log P\left(D_{LPS-c} > x\right)}{\log(x)} &\leq k \lim_{x\to\infty} -\frac{\log P\left(B_e > \tau_2 x\right)}{\log(x)} = k(\theta-1). \end{aligned} \tag{4.8}$$

For $\eta \in (1,\theta)$, $\mathbb{E}\left[B^\eta\right] < \infty$. Invoking Lemma 13, we conclude that $\mathbb{E}\left[D_{LPS-c}^{k(\eta-1)}\right] < \infty$. Using the Markov inequality,

$$P\left(D_{LPS-c} > x\right) \leq \mathbb{E}\left[D_{LPS-c}^{k(\eta-1)}\right] x^{-k(\eta-1)} \Rightarrow \liminf_{x\to\infty} -\frac{\log P\left(D_{LPS-c} > x\right)}{\log(x)} \geq k(\eta-1).$$

Letting $\eta \uparrow \theta$, we get

$$\liminf_{x\to\infty} -\frac{\log P\left(D_{LPS-c} > x\right)}{\log(x)} \geq k(\theta-1). \tag{4.9}$$

From (4.8) and (4.9), it follows that $\Gamma(D_{LPS-c}) = k(\theta-1)$.

Finally, note that

$$D_{LPS-c} + B \leq_{\text{st}} V_{LPS-c} \leq_{\text{st}} D_{LPS-c} + Bc. \tag{4.10}$$

For non-negative random variables X and Y with tail index $\Gamma(X)$ and $\Gamma(Y)$, respectively, it is easy to show that $\Gamma(X+Y) = \min\{\Gamma(X), \Gamma(Y)\}$. Returning to (4.10), since

$$\Gamma(D_{LPS-c} + B) = \Gamma(D_{LPS-c} + B) = \min\{k(\theta-1), \theta\},$$

it follows easily that $\Gamma(V_{LPS-c}) = \min\{\Gamma(X), \Gamma(Y)\}$. This completes the proof.

4.B Proofs for results in Section 4.4

In this appendix, we prove the results stated in Section 4.4. The first three sections are devoted to proving Theorem 14, which describes the decay rate V_{LPS-c} with light-tailed job sizes. The proof for the case $\gamma(B) \in (0,\infty)$ is completed by establishing matching (asymptotic) lower and upper bounds on the tail of $\gamma(V_{LPS-c})$; this is done in 4.B.1 and 4.B.2, respectively. The proof for the case $\gamma(B) = \infty$ is given in 4.B.3. In 4.B.4, we prove Lemma 11, which establishes the monotonicity of $\gamma(V_{LPS-c})$ with respect to c. Finally, we give the proof of Lemma 12 (which describes properties of $\gamma(V_{LPS-c})$) in 4.B.5.

4.B.1 Proof of Theorem 14: Lower bound for the case $\gamma(B) \in (0, \infty)$

In this section, we prove the following (asymptotic) lower bound on the tail of V_{LPS-c}.

Lemma 16. *Assuming* $\gamma(B) \in (0, \infty)$, $\liminf_{x\to\infty} \frac{1}{x} \log P\left(V_{LPS-c} > x\right) \geq -\min_{a\in[0,1]} f_c(a)$.

To begin the proof, note that since $f_c(\cdot)$ is continuous over $[0, 1]$, it suffices to prove that, for $a \in (0, 1)$, $\liminf_{x\to\infty} \frac{1}{x} \log P\left(V_{LPS-c} > x\right) \geq -f_c(a)$. Fix $a \in (0, 1)$. Intuitively, to prove the above statement, we construct a 'bad' event where the waiting time of a 'tagged' job exceeds ax and its residence time exceeds $(1 - a)x$. Recall that we assume the tagged job has size B_0 and enters the (stationary) system at time 0. We denote the sojourn time of the tagged job by V.

We use a truncation-based argument. For $z > 0$, define $B^{(z)} = B\mathbf{1}(B \leq z)$. Pick $y > 0$ large enough so that $P\left(B^{(y)} > A\right) > 0$. Consider a 'truncated' system in which, except for the tagged job, only jobs with size less than or equal to y are allowed to enter the system. Denote the total backlog in this 'truncated' system just before the arrival of the tagged job by $W^{(y)}$. The total work entering the 'truncated' system in the time interval $(0, u]$ is denoted by $A^{(y)}(u)$, i.e.,

$$A^{(y)}(u) = \sum_{i=1}^{N(u)} B_i \mathbf{1}(B_i \leq y).$$

For large x, small $\epsilon > 0$, consider the following event.

$$I(x) := \left(W^{(y)} > ax + (c-1)y\right) \cap (A_1 \leq \alpha) \cap \left(A^{(y)}(u) > (1-a)(1-\frac{1}{c})(1+\epsilon)(u - A_1)\ \forall\ u \in (A_1, x)\right)$$
$$:= I_1(x) \cap I_2 \cap I_3(x).$$

At the instant the tagged job begins service, the maximum work remaining in the system corresponding to arrivals before time 0 is $(c - 1)y$. Therefore, the event $W^{(y)} > ax + (c - 1)y$ (and therefore event $I(x)$) implies the tagged job and subsequent arrivals wait for at least time ax before beginning service. Moreover, at any time instant $u \in (ax, x)$, under $I(x)$, the remaining work in the system corresponding to arrivals after time 0 exceeds

$$(1-a)\left(\frac{c-1}{c}\right)(1+\epsilon)(u-\alpha) - (u-ax)\left(\frac{c-1}{c}\right)$$
$$> (1-a)\left(\frac{c-1}{c}\right)(1+\epsilon)(u-\alpha) - (u-au)\left(\frac{c-1}{c}\right) = \epsilon(1-a)\left(\frac{c-1}{c}\right)u - \alpha(1-a)\left(\frac{c-1}{c}\right)(1+\epsilon)$$
$$> \epsilon(1-a)\left(\frac{c-1}{c}\right)ax - \alpha(1-a)\left(\frac{c-1}{c}\right)(1+\epsilon) := \nu_1 x - \nu_2.$$

Therefore, the number of jobs that arrived after time 0 and are still in the system at time u exceeds $\frac{(\nu_1 x - \nu_2)}{y} > c - 1$ for large enough x. Therefore, under event $I(x)$, the tagged job gets no service until time ax and gets (at most) service at rate $1/c$ in the interval (ax, x). Therefore,

$$\left(B_0 > \frac{(1-a)x}{c}\right) \cap I(x) \Rightarrow V^{(y)} > x,$$

where $V^{(y)}$ denotes the sojourn time of the tagged job in the 'truncated' system. Since $V^{(y)} \leq_{\text{st}} V$, for large

enough x,

$$P\left(V>x\right) \geq P\left(V^{(y)}>x\right)$$
$$\geq P\left(B_0>\frac{(1-a)x}{c} \cap I(x)\right) = P\left(B_0>\frac{(1-a)x}{c}\right) P\left(I_1(x)\right) P\left(I_2\right) P\left(I_3(x)|I_2\right).$$

At this point, we note that $P\left(I_3(x)|I_2\right) \geq P\left(Z^{(y)}_{(1-a)\left(1-\frac{1}{c}\right)(1+\epsilon)}>x\right)$, where $Z^{(y)}_{(1-a)\left(1-\frac{1}{c}\right)(1+\epsilon)}$ denotes a busy period in a GI/GI/1 queue, with interarrival times A, job sizes $B^{(y)}$ and server speed $(1-a)\left(1-\frac{1}{c}\right)(1+\epsilon)$. Define $\Psi^{(y)}(s) := -\Phi_A^{-1}\left(\frac{1}{\Phi_{B^{(y)}}(s)}\right)$. Noting that

$$\lim_{x\to\infty}\frac{\log P\left(Z^{(y)}_{(1-a)\left(1-\frac{1}{c}\right)(1+\epsilon)}>x\right)}{x} = -\sup_{s\geq 0}\left[s(1-\frac{1}{c})(1-a)(1+\epsilon)-\Psi^{(y)}(s)\right],$$

we have

$$\liminf_{x\to\infty}\frac{1}{x}\log P\left(V>x\right) \geq -\left(\frac{\gamma_F^{(y)}}{a}+\frac{(1-a)\gamma(B)}{c}+\sup_{s\geq 0}\left[s(1-\frac{1}{c})(1-a)(1+\epsilon)-\Psi^{(y)}(s)\right]\right),$$

where $\gamma_F^{(y)} := \sup\{\theta : \Psi^{(y)}(\theta)-\theta\leq 0\}$. The proof is completed by letting $y\uparrow\infty$, $\epsilon\downarrow 0$. It can be shown that

$$\lim_{y\uparrow\infty}\gamma_F^{(y)} = \gamma_F, \tag{4.11}$$
$$\lim_{\epsilon\downarrow 0}\limsup_{y\uparrow\infty}\sup_{s\geq 0}\left[s(1-\frac{1}{c})(1-a)(1+\epsilon)-\Psi^{(y)}(s)\right] = \sup_{s\geq 0}\left[s(1-\frac{1}{c})(1-a)-\Psi(s)\right]. \tag{4.12}$$

(4.11) and (4.12) can be proved by mimicking the arguments in the proofs of Propositions 2.1 and 2.2, respectively, in [53]. We omit these proofs here.

4.B.2 Proof of Theorem 14: Upper bound for the case $\gamma(B)\in(0,\infty)$

The following lemma gives us a matching asymptotic upper bound on the tail of V_{LPS-c}.

Lemma 17. *Assuming* $\gamma(B)\in(0,\infty)$, $\limsup_{x\to\infty}\frac{1}{x}\log P\left(V_{LPS-c}>x\right)\leq -\min_{a\in[0,1]} f_c(a)$.

To begin the proof, denote the sojourn time of the tagged job by V. Then, we have the following upper bound for $P\left(V>x\right)$.

$$P\left(V>x\right)\leq P\left(W+A(x)+B>x, W+Bc>x\right) = P\left(W+\min\{A(x)+B, Bc\}>x\right).$$

Since W is independent of $\min\{A(x)+B, Bc\}$, and $P\left(W>x\right)\leq e^{-\gamma_F x}$ (this was proved by Kingman [36]), we can construct a random variable $\tilde{W}$ independent of $\min\{A(x)+B, Bc\}$ satisfying (i) $W\leq_{\text{a.s.}}\tilde{W}$,

(ii) $\tilde{W} \sim Exp(\gamma_F)$. Pick $\epsilon \in (0,1)$. For $x > 0$,

$$\begin{aligned} P(V > x) &\leq P\left(\tilde{W} + \min\{A(x) + B, Bc\} > x\right) \\ &\leq P\left(\tilde{W} > (1-\epsilon)x\right) + \int_{y=0}^{(1-\epsilon)x} P(\min\{Bc, A(x)+B\} > x - y)\, dF_{\tilde{W}}(y) \\ &\leq P\left(\tilde{W} > (1-\epsilon)x\right) + x\gamma_F \int_{a=0}^{1-\epsilon} P(\min\{Bc, A(x)+B\} > (1-a)x)\, e^{-a\gamma_F x} da. \end{aligned} \quad (4.13)$$

To continue, we apply the following Lemma, which we prove later.

Lemma 18. *Assume that $\gamma(B) \in (0,\infty)$. Given $\epsilon \in (0,1)$, there exists $x_0 > 0$ and a function $\eta(x) \in o(1)$ such that for all $b \in [\epsilon, 1]$, $x \geq x_0$,*

$$\frac{1}{x}\log P(\min\{A(x)+B, Bc\} > bx) \leq -\left\{\frac{b\gamma(B)}{c} + \sup_{s\geq 0}\left[bs\left(1-\frac{1}{c}\right) - \Psi(s)\right] + \eta(x)\right\}.$$

Invoking Lemma 18, it follows that there exists $x_0 > 0$ and a function $\eta(x) \in o(1)$ such that

$$P(\min\{Bc, A(x)+B\} > (1-a)x) \leq e^{-x\left\{\frac{(1-a)\gamma(B)}{c} + \sup_{s\geq 0}\left[(1-a)s\left(1-\frac{1}{c}\right) - \Psi(s)\right] + \eta(x)\right\}}$$

for all $a \in [0, 1-\epsilon]$, $x > x_0$. Substituting the above bound in (4.13), we conclude that for large enough x,

$$\begin{aligned} P(V > x) &\leq P\left(\tilde{W} > (1-\epsilon)x\right) + x\gamma_F e^{-x\eta(x)} \int_{a=0}^{1-\epsilon} e^{-xf_c(a)} da \\ &\leq e^{-\gamma_F(1-\epsilon)x} + x\gamma_F e^{-x\eta(x)} e^{-xf_c^*}, \end{aligned}$$

where $f_c^* = \min_{a\in[0,1]} f_c(a)$. This implies

$$\limsup_{x\to\infty} \frac{1}{x}\log P(V > x) \leq -\min\{f_c^*, \gamma_F(1-\epsilon)\}.$$

Letting $\epsilon \downarrow 0$ and noting that $f_c(1) = \gamma_F$, we obtain the desired result. To complete the proof, we need to prove Lemma 18. The proof of Lemma 18 depends on the following lemmas.

Lemma 19. *Assume that $\gamma(B) \in (0,\infty)$. For $s \geq 0$ satisfying $\mathbb{E}\left[e^{sB}\right] < \infty$,*

$$\lim_{x\to\infty} \frac{1}{x}\log \mathbb{E}\left[e^{s(B-x)} | B > x\right] = 0.$$

Proof. Pick $s \geq 0$ satisfying $\mathbb{E}\left[e^{sB}\right] < \infty$. Clearly, $s \leq \gamma(B)$.

$$\begin{aligned} \mathbb{E}\left[e^{sB}\right] &\geq P(B > x)\,\mathbb{E}\left[e^{sB} | B > x\right] = e^{-x(\gamma(B)-s+o(1))}\mathbb{E}\left[e^{s(B-x)} | B > x\right]. \\ \Rightarrow \frac{1}{x}\log \mathbb{E}\left[e^{sB}\right] &\geq -(\gamma(B) - s + o(1)) + \frac{1}{x}\log \mathbb{E}\left[e^{s(B-x)} | B > x\right]. \end{aligned}$$

Taking limits as $x \to \infty$, we obtain $\limsup_{x\to\infty} \frac{1}{x}\log \mathbb{E}\left[e^{s(B-x)} | B > x\right] \leq \gamma(B) - s$. Pick $\tilde{s} \geq s$ satisfying $\mathbb{E}\left[e^{\tilde{s}B}\right] < \infty$.

$$\limsup_{x\to\infty} \frac{1}{x}\log \mathbb{E}\left[e^{s(B-x)} | B > x\right] \leq \limsup_{x\to\infty} \frac{1}{x}\log \mathbb{E}\left[e^{\tilde{s}(B-x)} | B > x\right] \leq \gamma(B) - \tilde{s}.$$

The proof is completed by letting $\tilde{s} \uparrow \gamma(B)$. □

Lemma 20. *Suppose that function $\varphi(x)$ satisfies $\limsup_{x\to\infty} \varphi(x) = \omega \in \mathbb{R}$. Given $b_0 > 0$, there exists $x_0 > 0$ and a function $\eta(x) \in o(1)$ such that for all $b \geq b_0$, $x \geq x_0$, $\varphi(bx) \leq \omega + \eta(x)$.*

This lemma is easy to prove, so the proof is omitted. We are now ready to prove Lemma 18.

Proof of Lemma 18. Recall that for $r \geq 0$, $\hat{s}(r) := \arg\max_{s \geq 0} [rs - \Psi(s)]$. Since $\hat{s}(r)$ is increasing in r, and $b \leq 1$, we may restate the inequality stated in the lemma as follows.

$$\frac{1}{x} \log P\left(\min\{A(x) + B, Bc\} > bx\right) \leq -\left\{ \frac{b\gamma(B)}{c} + \sup_{s \in [0, \hat{s}(1)]} \left[bs\left(1 - \frac{1}{c}\right) - \Psi(s) \right] + \eta(x) \right\}.$$

We now prove that there exists $x_0 > 0$ and a function $\eta(x) \in o(1)$ such that for all $b \in [\epsilon, 1]$, $x \geq x_0$, the above inequality holds.

$$\begin{aligned} \log P\left(\min\{A(x) + B, Bc\} > bx\right) &= \log P\left(B > \frac{bx}{c}, A(x) + B > bx\right) \\ &= \log P\left(B > \frac{bx}{c}\right) + \log P\left(A(x) + B > bx | B > \frac{bx}{c}\right). \end{aligned}$$

For $s \geq 0$, we can use the Chernoff bound to bound the second term in the expression above.

$$\begin{aligned} \log P\left(\min\{A(x) + B, Bc\} > bx\right) &\leq \log P\left(B > \frac{bx}{c}\right) + \log \mathbb{E}\left[e^{s(A(x)+B-bx)} | B > \frac{bx}{c}\right] \\ &= \log P\left(B > \frac{bx}{c}\right) + \log \mathbb{E}\left[e^{s(B - \frac{bx}{c})} | B > \frac{bx}{c}\right] + \log \mathbb{E}\left[e^{sA(x)}\right] - sbx\left(1 - \frac{1}{c}\right). \end{aligned} \quad (4.14)$$

We now use Lemma 20 to bound the first two terms of the expression above.

1. Since $\lim_{x \to \infty} \frac{\log P(B > x)}{x} = -\gamma(B)$, there exists $x_1 > 0$, and $\eta_1(x) \in o(1)$ such that for all $b \geq \epsilon$, $x \geq x_1$,
$$\log P\left(B > \frac{bx}{c}\right) \leq \frac{bx}{c}\left(-\gamma(B) + \eta_1(x)\right). \quad (4.15)$$

2. Since $\mathbb{E}\left[e^{\hat{s}(1)B}\right] < \infty$, we know from Lemma 19 that $\lim_{x \to \infty} \frac{1}{x} \log \mathbb{E}\left[e^{\hat{s}(1)(B-x)} | B > x\right] = 0$. Therefore, there exists $x_2 > 0$, and $\eta_2(x) \in o(1)$ such that for all $b \geq \epsilon$, $x \geq x_2$,
$$\log \mathbb{E}\left[e^{\hat{s}(1)(B - \frac{bx}{c})} | B > \frac{bx}{c}\right] \leq \frac{bx}{c}\eta_2(x).$$
This in turn implies that for all $s \in [0, \hat{s}(1)]$, $b \geq \epsilon$, $x \geq x_2$,
$$\log \mathbb{E}\left[e^{s(B - \frac{bx}{c})} | B > \frac{bx}{c}\right] \leq \frac{bx}{c}\eta_2(x). \quad (4.16)$$

Finally, invoking Lemma 9, we note that

$$\log \mathbb{E}\left[e^{sA(x)}\right] = x(\Psi(x) + \eta_3(x)), \quad (4.17)$$

where $\eta_3(x) \in o(1)$. Substituting (4.15), (4.16), and (4.17) into (4.14), we obtain that for $s \in [0, \hat{s}(1)]$, $b \geq \epsilon$, $x \geq x_0$,

$$\frac{\log P\left(\min\{A(x) + B, Bc\} > bx\right)}{x} \leq -\left\{ \frac{b\gamma(B)}{c} + bs\left(1 - \frac{1}{c}\right) - \Psi(s) + \eta(x) \right\},$$

where $x_0 = \max\{x_1, x_2\}$ and $\eta(x) = -\frac{b\eta_1(x)}{c} - \frac{b\eta_2(x)}{c} - \eta_3(x)$. Tightening the bound with respect to s, we

conclude that

$$\frac{\log P\left(\min\{A(x)+B, Bc\} > bx\right)}{x} \leq -\left\{\frac{b\gamma(B)}{c} + \sup_{s\in[0,\hat{s}(1)]}\left[bs\left(1-\frac{1}{c}\right) - \Psi(s)\right] + \eta(x)\right\}$$

for all $b \geq \epsilon$, $x \geq x_0$. This completes the proof. □

This completes the proof of the asymptotic upper bound.

