Available online at www.sciencedirect.com
ScienceDirect
Performance Evaluation 00 (2010) 1–22
Performance Evaluation

Tail-robust Scheduling via Limited Processor Sharing

Jayakrishnan Nair[a], Adam Wierman[b], Bert Zwart[c]

[a] *Department of Electrical Engineering, California Institute of Technology*
[b] *Computing and Mathematical Sciences Department, California Institute of Technology*
[c] *CWI Amsterdam, VU University Amsterdam, Eurandom & Georgia Tech*

Abstract

From a rare events perspective, scheduling disciplines that work well under light (exponential) tailed workload distributions do not perform well under heavy (power) tailed workload distributions, and vice-versa, leading to fundamental problems in designing schedulers that are robust to distributional assumptions on the job sizes. This paper shows 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 load.

Keywords: GI/GI/1 queue, scheduling, limited processor sharing, large deviations, tail asymptotics, heavy-tailed job size, light-tailed job size, tail-robustness

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) [1], 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, $\mathbb{P}(V > x)$ 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 $\mathbb{P}(V > x)$ as $x \to \infty$, which provides a characterization of the

likelihood of large delays. From this work, which we survey briefly in Section 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 [2], while if the job size distribution is heavy-tailed, SRPT, Processor Sharing (PS), and many other policies (e.g. all SMART policies [3]) minimize (up to a constant) the sojourn time tail [4].

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 [5], 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 paper 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 paper. 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 [7, 8]. Some stochastic ordering results are derived in Nuyens & van der Weij [9]. Zhang,

Dai & Zwart [10, 11, 12] develop fluid, diffusion and heavy traffic approximations. Finally, Gupta & Harchol-Balter [13] 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 3 and 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 3 and 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 tradeoff necessary between optimality and robustness.

In Section 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, by choosing $c = \lfloor 1/(1-\rho) \rfloor + 1$ it is possible to guarantee that LPS-c is tail-robust, i.e., that the sojourn time tail 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 1). 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 2 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 paper is organized as follows. In Section 2, we introduce the model and notation for the paper, and discuss prior work studying the sojourn time asymptotics of scheduling policies. In Sections 3 and 4 we present our new results characterizing the sojourn time asymptotics of LPS-c. Then, in Section 5 we present the main results of the paper showing how to design the multiprogramming level c for LPS-c to ensure tail-robust performance. Finally, we conclude in Section 6.

2. Preliminaries

2.1. Model and Notation

Throughout this paper, 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) $\mathbb{P}(B > A) > 0$ (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 ex-

cess/residual lifetime of B, i.e., $\mathbb{P}(B_e > x) = \frac{1}{\beta}\int_x^\infty \bar{F}_B(t)dt$ 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 paper, our interest is centered around the asymptotic behavior of the sojourn time tail, i.e., the behavior of $\mathbb{P}(V_\pi > x)$ 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 $(0, x]$.

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., $F_X(x) = \mathbb{P}(X \leq x)$; $\Phi_X(\cdot)$ denotes the moment generating function of X, i.e., $\Phi_X(s) = \mathbb{E}\left[e^{sX}\right]$. 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 $\lim_{x\to\infty} \frac{\mathbb{P}(X>x+y)}{\mathbb{P}(X>x)} = 1$ 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 [14]. X (or its d.f. F_X) is said to be *regularly varying* with index $\theta > 1$ (denoted $X \in \mathcal{RV}(\theta)$) if $\mathbb{P}(X > x) = x^{-\theta}L(x)$, 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 [14].

We describe the (logarithmic) asymptotic tail behavior of heavy-tailed X, using its *tail index*, defined as $\Gamma(X) := \lim_{x\to\infty} -\frac{\log \mathbb{P}(X>x)}{\log(x)}$, when the limit exists. Note that if $\Gamma(X) \in (0, \infty)$, then for arbitrarily small $\epsilon > 0$, $x^{-(\Gamma(X)+\epsilon)} \leq \mathbb{P}(X > x) \leq x^{-(\Gamma(X)-\epsilon)}$ 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 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 light-tailed X, using its *decay rate*, defined as $\gamma(X) := \lim_{x\to\infty} -\frac{\log \mathbb{P}(X>x)}{x}$, when the limit exists. If $\gamma(X) \in (0, \infty)$, then for arbitrarily small $\epsilon > 0$, $e^{-(\Gamma(X)+\epsilon)x} \leq \mathbb{P}(X > x) \leq e^{-(\Gamma(X)-\epsilon)x}$ 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 is devoted to the analysis of $\gamma(V_{LPS-c})$ for the case $\gamma(B) \in (0, \infty]$.