4.B.3 Proof of Theorem 14: The case of $\gamma(B) = \infty$

We now characterize the sojourn time decay rate under LPS-c for the case $\gamma(B) = \infty$.

Lemma 21. *If* $\gamma(B) = \infty$, *then* $\gamma(V_{LPS-c}) = \gamma(V_{FCFS})$.

We prove the lemma by constructing matching upper and lower asymptotic bounds on the sojourn time tail. Let V denote the sojourn time of our tagged job. We start with the upper bound. Since $V \leq_{\text{st}} W + B_0 c$, where W and B_0 are independent,

$$\limsup_{x\to\infty} \frac{\log P\left(V > x\right)}{x} \leq -\gamma(W + B_0 c) = -\gamma(W) = -\gamma_F.$$

The last step above is based on the fact that if X and Y are independent random variables with decay rates $\gamma(X)$ and $\gamma(Y)$, respectively, then $\gamma(X+Y) = \min\{\gamma(X), \gamma(Y)\}$. To obtain the lower bound, we use the truncation argument used in 4.B.1. Reusing the notation developed there, the event $W^{(y)} > x + y(c-1) \Rightarrow V^{(y)} > x$. Therefore,

$$P\left(V > x\right) \geq P\left(V^{(y)} > x\right) \geq P\left(W^{(y)} > x + y(c-1)\right)$$

$$\Rightarrow \liminf_{x\to\infty} \frac{\log P\left(V > x\right)}{x} \geq -\gamma_F^{(y)} \quad \Rightarrow \quad \liminf_{x\to\infty} \frac{\log P\left(V > x\right)}{x} \geq -\gamma_F.$$

The last step uses the fact that $\lim_{y\to\infty} \gamma_F^{(y)} = \gamma_F$.

4.B.4 Proof of Lemma 11

Proof. To prove monotonicity, we prove that for all $a \in [0,1)$, $f_c(a)$ is monotone decreasing in c. To do this, we replace $\frac{1}{c}$ by a continuous parameter $\nu \in (0,1]$ in the definition of $f_c(a)$ and observe that $\frac{\partial f_c(a)}{\partial \nu} \geq 0$. Indeed,

$$\frac{\partial}{\partial \nu}\left[a\gamma_F + (1-a)\nu\gamma(B) + g\left((1-a)(1-\nu)\right)\right] = (1-a)\left(\gamma(B) - \hat{s}\left((1-a)(1-\nu)\right)\right) \geq 0.$$

Since f_c^* is monotonically decreasing in c, the limit $f^* := \lim_{c\to\infty} f_c^*$ exists. $f_c^* \geq \gamma_L$ (since LPC-c is work conserving); this implies $f^* \geq \gamma_L$. To prove the reverse inequality, we note that $f_c^* \leq f_c(0)$ and that $\lim_{c\to\infty} f_c(0) = \gamma_L$. This implies that $f_c^* \leq \gamma_L$, completing the proof. □

4.B.5 Proof of Lemma 12

Consider the expression for the LPS decay rate given by (4.5). Defining

$s_c^* = \arg\max_{s\in[0,\kappa_c]}\left[s\left(1-\frac{1}{c}\right) - \Psi(s)\right]$, we see that $s_c^* = \min\{\kappa_c, \hat{s}\left(1-\frac{1}{c}\right)\}$. It is easy to see that s_c^*

is monotonically increasing in c. Using KKT conditions, we get $a_c^* = 1 - \frac{\Psi'(s_c^*)}{1-\frac{1}{c}}$, which implies that a_c^* is monotonically decreasing with respect to c.

If $\gamma_F < \gamma(B)$, then $0 < \hat{s}(1) < \gamma_F < \gamma(B)$. Define $\hat{c} := \frac{\gamma(B)-\hat{s}(1)}{\gamma_F-\hat{s}(1)}$. Since $\hat{s}(1) \geq \hat{s}\left(1-\frac{1}{c}\right)$, it is easy to show that

$$c > \hat{c} \Rightarrow \kappa_c > \hat{s}(1) \Rightarrow \kappa_c > \hat{s}\left(1-\frac{1}{c}\right) \Rightarrow s_c^* = \hat{s}\left(1-\frac{1}{c}\right).$$

Since $\Psi'\left(\hat{s}\left(1-\frac{1}{c}\right)\right) = 1-\frac{1}{c}$, we conclude that for $c > \hat{c}$, $a_c^* = 0$ and $\gamma(V_{LPS-c}) = f_c(0)$. Moreover, from the proof of Lemma 11, it follows that if $\gamma(B) > \hat{s}(1)$, then $f_c(0)$ is strictly monotonically decreasing with respect to c. This, along with $\lim_{c\to\infty} f_c(0) = \gamma_L$ implies $f_c(0) > \gamma_L$ for all c.

4.C Proof of Corollaries 2 and 3 in Section 4.5

This section states and proves Lemma 22, which is used in the proofs of Corollaries 2 and 3 in Section 4.5. To state Lemma 22, we need the following notation. For $i > 1$, define $\tilde{c}(i,\rho)$ as the smallest multiprogramming level c such that we have at least i 'spare slots' under LPS-c under regularly varying job sizes, i.e., $\tilde{c}(i,\rho) := \min\{c \in \mathbb{N} \mid \lfloor c\rho \rfloor < c\rho,\ k_c \geq i\}$.

Lemma 22.

$$\tilde{c}(i,\rho) = \left\lfloor \frac{i-1}{1-\rho} \right\rfloor + 1. \tag{4.18}$$

Proof. Assuming $\lfloor c\rho \rfloor < c\rho$, let us first show that

$$k_c \geq i \quad \Longleftrightarrow \quad c > \frac{i-1}{1-\rho}. \tag{4.19}$$

First, we see that $\lfloor c\rho \rfloor = \lfloor c - c(1-\rho) \rfloor = c - \lceil c(1-\rho) \rceil$. This means $k_c = \lceil c(1-\rho) \rceil$, which implies (4.19). From (4.19), it is clear that

$$\tilde{c}(i,\rho) = \min\{c \in \mathbb{N} \mid c > \frac{i-1}{1-\rho},\ c\rho \text{ is not an integer}\};$$

it is easy to verify that this condition implies (4.18). □

Chapter 5

When Heavy-Tailed and Light-Tailed Flows Compete: Response Time Tail Under Generalized Max-Weight Scheduling

5.1 Introduction

In the previous two chapters, we considered the problem of designing robust scheduling policies that guarantee good response time tail behavior for both heavy-tailed and light-tailed workloads. In the present chapter, we consider a scenario in which heavy-tailed and light-tailed workloads interact. This is motivated by the observation that in communication networks, certain bursty traffic flows are best modeled using heavy-tailed distributions, while others are best modeled using light-tailed distributions. Such distinctly asymmetric flows naturally interact with one another when sharing the communication resources of the network. In this chapter, we study the problem of scheduling for good response time tail behavior in such a setting.

Specifically, we consider a system consisting of two queues contending for service from a single server. One of the queues is fed by a heavy-tailed traffic flow, whereas the other is fed by a light-tailed traffic flow. The queues experience a time varying connectivity with the server, and the server can serve a single packet from a connected queue in each slot. This captures a wireless uplink/downlink scenario with two nodes communicating with an access point or base station via fading channels. In this setting, our scheduling design goal is that each traffic flow must experience good response time tail behavior. Additionally, we seek scheduling policies that are throughput optimal, i.e., the policy must stabilize the queueing system over the largest possible set of arrival rates.

In the context of wireless networks, the most well studied throughput optimal scheduling policy is the max-weight policy proposed by Tassiulas et al. [66, 67]. In our setting, we show that the max-weight policy, which serves the longest connected queue in each slot, causes the light-tailed flow to experience heavy-tailed delays. This is because the max-weight algorithm throttles the light-tailed flow when the heavy-tailed flow

generates its (frequent) large bursts.

One way of ensuring that the light-tailed flow experiences light-tailed response times is, of course, to schedule it with a strict priority over the heavy-tailed flow. However, the main drawback of giving strict priority to the light-tailed flow is that the corresponding scheduling policy is not throughput optimal. This seems to suggest a tradeoff between throughput optimality and good response time tail performance for the light-tailed flow.

Our main contribution is to show that it is indeed possible to design a throughput optimal scheduling policy that guarantees light-tailed response times for the light-tailed flow. Our design entails a careful choice of *inter-queue* scheduling policy, as well as *intra-queue* scheduling policies. The inter-queue scheduling policy determines which queue to serve in each slot, given the current queue lengths and connectivity state, whereas the intra-queue scheduling policies specify which waiting packet to serve from the queue selected for service by the inter-queue scheduling policy. We consider inter-queue scheduling policies from a class of generalized max-weight policies, which guarantee throughput optimality, while providing a relative priority to the light-tailed flow. Our analysis highlights *how much* relative priority the inter-queue policy needs to award to the light-tailed flow to make light-tailed response times possible. Importantly, our results suggest that the response time tail of the heavy-tailed flow remains unaffected in this process; we prove this formally for the special case in which both queues are always connected to the server. Additionally, our analysis reveals that the correct choice of intra-queue scheduling policies is crucial in order to obtain good response time tail behavior.

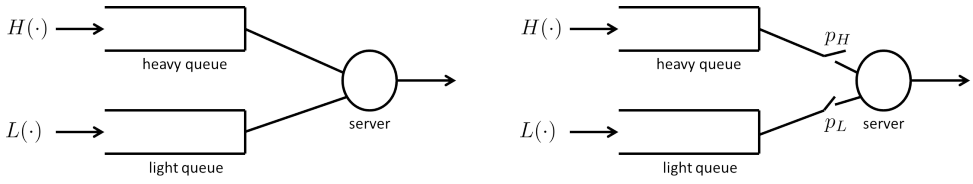
Our work builds on recent work by Markakis et al. [42] and Jagannathan et al. [31, 32]; our model is borrowed from these papers. The focus of these papers is on *queue length* asymptotics under different throughput optimal policies from the class of generalized max-weight policies. In contrast, in this chapter, we analyze the distribution of *response times* experienced by the heavy-tailed and the light-tailed flow. The motivation for studying the distribution of response times is twofold. Firstly, from the standpoint of the applications sending/receiving information, the response time is a more relevant performance metric than the queue length. Secondly, an analysis of response times brings into focus the effect of the *intra-queue* scheduling policies. Note that the evolution of queue lengths is insensitive to the intra-queue scheduling policies. However, our analysis shows that the intra-queue scheduling policies do significantly impact the response times experienced by both traffic flows.

The remainder of this chapter is organized as follows. In Section 5.2, we introduce our model and notation. We then summarize our main results in Section 5.3. The formal statements of our results, their interpretations, and proofs follow in Sections 5.4 and 5.5. Finally, we conclude in Section 5.6.

5.2 Model and preliminaries

5.2.1 System model

We now introduce our system model. In our model, two parallel queues contend for service from a single server. One of the queues sees a heavy-tailed arrival process, whereas the other sees a light-tailed arrival process. We refer to the former queue as the heavy queue, or Queue H, and the latter queue as the light queue, or Queue L. We consider two distinct scenarios for connectivity of the queues with the server: a *wireline scenario*, in which both queues are always connected to the server, and a *wireless scenario*, in which each queue experiences a stochastically time varying connectivity with the server. Fig. 5.1 provides an illustration of our setup. Time is slotted, and in each slot, the server can provide single unit of service to a connected queue. Henceforth, we refer to this unit of service as a packet, and say the server can process a single packet from a connected queue in each slot.



(a) *The wireline scenario* **(b)** *The wireless scenario*

Figure 5.1: *Model*

Before we formally define the model, we recall the following definitions for heavy-tailed and light-tailed distributions. For any non-negative random variable X, we use F_X to denote its distribution function (d.f.), i.e., $F_X(x) := P(X \leq x)$, and $\bar{F}_X$ to denote its tail distribution function, i.e., $\bar{F}_X(x) := P(X > x)$. The random variable X (or its d.f. F_X) is said to be *heavy-tailed* if

$$\limsup_{x\to\infty} \frac{\bar{F}_X(x)}{e^{-\phi x}} = \infty$$

for all $\phi > 0$. Conversely, X (or its d.f. F_X) is said to be *light-tailed* if it is not heavy-tailed, i.e., if there exists $\phi > 0$ such that

$$\lim_{x\to\infty} \frac{\bar{F}_X(x)}{e^{-\phi x}} = 0.$$

Intuitively, a d.f. is heavy-tailed if its tail is asymptotically heavier than that of any exponential distribution. The random variable X (or its d.f. F_X) is said to be *regularly varying* with index $\theta > 0$ (denoted $X \in \mathcal{RV}(\theta)$) if $P(X > x) = x^{-\theta}L(x)$, where $L(x)$ is a slowly varying function, i.e., $L(x)$ satisfies $\lim_{x\to\infty} \frac{L(xy)}{L(x)} = 1 \; \forall \; y > 0$. Regularly varying distributions, which constitute an important subclass of the class of heavy-tailed distributions, are a generalization of the class of Pareto distributions [64].

Returning now to the definition of our model, let t denote the time index. In each slot, a job, comprising

a burst of packets, can arrive stochastically into each queue. Let $H(t)$ and $L(t)$ denote, respectively, the size of the job (in number of packets) arriving into the heavy queue and the light queue in time slot t. We adopt the convention that the size of the incoming job is zero if there is no arrival in a slot.

Our stochastic model for the arrival processes is the following. The sequences $L(\cdot)$ and $H(\cdot)$ are i.i.d. and independent of one another. The random variable $L(t)$ is light-tailed, and the random variable $H(t)$ is regularly varying, with index $\theta_H > 1$. In other words, we assume that the sizes of jobs entering the light queue are light-tailed, the sizes of jobs entering the heavy queue are heavy-tailed (specifically, regularly varying), and inter-arrival times between jobs in each queue are geometrically distributed. Let $\lambda_H := \mathbb{E}[H(t)]$ and $\lambda_L := \mathbb{E}[L(t)]$ denote the mean arrival rates into the heavy queue and the light queue, respectively.

In the wireline scenario, the light queue and the heavy queue are always connected to the server, so that the server can serve either queue in each slot. In the wireless scenario, the connectivities of the heavy queue and the light queue are described, respectively, by Bernoulli sequences $\{\eta_H(t)\}$ and $\{\eta_L(t)\}$. $\eta_H(t), \eta_L(t) \in \{0, 1\}$, with a value of 1 indicating that the corresponding queue is connected to the server in time slot t. We assume that the sequences $\{\eta_H(t)\}$ and $\{\eta_L(t)\}$ are mutually independent and independent of the arrival processes. Let $p_H := P(\eta_H(t) = 1)$ and $p_L := P(\eta_L(t) = 1)$ denote, respectively, the probabilities that the heavy queue and the light queue are connected to the server in each time slot. We assume that $p_H, p_L \in (0, 1)$. Also, we assume that the server can detect the connectivity state of both queues, as well as the queue size of a connected queue in each slot. Note that our model for the wireless scenario captures an uplink/downlink setting with two wireless nodes connected to a base station or access point via independent fading channels.

Let $q_H(t)$ and $q_L(t)$ denote, respectively, the lengths (in number of packets) of the heavy queue and the light queue in the beginning of time slot t. The queue lengths evolve as follows.

$$\begin{aligned} q_H(t+1) &= H(t) + q_H(t) - \mathbf{1}_{\{\text{heavy queue got service in slot t}\}}, \\ q_L(t+1) &= L(t) + q_L(t) - \mathbf{1}_{\{\text{light queue got service in slot t}\}}. \end{aligned}$$

When both queues are connected to the server in a certain slot, the inter-queue scheduling policy determines which queue will receive service. In the wireless scenario, if only one queue is connected to the server in a certain slot, then that queue receives service if it has any waiting packets. We refer to such slots as exclusive slots. We use q_H and q_L to denote respectively the stationary queue lengths of the heavy queue and the light queue. We use V_H to denote the steady state response time experienced by a job in the heavy queue, and V_L to denote the steady state response time experienced by a job in the light queue.

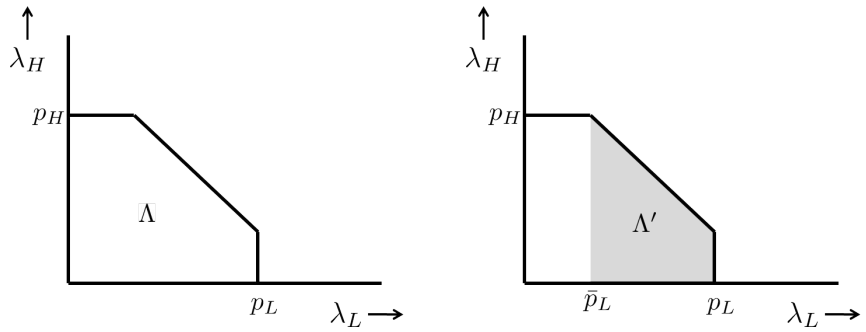
5.2.2 Stability region

The *stability region* for the queueing system defined above, i.e., the set Λ of (λ_H, λ_L) pairs that are stabilizable, is well understood. We discuss the stability region corresponding to the the wireline and the wireless scenario separately.

Wireless scenario: In the wireless scenario, it follows from [67] that

$$\Lambda = \{(\lambda_H, \lambda_L) \mid 0 \leq \lambda_H < p_H,\ 0 \leq \lambda_L < p_L,\ \lambda_H + \lambda_L < p_H + p_L - p_H p_L\}.$$

The constraints defining the stability region are intuitive: the average arrival rate into each queue cannot exceed its maximum possible service rate (defined by fraction of time it is connected to the server), and the sum total of the arrival rates cannot exceed the maximum possible aggregate service rate of the two queues (defined by the fraction of time at least one queue is connected to the server). This stability region is visualized in Fig. 5.2a. It is also well known that the class of generalized max-weight inter-queue policies are *throughput optimal* [21, 66, 67], i.e., they stabilize the queueing system over the entire stability region.