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. $B \leq_{\text{st}} V_\pi \leq_{\text{st}} Z^*$, where Z^* denotes the total time to emptyness 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 [4] 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 [15]).

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 1 (Mandjes-Zwart [16])**.** *For $s \geq 0$, $\lim_{x\to\infty} \frac{\log \mathbb{E}[e^{sA(x)}]}{x} = \Psi(s)$. Further, $\Psi(s)$ is strictly convex and lower semi-continuous.*

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

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

see Nuyens & Zwart [17]. 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 [2] that the sojourn time decay rate under FCFS is actually optimal. Moreover, it has been proved by Nuyens et al. [3] 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 dicussion 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.

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.

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 2. *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 [18]. Since $g(\cdot)$ is convex and differentiable, its derviative must be continuous; see Theorem 25.5 in [19]. □

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

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

$$\Gamma(V_{LPS-c}) = \lim_{x\to\infty} -\frac{\log \mathbb{P}\left(V_{LPS-c} > x\right)}{\log(x)} = \min\left\{\theta, (\theta - 1)(c - \lfloor c\rho \rfloor)\right\}. \tag{2}$$

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 A.1, a lower bound on the tail of delay in Appendix A.2, and then combine them to complete the proof of Theorem 1 in Appendix 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 offer 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 Appendix A.1) [21], thus it is perhaps not surprising that k determines the weight of the sojourn time tail in the LPS-c queue. [1] 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).

Another natural view of LPS-c is as a hybrid version of FCFS ($c = 1$) and PS ($c \to \infty$). In this view, Theorem 1 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 $\Gamma(V_{PS}) = \Gamma(B) = \theta$.

[1] A remark about Assumption 1: This assumption ensures that in the presence of $k - 1$ infinitely sized jobs in the system, the queue is positive recurrent (and not null recurrent). This assumption is also made in [21] for the analysis of the GI/GI/c queue.

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 5.

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

$$\gamma(V_{LPS-c}) = \lim_{x\to\infty} -\frac{\log \mathbb{P}(V_{LPS-c} > x)}{x} = \min_{a\in[0,1]} f_c(a), \tag{3}$$

$$\textit{where} \quad f_c(a) := a\gamma_F + \frac{(1-a)\gamma(B)}{c} + \sup_{s\geq 0}\left[(1-a)s\left(1-\frac{1}{c}\right) - \Psi(s)\right]. \tag{4}$$

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

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

[2] 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.

Given the complexity of the decay rate in Theorem 2, 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 [22] 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 (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.

4.1. Interpreting the decay rate under LPS-c

To build an understanding of (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). 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 to effects (ii) and (iii) dominate; and intermediate values of a represent different combinations of all three effects. The minimization operation in (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 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 $(0, x)$.

4.2. 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 3. *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 3 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 3 that the sojourn time tail gets 'lighter' with increasing c. We prove Lemma 3 in Appendix 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 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. [3] 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 2 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 $c \leq \frac{\gamma(B)}{\gamma_F}$, $f_c'(1) \leq 0$, implying $a_c^* = 1$ (recall that $f_c(\cdot)$ is convex) and $\gamma(V_{LPS-c}) = \gamma_F$. Therefore, for small enough c (specifically, $c \leq \frac{\gamma(B)}{\gamma_F}$), the decay rate of LPC-c matches that of FCFS. Moreover, long delays are most likely caused by effect (i).

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 Prop. 5.2.1 in [18]), we can rewrite the sojourn time decay rate as follows.