(a) *The stability region* Λ **(b)** *The subset* Λ'

Figure 5.2: *Wireless scenario: The stability region* Λ*, and its subset of interest* Λ'*.*

Let $\bar{p}_L := p_L(1 - p_H)$. Note that $\bar{p}_L$ is the probability that only the light queue is connected to the server in a slot, i.e., the probability that a slot is exclusive to the light queue. If $\lambda_L < \bar{p}_L$, then the arrivals into the light queue can be stably supported by just exclusive slots, implying the light queue essentially does not need to compete for service with the heavy queue. This case is uninteresting, since the response time distribution for the light queue is guaranteed to be light-tailed, irrespective of the inter-queue or intra-queue scheduling policy. For the same reason, the case $\lambda_H = 0$ is uninteresting. Accordingly, in the remainder of this chapter, we restrict our attention to the subset Λ' of the stability region over which $\lambda_L > \bar{p}_L$, and $\lambda_H > 0$. This subset is visualized in Fig. 5.2b.

Wireline scenario: In the wireline scenario, it is easy to see that the stability region is given by

$$\Lambda = \{(\lambda_H, \lambda_L) \mid \lambda_L, \lambda_H \geq 0,\ \lambda_L + \lambda_H < 1.\}.$$

Note that the stability region is defined by the constraint that the aggregate packet arrival rate into the system is less that the maximum service rate of the server, i.e., 1 packet per slot. Note, as before, that the cases $\lambda_L = 0$ and $\lambda_H = 0$ are uninteresting. Accordingly, in the remainder of this chapter, we restrict our attention to the subset Λ' of the stability region over which $\lambda_L, \lambda_H > 0$. Note that in the wireline scenario, any work

conserving inter-queue scheduling policy is throughput optimal. For example, the policy that gives strict priority to the light queue is throughput optimal. However, since our interest is in policies that are throughput optimal in more general settings, we restrict our attention to generalized max-weight inter-queue policies.

5.2.3 Notation and preliminaries

We now state some relevant definitions and properties related to heavy-tailed distributions. The following characterization of heavy-tailed distributions will be useful later. For any non-negative random variable X, define

$$\Psi_X(x) := \frac{-\log \bar{F}_X(x)}{x}.$$

Lemma 23. *Suppose X is non-negative random variable. Then X is heavy-tailed if and only if*

$$\liminf_{x\to\infty} \Psi_X(x) = 0.$$

We give the proof of Lemma 23 in Appendix 5.A.

A non-negative random variable X (or its d.f. F_X) is said to be *long-tailed* (denoted $X \in \mathcal{L}$) if

$$\lim_{x\to\infty} \frac{P(X > x+y)}{P(X > x)} = 1$$

for all $y > 0$. The class of long-tailed distributions includes most of the common heavy-tailed distributions, including the Pareto, the lognormal and the heavy-tailed Weibull distribution [64]. The following sufficient condition for a distribution to be long-tailed will be useful.

Lemma 24. *Suppose X is a non-negative random variable. If $\Psi_X(x)$ is non-increasing with*

$$\lim_{x\to\infty} \Psi_X(x) = 0,$$

then $X \in \mathcal{L}$.

We give the proof of Lemma 24 in Appendix 5.A. The class of regularly varying distributions is a strict subset of the class of long-tailed distributions, which in turn is a strict subset of the class of heavy-tailed distributions [64].

We describe the (logarithmic) asymptotic tail behavior of heavy-tailed X, using its *tail index*, defined as

$$\Gamma(X) := \lim_{x\to\infty} -\frac{\log P(X > x)}{\log(x)},$$

when the limit exists. The tail index is useful for describing the asymptotic tail behavior of distributions that exhibit a roughly 'power-law' tail. In particular, if $X \in \mathcal{RV}(\theta)$, then $\Gamma(X) = \theta$ [55, Prop. 2.6]. It is easy to check that if $\Gamma(X) < \infty$, then X is heavy-tailed. Moreover, it can be shown that

(i) if $\Gamma(X) > 0$, then $\mathbb{E}\left[X^{\beta}\right] < \infty$ for $0 \leq \beta < \Gamma(X)$,

(ii) if $\Gamma(X) < \infty$, then $\mathbb{E}\left[X^{\beta}\right] = \infty$ for $\beta > \Gamma(X)$.

Finally, note that a smaller value of tail index implies a 'heavier' tail.

To give a lower bound on the tail decay of a heavy-tailed random variable X, we use

$$\bar{\Gamma}(X) := \limsup_{x\to\infty} -\frac{\log P\left(X > x\right)}{\log(x)}.$$

It is easy to check that if $\bar{\Gamma}(X) < \infty$, then X is heavy-tailed. Moreover, if $\bar{\Gamma}(X) < \infty$, then $\mathbb{E}\left[X^{\beta}\right] = \infty$ for $\beta > \bar{\Gamma}(X)$.

We conclude this section by listing some notation used in the following sections. For functions $\varphi(x)$ and $\xi(x)$, the notation $\varphi(x) \sim \xi(x)$ means $\lim_{x\to\infty} \frac{\varphi(x)}{\xi(x)} = 1$. For $t_1, t_2 \in \mathbb{N}$,

$$\begin{aligned} A_L(t_1,t_2) &:= \sum_{t=t_1}^{t_2} L(t), \\ A_H(t_1,t_2) &:= \sum_{t=t_1}^{t_2} H(t). \end{aligned}$$

$A_L(t_1,t_2)$ and $A_H(t_1,t_2)$ denote, respectively, the number of packets entering the light queue and the heavy queue in slots t_1 through t_2. For $y \in \mathbb{N}$,

$$A_L^{(y)}(t_1,t_2) := \sum_{t=t_1}^{t_2} L(t)\mathbf{1}_{\{L(t)\leq y\}}.$$

$A_L^{(y)}(t_1,t_2)$ is the number of packets entering the light queue from jobs of size $\leq y$ in slots t_1 through t_2. Let $\lambda_L^{(y)} := \mathbb{E}\left[L(1)\mathbf{1}_{\{L(1)\leq y\}}\right]$. Clearly, $\lim_{y\to\infty} \lambda_L^{(y)} = \lambda_L$. In the wireless scenario, define $\bar{S}_L(t_1,t_2) := \sum_{t=t_1}^{t_2} \mathbf{1}_{\{\eta_L(t)=1,\eta_H(t)=0\}}$. $\bar{S}_L(t_1,t_2)$ is the number of exclusive slots available to the light queue in slots t_1 through t_2.

Finally, in the wireline scenario, we interpret $\bar{p}_L \equiv 0$, and $\bar{S}_L(t_1,t_2) \equiv 0$.

5.3 Summary of results

In this section, we outline the main technical contributions of this chapter. Our main results are summarized in Table 5.1. Before we interpret the results in this table, we define the inter-queue scheduling policies analyzed in this chapter. These policies belong to the class of generalized max-weight policies.

Max-weight-α scheduling: In each slot, the max-weight-α policy serves the queue that wins the comparison

$$q_L(t)^{\alpha_L}\eta_L(t) \gtreqless q_H(t)^{\alpha_H}\eta_H(t).$$

	Light queue	Heavy queue
Max-weight-α ($\alpha_L \geq \alpha_H = 1$)	$\bar{\Gamma}(V_L) \leq \alpha_L\theta_H - 1$ (Thm. 16) (for any intra-queue policy)	
	$\Gamma(V_{L,FCFS}) = \alpha_L(\theta_H - 1)$ (Thm. 17)	$\Gamma(V_{H,FCFS}) = \theta_H - 1$ (Thm. 19)
	$\bar{\Gamma}(V_{L,PLCFS}) \leq \theta_H - 1/\alpha_L$ (Thm. 18)	$\Gamma(V_{H,PLCFS}) = \theta_H$ (Thm. 20) (wireline only; for $\alpha_L > \frac{\theta_H}{\theta_H - 1}$)
Log-max-weight	$V_{L,FCFS}$ is light-tailed (Thm. 22)	$\Gamma(V_{H,FCFS}) = \theta_H - 1$ (Thm. 24) (wireline only)
	$V_{L,PLCFS}$ is heavy-tailed (Thm. 23)	$\Gamma(V_{H,PLCFS}) = \theta_H$ (Thm. 25) (wireline only)

Table 5.1: *Summary of main results*

We focus on the case $\alpha_L \geq \alpha_H = 1$; this corresponds to awarding a relative priority to the light queue. Note that a higher value of α_L implies a higher priority for the light queue. The case $\alpha_L = \alpha_H = 1$ corresponds to the classical max-weight policy.

Log-max-weight scheduling: In each slot, the log-max-weight policy serves the queue that wins the comparison

$$q_L(t)\eta_L(t) \gtreqless \log(1 + q_H(t))\eta_H(t).$$

Note that the log-max-weight policy awards an even higher priority to the light queue compared to the max-weight-α class of policies, since it awards service by comparing an exponential function of the light queue occupancy with the heavy queue occupancy.

The throughput optimality of these policies follows easily from Theorem 1 in [21]. We now interpret the statements in Table 5.1. In this table, note that we emphasize the dependence of V_L and V_H on the corresponding intra-queue scheduling policies. Unless otherwise stated, the results in the table apply to the wireless as well as the wireline scenario.

5.3.1 Max-weight-α scheduling

Let us first consider the original max-weight policy; this corresponds to $\alpha_L = \alpha_H = 1$. It follows from Theorem 16 that under this policy, $\bar{\Gamma}(V_L) \leq \alpha_L - 1$ for any intra-queue scheduling policy for the light queue. This means that under max-weight scheduling, the jobs in the light queue experience heavy-tailed response times, irrespective of the intra-queue scheduling policy for the light queue. Moreover, the response time tail is at least one order heavier than the tail of the job size distribution for the heavy queue. Informally, this is because under the max-weight policy, a large job arriving into the heavy queue causes the light queue to be starved of service (except in its exclusive slots) for a long time. This suggests that an inter-queue policy that can reduce this starvation period by awarding a higher priority to the light queue might lead to a better response time tail for the light queue. This motivates the analysis of max-weight-α inter-queue policies with

$\alpha_L > \alpha_H = 1$.

For the max-weight-α class of policies (with $\alpha_L > \alpha_H = 1$), Theorem 16 states that the response time tail index for the light queue is bounded above by $\alpha_L(\theta_H - 1)$ for any intra-queue scheduling policy. This implies that response times in the light queue remain heavy-tailed under the class of max-weight-α policies. However, since the upper bound on the tail index is an increasing function of α_L, approaching ∞ as $\alpha_L \to \infty$, this suggests the possibility of achieving an arbitrarily high response time tail index with the appropriate intra-queue scheduling policy. Indeed, Theorem 17 states that under first-come-first served (FCFS) scheduling in the light queue, the response time tail index grows linearly with α_L. This means that the response time tail index for the light queue can be made arbitrarily large by setting α_L large enough, i.e., by awarding the light queue sufficiently high priority. Crucially, the intra-queue scheduling policy needs to be appropriately chosen in order to exploit this priority. This is illustrated by Theorem 18, which states that under preemptive-last-come-first-served (PLCFS) scheduling in the light queue, the tail index of V_L remains bounded above by θ_H for all α_L.

Theorems 19 and 20 cover the response time tail for the heavy queue under max-weight-α scheduling between queues. Theorem 19 states that with FCFS scheduling in the heavy queue, the tail index of V_L equals $\theta_H - 1$. Note that $\theta_H - 1$ is also the response tail index if the heavy queue receives exclusive access to be server (with FCFS scheduling). For the wireline scenario, for $\alpha_L \geq \frac{\theta_H}{\theta_H - 1}$, Theorem 20 states that under PLCFS scheduling in the heavy queue, the response time tail index equals θ_H. This is the optimal tail index, since the response time tail index is bounded above by the tail index of the job size distribution. These results indicate that the response time tail for the heavy queue is unaffected by the higher priority awarded to the light queue under max-weight-α policies.

In conclusion, our analysis of the max-weight-α class of inter-queue policies reveals that the level of priority awarded by these policies to the light queue is insufficient to make its response time distribution light-tailed. This motivates the log-max-weight policy, which awards an even higher priority to the light queue.

5.3.2 Log-max-weight scheduling

The log-max-weight inter-queue scheduling policy determines which queue to serve in each slot by comparing an exponential function of the light queue occupancy with the heavy queue occupancy. Theorem 22 states the key result of this chapter: under log-max-weight scheduling between queues, and FCFS scheduling in the light queue, V_L is light-tailed. In other words, the level of priority awarded by the log-max-weight policy to the light queue is sufficient to make its response time distribution light-tailed. Remarkably, the correct choice of intra-queue policy is essential for this to happen. Theorem 23 states that with PLCFS scheduling in the light queue, the response time remains heavy-tailed.

Finally, Theorem 25 states that for the wireline scenario, with PLCFS scheduling within the heavy queue, its response time tail index is optimal. As before, this indicates that the response time tail for the heavy queue

remains unaffected by the higher priority awarded to the light queue by the log-max-weight policy.

In summary, our results show that by carefully designing intra-queue and inter-queue scheduling policies, it is possible to maintain throughput optimality and ensure light-tailed delays for the light-tailed flow, without affecting the response time tail performance of the heavy-tailed flow. In the following sections, we formally state and prove our results. In Section 5.4, we analyze max-weight-α inter-queue scheduling, and in Section 5.5, we analyze log-max-weight inter-queue scheduling.

5.4 Max-weight-α scheduling between queues

In this section, we study the tail behavior of the (stationary) response time in the light queue and the heavy queue under the max-weight-α scheduling policy between queues. The max-weight-α policy is a generalization of the max-weight policy, and is characterized by two positive parameters α_L and α_H. In each slot, the max-weight-α policy serves the queue that wins the comparison

$$q_L(t)^{\alpha_L}\eta_L(t) \gtreqless q_H(t)^{\alpha_H}\eta_H(t).$$

Ties may be broken arbitrarily, but we assume for concreteness that ties are broken in favor of the light queue. The throughput optimality of this policy follows easily from Theorem 1 in [21].

Note that when $\alpha_L = \alpha_H$, the max-weight-α policy is identical to the original max-weight policy. The parameters α_L and α_H determine the relative priorities of the two queues. Since we are interested in the scenario where the light queue receives a higher priority than the heavy queue, we focus on the case $\alpha_L \geq \alpha_H$. Moreover, it is easy to see that we may set $\alpha_H = 1$ without loss of generality. Accordingly, we focus on the range of parameters satisfying $\alpha_L \geq \alpha_H = 1$. Note that a higher value of α_L implies a higher priority for the light queue.

Unless otherwise stated, results in this section apply for the wireline as well as the wireless scenario An analysis of the (stationary) queue length tail asymptotics for our system with max-weight-α scheduling between queues is carried out in [31]. It is proved in [31] that

$$\Gamma(q_L) = \alpha_L(\theta_H - 1), \qquad \Gamma(q_H) = \theta_H - 1. \tag{5.1}$$

The goal of this section is to understand how the relative priority awarded to the light queue by the max-weight-α policy impacts the response time tail behavior for the light queue and the heavy queue. We study the light queue first.

5.4.1 The response time tail for the light queue

We begin our analysis of the light queue's response time tail under max-weight-α inter-queue scheduling by proving an upper bound on the response time tail index. The following theorem gives an upper bound on the tail index of V_L for any intra-queue scheduling policy in the light queue.

Theorem 16. *Suppose that the arrival rates lie in the subset Λ' of the stability region. Then under the max-weight-α scheduling policy between queues with $\alpha_L \geq \alpha_H = 1$,*

$$\bar{\Gamma}(V_L) = \limsup_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} \leq \alpha_L \theta_H - 1$$

for any intra-queue scheduling policy in the light queue.

The above theorem implies that under the max-weight-α scheduling between queues, the light queue sees heavy-tailed delays. Informally, this is because when there is a large arrival into the heavy queue, the light queue is denied service (except in exclusive slots) for a long time. Indeed, our proof of Theorem 16 is based on formalizing the intuition that if a job of size $\Theta(x^{\alpha_L})$ arrives into the heavy queue early in the busy period, then with high probability, the light queue is denied service for a period of $\Omega(x)$ slots, except in its exclusive slots.

Our proof of Theorem 16 relies on the following representation for the response time tail. Consider a tagged busy period of the system. Let N_L denote the number of jobs entering the light queue in this busy period, and $V_{L,i}$, for $i = 1, 2, \cdots, N_L$, denote the response time of the i'th arriving job. The tail of V_L has the following well-known representation.

$$P\left(V_L > x\right) = \frac{\mathbb{E}\left[N_L^{(x)}\right]}{\mathbb{E}\left[N_L\right]}, \tag{5.2}$$

where $N_L^{(x)} := \sum_{i=1}^{N_L} \mathbf{1}_{\{V_{L,i}>x\}}$ is the number of jobs in the light queue that experience a response time exceeding x in the busy period.[1]

Proof of Theorem 16. The proof proceeds by defining a 'bad' event $I(x)$ such that the bound

$$P\left(V_L > x\right) \geq \frac{P\left(I(x)\right)\mathbb{E}\left[N_L^{(x)} \mid I(x)\right]}{\mathbb{E}\left[N_L\right]} \tag{5.3}$$

leads us to the statement of the theorem.

Without loss of generality, assume that the busy period under consideration starts in time slot 1. Recall that over the subset Λ' of the stability region, $\bar{p}_L < \lambda_L$, and $\lim_{y\to\infty} \lambda_L^{(y)} = \lambda_L$. Pick y large enough so that $\bar{p}_L < \lambda_L^{(y)}$. Let $\delta := (\lambda_L^{(y)} - \bar{p}_L)/4$.

[1] That $\mathbb{E}\left[N_L\right] < \infty$ may be justified as follows. It follows from the Lyapunov analysis in [21] that under the max-weight-α policy, the tuple of queue occupancies evolves according to a positive recurrent Markov chain. This implies that busy periods of the queueing systems have finite mean, which in turn implies that $\mathbb{E}\left[N_L\right] < \infty$ via Wald's lemma.

We are now ready to define the event $I(x)$. Fix $\epsilon > 0$.

$$\begin{aligned} I(x) \quad := \quad & \left\{H(1) > \left\lceil \frac{xy}{\delta} \right\rceil + (\lambda_L + \epsilon)^{\alpha_L} \left\lceil \frac{xy}{\delta} \right\rceil^{\alpha_L} \right\} \bigcap \\ & \left\{A_L\left(1, \left\lceil \frac{xy}{\delta} \right\rceil\right) < (\lambda_L + \epsilon) \left\lceil \frac{xy}{\delta} \right\rceil \right\} \bigcap \\ & \left\{\bar{S}_L\left(1, \left\lceil \frac{xy}{\delta} \right\rceil\right) < (\bar{p}_L + \delta) \left\lceil \frac{xy}{\delta} \right\rceil \right\} \bigcap \\ & \left\{A_L^{(y)}\left(1, \left\lceil \frac{xy}{\delta} \right\rceil\right) > (\lambda_L^{(y)} - \delta) \left\lceil \frac{xy}{\delta} \right\rceil \right\} \\ & =: I_1(x) \cap I_2(x) \cap I_3(x) \cap I_4(x). \end{aligned}$$

Informally, the event $I_1(x)$ corresponds to the busy period starting with a 'large' job of size $O(x^{\alpha_L})$ entering the heavy queue. The events $I_2(x)$, $I_3(x)$, and $I_4(x)$ state that the number of packet arrivals into the light queue and number of exclusive slots for the light queue over the interval from slot 1 to slot $\left\lceil \frac{xy}{\delta} \right\rceil$ do not deviate much from their 'law of large numbers' estimates. Indeed, the weak law of large numbers implies that the events $I_2(x)$, $I_3(x)$, and $I_4(x)$ have a probability approaching 1 as $x \to \infty$.

Next, we show that the event $I(x)$ implies that at least x jobs entering the light queue in the busy period under consideration experience a response time exceeding x time slots. To see this, note that the event $I_1(x) \cap I_2(x)$ implies that the heavy queue has priority over the light queue in slots 1 through $\left\lceil \frac{xy}{\delta} \right\rceil$. Indeed, $I_1(x)$ implies that the length of the heavy queue remains greater than $(\lambda_L + \epsilon)^{\alpha_L} \left\lceil \frac{xy}{\delta} \right\rceil^{\alpha_L}$ over this interval, and $I_2(x)$ implies that the length of the light queue never exceeds $(\lambda_L + \epsilon) \left\lceil \frac{xy}{\delta} \right\rceil$ over the same interval. As a result, under event $I(x)$, the light queue receives service only in its exclusive slots until time $\left\lceil \frac{xy}{\delta} \right\rceil$. Note that $I_3(x)$ gives an upper bound on the number of exclusive slots received by the light queue until time $\left\lceil \frac{xy}{\delta} \right\rceil$. Finally, $I_4(x)$ gives a lower bound on the number packets arriving into the light queue until time $\left\lceil \frac{xy}{\delta} \right\rceil$ from jobs of size $\leq y$. Therefore, under event $I(x)$, the number of packets remaining in the light queue after time slot $\left\lceil \frac{xy}{\delta} \right\rceil$, corresponding to jobs of size $\leq y$ exceeds

$$\begin{aligned} & (\lambda_L^{(y)} - \delta) \left\lceil \frac{xy}{\delta} \right\rceil - (\bar{p}_L + \delta) \left\lceil \frac{xy}{\delta} \right\rceil \\ = \quad & 2\delta \left\lceil \frac{xy}{\delta} \right\rceil \\ \geq \quad & 2xy \end{aligned}$$

Now, since the corresponding jobs have a size of at most y, we conclude that under $I(x)$, the light queue contains at least $2x$ jobs at the end of $\left\lceil \frac{xy}{\delta} \right\rceil$ slots. Since each of these jobs requires at least one slot of service to complete, we conclude that under $I(x)$, at least $x - 1$ jobs experience a response time exceeding x in the busy period under consideration.

We return now to our bound on the tail of V_L :

$$P\left(V_L > x\right) \geq \frac{P\left(I(x)\right) \mathbb{E}\left[N_L^{(x)} \mid I(x)\right]}{\mathbb{E}\left[N_L\right]}.$$

We have defined the event $I(x)$ such that $\mathbb{E}\left[N_L^{(x)} \mid I(x)\right] \geq x-1$. To bound the probability of $I(x)$, note that

$$P\left(I(x)\right) = P\left(I_1(x)\right)\ P\left(I_2(x) \cap I_3(x) \cap I_4(x)\right),$$

since the arrival process into the heavy queue is independent of the arrival process into the light queue and the queue connectivity processes. Invoking the weak law of large numbers, we conclude that for $\nu \in (0,1)$, $P\left(I_2(x) \cap I_3(x) \cap I_4(x)\right) > (1-\nu)$ for large enough x. Therefore, for large enough x,

$$\begin{aligned} P\left(V_L > x\right) &\geq \frac{1-\nu}{\mathbb{E}\left[N_L\right]}(x-1)\ P\left(I_1(x)\right) \\ \Rightarrow -\frac{\log P\left(V_L > x\right)}{\log(x)} &\leq -\frac{\log\left(\frac{1-\nu}{\mathbb{E}[N_L]}\right)}{\log(x)} - \frac{\log P\left(I_1(x)\right)}{\log(x)} - \frac{\log(x-1)}{\log(x)}. \end{aligned}$$

As a result,

$$\begin{aligned} \limsup_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} &\leq \limsup_{x\to\infty} -\frac{\log\left(\frac{1-\nu}{\mathbb{E}[N_L]}\right)}{\log(x)} - \frac{\log P\left(I_1(x)\right)}{\log(x)} - 1 \\ &= \alpha_L\theta_H - 1, \end{aligned}$$

where the last step above uses the fact that $H(1) \in \mathcal{RV}(\theta_H)$, which implies that

$$\lim_{x\to\infty} -\frac{\log P\left(H(1) > \left\lceil \frac{xy}{\delta} \right\rceil + (\lambda_L+\epsilon)^{\alpha_L}\left\lceil \frac{xy}{\delta} \right\rceil^{\alpha_L}\right)}{\log(x)} = \alpha_L\theta_H.$$

This completes the proof. □

Theorem 16 states that the response time distribution corresponding to the light queue is heavy-tailed under the max-weight-α policy, irrespective of the intra-queue scheduling policy. However, the upper bound on the tail index is an increasing function of α_L, with the bound approaching ∞ as $\alpha \to \infty$. This suggests the possibility of an increasing response time tail index (i.e., a lighter response time tail) with increasing α_L (i.e., increasing priority for the light queue) with the appropriate intra-queue scheduling policy. We investigate this possibility next.

We study the response time tail behavior in the light queue, under max-weight-α scheduling between queues, and two specific intra-queue scheduling disciplines in the light queue: first-come-first-served (FCFS), and preemptive-last-come-first-served (PLCFS). We characterize the exact response time tail index under FCFS (see Theorem 17), and give an upper bound on the tail index under PLCFS (see Theorem 18).

Our results show that with FCFS scheduling within the light queue, the response time tail index increases linearly with α_L. This means that even though the response time in the light queue remains heavy-tailed for all α_L, its tail index can be made arbitrarily large by setting α_L to a large enough value, i.e., by giving the

light queue sufficient priority. In contrast, under PLCFS scheduling in the light queue, the tail index remains bounded above by θ_H for all values of α_L. This highlights the importance of choosing the right intra-queue scheduling policy in order to exploit the priority awarded to it by the inter-queue scheduling policy.

We now state and prove our results for the response time tail under FCFS and PLCFS intra-queue scheduling. The following theorem covers the FCFS case.

Theorem 17. *Suppose that the arrival rates lie in the subset Λ' of the stability region. Then under the max-weight-α scheduling policy between queues with $\alpha_L > \alpha_H = 1$, and first-come-first-served scheduling within the light queue,*

$$\Gamma(V_L) = \lim_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} = \alpha_L(\theta_H - 1).$$

The proof is relatively straightforward for the the wireless scenario, and follows from the tail asymptotics of q_L derived in [31, Chapter 5]. We give the proof of this case here. The proof for the wireline scenario is more involved; we do not present the proof for this case here as it is similar to the proof of Theorem 22 in the following section.

Proof of Theorem 17 for the wireless scenario. Consider a tagged job entering the light queue in slot 0 in steady state. The tagged job has size $L(0) > 0$ and sees a queue length $q_L(0)$ in the light queue. Let us denote the response time of the tagged job by V_L. We need to prove that

$$\lim_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} = \alpha_L(\theta_H - 1). \tag{5.4}$$

We do this by proving matching asymptotic lower and upper bounds on the tail of V_L.

The lower bound on the tail of V_L is easy: since packets in the light queue are served in a FCFS manner, $V_L \geq q_L(0)$. Therefore, $P\left(V_L > x\right) \geq P\left(q_L(x) > 0\right)$, which implies, using (5.1), that

$$\limsup_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} \leq \alpha_L(\theta_H - 1). \tag{5.5}$$

We now obtain the upper bound on the tail of V_L. Note that

$$q_L(1) = q_L(0) - \mathbf{1}_{\{\text{light queue for service in slot 0}\}} + L(0).$$

Since the light queue uses FCFS scheduling, V_L is simply equal to the time it takes for the light queue to receive service $q_L(1)$ times. Define $T := \min\{x \in \mathbb{N} \mid \bar{S}_L(1, x) \geq q_L(1)\}$. Note that T is the time it takes

after slot 0 for the light queue to see $q_L(1)$ exclusive slots. Clearly, $V_L \leq T$. Fix small $\epsilon > 0$. We have

$$\begin{aligned} P\left(V_L > x\right) &\leq P\left(T > x\right) \\ &= P\left(\bar{S}_L(1,x) < q_L(1)\right) \\ &= P\left(\bar{S}_L(1,x) < q_L(1);\ q_L(1) > (\bar{p}_L - \epsilon)x\right) + P\left(\bar{S}_L(1,x) < q_L(1);\ q_L(1) \leq (\bar{p}_L - \epsilon)x\right) \\ &\leq P\left(q_L(1) > (\bar{p}_L - \epsilon)x\right) + P\left(\bar{S}_L(1,x) < (\bar{p}_L - \epsilon)x\right). \end{aligned} \tag{5.6}$$

Since $q_L(0)$ is heavy-tailed with tail index $\alpha_L(\theta_H - 1)$, and $L(0)$ is light-tailed, it is easy to show that $q_L(1)$ is heavy-tailed with tail index $\alpha_L(\theta_H - 1)$. Moreover using the Chernoff bound, we conclude that there exists $\phi > 0$ such that $P\left(\bar{S}_L(1,x) < (\bar{p}_L - \epsilon)x\right) \leq e^{-\phi x}$. Using these two facts, the bound (5.6) implies that

$$\liminf_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} \geq \alpha_L(\theta_H - 1). \tag{5.7}$$

(5.5) and (5.7) of course imply (5.4). This completes the proof. □

Next, we cover the case with PLCFS scheduling in the light queue, with max-weight-α scheduling between queues.

Theorem 18. *Suppose that the arrival rates lie in the subset Λ' of the stability region. Then under max-weight-α scheduling policy between queues with $\alpha_L > \alpha_H = 1$, and preemptive-last-come-first-served scheduling within the light queue,*

$$\bar{\Gamma}(V_L) = \limsup_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} \leq \theta_H - \frac{1}{\alpha_L}.$$

Informally, the proof of this theorem is based on the following idea: a large arrival into the heavy queue early into the busy period can cause a large number of jobs in the light queue to experience a response time of the same order as the length of the busy period. We give the proof of this theorem in Appendix 5.C.

This completes the analysis of the response time tail for the light queue under max-weight-α scheduling. In the following subsection, we study the response time tail for the heavy queue.

5.4.2 The response time tail for the heavy queue

In this section, we analyze the response time tail for the heavy queue with max-weight-α scheduling between queues. We show first that with FCFS scheduling within the heavy queue, the response time tail index is the same as it would be if the heavy queue was being served exclusively by the server. In other words, with FCFS scheduling in the heavy queue, the response time tail index is insensitive to the level of priority awarded to the light queue by the max-weight-α policy.

Theorem 19. *Suppose that the arrival rates lie in the subset Λ' of the stability region. Then under max-weight-α scheduling between queues with $\alpha_L \geq \alpha_H = 1$, and first-come-first-served scheduling within the heavy queue,*

$$\Gamma(V_H) = \lim_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} = \theta_H - 1.$$

Note that in a $Geo/GI/1$ queue with FCFS scheduling, if the job size distribution belongs to $\mathcal{RV}(\theta_H)$, then it is well known that the response time tail index equals $\theta_H - 1$ (for example, see [13]). Theorem 19 implies that with FCFS intra-queue scheduling, the response time distribution for heavy queue has the same tail index. The proof of this theorem is based on the queue length tail asymptotics derived in [31].

Proof of Theorem 19. Consider a tagged job entering the heavy queue in slot 0 in steady state. The tagged job has size $H(0) > 0$. Let us denote the response time of the tagged job by V_H. We need to prove that

$$\lim_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} = \theta_H - 1.$$

We do this by proving matching asymptotic lower and upper bounds on the tail of V_H.

The lower bound is easy: since packets in the heavy queue are served in a FCFS manner, $V_H \geq q_H(0)$. Therefore, $P\left(V_H > x\right) \geq P\left(q_H(x) > 0\right)$, which implies, using (5.1), that

$$\limsup_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} \leq \theta_H - 1.$$

We now obtain a matching upper bound. We do this separately for the wireless and the wireline case.

Wireless case: Since the heavy queue uses FCFS scheduling, V_H is simply equal to the time it takes for the heavy queue to receive service $q_H(1)$ times. Define $T := \min\{x \in \mathbb{N} \mid \bar{S}_H(1,x) \geq q_H(1)\}$. Note that T is the time it takes after slot 0 for the heavy queue to see $q_H(1)$ exclusive slots. Clearly, $V_H \leq T$. Fix small $\epsilon > 0$. We have

$$\begin{aligned}
P\left(V_H > x\right) &\leq P\left(T > x\right) \\
&= P\left(\bar{S}_H(1,x) < q_H(1)\right) \\
&= P\left(\bar{S}_H(1,x) < q_H(1);\ q_H(1) > (\bar{p}_H - \epsilon)x\right) + P\left(\bar{S}_H(1,x) < q_H(1);\ q_H(1) \leq (\bar{p}_H - \epsilon)x\right) \\
&\leq P\left(q_H(1) > (\bar{p}_H - \epsilon)x\right) + P\left(\bar{S}_H(1,x) < (\bar{p}_H - \epsilon)x\right).
\end{aligned}$$

Since $q_H(0)$ is heavy-tailed with tail index $\theta_H - 1$, and $H(0)$ is heavy-tailed with tail index θ_H, it is easy to show that $q_H(1)$ is heavy-tailed with tail index $\theta_H - 1$. Moreover using the Chernoff bound, we conclude that there exists $\phi > 0$ such that $P\left(\bar{S}_H(1,x) < (\bar{p}_H - \epsilon)x\right) \leq e^{-\phi x}$. Using these two facts, our bound on

$P(V_H > x)$ above implies that

$$\liminf_{x\to\infty} -\frac{\log P(V_H > x)}{\log(x)} \geq \theta_H - 1.$$

This gives us the matching upper bound, and completes the proof for the wireless case.

Wireline case: In the wireline case, we use the fact that $V_H \leq Z$, where Z is the number of time slots following the arrival of the tagged job till the system empties. We argue below that $Z \in \mathcal{RV}(\theta_H - 1)$. This implies that

$$\liminf_{x\to\infty} -\frac{\log P(V_H > x)}{\log(x)} \geq \lim_{x\to\infty} -\frac{\log P(Z > x)}{\log(x)} = \theta_H - 1.$$

This gives us the required matching upper bound. It remains now to show that $Z \in \mathcal{RV}(\theta_H - 1)$. Note that in the wireline scenario, the sum of queue lengths evolves as a discrete time $Geo/GI/1$ queue in which the amount of work entering the queue in slot t equals $B(t) := L(t) + H(t)$. Z is simply the residual busy period for this $Geo/GI/1$ queue. Since $B(t)$ is regularly varying with index θ_H, it follows that the residual busy period is regularly varying with index $\theta_H - 1$. □

For a $Geo/GI/1$ queue fed by a heavy-tailed arrival process, it is well known that FCFS is not the optimal scheduling policy for the response time tail [13]. Instead, policies such as PLCFS are known to be optimal for the response time tail [13]. This motivates us to analyze the response time tail for the heavy queue with PLCFS scheduling intra-queue scheduling. We are able to do this for the wireline scenario.

Theorem 20. *In the wireline scenario, suppose that the arrival rates lie in the subset Λ' of the stability region. Then under max-weight-α scheduling policy between queues with $\alpha_L > \frac{\theta_H}{\theta_H - 1}$ and $\alpha_H = 1$, and preemptive-last-come-first-served scheduling within the heavy queue,*

$$\Gamma(V_H) = \lim_{x\to\infty} -\frac{\log P(V_H > x)}{\log(x)} = \theta_H.$$

Note that since the response time distribution stochastically dominates the job size distribution, θ_H is the maximum possible tail index for the response time distribution of the heavy queue. Theorem 20 states that in the wireline scenario, PLCFS scheduling in the heavy queue does indeed produce the optimal response time tail index when $\alpha_L \geq \frac{\theta_H}{\theta_H - 1}$. This means that as we tune the max-weight-α policy to award increasing priority to the light queue, the response time tail index for the heavy queue remains optimal with PLCFS intra-queue scheduling. We conjecture that the same is true in the wireless scenario. The main difficulty in extending Theorem 20 to the wireless scenario is that the busy period tail behavior is unknown for this case. On the other hand, in the wireline scenario, the busy period behaves identically to busy periods in a $Geo/GI/1$ queue which sees the combined arrival processes of the heavy and the light queue in our model; this busy period is well understood.

Proof of Theorem 20. As in the previous proof, consider a tagged job entering the heavy queue in slot 0 in

steady state. The tagged job has size $H(0) > 0$. Let us denote the response time of the tagged job by V_H. We need to prove that

$$\lim_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} = \theta_H.$$

We do this by proving matching asymptotic lower and upper bounds on the tail of V_H.

The lower bound is easy: it is clear that $V_H \geq H(0)$. Therefore,

$$\limsup_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} \leq \limsup_{x\to\infty} -\frac{\log P\left(H(0) > x\right)}{\log(x)} = \theta_H.$$

To prove the upper bound, define

$$Z := \min\{x \in \mathbb{N} \mid (H(0) + q_L(1) + \sum_{i=1}^{x}(L(i) + H(i)) \leq x))\}.$$

Z is defined so that at the end of time slot Z, the total occupancy of both queues equals the number of packets waiting in the heavy queue at the time of the tagged job's arrival. Since the heavy queue uses PLCFS scheduling, it follows that $V_H \leq Z$.

We now use Lemma 30 in Appendix 5.B to show that $Z \in \mathcal{RV}(\theta)$. Note that $H(0) \in \mathcal{RV}(\theta_H)$, and $\Gamma(q_H(1)) = \alpha_L(\theta_H - 1)$. For the range of α_L under consideration, $\theta_H < \alpha_L(\theta_H - 1)$, i.e., $H(0)$ has a heavier tail than $q_H(1)$. This implies that $H(0) + q_H(1) \in \mathcal{RV}(\theta_H)$. Therefore, Lemma 30 in Appendix 5.B implies that $Z \in \mathcal{RV}(\theta)$.