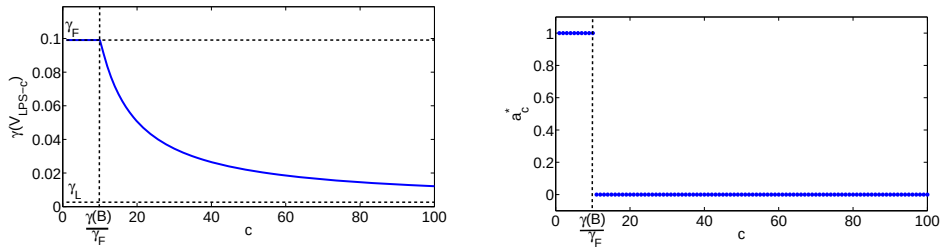
$$\gamma(V_{LPS-c}) = \frac{\gamma(B)}{c} + \max_{s\in[0,\kappa_c]}\left[s\left(1-\frac{1}{c}\right) - \Psi(s)\right]. \tag{5}$$

Note that the above form for the decay rate is more computationally convenient than that in the statement of Theorem 2. Additionally, it allows us to characterize the value of a_c^*; this is summarized in the following lemma.

Lemma 4. *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) $\gamma(V_{LPS-c}) = \frac{\gamma(B)}{c} + g\left(1 - \frac{1}{c}\right) > \gamma(V_{PS})$.

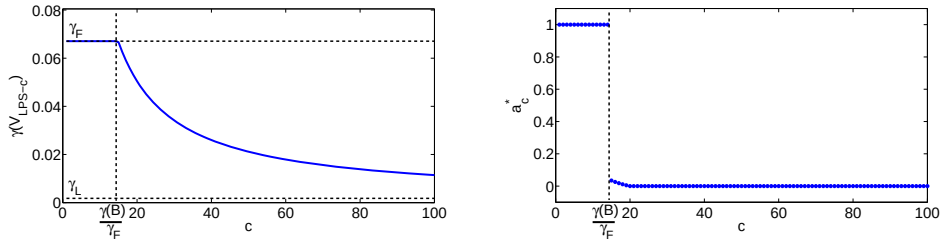
From the standpoint of tail-robust scheduling using LPS, which is the focus of Section 5, the above lemma has the following important implication: For the class of workload distributions

[3] 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 $\gamma_F < \gamma(B)$.



(a) (b)

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



(a) (b)

Figure 2: M/Erlang-2/1 example

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 4 in Appendix B.5.

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

Example 1: Consider first the M/M/1 case; $A \sim Exp(\lambda)$, $B \sim Exp(\mu)$. [4] 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 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 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, $a_c^* \in (0, 1)$.

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

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 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 [5] 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 in 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 3 and 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 1 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 it only needs 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 distribution (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 1. *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., $\gamma(V_{LPS-c}) > \gamma(V_{PS})$. [5]

(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.,* $\gamma(V_{LPS-c}) = \gamma(V_{FCFS})$.

[5] 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 combining Theorems 1 and 2 with Lemma 14 (see Appendix 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 2. *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., $\gamma(V_{LPS-c}) > \gamma(V_{PS})$. [5]

(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 combining Theorems 1 and 2 with Lemma 14 (see Appendix C).

Design 2 provides a parameter, Θ, that allows the scheduling designer to tradeoff 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.

6. Concluding remarks

The contributions in this paper 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 non-work-conserving case where the service rate is a function of the number of jobs in service, as studied in [13]. 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 paper should extend to this case naturally, but there is not room in this paper to describe the details. 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 tradeoffs between robustness and optimality.

Appendix A. Proofs for results in Section 3

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

Appendix 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

$$D_{LPS-c} \leq_{\text{st}} D_{GI/GI/c}. \tag{A.1}$$

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 [21]. The following lemma follows directly from (A.1) and Theorem 2.1 in [21].

Lemma 5. *Under Assumption 1, for* $\eta > 1$, $\mathbb{E}[B^\eta] < \infty \Rightarrow \mathbb{E}\left[D_{LPS-c}^{kc(\eta-1)}\right] < \infty$.

Appendix 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 3. *Under Assumption 1, if* $B_e \in \mathcal{L}$ [6] *and* $\mathbb{E}[B^\eta] < \infty$ *for some* $\eta > 1$, *then*

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

for positive constants τ_1 *and* τ_2.

[6] $B \in \mathcal{L} \Rightarrow B_e \in \mathcal{L}$, but the converse is not true; see Section 3 of [14].

To prove Theorem 3, 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 $x > 0$,

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

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

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

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

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

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 $n_1, n_2 \in \mathcal{N}_1(n(x))$, the events H_{n_1} and H_{n_2} are mutually exclusive.

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

Lemma 6. *$I(x) \Rightarrow D > x$.*

Proof. Since $I(x) \Rightarrow H_n(x)$ for some $n \in \mathcal{N}_1(n(x))$, $I(x)$ implies k large jobs with indices strictly less than $-n(x)$ remain in the system till time x. This means the $n(x)$ 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 $n(x)$ arrivals before the tagged job having an original service requirement bounded above by y strictly exceeds

$$n(x)(\beta^{(y)}-\epsilon)-\frac{\lfloor c\rho\rfloor}{c}\left(x+n(x)(\alpha+\epsilon)\right)=n(x)\left((\beta^{(y)}-\epsilon)-\frac{\lfloor c\rho\rfloor}{c}(\alpha+\epsilon)\right)-x\frac{\lfloor c\rho\rfloor}{c}\geq y(\lfloor c\rho\rfloor-1).$$

This in turn implies the tagged job can receive no service until time x. □

All that remains is to bound $\mathbb{P}(I(x))$, and thus $\mathbb{P}(D>x)$:

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

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,

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

Also, invoking Lemma 7 (stated and proved below), $\mathbb{P}\left(B_{-p}\leq x+p(\alpha+\epsilon)\ \forall\ p\in\mathbb{N}\right)\geq\alpha>0$ for large enough x. Therefore, for large enough x,

$$\mathbb{P}(D>x)\geq\alpha(1-\delta)\sum_{n\in\mathcal{N}_1(n(x))}\prod_{i=1}^{k}\mathbb{P}\left(B>x+n_i(\alpha+\epsilon)\right). \tag{A.2}$$

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

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

From (A.2),

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

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

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

This completes the proof.

Lemma 7. *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 $\tilde{a} > 0$,*

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

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 $x \geq x_0$.

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

□

Appendix A.3. Proof of Theorem 1

We can now complete the proof of Theorem 1 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 3, for large enough x,

$$\begin{aligned} \mathbb{P}(D_{LPS-c} > x) &\geq (1-\delta)\tau_1 \mathbb{P}(B_e > \tau_2 x)^k \\ \Rightarrow \limsup_{x\to\infty} -\frac{\log \mathbb{P}(D_{LPS-c} > x)}{\log(x)} &\leq k \lim_{x\to\infty} -\frac{\log \mathbb{P}(B_e > \tau_2 x)}{\log(x)} = k(\theta - 1). \end{aligned} \tag{A.3}$$

For $\eta \in (1, \theta)$, $\mathbb{E}[B^\eta] < \infty$. Invoking Lemma 5, we conclude that $\mathbb{E}\left[D_{LPS-c}^{k(\eta-1)}\right] < \infty$. Therefore, for large enough x,

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

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

$$\liminf_{x\to\infty} -\frac{\log \mathbb{P}(D_{LPS-c} > x)}{\log(x)} \geq k(\theta - 1). \tag{A.4}$$

Finally, we complete the proof of Theorem 1 by noting that (i) (1) follows from (A.3) and (A.4), and (ii) (2) follows from (1) easily since

$$D_{LPS-c} + B \leq_{\text{st}} V_{LPS-c} \leq_{\text{st}} D_{LPS-c} + Bc,$$

where D_{LPS-c} and B are independent. We omit the details due to space limitations.