Finally, using the fact that $V_H \leq Z$, we have

$$\liminf_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} \geq \lim_{x\to\infty} -\frac{\log P\left(Z > x\right)}{\log(x)} = \theta_H.$$

This gives us the desired matching upper bound, and completes the proof. □

Together, Theorems 19 and 20 suggest that with increasing α_L, i.e., with increasing priority given to the light queue, the response time tail for the heavy queue remains unaffected. Moreover, the response time tail for the heavy queue behaves like the response time tail in an isolated $Geo/GI/1$ queue, PLCFS scheduling being optimal for the response time tail.

In conclusion, in this section, we analyze the tail behavior of the stationary response time seen by the light queue and the heavy queue under max-weight-α scheduling between queues. Specifically, we study the impact of the higher priority awarded to the light queue by the max-weight-α policy on the response time tail of both queues. Our results show that the response times in the light queue are heavy-tailed in this case. However, we show that by setting α_L large enough, i.e., by awarding sufficiently high priority to the light queue, its response time tail index can be made arbitrarily large. Moreover, our results suggest that the response time tail index of the heavy queue is insensitive to the level of priority awarded to the light queue.

Since our goal is to guarantee light-tailed response times for the light queue, we must therefore study

inter-queue scheduling policies that provide an even higher level of priority to the light queue than the max-weight-α class of policies. This motivates our analysis of the log-max-weight policy in the following section.

5.5 Log-max-weight scheduling between queues

In this section, we study the tail behavior of the (stationary) response time in the light queue and the heavy queue under the log-max-weight scheduling policy between queues. The log-max-weight policy [31] is defined as follows. In each slot t, it serves the queue that wins the comparison

$$q_L(t)\eta_L(t) \gtreqless \log(1+q_H(t))\eta_H(t). \tag{5.8}$$

As before, we assume for concreteness that ties are broken in favor of the light queue. The throughput optimality of this policy once again follows easily from Theorem 1 in [21].

The log-max-weight policy awards an even higher degree of priority to the light queue than the max-weight-α policy. Note that in order to determine which queue to serve in a slot, the max-weight-α policy compares $q_H(t)$ with $q_L(t)^{\alpha_L}$, whereas the log-max-weight policy compares $q_H(t)$ with $e^{q_L(t)}-1$.

The goal of this section is to analyze the response time tail for the light queue as well as the heavy queue under log-max-weight inter-queue scheduling. Unless otherwise stated, results in this section apply for the wireline as well as the wireless scenario.

Before we prove our results on response time tail behavior, we need the following intermediate result. The following theorem states that under log-max-weight scheduling, the stationary queue length distribution of the light queue is light-tailed. This statement is proved in [31] for only the wireline scenario. We give a proof here that applies to the wireline as well as the wireless scenario.

Theorem 21. *Under log-max-weight scheduling between queues, q_L is light-tailed.*

Our proof of Theorem 21 relies on the following lemma.

Lemma 25. *Suppose that F_X is the distribution function corresponding to a non-negative random variable. If $F_X \in \mathcal{L}$, and $\bar{F}_X(x) := 1 - F_X(x)$ is strictly decreasing over $x \geq 0$, then*

$$\mathbb{E}\left[\frac{1}{\bar{F}_X(q_L)}\right] < \infty \text{ and } \mathbb{E}\left[\frac{1}{\bar{F}_X(\log(1+q_H))}\right] < \infty.$$

The above lemma is a consequence of Theorem 1 in [21]. We give its proof in Appendix 5.D. Note that if $F_X \in \mathcal{L}$, then $1/\bar{F}(x)$ grows sub-exponentially. Therefore, Lemma 25 states that certain *sub-exponential* moments of q_L are finite. However, in order to prove that q_L is light-tailed, we need to show that certain *exponential* moments of q_L are finite, i.e., $\mathbb{E}\left[e^{\beta q_L}\right] < \infty$ for some $\beta > 0$. We do this as follows.

Proof of Theorem 21. For the purpose of obtaining a contradiction, let us assume that q_L is heavy-tailed.

Invoking Lemma 23, we conclude that $\liminf_{x\to\infty} \Psi_{q_L}(x) = 0$. Fix $\delta \in (0,1)$. It is easy to see that there exists a strictly increasing integer sequence $\{x_k\}_{k\geq 1}$, with $x_1 = 0$, and $x_k \overset{k\uparrow\infty}{\to} \infty$ such that

(i) $\Psi_{q_L}(x_k)$ is non-decreasing in k, with $\lim_{k\to\infty} \Psi_{q_L}(x_k) = 0$,

(ii) $\bar{F}_{q_L}(x_{k+1}) \leq (1-\delta)\bar{F}_{q_L}(x_k)$ for $k \geq 1$.

We now define a distribution F_Y that agrees with F_{q_L} along the sequence $\{x_k\}$ such that F_Y satisfies the conditions of Lemma 25, implying that $\mathbb{E}\left[1/\bar{F}_Y(q_L)\right] < \infty$. We then show via a direct computation that $\mathbb{E}\left[1/\bar{F}_Y(q_L)\right] = \infty$. This gives us a contradiction, proving that q_L is light-tailed.

We define the distribution F_Y as follows. $\bar{F}_Y(x_k) = \bar{F}_{q_L}(x_k)$ for all $k \geq 1$. For $x \in (x_k, x_{k+1})$,

$$\log(\bar{F}_Y(x)) = \log(\bar{F}_Y(x_k)) + \frac{x - x_k}{x_{k+1} - x_k}\left(\log(\bar{F}_Y(x_{k+1})) - \log(\bar{F}_Y(x_k))\right). \tag{5.9}$$

Note that for $x \in (x_k, x_{k+1})$, $\log(\bar{F}_Y(x))$ is defined by linearly interpolating between $\log(\bar{F}_Y(x_k))$ and $\log(\bar{F}_Y(x_{k+1}))$. Equation (5.9) implies, via simple algebraic manipulations that for $x \in (x_k, x_{k+1})$,

$$\begin{aligned}\Psi_Y(x) &= \frac{\log(\bar{F}_Y(x_k)) - \log(\bar{F}_Y(x_{k+1}))}{x_{k+1} - x_k} + \frac{1}{x}\frac{x_k x_{k+1}\left(\Psi_Y(x_k) - \Psi_Y(x_{k+1})\right)}{x_{k+1} - x_k} \\ &=: \nu_1 + \frac{\nu_2}{x},\end{aligned}$$

where $\nu_1 > 0$, $\nu_2 \geq 0$. This implies that $\Psi_Y(x)$ is non-decreasing over $x \geq 0$, with $\lim_{x\to\infty} \Psi_Y(x) = 0$. From Lemma 24, we conclude that then $F_Y \in \mathcal{L}$. Moreover, since $\bar{F}_Y(x)$ is strictly decreasing over $x \geq 0$ by definition, Lemma 25 implies that $\mathbb{E}\left[1/\bar{F}_Y(q_L)\right] < \infty$.

We now show through a direct computation that $\mathbb{E}\left[1/\bar{F}_Y(q_L)\right] = \infty$. Pick $k_0 \in \mathbb{N}$.

$$\begin{aligned}\mathbb{E}\left[\frac{1}{\bar{F}_Y(q_L)}\right] &\geq \sum_{x=1}^{x_{k_0+1}} \frac{1}{\bar{F}_Y(x)} P\left(q_L = x\right) \\ &= \sum_{k=1}^{k_0} \sum_{x=x_k+1}^{x_{k+1}} \frac{1}{\bar{F}_Y(x)} P\left(q_L = x\right) \\ &\geq \sum_{k=1}^{k_0} \sum_{x=x_k+1}^{x_{k+1}} \frac{1}{\bar{F}_Y(x_k)} P\left(q_L = x\right) \\ &= \sum_{k=1}^{k_0} \frac{\bar{F}_{q_L}(x_k) - \bar{F}_{q_L}(x_{k+1})}{\bar{F}_Y(x_k)}.\end{aligned}$$

Now, since $\bar{F}_Y$ and $\bar{F}_{q_L}$ agree along the sequence $\{x_k\}$,

$$\begin{aligned}\mathbb{E}\left[\frac{1}{\bar{F}_Y(q_L)}\right] &\geq \sum_{k=1}^{k_0} \frac{\bar{F}_Y(x_k) - \bar{F}_Y(x_{k+1})}{\bar{F}_Y(x_k)} \\ &\geq \sum_{k=1}^{k_0} \delta = k_0\delta,\end{aligned}$$

where the last step above uses the fact that $\bar{F}_Y(x_{k+1}) \leq (1-\delta)\bar{F}_Y(x_k)$ for $k \geq 1$. Since $\mathbb{E}\left[1/\bar{F}_Y(q_L)\right] \geq k_0\delta$ for any $k_0 \in \mathbb{N}$, it follows that $\mathbb{E}\left[1/\bar{F}_Y(q_L)\right] = \infty$. This gives us a contradiction, which proves that q_L is light-tailed. □

Using Lemma 25 and the same arguments as in the above proof, one can prove the following statement about the tail of q_H.

Corollary 4. *Under log-max-weight scheduling between queues,* $\log(1+q_H)$ *is light-tailed.*

This corollary will be useful later.

We are now ready to begin our analysis of the response time tail for the light queue and the heavy queue under log-max-weight scheduling between queues. We start with the light queue.

5.5.1 The response time tail for the light queue

In this section, we analyze the response time tail for the light queue under log-max-weight scheduling between queues. As before, we consider two intra-queue scheduling policies: FCFS and PLCFS.

The key result in this section is that with FCFS scheduling within the light queue, V_L is light-tailed (see Theorem 22). This means that *the log-max-weight policy indeed provides sufficient priority to the light queue to make its response time distribution light-tailed*. However, for this to happen, the intra-queue scheduling policy cannot be chosen arbitrarily. In fact, we show that with PLCFS scheduling within the light queue, its response time distribution remains heavy-tailed (see Theorem 23). This extreme contrast between the two policies highlights once again the importance of correctly choosing the intra-queue scheduling policy to exploit the priority awarded to the light queue by the inter-queue scheduling policy.

We now state and prove our results for FCFS and PLCFS intra-queue scheduling. The following theorem covers the FCFS case.

Theorem 22. *Suppose that the arrival rates lie in the subset* Λ' *of the stability region. Then under the log-max-weight scheduling policy between queues, and first-come-first-served scheduling within the light queue,* V_L *is light-tailed.*

The proof of this theorem for the wireless case is easy, and follows the same steps as the proof of Theorem 17. We do not repeat the steps here. Instead, we focus on the proof for the wireline case, which is much more involved. The proof involves bounding V_L from above by considering a fictitious system in which the heavy queue is never served, and the light queue is only served when it wins the comparison (5.8). We sketch the main steps of this proof here, and supply the remaining details in Appendix 5.E.

Proof of Theorem 22 for the wireline scenario. In this proof, for simplicity, we write $A_L(t)$ to denote $A_L(1,t)$ and $A_H(t)$ to denote $A_H(1,t)$ for $t \in \mathbb{N}$. Consider a tagged job entering Queue L in slot 0 when the system is in steady state. Let V_L denote the sojourn time of this tagged job. For $t \in \mathbb{N}$, let $S_L(t)$ and $S_H(t)$ denote

the total number of packets served in Queues L and H, respectively, in slots 1 through t. Since Queue L uses FCFS scheduling,

$$V_L = \min\{t \geq 1 \mid S_L(t) = q_L(1)\},$$

which means $V_L > t \iff S_L(t) < q_L(1)$.

To prove that V_L is light-tailed, we construct an upper bound on V_L using a fictitious queueing system. The fictitious system also consists of two queues, fed by the *same* input processes as the original queues. Let $\widetilde{L}$ and $\widetilde{H}$ denote, respectively, the fictitious queues fed by the arrival processes corresponding to queues L and H. The fictitious queue $\widetilde{H}$ has length one more than queue H in the beginning of slot 1, and subsequently never receives service, i.e., the size of queue $\widetilde{H}$ evolves as

$$\begin{aligned} q_{\widetilde{H}}(1) &= q_H(1) + 1, \\ q_{\widetilde{H}}(t+1) &= q_{\widetilde{H}}(1) + A_H(t) \quad \text{for } t \geq 1. \end{aligned}$$

The queue $\widetilde{L}$ on the other hand has length 0 in the beginning of slot 1, and receives service in a slot $t \geq 1$ iff $q_{\widetilde{L}}(t) \geq \log(1 + q_{\widetilde{H}}(t))$. For $t \geq 1$, let $S_{\widetilde{L}}(t)$ denote the number of packets served in queue $\widetilde{L}$ in slots 1 through t. With this notation, the queue length of $\widetilde{L}$ evolves as

$$\begin{aligned} q_{\widetilde{L}}(1) &= 0, \\ q_{\widetilde{L}}(t+1) &= A_L(t) - S_{\widetilde{L}}(t) \quad \text{for } t \geq 1. \end{aligned}$$

The following lemma states that the fictitious queue $\widetilde{L}$ receives less service than queue L.

Lemma 26. *For all $t \geq 1$, $S_{\widetilde{L}}(t) \leq S_L(t)$.*

Proof. We assume the contrary and obtain a contradiction. Note that since $q_{\widetilde{L}}(1) \leq q_L(1)$ and $q_{\widetilde{H}}(1) > q_H(1)$, $S_{\widetilde{L}}(1) \leq S_L(1)$. Let $\hat{t} \geq 2$ be the first slot in which $S_{\widetilde{L}}(\hat{t}) > S_L(\hat{t})$. Since increments in $S_L(\cdot)$ and $S_{\widetilde{L}}(\cdot)$ can only be 0 or 1, it follows that (i) $S_{\widetilde{L}}(\hat{t}-1) = S_L(\hat{t}-1)$, (ii) $\widetilde{L}$ received service in slot $\hat{t}$, (iii) L did not receive service in slot $\hat{t}$. Conditions (ii) and (iii) imply that

$$\begin{aligned} \left[A_L(\hat{t}-1) - S_{\widetilde{L}}(\hat{t}-1)\right] &\geq \log\left(1 + q_{\widetilde{H}}(1) + A_H(\hat{t}-1)\right), \\ \left[q_L(1) + A_L(\hat{t}-1) - S_L(\hat{t}-1)\right] &< \log\left(q_H(1) + A_H(\hat{t}-1) - S_H(\hat{t}-1)\right). \end{aligned}$$

From Condition (i), it is easy to see that the above two statements are contradictory. □

Lemma 26 gives us an upper bound on V_L based on the service process of the fictitious queue $\widetilde{L}$ as follows. Define $\tilde{V}_L := \min\{t \geq 1 \mid S_{\widetilde{L}}(t) = q_L(1)\}$. Lemma 26 implies that $V_L \leq \tilde{V}_L$. It therefore suffices to prove that $\tilde{V}_L$ is light-tailed.

To prove that $\tilde{V}_L$ is light-tailed, we analyze the following two components of $\tilde{V}_L$ separately. Define

$$T_1 := \min\{t \geq 1 \mid A_L(t) \geq \log(1 + q_{\widetilde{H}}(1) + A_H(t))\}.$$

It is easy to verify that the fictitious queue $\widetilde{L}$ first receives service in slot $T_1 + 1$. Next, define

$$T_2 := \min\{t \geq 1 \mid S_{\widetilde{L}}(T_1 + t) = q_L(1)\}.$$

Clearly, $\tilde{V}_L = T_1 + T_2$. That $\tilde{V}_L$ is light-tailed follows from the following lemmas.

Lemma 27. *T_1 is light-tailed.*

Lemma 28. *T_2 is light-tailed.*

Since the sum of light-tailed random variables is light-tailed, the above lemmas imply that $\tilde{V}_L$ is light-tailed, which in turn implies that V_L is light-tailed.

To complete the proof of Theorem 22, we need to prove Lemmas 27 and 28. We give these proofs in Appendix 5.E. □

Next, we consider the case of PLCFS intra-queue scheduling.

Theorem 23. *Suppose that the arrival rates lie in the subset Λ' of the stability region. Then under the log-max-weight scheduling policy between queues, and preemptive-last-come-first-served scheduling within the light queue, V_L is heavy-tailed.*

We prove this theorem in Appendix 5.F. The proof is based on formalizing the following idea: for appropriately chosen $\gamma > 0$, an arrival of size $O(x^\gamma)$into the heavy queue early in a busy period can cause $\Omega(\log(x))$ jobs in the light queue to experience a response time of $\Omega(x)$ slots in the busy period.

Theorems 22 and 23 demonstrate a remarkable phenomenon: with the same service process for the light queue, one intra-queue scheduling discipline results in heavy-tailed response times, whereas another leads to light-tailed response times.

This completes our analysis of the response time tail for the light queue under log-max-weight scheduling between queues. Next, we study the heavy queue.

5.5.2 The response time tail for the heavy queue

In this section, we study the response time tail for the heavy queue under log-max-weight inter-queue scheduling. We are only able to analyze the wireline scenario here. For the wireline case, we prove that with FCFS as well as PLCFS intra-queue scheduling, V_H has the same tail index as it would if the heavy queue was being served exclusively by the server. These results show that in the wireline scenario, the response time tail index is unaffected by the priority given to the light queue by the log-max-weight policy. This means

that it is possible to achieve light tailed delays in the light queue, and also the best possible response time tail index for the heavy queue. We conjecture that the same is true for the wireless scenario. As before, the main difficulty in extending our results to the wireless scenario is that the behavior of the busy period tail is unknown for this case.

The following theorems summarize our results.

Theorem 24. *In the wireline scenario, suppose that the arrival rates lie in the subset Λ' of the stability region. Then under log-max-weight scheduling between queues, and first-come-first-served scheduling within the heavy queue,*

$$\lim_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} = \theta_H - 1.$$

Proof. The proof follows along similar lines as the corresponding proof for max-weight-α scheduling (i.e., the proof of Theorem 19), except that for the lower bound on the tail of V_H, we need to prove that $\bar{\Gamma}(q_H) \leq \theta_H - 1$. This is easy to argue, since $q_H \geq_{\text{st}} \hat{q}_H$, where $\hat{q}_H$ is the stationary queue length of a $Geo/GI/1$ queue fed by the same arrival process as the heavy queue. Since $\Gamma(\hat{q}_H) = \theta_H - 1$, it follows that $\bar{\Gamma}(q_H) \leq \theta_H - 1$. □

Theorem 25. *In the wireline scenario, suppose that the arrival rates lie in the subset Λ' of the stability region. Then under log-max-weight scheduling between queues, and preemptive-last-come-first-served scheduling within the heavy queue,*

$$\lim_{x\to\infty} -\frac{\log P\left(V_H > x\right)}{\log(x)} = \theta_H.$$

The proof of Theorem 25 is similar to the proof of Theorem 20 in Section 5.4, and is omitted. This completes our discussion of the response time tail for the heavy queue.