Appendix B. Proofs for results in Section 4

In this appendix, we prove the results stated in Section 3. The first three sections are devoted to proving Theorem 2, 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 Appendix B.1 and Appendix B.2 respectively. The proof for the case $\gamma(B) = \infty$ is given in Appendix B.3. In Appendix B.4, we prove Lemma 3, which establishes the monotonicity of $\gamma(V_{LPS-c})$ with respect to c. Finally, we give the proof of Lemma 4 (which describes properties of $\gamma(V_{LPS-c})$) in Appendix B.5.

Appendix B.1. Proof of Theorem 2: 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 8. *Assuming $\gamma(B) \in (0, \infty)$, $\liminf_{x\to\infty} \frac{1}{x} \log \mathbb{P}(V_{LPS-c} > x) \geq -\min_{a\in[0,1]} f_c(a)$.*

To begin the proof, note that since $f_c(\cdot)$ is continuous over $[0, 1]$, it suffices to prove that, for $a \in (0, 1)$, $\liminf_{x\to\infty} \frac{1}{x} \log \mathbb{P}(V_{LPS-c} > x) \geq -f_c(a)$. 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 $\mathbb{P}\left(B^{(y)} > A\right) > 0$. 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,

$$\mathbb{P}(V > x) \geq \mathbb{P}\left(V^{(y)} > x\right)$$

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

At this point, we note that $\mathbb{P}(I_3(x)|I_2) \geq \mathbb{P}\left(Z^{(y)}_{(1-a)(1-\frac{1}{c})(1+\epsilon)} > x\right)$, where $Z^{(y)}_{(1-a)(1-\frac{1}{c})(1+\epsilon)}$ 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

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

we have

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

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

$$\lim_{y\uparrow\infty} \gamma_F^{(y)} = \gamma_F, \tag{B.1}$$

$$\lim_{\epsilon\downarrow 0}\limsup_{y\uparrow\infty}\sup_{s\geq 0}\left[s(1-\frac{1}{c})(1-a)(1+\epsilon) - \Psi^{(y)}(s)\right] = \sup_{s\geq 0}\left[s(1-\frac{1}{c})(1-a) - \Psi(s)\right]. \tag{B.2}$$

(B.1) and (B.2) can be proved by mimicking the arguments in the proofs of Propositions 2.1 and 2.2 respectively in [17]. We omit these proofs due to space constraints.

Appendix B.2. Proof of Theorem 2: 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 9. *Assuming $\gamma(B) \in (0,\infty)$, $\limsup_{x\to\infty} \frac{1}{x}\log \mathbb{P}(V_{LPS-c} > x) \leq -\min_{a\in[0,1]} f_c(a)$.*

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

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

Since W is independent of $\min\{A(x)+B, Bc\}$, and $\mathbb{P}(W > x) \leq e^{-\gamma_F x}$ (this was proved by Kingman [23]), 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 $x > 0$,

$$\begin{aligned}
\mathbb{P}(V > x) &\leq \mathbb{P}\left(\tilde{W} + \min\{A(x) + B, Bc\} > x\right) \\
&\leq \mathbb{P}\left(\tilde{W} > (1-\epsilon)x\right) + \int_{y=0}^{(1-\epsilon)x} \mathbb{P}(\min\{Bc, A(x)+B\} > x - y)\, dF_{\tilde{W}}(y) \\
&\leq \mathbb{P}\left(\tilde{W} > (1-\epsilon)x\right) + x\gamma_F \int_{a=0}^{1-\epsilon} \mathbb{P}(\min\{Bc, A(x)+B\} > (1-a)x)\, e^{-a\gamma_F x} da. \qquad \text{(B.3)}
\end{aligned}$$

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

Lemma 10. *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 $b \in [\epsilon, 1]$, $x \geq x_0$,*

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

Invoking Lemma 10, it follows that there exists $x_0 > 0$ and a function $\eta(x) \in o(1)$ such that

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

for all $a \in [0, 1-\epsilon]$, $x > x_0$. Substituting the above bound in (B.3), we conclude that for large enough x,

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

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

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

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 10. The proof of Lemma 10 depends on the following lemmas.

Lemma 11. *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, $s \leq \gamma(B)$.

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

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 12. *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 10.

Proof of Lemma 10. 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.

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

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.

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

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

$$\begin{aligned}\log \mathbb{P}(\min\{A(x) + B, Bc\} > bx) &\leq \log \mathbb{P}\left(B > \frac{bx}{c}\right) + \log \mathbb{E}\left[e^{s(A(x)+B-bx)} | B > \frac{bx}{c}\right] \\ &= \log \mathbb{P}\left(B > \frac{bx}{c}\right) + \log \mathbb{E}\left[e^{s(B-\frac{bx}{c})} | B > \frac{bx}{c}\right] + \log \mathbb{E}\left[e^{sA(x)}\right] - sbx\left(1 - \frac{1}{c}\right). \quad \text{(B.4)}\end{aligned}$$

We now use Lemma 12 to bound the first two terms of the expression above.

- Since $\lim_{x\to\infty} \frac{\log \mathbb{P}(B>x)}{x} = -\gamma(B)$, there exists $x_1 > 0$, and $\eta_1(x) \in o(1)$ such that for all $b \geq \epsilon$, $x \geq x_1$,
$$\log \mathbb{P}\left(B > \frac{bx}{c}\right) \leq \frac{bx}{c}\left(-\gamma(B) + \eta_1(x)\right). \tag{B.5}$$
- Since $\mathbb{E}\left[e^{\hat{s}(1)B}\right] < \infty$, we know from Lemma 11 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 $s \in [0, \hat{s}(1)]$, $b \geq \epsilon$, $x \geq x_2$,
$$\log \mathbb{E}\left[e^{s(B-\frac{bx}{c})}|B > \frac{bx}{c}\right] \leq \frac{bx}{c}\eta_2(x). \tag{B.6}$$

Finally, invoking Lemma 1, we note that
$$\log \mathbb{E}\left[e^{sA(x)}\right] = x(\Psi(x) + \eta_3(x)), \tag{B.7}$$
where $\eta_3(x) \in o(1)$. Substituting (B.5), (B.6) and (B.7) into (B.4), we obtain that for $s \in [0, \hat{s}(1)]$, $b \geq \epsilon$, $x \geq x_0$,
$$\frac{\log \mathbb{P}\left(\min\{A(x) + B, Bc\} > bx\right)}{x} \leq -\left\{\frac{b\gamma(B)}{c} + bs\left(1 - \frac{1}{c}\right) - \Psi(s) + \eta(x)\right\},$$
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
$$\frac{\log \mathbb{P}\left(\min\{A(x) + B, Bc\} > bx\right)}{x} \leq -\left\{\frac{b\gamma(B)}{c} + \sup_{s\in[0,\hat{s}(1)]}\left[bs\left(1 - \frac{1}{c}\right) - \Psi(s)\right] + \eta(x)\right\}$$
for all $b \geq \epsilon$, $x \geq x_0$. This completes the proof. □

This completes the proof of the asymptotic upper bound.

Appendix B.3. Proof of Theorem 2: The case of $\gamma(B) = \infty$

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

Lemma 13. *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,
$$\limsup_{x\to\infty} \frac{\log \mathbb{P}(V > x)}{x} \leq -\gamma(W + B_0 c) = -\gamma(W) = -\gamma_F.$$
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 Appendix B.1. Reusing the notation developed there, the event $W^{(y)} > x + y(c-1) \Rightarrow V^{(y)} > x$. Therefore,
$$\mathbb{P}(V > x) \geq \mathbb{P}\left(V^{(y)} > x\right) \geq \mathbb{P}\left(W^{(y)} > x + y(c-1)\right)$$
$$\Rightarrow \liminf_{x\to\infty} \frac{\log \mathbb{P}(V > x)}{x} \geq -\gamma_F^{(y)} \quad \Rightarrow \quad \liminf_{x\to\infty} \frac{\log \mathbb{P}(V > x)}{x} \geq -\gamma_F.$$
The last step uses the fact that $\lim_{y\to\infty} \gamma_F^{(y)} = \gamma_F$.

Appendix B.4. Proof of Lemma 3

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. □

Appendix B.5. Proof of Lemma 4

Consider the expression for the LPS decay rate given by (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 3, 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.