In conclusion, in this section, we show in this section that with log-max-weight inter-queue scheduling, it is possible to achieve light-tailed response times in the light queue. In other words, it is possible to design inter-queue and intra-queue scheduling policies for our system such that we maintain throughput optimality, and achieve light-tailed delays for the light queue. Moreover, our results suggest that this can be done without affecting the response time tail for the heavy queue.

5.6 Concluding remarks

In this chapter, we investigate the behavior of generalized max-weight policies in a simple network setting where heavy-tailed and light-tailed flows compete. We begin with the observation that under the classical max-weight policy, the light-tailed flow experiences heavy-tailed response times. This is because the max-weight algorithm throttles the light-tailed flow whenever there is a large burst of traffic from the heavy-tailed flow. The main contribution of this chapter is to show that by carefully designing inter-queue and intra-queue scheduling policies, it is possible to maintain throughput optimality, and guarantee light-tailed response times

for the light-tailed flow. Moreover, our analysis suggests that this can be achieved without affecting the response time tail for the heavy-tailed flow.

5.A Proofs of Lemmas 23 and 24

This section is devoted to the proofs of Lemmas 23 and 24.

Proof of Lemma 23. We have to prove that a non-negative random variable X is heavy-tailed iff.

$$\liminf_{x\to\infty} \Psi_X(x) = 0. \tag{5.10}$$

We first prove that Condition (5.10) implies that X is heavy-tailed. Fix $\mu > 0$. Condition (5.10) implies that there exists a positive, increasing sequence $\{x_k\}$ such that $\lim_{k\to\infty} x_k = \infty$ and $\Psi_X(x_k) < \mu/2$ for all k. Now,

$$\begin{aligned}
& \frac{-\log \bar{F}_X(x_k)}{x_k} < \frac{\mu}{2} \quad \forall\, k \\
\Rightarrow \quad & \bar{F}_X(x_k) > e^{-(\mu/2)x_k} \quad \forall\, k \\
\Rightarrow \quad & \frac{\bar{F}(x_k)}{e^{-\mu x_k}} > e^{(\mu/2)x_k} \quad \forall\, k \\
\Rightarrow \quad & \lim_{k\to\infty} \frac{\bar{F}(x_k)}{e^{-\mu x_k}} = \infty \\
\Rightarrow \quad & \limsup_{x\to\infty} \frac{\bar{F}(x)}{e^{-\mu x}} = \infty.
\end{aligned}$$

Since the above holds for any $\mu > 0$, we conclude that X is heavy-tailed.

Next, we prove that if X is heavy-tailed, then Condition (5.10) holds. Fix $\mu > 0$. Since X is heavy-tailed, we have $\limsup_{x\to\infty} \frac{\bar{F}(x)}{e^{-\mu x}} = \infty$. Therefore, there exists a positive, increasing sequence $\{x_k\}$ such that $\lim_{k\to\infty} x_k = \infty$ and $\frac{\bar{F}(x_k)}{e^{-\mu x_k}} > 1$ for all k. This implies that

$$\begin{aligned}
& \log \bar{F}(x_k) > -\mu x_k \quad \forall\, k \\
\Rightarrow \quad & \liminf_{k\to\infty} \frac{-\log \bar{F}_X(x_k)}{x_k} \le \mu \\
\Rightarrow \quad & \liminf_{x\to\infty} \frac{-\log \bar{F}_X(x)}{x} \le \mu.
\end{aligned}$$

Since the above holds for any $\mu > 0$, it follows that Condition (5.10) holds. This completes the proof. □

Proof of Lemma 24. Fix $y > 0$.

$$\begin{aligned}\frac{\bar{F}_X(x+y)}{\bar{F}_X(x)} &= \frac{e^{-(x+y)\Psi_X(x+y)}}{e^{-(x)\Psi_X(x)}} \\ &\geq e^{-[(x+y)\Psi_X(x)-x\Psi_X(x)]} \\ &= e^{-y\Psi_X(x)}.\end{aligned}$$

This implies that

$$\begin{aligned}\liminf_{x\to\infty} \frac{\bar{F}_X(x+y)}{\bar{F}_X(x)} &\geq \liminf_{x\to\infty} e^{-y\Psi_X(x)} \\ &= 1.\end{aligned}$$

Since $\frac{\bar{F}_X(x+y)}{\bar{F}_X(x)} \leq 1$, it is obvious that

$$\limsup_{x\to\infty} \frac{\bar{F}_X(x+y)}{\bar{F}_X(x)} \leq 1.$$

It therefore follows that

$$\lim_{x\to\infty} \frac{\bar{F}_X(x+y)}{\bar{F}_X(x)} = 1.$$

This completes the proof. □

5.B Technical lemmas

In this section, we state two technical lemmas that are used in our proofs.

The first concerns the probability of extremely large deviations of the running sum of regularly varying i.i.d. random variables from the mean.

Lemma 29. *Suppose that $\{Y(i)\}_{i\geq 1}$ is an i.i.d. sequence of non-negative random variables with $Y(1) \in \mathcal{RV}(\beta), \beta > 1$. Also, suppose $\{\psi(n)\}_{n\geq 1}$ is a positive, increasing sequence that is superlinear, i.e., $\lim_{n\to\infty} {}^{\psi(n)}/_{n} = \infty$. Then*

$$P\left(\sum_{i=1}^{n} Y(i) > \psi(n)\right) \sim nP\left(Y(1) > \psi(n)\right).$$

Intuitively, the above lemma means that extremely large deviations of the running sum from its expected value occur most likely because of one extremely large value. The proof is quite involved, but it omitted here as it follows along the same lines as the proof of Example 23 in [58].

Lemma 30. *Suppose $\{Y_i\}_{i\geq 1}$ is an i.i.d. sequence of non-negative random variables taking values in $\{0\}\cup\mathbb{N}$ satisfying $Y_i \in \mathcal{RV}(\beta)$ for $\beta > 1$, and $\mathbb{E}[Y_i] < 1$. Also, suppose that X is a non-negative random variable*

independent of $\{Y_i\}_{i\geq 1}$, *such that* X *takes values in* $\mathbb{N}$, *and* $X \in \mathcal{RV}(\theta)$, *where* $\theta > 0$. *Define*

$$T := \{t \in \mathbb{N} \mid X + \sum_{i=1}^{t} Y_i \leq t\}.$$

If $\theta \leq \beta$, *then* $T \in \mathcal{RV}(\theta)$.

In the above lemma, T may be interpreted as a busy period in a discrete-time $Geo/GI/1$ queue, started by a job of size X in the queue in time slot 0, with Y_i denoting the amount of work entering the queue in time slot i. The lemma states that if the busy period is started by a random variable X with tail heavier than the job size distribution, then the residual busy period has the same index as X. The proof is a straightforward application of the relationship between the tail of a regularly varying distribution and the behavior of its Laplace-Stieltjes transform around the origin [44] (see also Section 3 in [11]). We omit the proof here.

5.C Proof of Theorem 18

This section is devoted to the proof of Theorem 18. The proof is based on the representation (5.2) of the response time tail. In a tagged busy period, we define a 'bad' event $I(x)$ such that the bound

$$P\left(V_L > x^{\alpha_L}\right) \geq \frac{P\left(I(x)\right)\mathbb{E}\left[N_L^{(x^{\alpha_L})} \mid I(x)\right]}{\mathbb{E}\left[N_L\right]} \tag{5.11}$$

leads us to the statement of the theorem. Informally, the event $I(x)$ involves a job of size $\Theta(x^{\alpha_L})$ arriving into the heavy queue to start the busy period, resulting in $\Omega(x)$ jobs in the light queue experiencing a response time of $\Omega(x^{\alpha_L})$ slots in the busy period.

Without loss of generality, assume that the busy period under consideration starts in time slot 1. Recall that over the subset Λ' of the stability region, $\bar{p}_L < \lambda_L$, and $\lim_{y\to\infty} \lambda_L^{(y)} = \lambda_L$. Pick y large enough so that $\bar{p}_L < \lambda_L^{(y)}$. Pick $\delta > 0$ such that $\delta \leq (\lambda_L^{(y)} - \bar{p}_L)/3$, and δ divides 1 (this last requirement is for notational convenience alone). Define $\beta := \frac{\lambda_L+\delta}{\delta}$. Pick $\mu \in (\bar{p}_L + \delta, 1)$.

Our 'bad' event $I(x) := G(x) \cap H(x)$, where we define and interpret the events $G(x)$ and $H(x)$ below. We start with the definition of $G(x)$.

$$\begin{aligned} G(x) \quad := \quad & \left\{H(1) > \frac{\lceil x \rceil}{\delta} + (\beta+\mu)^{\alpha_L}\lceil x \rceil^{\alpha_L} + \lceil x \rceil^{\alpha_L}\right\} \bigcap \\ & \left\{A_L\left(1, \frac{\lceil x \rceil}{\delta}\right) < \beta \lceil x \rceil\right\} \bigcap \\ & \left\{\bar{S}_L\left(1, \frac{\lceil x \rceil}{\delta}\right) < (\bar{p}_L + \delta)\frac{\lceil x \rceil}{\delta}\right\} \bigcap \\ & \left\{A_L^{(y)}\left(1, \frac{\lceil x \rceil}{\delta}\right) > (\lambda_L^{(y)} - \delta)\frac{\lceil x \rceil}{\delta}\right\} \\ =: \quad & G_1(x) \cap G_2(x) \cap G_3(x) \cap G_4(x). \end{aligned}$$

Roughly, $G(x)$ states that a job of size $\Theta(x^{\alpha_L})$ arrives into the heavy queue at the start of the busy period, and the number of arrivals in the light queue, as well as the number of exclusive slots seen by it in slots 1 through $\frac{\lceil x \rceil}{\delta}$ do not deviate much from their 'law of large numbers' estimates. The following lemma states the key implications of the event $G(x)$.

Lemma 31. *$G(x)$ implies that at the end of $\frac{\lceil x \rceil}{\delta}$ slots,*

(i) *the occupancy of the heavy queue strictly exceeds $(\beta+\mu)^{\alpha_L}\lceil x \rceil^{\alpha_L} + \lceil x \rceil^{\alpha_L}$, i.e., $H\left(\frac{\lceil x \rceil}{\delta}+1\right) > (\beta+\mu)^{\alpha_L}\lceil x \rceil^{\alpha_L} + \lceil x \rceil^{\alpha_L}$,*

(ii) *the occupancy of the light queue is strictly less that $\beta\lceil x \rceil$, i.e., $L\left(\frac{\lceil x \rceil}{\delta}+1\right) < \beta\lceil x \rceil$,*

(iii) *the light queue contains at least $\lceil x \rceil$ packets from jobs of size $\leq y$.*

Proof. The first two claims of the lemma are easy to verify. Indeed, Claim (i) is a consequence of event $G_1(x)$, and Claim (ii) is a consequence of event $G_2(x)$. We give the arguments for Claim (iii) below.

Note that $G(x)$ implies that the light queue does not receive service, except in its exclusive slots, in slots 1 through $\frac{\lceil x \rceil}{\delta}$. Indeed, the event $G_2(x)$ guarantees that the occupancy of the light queue stays strictly below $\beta\lceil x \rceil$ during this period, whereas the event $G_1(x)$ implies that the occupancy of the heavy queue stays above $\beta^{\alpha_L}\lceil x \rceil^{\alpha_L}$ over the same period. Also, during the period from slot 1 to slot $\frac{\lceil x \rceil}{\delta}$, $G_3(x)$ gives an upper bound on the number of exclusive slots received by the light queue, and $G_4(x)$ gives a lower bound on the number packets arriving into the light queue from jobs of size $\leq y$. Therefore, under event $G(x)$, the number of packets remaining in the light queue after time slot $\frac{\lceil x \rceil}{\delta}$, corresponding to jobs of size $\leq y$ exceeds

$$(\lambda_L - \delta)\frac{\lceil x \rceil}{\delta} - (\bar{p}_L + \delta)\frac{\lceil x \rceil}{\delta} \geq \delta\frac{\lceil x \rceil}{\delta} = \lceil x \rceil .$$

This verifies Claim (iii). □

Invoking the weak law of large numbers, we know that $P\left(G_2(x) \cap G_3(x) \cap G_4(x)\right)$ approaches 1 as $x \to \infty$. Therefore, fixing $\nu \in (0,1)$,

$$P\left(G(x)\right) \geq (1-\nu)P\left(G_1(x)\right) \text{ for large enough } x. \tag{5.12}$$

Next, we define the event $H(x)$. This event concerns arrivals into the light queue, and exclusive slots available to it over $\lceil x \rceil^{\alpha_L}$ slots following slot $\frac{x}{\delta}$. Specifically, the event $H(x)$ states that the number of arrivals in the light queue, as well as the number of exclusive slots available to it, do not deviate much from the corresponding 'law of large numbers' estimates over $\lceil x \rceil^{\alpha_L - 1}$ periods, each period being $\lceil x \rceil$ slots long. Formally,

$$H(x) := \bigcap_{k=1,2,\cdots,\lceil x \rceil^{\alpha_L - 1}} H_k(x),$$

where

$$\begin{aligned} H_k(x) &:= \left\{ \bar{S}_L\left(\frac{\lceil x \rceil}{\delta} + (k-1)\lceil x \rceil + 1, \frac{\lceil x \rceil}{\delta} + k\lceil x \rceil\right) < (\bar{p}_L + \delta)\lceil x \rceil \right\} \bigcap \\ &\quad \left\{ A_L\left(\frac{\lceil x \rceil}{\delta} + (k-1)\lceil x \rceil + 1, \frac{\lceil x \rceil}{\delta} + k\lceil x \rceil\right) > (\lambda_L - \delta)\lceil x \rceil \right\} \\ &=: H_{k,1}(x) \cap H_{k,2}(x). \end{aligned}$$

The following lemma states the key implications of our 'bad' event $I(x) = G(x) \cap H(x)$.

Lemma 32. *The event $I(x) = G(x) \cap H(x)$ implies that for $k = 1, \cdots, \lceil x \rceil^{\alpha_L - 1}$,*

(i) $L\left(\frac{\lceil x \rceil}{\delta} + (k-1)\lceil x \rceil + 1\right) \geq L(\frac{\lceil x \rceil}{\delta} + 1)$

(ii) $L(\frac{\lceil x \rceil}{\delta} + (k-1)\lceil x \rceil + m) \geq L(\frac{\lceil x \rceil}{\delta} + 1) - \mu\lceil x \rceil$ *for* $m = 2, 3, \cdots, \lceil x \rceil$.

Proof. We first point out that over the $\lceil x \rceil^{\alpha_L}$ slots following slot $\frac{\lceil x \rceil}{\delta}$, the event $I(x)$ implies that the heavy queue occupancy remains greater than $(\beta + \mu)^{\alpha_L}\lceil x \rceil^{\alpha_L}$. Therefore, if the light queue is to win service in a non-exclusive slot during this period, its occupancy must strictly exceed $(\beta + \mu)\lceil x \rceil$.

Note that Claim (i) above is trivially true for $k = 1$. We prove the lemma inductively as follows. We show that if Claim (i) is true for $k = k_0$, then Claim (ii) is true for $k = k_0$, and Claim (i) is true for $k = k_0 + 1$. Accordingly, let us assume that Claim (i) is true for some $k = k_0$. For notational convenience, define $t[k] := \frac{\lceil x \rceil}{\delta} + (k_0 - 1)\lceil x \rceil$.

We consider two cases.

Case 1: In slots $t[k_0] + 1$ through $t[k_0 + 1]$, the light queue received service only in exclusive slots.

In this case, in the interval from slot $t[k_0] + 1$ to slot $t[k_0 + 1]$, the event $H_{k_0}(x)$ implies that the light queue received service in at most $(\bar{p}_L + \delta)\lceil x \rceil$ slots, which implies that the light queue occupancy cannot decrease by more than $\mu\lceil x \rceil$ over this period. This proves Claim (ii) for $k = k_0$. Moreover, since $H_{k_0}(x)$ implies that the arrivals into the light queue outnumber the number of free slots over this period, Claim (i) follows for $k = k_0 + 1$.

Case 2: In slots $t[k_0] + 1$ through $t[k_0 + 1]$, the light queue received service in a non-exclusive slot.

In this case, let $\tilde{t}$ denote the first non-exclusive slot in which the light queue receives service. Using the same argument as in Case 1, it follows that the light queue occupancy cannot decrease by more than $\mu\lceil x \rceil$ in slots $t[k_0] + 1$ through $\tilde{t} - 1$. As we have argued before, the light queue can win service in a non-exclusive slot only when its occupancy strictly exceeds $(\beta + \mu)\lceil x \rceil$. This means that in the interval from slot $\tilde{t}$ to slot $t[k_0 + 1]$, the queue level can drop below $(\beta + \mu)\lceil x \rceil$ only via service in exclusive slots. Given the upper bound on the number of exclusive slots, it follows that the light queue occupancy stays above $\beta\lceil x \rceil$ slots $\tilde{t}$ through slot $t[k_0 + 1]$. Since we know from Lemma 31 that $L(\frac{\lceil x \rceil}{\delta} + 1) < \beta\lceil x \rceil$, it now follows that Claim (ii) holds for $k = k_0$, and Claim (i) holds for $k = k_0 + 1$.

This completes the proof. □

The above lemma states that under event $I(x)$, over $\lceil x \rceil^{\alpha_L}$ slots following slot $\frac{\lceil x \rceil}{\delta}$, the occupancy of the light queue never dips more than $\mu \lceil x \rceil$ below its level after slot $\frac{\lceil x \rceil}{\delta}$. Moreover, we know from Lemma 31 that under event $I(x)$, at the end of slot $\frac{x}{\delta}$, there are at least $\lceil x \rceil$ packets in the light queue from jobs of size $\leq y$. Therefore, since the light queue uses PLCFS, we conclude that at least $(1-\mu)\lceil x \rceil$ packets, from jobs of size $\leq y$ stay in queue for more than $\lceil x \rceil^{\alpha_L}$ slots. This in turn implies that under event $I(x)$, at least $\frac{(1-\mu)x}{y}$ jobs in the light queue experience a response time exceeding x^{α}, i.e.,

$$\mathbb{E}\left[N_L^{(x^{\alpha_L})} \mid I(x)\right] \geq \frac{(1-\mu)x}{y}. \tag{5.13}$$

Note that the Chernoff bound implies that there exists $\tau > 0$ such that $P\left(H_{k,1}(x)\right) \geq 1 - e^{-\tau\lceil x \rceil}$ and $P\left(H_{k,2}(x)\right) \geq 1 - e^{-\tau\lceil x \rceil}$. Therefore, $P\left(H_k(x)\right) \geq 1 - 2e^{-\tau\lceil x \rceil}$, implying that

$$P\left(H(x)\right) \geq \left(1 - 2e^{-\tau\lceil x \rceil}\right)^{\lceil x \rceil^{\alpha_L - 1}}.$$

It is easy to show that $P\left(H(x)\right) \overset{x\uparrow\infty}{\to} 1$, implying that

$$P\left(H(x)\right) \geq (1-\nu) \text{ for large enough } x. \tag{5.14}$$

Returning to our bound (5.11), we now have, using (5.12), (5.13), and (5.14):

$$\begin{aligned}
P\left(V_L > x^{\alpha_L}\right) &\geq \frac{(1-\nu)^2}{\mathbb{E}\left[N_L\right]} \frac{(1-\mu)x}{y} P\left(G_1(x)\right) \\
\Rightarrow \limsup_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} &= \limsup_{x\to\infty} -\frac{\log P\left(V_L > x^{\alpha_L}\right)}{\log(x^{\alpha_L})} \\
&\leq \limsup_{x\to\infty} -\frac{\log\left(\frac{(1-\nu)^2(1-\mu)}{y\mathbb{E}[N_L]}\right)}{\alpha_L \log(x)} - \frac{\log P\left(G_1(x)\right)}{\alpha_L \log(x)} - \frac{1}{\alpha_L} \\
&= \theta_H - \frac{1}{\alpha_L},
\end{aligned}$$

where the last step uses the fact that $H(1) \in \mathcal{RV}(\theta_H)$, which implies that $\lim_{x\to\infty} -\frac{\log P(G_1(x))}{\log(x)} = \alpha_L \theta_H$.

This completes the proof.

5.D Proof of Lemma 25

This section is devoted to the proof of Lemma 25.

This lemma is a consequence of Theorem 1 in [21]. It is easy to see that the log-max-weight policy is equivalent to a policy that serves in each slot t, the queue that wins the comparison

$$f_L(q_L(t))\eta_L(t) \gtreqless f_H(q_H(t))\eta_H(t),$$

where $f_L(x) = 1/\bar{F}_X(x)$, and $f_H(x) = 1/\bar{F}_X(\log(1+x))$.

Theorem 1 in [21] states that $\mathbb{E}[f_L(q_L)] < \infty$ and $\mathbb{E}[f_H(q_H)] < \infty$, which is the statement of this lemma, if the following conditions hold.

(i) For any $y > 0$, $\lim_{x\to\infty} \frac{f_L(x+y)}{f_L(x)} = 1$.

(ii) For any $y > 0$, $\lim_{x\to\infty} \frac{f_H(x+y)}{f_H(x)} = 1$.

(iii) $\lim_{x\to\infty} xP(L(t) > x) f_L(x) = 0$.

(iv) $\lim_{x\to\infty} xP(H(t) > x) f_H(x) = 0$.

Therefore, to prove the lemma, it suffices to check that the above conditions hold.

Conditions (i) and (ii) hold because $F_X \in \mathcal{L}$. Indeed, Condition (i) follows directly from the definition of the class $\mathcal{L}$. Condition (ii) follows from the uniform convergence theorem for long-tailed functions [10].

Since $L(t)$ is light-tailed, it is easy to argue that there exists $\beta > 0$ such that $xP(L(t) > x) \leq e^{-\beta x}$ for large enough x. This implies that

$$\liminf_{x\to\infty} \frac{xP(L(t) > x)}{\bar{F}_X(x)} \leq \liminf_{x\to\infty} \frac{e^{-\beta x}}{\bar{F}_X(x)} = 0,$$

where the last step follows from the fact that F_X is heavy-tailed. Therefore, Condition (iii) holds.

Finally, since $H(t) \in \mathcal{RV}(\theta_H)$, where $\theta_H > 1$, it is easy to see that for $\epsilon \in (0, \theta_H - 1)$, $xP(H(t) > x) \leq x^{-(\theta_H - 1 - \epsilon)}$ for large enough x. Therefore,

$$\liminf_{x\to\infty} \frac{xP(H(t) > x)}{\bar{F}_X(log(1+x))} \quad \leq \quad \liminf_{x\to\infty} \frac{x^{-(\theta_H-1-\epsilon)}}{\bar{F}_X(log(2x))}.$$

Making the change of variable $x \to e^y$, we obtain

$$\liminf_{x\to\infty} \frac{xP(H(t) > x)}{\bar{F}_X(log(1+x))} \quad \leq \quad \liminf_{y\to\infty} \frac{e^{-(\theta_H-1-\epsilon)y}}{\bar{F}_X(log(2)+y)} = 0,$$

where we have used again the fact that $F_X \in \mathcal{L}$. Therefore, Condition (iv) holds. This completes our proof.

5.E Proof of Theorem 22

In this section, we complete the proof of Theorem 22 by proving Lemmas 27 and 28.

5.E.1 Proof of Lemma 27

We now prove Lemma 27. Pick small $\epsilon \in (0, \lambda_L)$. Interpreting $A_L(0) = 0$, define

$$J := \max\{t \geq 0 \mid A_L(t) \leq (\lambda_L - \epsilon)t\}.$$

We first prove that J is light-tailed.

Lemma 33. *J is light-tailed.*

Proof. For $t > 0$,

$$\begin{aligned} P(J \geq t) &= P(\exists \tau \geq t \mid A_L(\tau) \leq (\lambda_L - \epsilon)\tau) \\ &\leq \sum_{\tau \geq t} P(A_L(\tau) \leq (\lambda_L - \epsilon)\tau) \\ &\leq \sum_{\tau \geq t} e^{-\phi\tau} \quad \text{(for some } \phi > 0) \\ &= \left(\frac{1}{1 - e^{-\phi}}\right) e^{-\phi t}. \end{aligned}$$

The first inequality above follows from the union bound, the second follows from the Chernoff bound. Since we have an exponentially decaying upper bound on the tail d.f. of J, it follows that J is light-tailed. □

Now, we bound the tail d.f. of T_1 from above as follows. For $t > 0$,

$$\begin{aligned} P(T_1 > t) &= P(T_1 > t;\ J < t) + P(T_1 > t;\ J \geq t) \\ &\leq P\left(A_L(t) < \log(1 + q_{\widetilde{H}}(1) + A_H(t));\ J < t\right) + P(J \geq t). \end{aligned}$$

Noting that the event $J < t$ implies that $A_L(t) > (\lambda_L - \epsilon)t$, we have

$$\begin{aligned} P(T_1 > t) &\leq P\left(\log(1 + q_{\widetilde{H}}(1) + A_H(t)) > (\lambda_L - \epsilon)t\right) + P(J \geq t) \\ &= P\left(1 + q_H(0) + H(0) + A_H(t) > e^{(\lambda_L - \epsilon)t} - 1\right) + P(J \geq t). \end{aligned}$$

For large enough t, $e^{(\lambda_L - \epsilon)t} - 1 \geq \frac{2}{3}e^{(\lambda_L - \epsilon)t}$. Therefore, for large enough t,

$$\begin{aligned} P(T_1 > t) &\leq P\left(1 + q_H(0) + H(0) + A_H(t) > \frac{2}{3}e^{(\lambda_L - \epsilon)t}\right) + P(J \geq t) \\ &\leq P\left(1 + q_H(0) > \frac{1}{3}e^{(\lambda_L - \epsilon)t}\right) + P\left(H(0) + A_H(t) > \frac{1}{3}e^{(\lambda_L - \epsilon)t}\right) + P(J \geq t) \\ &=: I + II + III. \end{aligned}$$

To prove that T_1 is light-tailed, it suffices to show that for large enough t, each of the above three terms is bounded above by a an exponentially decaying function of the form $\nu e^{-\phi t}$, where $\nu, \phi > 0$. That this is true for Term I follows from Corollary 4, which states that $\log(1 + q_H(0))$ is light-tailed. That Term II is similarly bounded follows from Lemma 29, which states that $II \sim tP\left(H(1) > \frac{1}{3}e^{(\lambda_L - \epsilon)t}\right)$. Finally, that Term III is bounded above by an exponentially decaying function follows from the fact that J is light-tailed. This completes the proof of Lemma 27.

5.E.2 Proof of Lemma 28

For this proof, we find it convenient to shift the time origin to T_1. Define, for $t \geq 1$,

$$\begin{aligned}\bar{L}(t) := L(T_1 + t), \quad \bar{H}(t) := H(T_1 + t), \quad \bar{S}_{\widetilde{L}}(t) = S_{\widetilde{L}}(T_1 + t)\\ \bar{q}_{\widetilde{L}}(t) := q_{\widetilde{L}}(T_1 + t), \quad \bar{q}_{\widetilde{H}}(t) := q_{\widetilde{H}}(T_1 + t),\\ \bar{A}_L(t) := \textstyle\sum_{i=1}^{t} \bar{L}(i), \quad \bar{A}_H(t) := \sum_{i=1}^{t} \bar{H}(i).\end{aligned}$$

Since T_1 is a stopping time for the fictitious queueing system, $\{\bar{L}(i)\}_{i\geq 1}$ and $\{\bar{H}(i)\}_{i\geq 1}$ are mutually independent sequences, and distributed identically to the sequences $\{L(i)\}_{i\geq 1}$ and $\{H(i)\}_{i\geq 1}$. With this time shift, the virtual queue lengths evolve as

$$\begin{aligned}\bar{q}_{\widetilde{L}}(t+1) &= \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t) - \bar{S}_{\widetilde{L}}(t) \quad \text{for } t \geq 1,\\ \bar{q}_{\widetilde{H}}(t+1) &= \bar{q}_{\widetilde{H}}(1) + \bar{A}_H(t) \quad \text{for } t \geq 1.\end{aligned}$$

Note also that

$$T_2 = \min\{t \geq 1 \mid \bar{S}_{\widetilde{L}}(t) = q_L(1)\}.$$

By the definition of T_1, $\bar{q}_{\widetilde{L}}(1) \geq \log(1 + \bar{q}_{\widetilde{H}}(1)) > 0$. By decreasing $\bar{q}_{\widetilde{L}}(1)$ if needed, we assume that $\bar{q}_{\widetilde{L}}(1) = \log(1+\bar{q}_{\widetilde{H}}(1))$. Using an argument analogous to that in the proof of Lemma 26, it can be shown that decreasing the length of the virtual queue in this manner can only decrease its service process $\bar{S}_{\widetilde{L}}(\cdot)$, thereby possibly increasing the value of T_2. An upper bound on the tail of this 'larger' T_2 is therefore also an upper bound on the tail of the original.

Pick $\sigma \in (0, \frac{\lambda_L}{4})$.

$$\begin{aligned}P(T_2 > t) &= P\left(q_L(1) > \bar{S}_{\widetilde{L}}(t)\right)\\ &= P\left(q_L(1) > \bar{S}_{\widetilde{L}}(t); \bar{S}_{\widetilde{L}}(t) < \sigma t\right) + P\left(q_L(1) > \bar{S}_{\widetilde{L}}(t); \bar{S}_{\widetilde{L}}(t) \geq \sigma t\right)\\ &\leq P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t\right) + P\left(q_L(1) > \sigma t\right).\end{aligned}$$

From Theorem 21, we know that $q_L(0)$ is light-tailed, which implies that $q_L(1)$ is light-tailed. Therefore, there exist constants $\nu, \phi > 0$ such that $P\left(q_L(1) > \sigma t\right) \leq \nu e^{-\phi t}$. To prove that T_2 is light-tailed, it suffices to prove a similar exponentially decaying upper bound on $P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t\right)$. We now proceed to do this.

Pick $\delta \in (\frac{\lambda_L}{2}, \frac{3\lambda_L}{4})$.

$$\begin{aligned} P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t\right) &= P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t; \frac{\bar{A}_L(t) - \log(2+\bar{A}_H(t))}{t} - \delta \leq \frac{\bar{S}_{\widetilde{L}}(t)}{t}\right) \\ &\quad + P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t; \frac{\bar{A}_L(t) - \log(2+\bar{A}_H(t))}{t} - \delta > \frac{\bar{S}_{\widetilde{L}}(t)}{t}\right) \\ &\leq P\left(\frac{\bar{A}_L(t) - \log(2+\bar{A}_H(t))}{t} < \sigma + \delta\right) \\ &\quad + P\left(\frac{\bar{A}_L(t) - \log(2+\bar{A}_H(t))}{t} - \delta > \frac{\bar{S}_{\widetilde{L}}(t)}{t}\right) \\ &=: I + II. \end{aligned} \tag{5.15}$$

To establish an exponentially decaying upper bound on $P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t\right)$, we now prove exponentially decaying upper bounds on Terms I and II separately.

Let us first consider Term I. Let $\epsilon := (\lambda_L - (\sigma + \delta))/2$. Note that our restrictions on δ and σ imply that $\epsilon > 0$.

$$\begin{aligned} I &= P\left(\log(2+\bar{A}_H(t)) > \bar{A}_L(t) - (\lambda_L - 2\epsilon)t\right) \\ &= P\left(\log(2+\bar{A}_H(t)) > \bar{A}_L(t) - (\lambda_L - 2\epsilon)t; \bar{A}_L(t) > (\lambda_L - \epsilon)t\right) \\ &\quad + P\left(\log(2+\bar{A}_H(t)) > \bar{A}_L(t) - (\lambda_L - 2\epsilon)t; \bar{A}_L(t) \leq (\lambda_L - \epsilon)t\right) \\ &\leq P\left(\log(2+\bar{A}_H(t)) > \epsilon t\right) + P\left(\bar{A}_L(t) \leq (\lambda_L - \epsilon)t\right) \\ &=: Ia + Ib. \end{aligned}$$

Invoking Lemma 29, we conclude that $Ia \sim tP\left(H(1) > e^{\epsilon t}\right)$, which implies that Term Ia is bounded above by an exponentially decaying function of t. The Chernoff bound implies that Term Ib is similarly bounded. Therefore, there exist $\nu_1, \phi_1 > 0$ such that $I \leq \nu_1 e^{-\phi_1 t}$ for large enough t.

Let us now consider term II. Define

$$\kappa(t) := \min\{\kappa \geq 1 \mid \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t-\kappa) - \bar{S}_{\widetilde{L}}(t-\kappa) < \log(1 + \bar{q}_{\widetilde{H}}(1) + \bar{A}_H(t-\kappa))\}.$$

$\kappa(t)$ is defined so that by the end of slot t, the fictitious queue $\widetilde{L}$ would have received service for $\kappa(t) - 1$ consecutive slots. Therefore, $\bar{S}_{\widetilde{L}}(t) - \bar{S}_{\widetilde{L}}(t - \kappa(t)) = \kappa(t) - 1$. Let $E(t)$ denote the event in II.

$$\begin{aligned} E(t) &\iff \bar{A}_L(t) - \bar{S}_{\widetilde{L}}(t) > \log(2+\bar{A}_H(t)) + \delta t \\ &\Rightarrow \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t) - \bar{S}_{\widetilde{L}}(t) > \log(1 + q_{\widetilde{H}}(1)) + \log(2+\bar{A}_H(t)) + \delta t \\ &\Rightarrow \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t) - \bar{S}_{\widetilde{L}}(t) > \log(3 + q_{\widetilde{H}}(1) + \bar{A}_H(t)) + \delta t \\ &\Rightarrow \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t) - \bar{S}_{\widetilde{L}}(t) > \log(1 + q_{\widetilde{H}}(1) + \bar{A}_H(t - \kappa(t))) + \delta t. \end{aligned}$$

The first implication above follows from our assumption that $\bar{q}_{\widetilde{L}}(1) = \log(1+q_{\widetilde{H}}(1))$. The second implication uses the fact that for $x, y \geq 2$, $\log(xy) \geq \log(x+y)$. Now, from the definition of $\kappa(t)$, we conclude that

$$\begin{aligned} E(t) \quad &\Rightarrow \quad \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t) - \bar{S}_{\widetilde{L}}(t) > \bar{q}_{\widetilde{L}}(1) + \bar{A}_L(t-\kappa(t)) - \bar{S}_{\widetilde{L}}(t-\kappa(t)) + \delta t \\ &\Rightarrow \quad \bar{A}_L(t) - \bar{A}_L(t-\kappa(t)) > \bar{S}_{\widetilde{L}}(t) - \bar{S}_{\widetilde{L}}(t-\kappa(t)) + \delta t. \\ &\Longleftrightarrow \quad \bar{A}_L(t) - \bar{A}_L(t-\kappa(t)) > \kappa(t) - 1 + \delta t. \end{aligned}$$

Pick $\epsilon' > 0$ such that $\lambda_L + \epsilon' < 1$. We can now bound II as follows.

$$\begin{aligned} II \quad &\leq \quad P\left(\bar{A}_L(t) - \bar{A}_L(t-\kappa(t)) > \kappa(t) - 1 + \delta t\right) \\ &= \quad P\left(\bar{A}_L(t) - \bar{A}_L(t-\kappa(t)) > \kappa(t) - 1 + \delta t;\ \kappa(t) - 1 \leq \left\lfloor \frac{\delta t}{\lambda_L + \epsilon'} \right\rfloor\right) \\ &\qquad P\left(\bar{A}_L(t) - \bar{A}_L(t-\kappa(t)) > \kappa(t) - 1 + \delta t;\ \kappa(t) - 1 > \frac{\delta t}{\lambda_L + \epsilon'}\right) \\ &\leq \quad P\left(\bar{A}_L(t) - \bar{A}_L\left(t - \left\lfloor \frac{\delta t}{\lambda_L + \epsilon'} \right\rfloor - 1\right) > \delta t\right) + P\left(A_L(t) > \frac{\delta t}{\lambda_L + \epsilon'} + \delta t\right) \\ leq \quad &P\left(\bar{A}_L(t) - \bar{A}_L\left(t - \left\lfloor \frac{\delta t}{\lambda_L + \epsilon'} \right\rfloor - 1\right) > \delta t\right) + P\left(A_L(t) > 2\delta t\right). \end{aligned}$$

Using the Chernoff bound on both the above terms, we conclude that there exist $\nu_2, \phi_2 > 0$ such that $II \leq \nu_2 e^{-\phi_2 t}$.

Since we have proved exponentially decaying upper bounds on Terms I and II in (5.15), it follows that $P\left(\bar{S}_{\widetilde{L}}(t) < \sigma t\right)$ decays exponentially in t, which implies that T_2 is light-tailed. This completes the proof.

5.F Proof of Theorem 23

This section is devoted to the proof of Theorem 23. The proof is structurally similar to the proof of Theorem 18.

The proof is based on the representation (5.2) of the response time tail. In a tagged busy period, we define a 'bad' event $I(x)$ such that the bound

$$P\left(V_L > x\right) \geq \frac{P\left(I(x)\right) \mathbb{E}\left[N_L^{(x)} \mid I(x)\right]}{\mathbb{E}\left[N_L\right]} \tag{5.16}$$

leads us to the statement of the theorem. Informally, the event $I(x)$ involves a large enough job arriving into the heavy queue to start the busy period, resulting in $\Omega(\log(x))$ jobs in the light queue experiencing a response time of $\Omega(x)$ slots in the busy period.