Appendix C. Proof of Corollaries 1 and 2 in Section 5

This section states and proves Lemma 14, which is used in the proofs of Corollaries 1 and 2 in Section 5. To state Lemma 14, 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 14.

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

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

$$k_c \geq i \quad \Longleftrightarrow \quad c > \frac{i-1}{1-\rho}. \tag{C.2}$$

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 (C.2). From (C.2), 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 (C.1). □

References

[1] L. E. Schrage, A proof of the optimality of the shortest remaining processing time discipline., Operations Research 16.

[2] K. Ramanan, A. L. Stolyar, Largest weighted delay first scheduling: large deviations and optimality, Annals of Applied Probability 11 (2001) 1–48.
[3] M. Nuyens, A. Wierman, B. Zwart, Preventing large sojourn times using SMART scheduling, Operations Research 56 (1) (2008) 88–101.
[4] O. Boxma, B. Zwart, Tails in scheduling, Performance Evaluation Review 34 (4) (2007) 13–20.
[5] A. Wierman, B. Zwart, Is tail-optimal scheduling possible?, Under submission.
[6] B. Avi-Itzhak, 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] F. Zhang, L. Lipsky, Modelling restricted processor sharing., in: Proc. of the 2006 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA06), 2006.
[8] F. Zhang, L. Lipsky, An analytical model for computer systems with non-exponential service times and memory thrashing overhead., in: Proc. of the 2007 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA07), 2007.
[9] M. Nuyens, W. van der Weij, Monotonicity in the limited processor-sharing queue, Stochastic Models 25 (3) (2009) 408–419.
[10] J. Zhang, J. Dai, B. Zwart, Diffusion limits of limited processor sharing queues., Tech. rep., Georgia Institute of Technology (2007).
URL `http://www.isye.gatech.edu/ jzhang/research/lps-ht.pdf`
[11] J. Zhang, J. Dai, B. Zwart, Law of large number limits of limited processor sharing queues., Tech. rep., Georgia Institute of Technology (2007).
URL `http://www.isye.gatech.edu/ jzhang/research/ fl-lps.pdf`
[12] J. Zhang, B. Zwart, Steady state approximations of limited processor sharing queues in heavy traffic, Queueing Syst. Theory Appl. 60 (3-4).
[13] V. Gupta, M. Harchol-Balter, Self-adaptive admission control policies for resource-sharing systems, in: SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems, 2009.
[14] K. Sigman, Appendix: A primer on heavy-tailed distributions, Queueing Syst. Theory Appl. 33 (1-3) (1999) 261–275.
[15] V. Anantharam, Scheduling strategies and long-range dependence, Queueing Systems Theory Appl. 33 (1-3) (1999) 73–89, Queues with heavy-tailed distributions.
[16] M. Mandjes, B. Zwart, Large deviations of sojourn times in processor sharing queues, Queueing Syst. Theory Appl. 52 (4) (2006) 237–250.
[17] M. Nuyens, B. Zwart, A large-deviations analysis of the GI/GI/1 SRPT queue, Queueing Syst. Theory Appl. 54 (2) (2006) 85–97.
[18] D. P. Bertsekas, Nonlinear Programming, 2nd Edition, Athena Scientific, 1999.
[19] R. T. Rockafellar, Convex Analysis (Princeton Landmarks in Mathematics and Physics), Princeton University Press, 1996.
[20] O. Boxma, D. Denisov, Sojourn time tails in the single server queue with heavy-tailed service times., Tech. rep., EURANDOM (2009).
URL `http://www.eurandom.nl/reports/2009/057-report.pdf`
[21] A. Scheller-Wolf, R. Vesilo, Structural interpretation and derivation of necessary and sufficient conditions for delay moments in FIFO multiserver queues, Queueing Syst. Theory Appl. 54 (3) (2006) 221–232.
[22] 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 27 (2) (1995) 567–583.
[23] J. F. C. Kingman, A martingale inequality in the theory of queues, Mathematical Proceedings of the Cambridge Philosophical Society 60 (02) (1964) 359–361.