Without loss of generality, assume that the busy period under consideration starts in time slot 1. Recall that over the subset Λ' of the stability region, $\bar{p}_L < \lambda_L$, and $\lim_{y\to\infty} \lambda_L^{(y)} = \lambda_L$. Pick y large enough so that

$\bar{p}_L < \lambda_L^{(y)}$. Pick $\delta > 0$ such that $\delta \leq (\lambda_L^{(y)} - \bar{p}_L)/4$.

Our 'bad' event $I(x) := G(x) \cap H(x)$, where we define and interpret the events $G(x)$ and $H(x)$ below. We start with the definition of $G(x)$. This event is parameterized by $\beta \in \mathbb{N}$, whose value we fix later.

$$\begin{aligned} G(x) &:= \left\{H(1) > \beta \lfloor \log(x) \rfloor + x^{\beta(\lambda_L + 2\delta)} + x + \beta\delta \log(x)\right\} \bigcap \\ &\quad \left\{A_L\left(1, \beta \lfloor \log(x) \rfloor\right) < (\lambda_L + \delta)\beta \lfloor \log(x) \rfloor\right\} \bigcap \\ &\quad \left\{\bar{S}_L\left(1, \beta \lfloor \log(x) \rfloor\right) < (\bar{p}_L + \delta)\beta \lfloor \log(x) \rfloor\right\} \bigcap \\ &\quad \left\{A_L^{(y)}\left(1, \beta \lfloor \log(x) \rfloor\right) > (\lambda_L^{(y)} - \delta)\beta \lfloor \log(x) \rfloor\right\} \\ &=: G_1(x) \cap G_2(x) \cap G_3(x) \cap G_4(x). \end{aligned}$$

Roughly, $G(x)$ states that a job of size $\Theta(x^{\max\{\beta(\lambda_L+2\delta),1\}})$ arrives into the heavy queue at the start of the busy period, and the number of arrivals in the light queue, as well as the number of exclusive slots seen by it in slots 1 through $\beta \lfloor \log(x) \rfloor$ do not deviate much from their 'law of large numbers' estimates. The following lemma states the key implications of the event $G(x)$.

Lemma 34. *$G(x)$ implies that at the end of $\beta \lfloor \log(x) \rfloor$ slots,*

(i) the occupancy of the heavy queue strictly exceeds $x^{\beta(\lambda_L+2\delta)} + x + \beta\delta \log(x)$, i.e., $H(\beta \lfloor \log(x) \rfloor + 1) > x^{\beta(\lambda_L+2\delta)} + x + \beta\delta \log(x)$,

(ii) the occupancy of the light queue is strictly less that $(\lambda_L + \delta)\beta \lfloor \log(x) \rfloor$, i.e., $L\left(\lceil \log(x) \rceil + 1\right) < (\lambda_L + \delta)\beta \lfloor \log(x) \rfloor$,

(iii) the light queue contains at least $2\beta\delta \lfloor \log(x) \rfloor$ packets from jobs of size $\leq y$.

The proof of Lemma 34 is very similar to the proof of Lemma 31, and is omitted.

Invoking the weak law of large numbers, we know that $P\left(G_2(x) \cap G_3(x) \cap G_4(x)\right)$ approaches 1 as $x \to \infty$. Therefore, fixing $\nu \in (0, 1)$,

$$P\left(G(x)\right) \geq (1 - \nu) P\left(G_1(x)\right) \text{ for large enough } x. \tag{5.17}$$

Next, we define the event $H(x)$. Let

$$n(x) := \left\lceil \frac{x}{\lfloor \beta\delta \lfloor \log(x) \rfloor \rfloor} \right\rceil, \quad m(x) := \lfloor \beta\delta \lfloor \log(x) \rfloor \rfloor.$$

The event $H(x)$ concerns arrivals into the light queue, and exclusive slots available to it over $n(x)m(x)$ slots following slot $\beta \lfloor \log(x) \rfloor$. Specifically, the event $H(x)$ states that the number of arrivals in the light queue, as well as the number of exclusive slots available to it, do not deviate much from the corresponding 'law of large numbers' estimates over $n(x)$ periods, each period being $m(x)$ slots long. For notational convenience,

define $t[k] := \beta \lfloor \log(x) \rfloor + (k-1)m(x)$. Formally,

$$H(x) := \bigcap_{k=1,2,\cdots,n(x)} H_k(x),$$

where

$$\begin{aligned} H_k(x) &:= \left\{ \bar{S}_L\left(t[k]+1, t[k+1]\right) < (\bar{p}_L + \delta)m(x) \right\} \bigcap \\ &\quad \left\{ A_L\left(t[k]+1, t[k+1]\right) > (\lambda_L - \delta)m(x) \right\} \\ &=: H_{k,1}(x) \cap H_{k,2}(x). \end{aligned}$$

The following lemma states the key implications of our 'bad' event $I(x) = G(x) \cap H(x)$.

Lemma 35. *The event $I(x) = G(x) \cap H(x)$ implies that for $k = 1, \cdots, n(x)$,*

(i) $L\left(t[k]+1\right) \geq L(t[1]+1)$

(ii) $L(t[k]+m) \geq L(t[1]+1) - m(x)$ *for* $m = 2, 3, \cdots, \lceil x \rceil$.

The proof of this lemma is very similar to that of Lemma 32, and is omitted. The above lemma states that under event $I(x)$, over $n(x)m(x)$ slots following slot $\beta \lfloor \log(x) \rfloor$, the occupancy of the light queue never dips more than $m(x)$ below its level after slot $\beta \lfloor \log(x) \rfloor$. Moreover, we know from Lemma 34 that under event $I(x)$, at the end of slot $\beta \lfloor \log(x) \rfloor$, there are at least $2\beta\delta \lfloor \log(x) \rfloor$ packets in the light queue from jobs of size $\leq y$. Therefore, since the light queue uses PLCFS, we conclude that at least $\beta\delta \lfloor \log(x) \rfloor$ packets, from jobs of size $\leq y$, stay in queue for more than $n(x)m(x)$ slots. Since $n(x)m(x) \geq x$, this in turn implies that under event $I(x)$, at least $\frac{2\beta\delta\lfloor \log(x) \rfloor}{y}$ jobs in the light queue experience a response time exceeding x, i.e.,

$$\mathbb{E}\left[N_L^{(x)} \mid I(x) \right] \geq \frac{2\beta\delta \lfloor \log(x) \rfloor}{y}. \tag{5.18}$$

Note that the Chernoff bound implies that there exists $\tau > 0$ such that $P\left(H_{k,1}(x)\right) \geq 1 - e^{-\tau m(x)}$ and $P\left(H_{k,2}(x)\right) \geq 1 - e^{-\tau m(x)}$. Therefore, $P\left(H_k(x)\right) \geq 1 - 2e^{-\tau m(x)}$, implying that

$$P\left(H(x)\right) \geq \left(1 - 2e^{-\tau m(x)}\right)^{n(x)}.$$

Let us fix $\beta > 1/\tau\delta$. For this choice of β, it is easy to show that $P\left(H(x)\right) \overset{x\uparrow\infty}{\to} 1$, implying that

$$P\left(H(x)\right) \geq (1-\nu) \text{ for large enough } x. \tag{5.19}$$

Returning to our bound (5.16), we now have, using (5.17), (5.18), and (5.19):

$$\begin{aligned} P\left(V_L > x\right) &\geq \frac{(1-\nu)^2}{\mathbb{E}\left[N_L\right]} \frac{2\beta\delta \left\lfloor \log(x) \right\rfloor}{y} P\left(G_1(x)\right) =: c\, P\left(G_1(x)\right) \\ \Rightarrow \limsup_{x\to\infty} -\frac{\log P\left(V_L > x\right)}{\log(x)} &\leq \limsup_{x\to\infty} -\frac{\log(c)}{\log(x)} - \frac{\log P\left(G_1(x)\right)}{\log(x)} \\ &= \theta_H \max\{\beta(\lambda_L + 2\delta), 1\}, \end{aligned}$$

where the last step uses the fact that $H(1) \in \mathcal{RV}(\theta_H)$.

Since $\limsup_{x\to\infty} -\frac{\log P(V_L>x)}{\log(x)} < \infty$, V_L is heavy-tailed. This completes the proof.

Bibliography

[1] L. Amaral, S. Buldyrev, S. Havlin, H. Leschhorn, P. Maass, M. Salinger, H. Stanley, and M. Stanley, "Scaling behavior in economics: I. Empirical results for company growth," *Journal de Physique I*, vol. 7, pp. 621–633, 1997.

[2] V. Anantharam, "How large delays build up in a GI/G/1 queue," *Queueing Systems*, vol. 5, no. 4, pp. 345–367, 1989.

[3] ——, "Scheduling strategies and long-range dependence," *Queueing Systems*, vol. 33, no. 1, pp. 73–89, 1999.

[4] M. Arlitt and C. Williamson, "Internet web servers: Workload characterization and performance implications," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 631–645, 1997.

[5] S. Asmussen, P. Fiorini, L. Lipsky, T. Rolski, and R. Sheahan, "Asymptotic behavior of total times for jobs that must start over if a failure occurs," *Mathematics of Operations Research*, vol. 33, no. 4, pp. 932–944, 2008.

[6] B. Avi-Itzhak and S. Halfin, "Expected response times in a non-symmetric time sharing queue with a limited number of service positions," in *Proceedings of the 12th International Teletraffic Congress*, 1988.

[7] M. Bagnoli and T. Bergstrom, "Log-concave probability and its applications," Department of Economics, University of California Santa Barbara, Economics Working Paper Series, 2004. [Online]. Available: http://ideas.repec.org/p/cdl/ucsbec/1989d.html

[8] J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1566 –1579, 1995.

[9] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[10] N. Bingham, C. Goldie, and J. Teugels, *Regular variation*. Cambridge University Press, 1989.

[11] S. Borst, O. Boxma, R. Nunez-Queija, and B. Zwart, "The impact of the service discipline on delay asymptotics," *Performance Evaluation*, vol. 54, pp. 175–206, 2003.

[12] O. Boxma and D. Denisov, "Sojourn time tails in the single server queue with heavy-tailed service times," EURANDOM, Tech. Rep., 2009. [Online]. Available: http://www.eurandom.nl/reports/2009/057-report.pdf

[13] O. Boxma and B. Zwart, "Tails in scheduling," *Performance Evaluation Review*, vol. 34, no. 4, pp. 13–20, 2007.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] D. Champernowne, "A model of income distribution," *The Economic Journal*, vol. 63, no. 250, pp. 318–351, 1953.

[16] V. Chistyakov, "A theorem on sums of independent positive random variables and its applications to branching random processes," *Theory of Probability and its Applications*, vol. 9, pp. 640–648, 1964.

[17] R. Cooper, *Introduction to queueing theory*. North-Holland, 1981.

[18] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.

[19] C. Cunha, A. Bestavros, and M. Crovella, "Characteristics of WWW client-based traces," Boston University Computer Science Department, Tech. Rep., 1995.

[20] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*. Springer Verlag, 2009.

[21] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, 2005.

[22] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *Proceedings of ACM SIGCOMM*, 1999.

[23] X. Gabaix, "Zipf's law for cities: An explanation," *The Quarterly Journal of Economics*, vol. 114, no. 3, pp. 739–767, 1999.

[24] A. Ganesh, N. O'Connell, and D. Wischik, *Big Queues*. Springer, 2004.

[25] C. Goldie and C. Klüppelberg, "Subexponential distributions," in *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Birkhäuser, Boston, 1998.

[26] F. Guillemin, P. Robert, and B. Zwart, "Tail asymptotics for processor-sharing queues," *Advances in Applied Probability*, vol. 36, no. 2, pp. 525–543, 2004.

[27] V. Gupta and M. Harchol-Balter, "Self-adaptive admission control policies for resource-sharing systems," in *Proceedings of ACM SIGMETRICS*, 2009.

[28] J. Hamilton, "The cost of latency," October 2009, uRL:http://perspectives.mvdirona.com/2009/10/31/TheCostOfLatency.asp.

[29] M. Harchol-Balter and A. B. Downey, "Exploiting process lifetime distributions for dynamic load balancing," *ACM Transactions on Computer Systems*, vol. 15, no. 3, pp. 253–285, 1997.

[30] D. Heath, S. Resnick, and G. Samorodnitsky, "Heavy tails and long range dependence in ON/OFF processes and associated fluid models," *Mathematics of Operations Research*, vol. 23, no. 1, pp. 145–165, 1998.

[31] K. Jagannathan, "Asymptotic performance of queue length based network control policies," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

[32] K. Jagannathan, M. Markakis, E. Modiano, and J. Tsitsiklis, "Throughput optimal scheduling in the presence of heavy-tailed traffic," in *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*, 2010.

[33] P. R. Jelenković and J. Tan, "Can retransmissions of superexponential documents cause subexponential delays?" in *Proceedings of IEEE INFOCOM*, 2007.

[34] ——, "Characterizing heavy-tailed distributions induced by retransmissions," Department of Electrical Engineering, Columbia University, Tech. Rep. EE2007-09-07, 2007.

[35] ——, "Dynamic packet fragmentation for wireless channels with failures," in *Proceedings of ACM MobiHoc*, 2008.

[36] J. F. C. Kingman, "A martingale inequality in the theory of queues," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 60, pp. 359–361, 1964.

[37] L. Kleinrock, *Queueing Systems. Volume 1: Theory*. J. Wiley & Sons, 1975.

[38] G. Latouche and V. Ramaswami, *Introduction to matrix analytic methods in stochastic modeling. 1999*. ASA-SIAM, Philadelphia, 1999.

[39] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[40] S. Lohr, "For impatient web users, an wye blink is just too long to wait," *New York Times*, 2012, published Feb. 29. [Online]. Available: http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html

[41] M. Mandjes and B. Zwart, "Large deviations of sojourn times in processor sharing queues," *Queueing Systems*, vol. 52, no. 4, pp. 237–250, 2006.

[42] M. Markakis, E. Modiano, and J. Tsitsiklis, "Scheduling policies for single-hop networks with heavy-tailed traffic," in *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, 2009.

[43] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet mathematics*, vol. 1, no. 2, pp. 226–251, 2004.

[44] N. N. H. Bingham and R. A. Doney, "Asymptotic properties of supercritical branching processes I: The Galton-Watson process," *Advances in Applied Probability*, pp. 711–731, 1974.

[45] J. Nair, M. Andreasson, L. Andrew, S. Low, and J. Doyle, "File fragmentation over an unreliable channel," in *Proceedings of IEEE INFOCOM*, 2010.

[46] J. Nair and S. H. Low, "Optimal job fragmentation," *SIGMETRICS Performance Evaluation Review*, vol. 37, no. 2, pp. 21–23, 2009.

[47] J. Nair, A. Wierman, and B. Zwart, "Scheduling for the tail: robustness versus optimality," in *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*, 2010.

[48] ——, "Tail-robust scheduling via limited processor sharing," *Performance Evaluation*, vol. 67, no. 11, pp. 978–995, 2010.

[49] R. Nelson, *Probability, stochastic processes, and queueing theory: The mathematics of computer performance modelling*. Springer, 1995.

[50] R. Núñez-Queija, "Queues with equally heavy sojourn time and service requirement distributions," *Annals of Operations Research*, vol. 113, no. 1, pp. 101–117, 2002.

[51] M. Nuyens and W. van der Weij, "Monotonicity in the limited processor-sharing queue," *Stochastic Models*, vol. 25, no. 3, pp. 408–419, 2009.

[52] M. Nuyens, A. Wierman, and B. Zwart, "Preventing large sojourn times using SMART scheduling," *Operations Research*, vol. 56, no. 1, pp. 88–101, 2008.

[53] M. Nuyens and B. Zwart, "A large-deviations analysis of the GI/GI/1 SRPT queue," *Queueing Systems*, vol. 54, no. 2, pp. 85–97, 2006.

[54] K. Ramanan and A. L. Stolyar, "Largest weighted delay first scheduling: Large deviations and optimality," *Annals of Applied Probability*, vol. 11, pp. 1–48, 2001.

[55] S. Resnick, *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Springer, 2007.

[56] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1996.

[57] J. S. Sadowsky, "The probability of large queue lengths and waiting times in a heterogeneous multiserver queue II: Positive recurrence and logarithmic limits," *Advances in Applied Probability*, vol. 27, no. 2, pp. 567–583, 1995.

[58] G. Samorodnitsky, "Long range dependence, heavy tails and rare events," 2002, Lecture notes. [Online]. Available: http://dspace.library.cornell.edu/bitstream/1813/9228/1/TR001350.pdf

[59] A. Scheller-Wolf and R. Vesilo, "Structural interpretation and derivation of necessary and sufficient conditions for delay moments in FIFO multiserver queues," *Queueing Systems*, vol. 54, no. 3, pp. 221–232, 2006.

[60] L. E. Schrage, "A proof of the optimality of the shortest remaining processing time discipline." *Operations Research*, vol. 16, no. 3, pp. 678–690, 1968.

[61] B. Schroeder and G. Gibson, "A large-scale study of failures in high-performance computing systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 337–351, 2010.

[62] M. Schwartz, *Telecommunication networks: Protocols, modeling and analysis*. Addison-Wesley Longman Publishing Co., 1986.

[63] R. Sheahan, L. Lipsky, P. M. Fiorini, and S. Asmussen, "On the completion time distribution for tasks that must restart from the beginning if a failure occurs," *ACM SIGMETRICS Performance Evaluation Review*, vol. 34, no. 3, pp. 24–26, 2006.

[64] K. Sigman, "Appendix: A primer on heavy-tailed distributions," *Queueing Systems*, vol. 33, no. 1, pp. 261–275, 1999.

[65] J. Tan, W. Wei, B. Jiang, N. Shroff, and D. Towsley, "Can multipath mitigate power law delays? - Effects of parallelism on tail performance," *SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1, pp. 381–382, 2010.

[66] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[67] ——, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, 1993.

[68] A. Wierman and B. Zwart, "Is tail-optimal scheduling possible?" *Operations Research*, to appear.

[69] F. Zhang and L. Lipsky, "Modelling restricted processor sharing," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, 2006.

[70] ——, "An analytical model for computer systems with non-exponential service times and memory thrashing overhead." in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, 2007.

[71] J. Zhang, J. Dai, and B. Zwart, "Diffusion limits of limited processor sharing queues," Georgia Institute of Technology, Tech. Rep., 2007. [Online]. Available: http://www.isye.gatech.edu/~jzhang/research/lps-ht.pdf

[72] ——, "Law of large number limits of limited processor sharing queues," Georgia Institute of Technology, Tech. Rep., 2007. [Online]. Available: http://www.isye.gatech.edu/~jzhang/research/fl-lps.pdf

[73] J. Zhang and B. Zwart, "Steady state approximations of limited processor sharing queues in heavy traffic," *Queueing Systems*, vol. 60, no. 3, pp. 227–246, 2008.

[74] B. Zwart, "Queueing systems with heavy tails," Ph.D. dissertation, Eindhoven University of Technology, 2001.